



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Sven Boris Bornemann

Android-basierte Smart Home Interaktion am
Beispiel einer Gegensprechanlage

Sven Boris Bornemann
Android-basierte Smart Home Interaktion am
Beispiel einer Gegensprechanlage

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter : Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 1. August 2011

Sven Boris Bornemann

Thema der Bachelorarbeit

Android-basierte Smart Home Interaktion am Beispiel einer Gegensprechanlage

Stichworte

Android™, Smartphone, Türklingel, Smart Home, Ubiquitous Computing, Seamless Interaction

Kurzzusammenfassung

Die steigende Allgegenwärtigkeit von Computern spielt eine tragende Rolle bei der Entwicklung neuer Systeme. Im Rahmen dieser Bachelorarbeit werden Interaktionsmöglichkeiten hinsichtlich eines intelligenten Türklingelsystems in einer Smart Home Umgebung untersucht. Nach der Identifikation verschiedenster Interaktionen durch die Schilderung von Beispielszenarien, werden diverse Architekturmuster auf ihre mögliche Verwendung hin geprüft. Anschließend wird die Realisierbarkeit des Systems anhand von Grundfunktionalitäten einer Gegensprechanlage nachgewiesen.

Sven Boris Bornemann

Title of the paper

Android-based smart home interaction on the example of an intercom system

Keywords

Android™, Smartphone, Doorbell, Smart Home, Ubiquitous Computing, Seamless Interaction

Abstract

The increasing ubiquity of computers plays a crucial role in the development of new systems. As part of this bachelor thesis, interactions are examined for an intelligent doorbell system in a smart home environment. After the identification of various interactions as described by example scenarios, various architectural patterns and their possible use are evaluated. Then, the feasibility of the system using basic functionality of an intercom system is proven.

Inhaltsverzeichnis

1	Einleitung	6
1.1	Zielsetzung	7
1.2	Gliederung der Arbeit	8
2	Grundlagen	9
2.1	Ubiquitous Computing	9
2.1.1	Pervasive Computing	11
2.1.2	Ambient Intelligence	11
2.2	Verteilte Systeme	11
2.3	Smart Home	13
2.4	Kontext	14
2.5	Android™	15
3	Analyse	18
3.1	Szenarien	18
3.1.1	Besucher klingelt	18
3.1.2	Benutzerauthentifizierung	20
3.1.3	Mobile Türklingel	22
3.2	Anforderungsanalyse	23
3.2.1	Funktionale Anforderungen	23
3.2.2	Nicht-Funktionale Anforderungen	28
3.3	Kontextabhängigkeit	30
3.3.1	Identifizierte Kontexte	30
3.3.2	Informationsgewinnung	31
3.3.3	Entscheidungsfindung	31
3.4	Interaktionen	32
3.5	Machbarkeitsstudie	34
3.5.1	Anforderungen	34
3.5.2	Beispielprojekte	34
3.6	Fazit	35
4	Design	36
4.1	Architektur	36

4.1.1	Systemüberblick	36
4.1.2	Abgrenzungen	42
4.1.3	Komponentenarchitektur	44
4.1.4	Model-View-Controller	44
4.2	Kommunikation	47
4.2.1	Synchrone Kommunikation	48
4.2.2	Asynchrone Kommunikation	48
4.3	Dienste	50
4.4	Fazit	53
5	Realisierung	57
5.1	Technische Umsetzung	57
5.1.1	Türklingel	57
5.1.2	Gegenstellen	59
5.1.3	Kommunikation	60
5.2	Evaluation	63
5.2.1	Erfüllung der Anforderungen	63
5.2.2	Funktionsweise des Prototypen	65
5.2.3	Erkenntnisse aus der Evaluation	69
6	Schluss	70
6.1	Zusammenfassung	70
6.2	Ausblick	71
	Literaturverzeichnis	73

1 Einleitung

Die Präsenz von Computern ist in der heutigen Zeit ubiquitär. Sie befinden sich in Chipkarten, Autos, Smartphones, Haushaltsgeräten und noch in vielen weiteren Gegenständen. Ziel für die Zukunft ist, die verschwindende Allgegenwärtigkeit der Computer. Mit diesem Aspekt beschäftigt sich das Smart Environment, welches dafür zwei Hauptziele verfolgt. Ein Aspekt ist hier die Verdrängung der Geräte in den Hintergrund, sodass der Anwender diese gar nicht mehr als solche wahrnehmen kann. Der zweite Gesichtspunkt beschäftigt sich mit der Unterstützung des Menschen im alltäglichen Leben. Hierdurch sollen Sicherheit sowie der Komfort erhöht werden.

Im Laufe der Zeit haben sich viele verschiedene Ausprägungen dieser Idee entwickelt. Ein Beispiel hierfür, ist das Ambient Assisted Living (BMBF). Der Fokus in dieser Ausprägung liegt bei der Unterstützung älterer Menschen. Durch den Einsatz technischer Hilfsmittel soll die Lebensqualität und die Unabhängigkeit älterer Menschen gesteigert werden.

Diese Systeme sind aber nicht nur in den Wohnungsumgebungen des Menschen zu finden, sondern auch in anderen Bereichen. So zum Beispiel auch in der Automobilindustrie. Hier werden immer mehr Assistenzsysteme konzipiert, die den Komfort und die Sicherheit des Autofahrers erhöhen sollen. Beispiele hierfür sind die Fußgängererkennung von Volvo¹ oder das farbige Head-Up Display von BMW².

All diese Ausprägungen haben ihren Ursprung in dem Begriff Ubiquitous Computing, der von Mark Weiser (Weiser (1991)) geprägt wurde. In dieser Vision wird die Allgegenwärtigkeit kleinster unsichtbarer Computer beschrieben. Im europäischen Raum wird dieses Forschungsgebiet auch unter dem Namen Ambient Intelligence zusammengefasst. Dabei sollen die Systeme aktiv auf die Bedürfnisse des Anwenders eingehen und ihm Aufgaben abnehmen. Hierzu werden alltägliche Gegenstände in sogenannte "schlaue" Gegenstände verwandelt. Aus einer Türklingel wird also eine Smart Doorbell, in dem diese mittels moderner Technologie erweitert wird. Dazu zählt unter anderem die Vernetzung mit anderen Gegenständen, um die Funktionalitäten zu erhöhen.

Durch die ständig fortschreitende Miniaturisierung sind wir im Alltag schon jetzt von unzähligen Computern umgeben. Von denen wir manche noch als solche wahrnehmen und andere

¹Quelle: <http://www.volvocars.com/at/explore/pages/pedestrian-detection.aspx> (abgerufen am 20.07.2011)

²Quelle: http://www.bmw.de/de/de/newvehicles/6series/coupe/2011/showroom/connectivity/coloured_hud.html (abgerufen am 20.07.2011)

wiederum nicht. Um die Visionen wahr werden zu lassen fehlt aber noch ein entscheidender Faktor: die Kommunikationsfähigkeit. Das bedeutet, die Geräte müssen untereinander kommunizieren, um die gewollten Funktionalitäten abbilden zu können.

Aber nicht nur die Eigenschaften der Geräte sind entscheidend, sondern auch wie in Zukunft mit diesen interagiert wird. Damit diese in den Hintergrund treten können, müssen neue intuitive Bedienkonzepte und Interaktionsmöglichkeiten entstehen. Außerdem müssen die Systeme Informationen über ihre Umgebung sammeln. Anhand derer sie entscheiden, welche Funktionen dem Anwender angeboten oder welche Arbeitsschritte dem Anwender abgenommen werden können.

1.1 Zielsetzung

Ziel dieser Arbeit ist es, eine „smarte“ Türklingel zu entwickeln, die sich auch durch eine einfache Integration in eine Smart Home Umgebung auszeichnet. Dazu werden mögliche Interaktionen der Anwender, also Bewohner und Besucher, identifiziert. Hierbei spielt die Kommunikation zwischen den Anwendern eine übergeordnete Rolle. Diese soll im Vergleich zu konventionellen Systemen nicht nur einfacher gestaltet sein, sondern auch einen höheren Synthesegrad erreichen. Ziel ist es also, den Anwendern das Gefühl zu vermitteln, dass sie sich bei ihrem Gespräch direkt gegenüber stehen. Dies soll erreicht werden, indem alle möglichen Geräte, innerhalb einer Wohnung als Kommunikationseinheit Verwendung finden. Anhand zusätzlicher Kontextinformationen wird es dem System ermöglicht, sich auf die Bedürfnisse und Situation des Bewohners anzupassen. Diese Informationen werden allerdings nicht von der Türklingel erfasst, sondern von der Umgebung in der sich diese befindet bereitgestellt.

Um die Umsetzbarkeit zu demonstrieren, soll eine prototypische Anwendung entwickelt werden. Dazu werden grundlegende Funktionen, wie die Übertragung eines Live-Videos realisiert. In Folge dessen werden audiovisuelle Daten von einer Gegenstelle, innerhalb der Wohnung, zur Türklingel und umgekehrt übermitteln. Des Weiteren soll das Hinterlassen von Videobotschaften realisiert werden.

In der anfänglichen Entstehungsphase der Arbeit wurden einige Aspekte und Themengebiete dieser Arbeit zusammen mit Sebastian Rösch (Rösch (2011)) erarbeitet. Beide Arbeiten werden im Kontext des Living Place Hamburg erstellt und beschäftigen sich mit der Übertragung audiovisueller Daten. Rösch untersucht in seiner Arbeit eine mögliche audiovisuelle Kommunikation in einer Smart Home Umgebung.

1.2 Gliederung der Arbeit

Die Arbeit gliedert sich in sechs Bestandteile, wobei der erste Teil die Einleitung darstellt. Hier wird ein kurzer Überblick in das Themengebiet gegeben und die Zielsetzung der Arbeit definiert.

Den zweiten Abschnitt der Arbeit bilden die Grundlagen. Hier werden Begriffe und Konzepte erläutert, die für das weitere Verständnis der Arbeit von Bedeutung sind.

Die Analyse des zuvor definierten Ziels ist Bestandteil des dritten Kapitels. Dazu werden Szenarien erläutert, die verschiedenste Interaktionsmöglichkeiten in Smart Home Umgebungen beinhalten. Aus diesen werden funktionale und nicht-funktionale Anforderungen identifiziert und beschrieben. Anschließend wird die Kontextabhängigkeit des Systems untersucht. Zum Schluss dieses Kapitels, wird anhand von Beispielprojekten die Durchführbarkeit dieser Arbeit verifiziert.

Aufgrund der Ergebnisse aus Teil drei, werden im vierten Kapitel verschiedene Designmöglichkeiten vorgestellt und miteinander verglichen. Anhand der identifizierten Anforderungen wird eine Designentscheidung getroffen und anhand von UML-Diagrammen erläutert.

Im fünften Abschnitt der Arbeit wird die Realisierung der Designentscheidung anhand von grundlegenden Funktionalitäten durchgeführt. Dazu werden Funktionen identifiziert, die das mobile Betriebssystem Android™ zur Verfügung stellt und solche, die noch implementiert werden müssen. Anschließend wird die Realisierung evaluiert. Hierfür wird untersucht, welche Anforderungen in der Realisierung umgesetzt werden konnten. Des Weiteren wird die Funktionsweise des entwickelten Prototyps vorgestellt.

Der sechste und letzte Teil der Arbeit beinhalten die Reflexion, im Hinblick auf die zum Anfang der Arbeit definierten Zielsetzung. Des Weiteren wird ein Ausblick gegeben, bezüglich der in der Zukunft möglichen Erweiterungen.

2 Grundlagen

2.1 Ubiquitous Computing

Der Begriff Ubiquitous Computing beschreibt die Allgegenwertigkeit von rechnergestützten Informationssystemen. Geprägt wurde dieser Begriff von Mark Weiser in seinem Artikel „The Computer for the 21st Century“ ([Weiser \(1991\)](#)). In dieser Vision werden die heute üblichen Personal Computer durch „intelligente Gegenstände“ ersetzt, die den Menschen unterstützen ohne ihn abzulenken. Es geht darum, die Interaktion zwischen Mensch und Maschine so zu gestalten, dass sich die Maschine auf den Menschen einstellt und nicht, wie heute üblich, der Mensch auf die Maschine.

John Krumm beschrieb drei elementare Zeitabschnitte der Rechenleistung in seinem Buch Ubiquitous Computing Fundamentals ([Krumm \(2009\)](#)). Das Zeitalter des Mainframe Computings, der Ära des Personal Computings und die des Ubiquitous Computing. Durch die ständige Weiterentwicklung der Mikroelektronik, wird Rechenleistung immer erschwinglicher und ist auf einem geringeren Platz realisierbar. Mittlerweile sind wir von Computern umgeben, die wir nicht mehr als solche wahrnehmen. Sie stecken in Kreditkarten, Autos, Küchengeräten und einer Vielzahl anderer Geräte. Daraus wird ersichtlich, dass wir uns schon in den Anfängen des Ubiquitous Computing Zeitalters befinden.

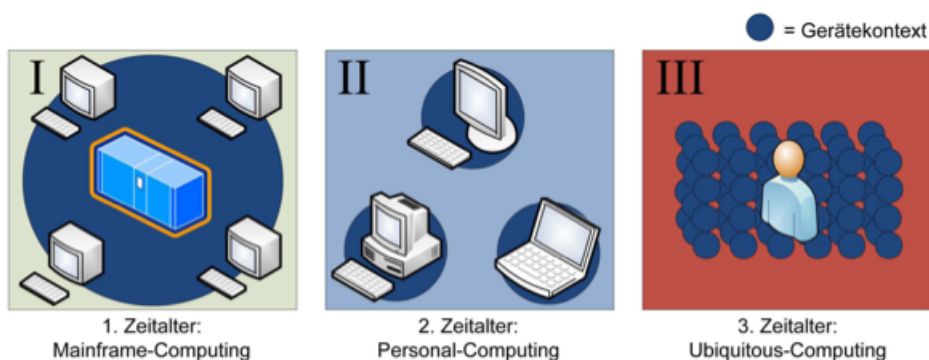


Abbildung 2.1: Zeitalter der Rechenleistung ([Witt \(2011\)](#), S. 12)

Aber wie wünschenswert ist eine Welt, in der nicht nur jeder mit jedem, sondern alles mit allem vernetzt ist? Mark Weiser sagte einmal:

As technology becomes more embedded and invisible, calms our lives by removing the annoyances while keeping us connected with what is truly important.

Mark Weiser
(Weiser, 1991)

Es besteht kein Zweifel, dass sich die Welt auf das Ubiquitous Computing zubewegt. Wie schnell das Ziel allerdings erreicht wird, hängt von der Zuverlässigkeit der Technik und dem daraus resultierenden Vertrauen ab. So schön und komfortabel diese Vision auch klingen mag, wirft sie doch einige gesellschaftliche, ethische und rechtliche Fragen auf.

Angesichts der großen Menge an Mikrosensoren, die ihre Daten beliebig über das Internet verteilen, stellt die Realisierung des Datenschutzes und das Sichern der Privatsphäre eine eminente Herausforderung dar. Auch sollte man sich vergegenwärtigen, zu welchen Folgen und Abhängigkeiten es führen kann, wenn der Technik immer mehr Kontrollfunktionen des täglichen Lebens übertragen werden.

Ubiquitous Computing wird durch eine Reihe von (technischen) Merkmalen identifiziert. Die folgenden Merkmale wurden von dem Büro für Technologiefolgen-Abschätzungen beim Deutschen Bundestag (TAB (2009)) in einem Zukunftsreport zusammengefasst:

- Dezentralität bzw. Modularität
- Einbettung
- Mobilität
- (Spontane) Vernetzung
- Kontextsensitivität
- Autonomie
- Energieautarkie

Im Laufe der Zeit wurden auf der Basis von Mark Weisers Theoremen weitere Ausprägungen entwickelt. Dazu zählen das Pervasive Computing und Ambient Intelligence. Diese beiden Ausprägungen werden im Folgenden kurz erläutert.

2.1.1 Pervasive Computing

Der Begriff Pervasive Computing wurde in der zweiten Hälfte der 1990er Jahre von der Industrie geprägt. Ziel soll es sein, die Allgegenwärtigkeit von Informationen mit schon vorhandenen Techniken kurzfristig nutzbar zu machen. Bregman beschrieb die pragmatische Variante des Ubiquitous Computing folgendermaßen:

Pervasive computing is about enabling people to gain immediate access to information and services anywhere, anytime, without having to scrounge for a phone jack. However, while mobility and wireless technology are a big part of it, it's really about making e-business personal. Thanks to the explosive growth of the Internet, people will soon expect to be able to engage in electronic business effortlessly.
Mark Bregman
(Bregman)

Im Vordergrund steht hier nicht die Verdrängung des Personal Computer, wie in dem Konzept von Mark Weiser. Im Gegenteil: dem Nutzer soll es ermöglicht werden, mit einem beliebigen Endgerät von überall auf Daten zuzugreifen. Für die Kommunikation soll ein verteiltes Netzwerk verwendet werden, in diesem Fall das Internet.

2.1.2 Ambient Intelligence

Im Vordergrund von Ambient Intelligence steht nicht unbedingt der technische Aspekt wie beim Ubiquitous Computing oder Pervasive Computing, sondern die Mensch-Maschinen-Kommunikation und die künstliche Intelligenz. Alltagsgegenstände sollen zu aktiven und kommunikationsfähigen Geräten heranreifen und somit den Menschen in seinem alltäglichen Leben unterstützen. Dafür müssen diese eine gewisse Reaktionsfähigkeit besitzen, um sich den veränderten Bedürfnissen eines Menschen anzupassen.

Zu den Ausprägungen, die aus dem Begriff Ambient Intelligence hervorgehen, gehören Smart Homes (siehe Abschnitt 2.3) sowie Ambient Assistent Living.

2.2 Verteilte Systeme

Verteilte Systeme bilden die Basis für Ubiquitous Computing und all seine Forschungsgebiete. Tanenbaum und van Steen charakterisieren Verteilte Systeme wie folgt:

Ein verteiltes System ist eine Ansammlung unabhängiger Computer, die den Benutzern wie ein einzelnes kohärentes System erscheinen

*Andrew S. Tannenbaum
(Tanenbaum und van Steen, 2007), S.19*

Hieraus geht hervor, dass Verteilte Systeme aus einer Ansammlung von Komponenten bestehen, die untereinander verbunden sind. Wobei dem Benutzer das Gefühl vermittelt werden soll, dass er nur mit einem einzigen System interagiert. Ziel ist es, einen leichteren Zugriff auf Ressourcen zu realisieren und die Tatsache zu verbergen, dass diese über ein Netzwerk verteilt sind.

Um dies zu erreichen, muss ein Verteiltes System in der Lage sein, sich einer Anwendung oder einem Benutzer gegenüber als ein einziges System zu präsentieren. Diese Eigenschaft wird als Transparenz bezeichnet. Tanenbaum u. a. identifizierten hier mehrere Arten der Transparenz, die in Verteilten Systemen Anwendung finden. Die Wichtigsten sind in Tabelle 2.1 aufgelistet.

Transparenz	Beschreibung
Zugriff	Verbirgt Unterschiede in der Darstellung und die Art und Weise, wie auf eine Ressource zugegriffen wird
Ort ¹	Verbirgt, wo sich eine Ressource befindet
Migration	Verbirgt, dass eine Ressource an einen anderen Ort verschoben werden kann
Relokation	Verbirgt, dass eine Ressource an einen anderen Ort verschoben werden kann, während sie genutzt wird
Replikation	Verbirgt, dass eine Ressource repliziert ist
Nebenläufigkeit	Verbirgt, dass eine Ressource von mehreren konkurrierenden Benutzern gleichzeitig genutzt werden kann
Fehler	Verbirgt den Ausfall und die Wiederherstellung einer Ressource

Tabelle 2.1: Unterschiedliche Formen der Transparenz (Tanenbaum und van Steen (2007), S. 22)

Verteilte Systeme lassen sich in verschiedenen Arten klassifizieren. Dazu zählen die Ausrichtung auf Kommunikationsverhalten, Informationsverarbeitung und Pervasivität. Anwendung finden Verteilte Rechensysteme oft in Hochleistungsanwendungen, im Bereich des parallelen Rechnens oder in Büroumgebungen, bei denen der Zugriff auf Datenbanksysteme eine entscheidende Rolle spielt.

¹Ressourcen sind abstrakte Konzepte, die nicht notwendigerweise eine Speicherung in irgendeiner Form beinhalten müssen.

2.3 Smart Home

Unter einem Smart Home versteht man eine intelligente Umgebung, die dem Anwender durch Sensorik, Analyse und Aktorik gewisse Dienste zur Verfügung stellt. Damit sollen dem Anwender Aufgaben abgenommen werden und er soll in seinem täglichen Leben unterstützt werden. Das Smart Home wird durch die Sensorik über die aktuellen Situationen informiert, kann diese dann analysieren und aktiv auf die Geschehnisse mittels der Aktorik eingehen. Die Smart Homes bilden eine Untergruppe der Ambient Intelligence (2.1.2) und sind durch den Begriff "Home" einer festen Umgebung zugeteilt, nämlich einer Wohnung oder einem Haus. In dem Artikel Ambient Intelligence: the Confluence of Ubiquitous/Pervasive Computing and Artificial Intelligence (Augusto (2007)) identifizierte Augusto drei Vorteile für den Anwender:

- Erhöhte Sicherheit
Durch Überwachung der Lebensgewohnheiten und Ermittlung eventuell schädlicher Situation
- Komfort
Beispielsweise durch die automatische Regulierung der Temperatur
- Wirtschaft
Kostenminimierung durch Kontrolle und Regulation

Living Place Hamburg

Die Realisierung des Smart Home Konzeptes stellt das Living Place in Hamburg dar. Seit 2009 steht den Studenten der Hochschule für Angewandte Wissenschaften Hamburg eine 130m² große Wohnung 2.2 zur Verfügung, um den Umgang mit neuen Technologien im Alltag zu erforschen.

Für die Erforschung von Aktivitäten und Technik ist das Living Place mit einem offenen Raumkonzept ausgestattet, dass in unterschiedliche Bereiche, wie Wohn-, Ess- und Schlafbereich aufgeteilt ist. Diese können sich je nach Kontext dem Bewohner anpassen. Neben dem Loft wurden zusätzlich 100m² Büro und Entwicklerfläche ausgebaut, die sich in zwei Arbeitsplätzen, Werkstatt, Serverraum und Usability Labor unterteilen.

Die aufgeführten Informationen und das Bildmaterial stammen aus einem Projektbericht von Rahimi und Vogt (Rahimi und Vogt (2009/2010)). Für einen tieferen Einblick in den Aufbau und die Konzeption des Living Place sei hier auf diesen Bericht verwiesen. Des Weiteren

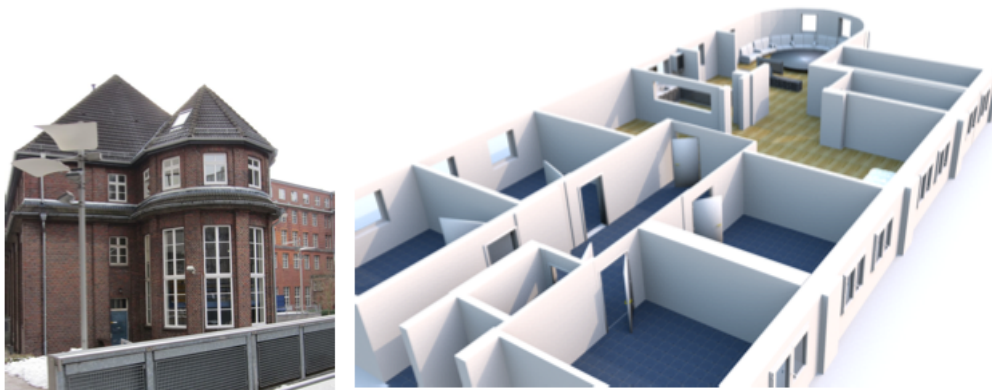


Abbildung 2.2: links: Außenansicht des Living Place, rechts: 3D Rendering

können aktuelle Entwicklungen im Living Place über dessen Internetpräsenz² verfolgt werden.

2.4 Kontext

Gerade in Smart Home Umgebungen werden dem Anwender eine große Vielfalt an Interaktionsmöglichkeiten geboten. Anhand von Kontextinformationen werden dem Anwender nur die momentan benötigten Eingaben oder Ausgaben angezeigt. Dadurch wird eine mögliche Überflutung von Informationen und Interaktionsmöglichkeiten verringert. Dey definiert den Begriff Kontext wie folgt:

Context is any information that can be used to characterise the of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Anind K. Dey
(Dey, 2001)

Ein Kontext besteht also aus Informationen, die die Situation einer Person, eines Ortes oder Objektes beschreiben. Wobei hier die Situation als eine relevante Interaktion zwischen Benutzer und Anwendung zu verstehen ist.

²<http://livingplace.informatik.haw-hamburg.de/blog/>

Anwendungen, die ihr Verhalten an Kontextinformationen ausrichten, werden kontextbewusste (context-aware) Anwendungen genannt. Die Möglichkeiten für kontextbewusste Anwendungen schlüsselte Dey ((Dey, 2001)) in drei Kategorien auf:

- Präsentation
Anpassung der Darstellungsweise von Informationen und Services anhand von Kontextinformationen
- Automatische Ausführung
Die automatische Ausführung einer Aktion, gestützt durch Kontextinformationen
- Historie
Speicherung der Kontextinformationen, um diese zu einem späteren Zeitpunkt wieder abrufen zu können

2.5 Android™

Am 05. November 2007 erklärte Google, dass sie mit einem Zusammenschluss verschiedener Hersteller das mobile Betriebssystem Android™ entwickeln. Dieser Zusammenschluss ist bekannt unter dem Namen Open Handset Alliance (OHA) und beinhaltet Hardware Hersteller wie HTC und Samsung, Provider wie T-Mobile und diverse Software Hersteller³. Seitdem das erste Handy mit dem Android™ Betriebssystem auf den Markt kam, hat sich viel verändert. Das Betriebssystem hat sich über viele verschiedene Versionen hin weiterentwickelt und ist nun auch für Tablet-PCs verfügbar. Momentan sind die Versionen des Betriebssystems für Handy und Tablet-PC noch voneinander getrennt, zukünftig werden diese beiden Stränge aber zusammengeführt.

Die Abbildungen 2.3 stellen die Anzahl aktiver Android™-Geräte zu unterschiedlichen Zeitpunkten dar. Hier ist innerhalb eines Jahres eine signifikante Steigerung zu erkennen, wodurch das enorme Potential von Android™ ersichtlich wird.

Für die Entwicklung von Applikationen wird die Programmiersprache Java verwendet. Dabei wird ein Entwickler durch ein umfangreiches Applikation Programming Interface (API) unterstützt, welches weitestgehend der Java Standard Edition (J2SE) entspricht (Meier (2010)). Lediglich geschwindigkeitskritische Bereiche werden in der Programmiersprache C oder C++ implementiert. Diese nativen Bibliotheken werden beispielsweise für Bildbearbeitung und Medienwiedergabe verwendet.

³Eine vollständige Liste aller Firmen kann unter folgender URL abgerufen werden: http://www.openhandsetalliance.com/oha_members.html (aufgerufen am 07.07.2011)

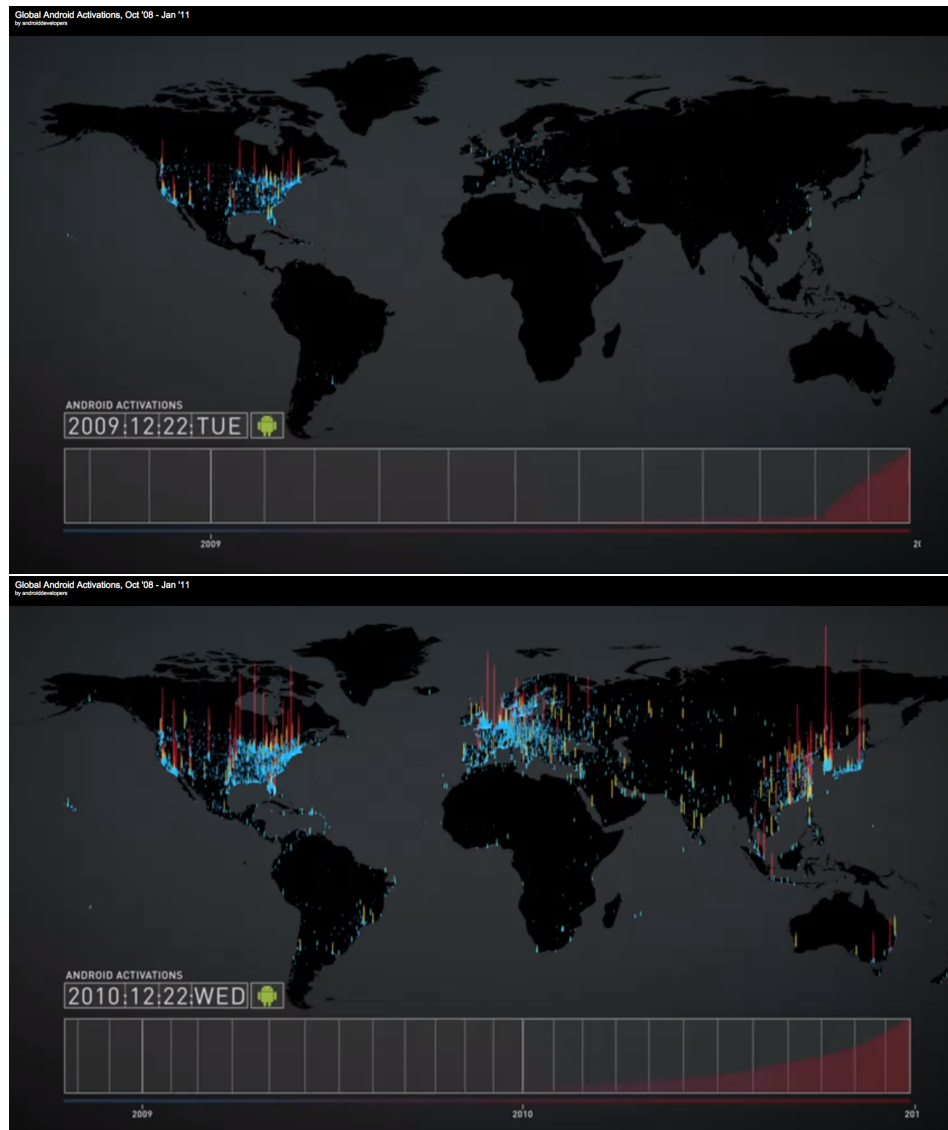


Abbildung 2.3: Aktive Android™-Geräte (oben: Dezember 2009, unten: Dezember 2010)
(Savov, 2011)

Im Vergleich zu anderen mobilen Betriebssystemen versucht Google mit Android™ ein möglichst frei konfigurierbares System bereitzustellen. So können die Funktionalitäten nicht nur durch zusätzliche Applikationen erweitert werden, sondern man kann diese auch ersetzen. Beispielsweise kann die vorhandene Oberfläche durch eine Adaptierte ersetzt werden. Dies nutzen auch die diversen Hersteller und statten ihre Geräte mit einer eigenen Oberfläche aus, um sich von der Konkurrenz abzuheben.

Dieser Ansatz bietet allerdings nicht nur Vorteile. Gerade seitens der Verbraucher wird hier ein erheblicher Nachteil generiert. Diese müssen nämlich nicht selten sehr lange auf Updates des Betriebssystems warten. Bevor die Handy Hersteller die neue Version veröffentlichen, müssen diese erst einmal prüfen, ob die spezifischen Anpassung wie die der Oberfläche noch lauffähig sind.

Philosophie

Das Android Open Source Project (AOSP) ([Google \(2011b\)](#)), wie es von Google genannt wird, schlägt einen ganz anderen Weg ein, als die am Markt vetretene Konkurrenz. Dies hat laut Google verschiedene Gründe ([Google \(2011d\)](#)). Zum einen soll hierdurch sichergestellt werden, dass möglichst viele innovative Ideen der Entwickler zur Realität werden. Zum anderen möchte Google einen "central point of failure" vermeiden. Dieser beschreibt die Einschränkung und Kontrolle von Innovationen durch die Industrie.

Android™ hat das Ziel, eine offene Plattform zu etablieren, um es Entwicklern zu ermöglichen, innovative Applikationen zu entwickeln. Dafür definiert das Android Compatibility Program ([Google \(2011a\)](#)) drei Komponenten, zu denen technische Details, das Android SDK⁴ und der Android Market gehören. Die technischen Details, die für diese Plattform definiert werden, sollen sicherstellen, dass die von den Entwicklern stammenden Anwendungen auf möglichst vielen Geräten laufen. Als Zweites werden Werkzeuge bereitgestellt, um die Möglichkeiten der Geräte voll auszunutzen. Außerdem bietet der Android Market dem Anwender nur Applikationen an, die problemlos auf dem Gerät laufen.

⁴Software Development Kit

3 Analyse

Ziel dieser Analyse ist es, die Anforderungen und Interaktionen am Beispiel einer intelligenten Türklingel zu identifizieren und zu untersuchen. Hierfür werden zunächst einige Beispielszenarien vorgestellt und anschließend dessen Anforderungen formuliert.

3.1 Szenarien

In diesem Kapitel sollen mögliche Szenarien erläutert werden, anhand derer unterschiedlichste Interaktionen erkennbar werden. Dabei handelt es sich um Interaktionen zwischen Menschen über das zu entwickelnde System, zwischen Mensch und Maschine und auch über Interaktionen innerhalb des Systems.

3.1.1 Besucher klingelt

In einer konventionellen Wohnung befinden sich die Klingelsysteme meist an fest definierten Positionen. Die Komponente auf die der Bewohner in seiner Wohnung Zugriff hat, befindet sich an der Wohnungstür und die für den Besucher befindet sich an der Haustür. Ziel soll es nun sein, das autarke System in die Umgebung eines Smart Homes zu integrieren.

Wenn sich ein Besucher vor der Haustür befindet, stehen ihm je nach der aktuellen Situation unterschiedliche Interaktionen zur Verfügung. Ist der Bewohner zu Hause, wird dem Besucher der Knopf zum Klingeln bereitgestellt. Dabei kann hier zwischen einem Einpersonnen- und Mehrpersonnenhaushalt unterschieden werden. Dem Besucher wird bei einem Mehrpersonnenhaushalt angeboten, für jede Personen, die in der Wohnung lebt, zu klingeln. Hat sich der Besucher entschieden, für eine Person zu klingeln, wählt er diese über die Oberfläche der Türklingel aus. Reagiert der Bewohner auf das Signal, wird eine Kommunikation zwischen den Personen aufgebaut. Reagiert dieser nicht oder lehnt die Kommunikation ab, wird dem Besucher eine neue Oberfläche geöffnet. Durch diese wird der Person mitgeteilt, dass der Bewohner momentan nicht verfügbar ist. Zusätzlich wird das Hinterlassen einer

Videonachricht angeboten, um den Bewohner über den Grund des Besuches zu informieren.

Wenn der Bewohner im Moment nicht zu Hause ist, wird dem Besucher gar nicht erst die Möglichkeit des Klingelns angeboten. Stattdessen wird eine vom Bewohner eingegebene Nachricht angezeigt und dem Besucher wird das Hinterlassen einer Nachricht angeboten. Welche weiteren Interaktionsmöglichkeiten einem Besucher zur Verfügung stehen können, wenn der Wohnungsbesitzer nicht zu Hause ist, wird in Szenario 3.1.2 untersucht.

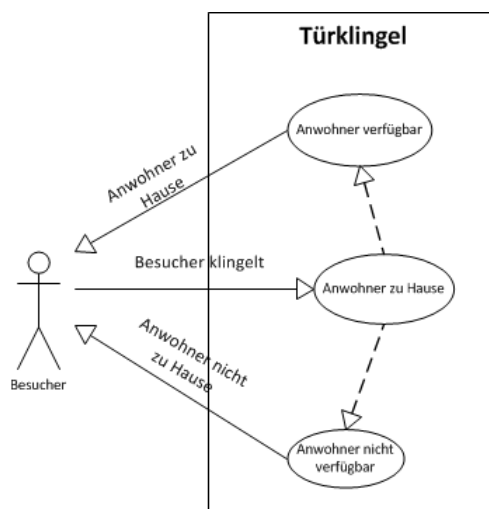


Abbildung 3.1: Anwendungsfall: Besucher klingelt

Gänzlich andere Abläufe und Interaktionen stehen dem Bewohner zur Verfügung. Ist dieser zu Hause und verfügbar, wird ihm das Klingeln über eine visuelle Benachrichtigung signalisiert. Die Visualisierung kann hierbei auf einem Fernseher, Computer, Smartphone, Küchengerät oder einem anderen elektronischen Gerät stattfinden. Innerhalb eines Fensters wird das Live-Video vom Smartphone angezeigt, welches sich vor der Haustür befindet. Ist die Position des Bewohners bekannt, weil diese durch ein vorhandenes Indoor-Positioning-System (Otto und Voskuhl (2010/2011)) erkannt wurde, wird die Visualisierung auf ein in der Nähe befindliches Gerät beschränkt. Ist die Position zur Zeit nicht bekannt, wird die Benachrichtigung auf jedem Gerät in der Wohnung angezeigt, welches die Möglichkeit zur Signalisierung bietet.

Wurde vom System ein geeignetes Gerät zur Anzeige ermittelt, wird hier dem Bewohner eine Nachricht angezeigt. Hier kann er sich nun entscheiden, ob er mit dem Besucher sprechen möchte oder nicht. Reagiert der Bewohner nicht, wird das Fenster nach einer gewissen Zeitspanne geschlossen und der Besucher wird über die Abwesenheit informiert. Das passiert auch, wenn der Bewohner die Kommunikation mit dem Besucher ablehnt. Sollte der

Kommunikation zugestimmt werden, wird eine Verbindung zwischen der Türklingel und der Komponente aufgebaut. Um die Privatsphäre des Bewohners weitestgehend zu schützen, wird in manchen Situationen nur eine Audioübertragung aufgebaut. Zum Beispiel, wenn dieser sich im Schlafzimmer oder im Badezimmer befindet. Des Weiteren kann er auch bei der Klingelvisualisierung entscheiden, ob eine Video- oder nur eine Audioübertragung stattfinden soll. Für welche Art der Übertragung sich der Bewohner entscheidet hängt dann von der Person ab, die sich vor der Tür befindet. Bei Freunden und Familie wird man eine Videoübertragung aus der Wohnung als unproblematisch empfinden. Steht hingegen eine unbekannte Person vor der Tür, wird man auf die Videoübertragung vermutlich verzichten.

Nach einer Kommunikation kann sich der Bewohner nun entscheiden, ob die Tür geöffnet werden soll oder nicht. Für den Fall, dass man eine Person erwartet oder kennt, ist es in den meisten Fällen überflüssig erst über die Gegensprechanlage zu kommunizieren. Daher kann der Bewohner nach der Klingelsignalisierung auch direkt die Tür öffnen.

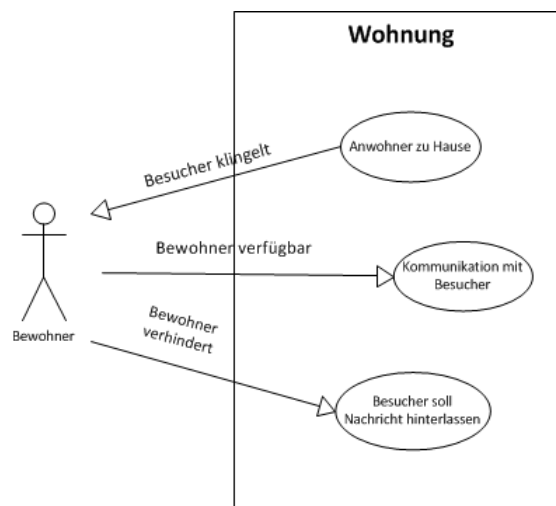


Abbildung 3.2: Anwendungsfall: Dem Bewohner wird das Klingeln signalisiert

3.1.2 Benutzerauthentifizierung

In diesem Szenario wird beschrieben, wie sich Personen beim System authentifizieren können. Wie im vorherigen Szenario beschrieben, dient eine smarte Türklingel nicht nur dazu, eine konventionelle Türklingel zu ersetzen und somit einen weiteren Schritt zum Smart Home und der Vision des Ubiquitous Computing zu erreichen. Es wäre zum Beispiel denkbar, durch geeignete Authentifizierungsmechanismen den Wohnungsschlüssel durch ein Smartphone zu ersetzen, Nachrichten für bestimmte Personen zu hinterlassen oder dem Briefträ-

ger das Öffnen einer speziellen Paketklappe zu erlauben. Geeignete Authentifizierungsmechanismen können beispielsweise der Abgleich biometrischer Daten oder die Eingabe eines Passwortes sein.

Über die Kontakte, die im Smart Home über einen Dienst verwaltet werden, hat der Bewohner die Möglichkeit, bestimmte Personen mit gesonderten Rechten auszustatten. Dies kann beispielsweise eine Zutrittsberechtigung zur Wohnung oder eine Nachrichten für bestimmte Besucher sein. Wird ein Kontakt mit solchen Eigenschaften ausgestattet, wird ein eindeutiger Schlüssel generiert und auf das Smartphone der Person transferiert. Wie eine intuitive Bedienung eines Kontaktdienstes aussehen kann, zeigt Google momentan mit Google+¹. Hier können Personen auf simple Art gruppiert werden, mittels sogenannter Circles (Abbildung 3.3). Für den Einsatz in einem Smart Home könnte dann ein Circle für eine bestimmte Berechtigung stehen.



Abbildung 3.3: Circles in Google+ (Quelle: Google+)

Hält eine Person ihr Smartphone an die Türklingel, wird dieser Schlüssel übertragen und das System prüft, ob ein entsprechender Datensatz zur Verfügung steht. Existiert ein Datensatz, wird die Person aufgefordert, ein Passwort einzugeben, um die Identität zu bestätigen. Je nach gewünschter Sicherheitsstufe sind hier verschiedene Verfahren denkbar. Die einfachste Möglichkeit stellt die Eingabe eines simplen Passwortes dar. Möglich ist aber auch das Aufzeichnen eines komplexen Symbols auf das Display oder die Authentifizierung über die Stimme. Konnte der Besucher kein Passwort anlegen, weil die Berechtigung vom Bewohner spontan angelegt wurde, generiert das System automatisch eine PIN-Nummer und versendet diese an des Smartphone des Berechtigten.

Wurde verifiziert, dass es sich um die richtige Person handelt, werden die für die Person spezifischen Eigenschaften angezeigt. Dies können hinterlassene Nachrichten sein oder die Möglichkeit, die Tür zu öffnen. Betritt eine Person die Wohnung, wird der Bewohner automatisch darüber informiert. So kann zusätzlich sichergestellt werden, dass sich keine unbefugte Person in der Wohnung aufhält.

¹Quelle: <https://plus.google.com/up/start/?continue=https://plus.google.com/&type=st&gpcaz=a3c1f476>

Weitere denkbare Möglichkeiten für einen Besucher nach der Authentifizierung:

- Nachrichten für bestimmte Personen hinterlassen
- Die Ortung des Bewohners anhand dessen Smartphones ermöglichen
- Das Öffnen des Paketfaches gestatten
- Den Bewohner von der Türklingel aus anrufen, wenn dieser unterwegs ist

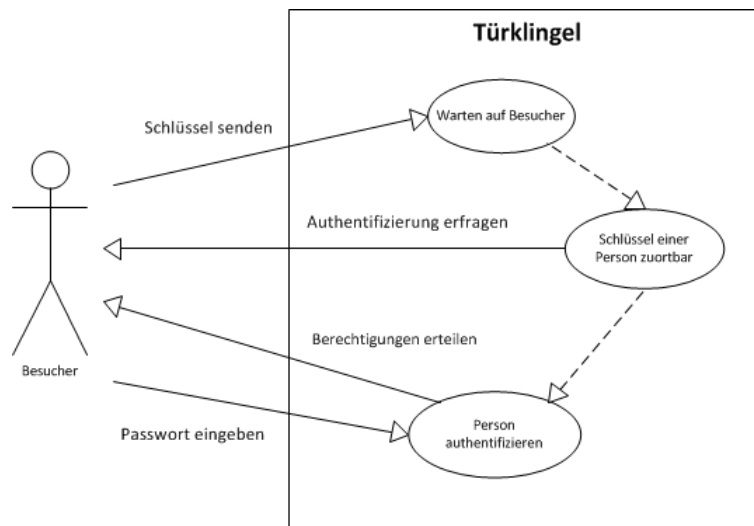


Abbildung 3.4: Anwendungsfall: Interaktion mit autorisierten Personen

3.1.3 Mobile Türklingel

Die vorherigen Szenarien gingen immer von einem fest installierten Smartphone an der Haustür aus, welches alle notwendigen Funktionen zur Verfügung stellt und fest in die Umgebung des Smart Homes integriert ist. Denkbar ist aber auch die Verlagerung der festen Systemkomponente vor der Haustür in eine mobile Applikation, die auf jedem Smartphone installiert werden kann. Somit könnte jeder Anwender immer eine personalisierte Türklingel mit sich führen.

Sobald eine Person an einer Tür steht und klingeln möchte, hält diese das Smartphone an ein Feld. Durch dieses Feld werden der mobilen Applikation automatisch alle notwendigen Daten übermittelt, um sich in die Wohnungsumgebung zu integrieren. Ein Beispiel für eine mögliche Übertragungsart stellt NFC² dar. Zusätzlich übersendet die Anwendung dem System in der Wohnung Daten, um den Besucher zu identifizieren. Handelt es sich hierbei um

²Near Field Communication ist ein Übertragungsstandard zum kontaktlosen Austausch von Daten und stellt eine Erweiterung von RFID dar.

eine dem System gegenüber unbekannte Person, stehen wie in dem Szenario "Besucher klingelt" 3.1.1 dem Besucher nur eingeschränkte Möglichkeiten, wie das Klingeln für eine in der Wohnung lebenden Person oder das Hinterlassen einer Nachricht zur Verfügung. Ist die Person dem System gegenüber aber bekannt, können nach einer Authentifizierung, wie in Szenario 3.1.2 beschrieben, weitere Informationen übermittelt werden. Dazu kann das Anzeigen von Nachrichten, die für diese Person hinterlassen wurden oder das Öffnen der Tür gehören.

Durch die automatische Integration unterschiedlichster Fremdgeräte in das bestehende System wird hier aber auch ein schwer zu kontrollierende Sicherheitslücke generiert. Smartphones stellen im Wesentlichen kleine mobile Computer dar, die wie normale Personal Computer auch anfällig für den Befall von Viren, Trojanern, etc. sind. Werden diese in die Wohnumgebung integriert, können sie auch dieses System infizieren. Die Entwicklung eines geeigneten Sicherheitskonzeptes wird hier nicht weiter verfolgt, da der Fokus der Arbeit auf den Interaktionen des Systems liegt und nicht deren Sicherheitsaspekte beinhaltet.

3.2 Anforderungsanalyse

Aufgrund der erstellten Szenarien ergeben sich funktionale- und nicht-funktionale Anforderungen, die hier aufgeführt und beschrieben werden.

3.2.1 Funktionale Anforderungen

Anhand der funktionalen Anforderungen werden die gewünschten Funktionalitäten und das Verhalten des Systems beschrieben.

Klingeln senden

Das System soll einem Besucher ermöglichen, Personen innerhalb der Wohnung über ein Klingeln zu erreichen.

Der Besucher soll:

- Eine Auswahl der in der Wohnung lebenden Personen erhalten, für die er klingeln darf
- Durch Berührung des Klingelknopfes das Signal absetzen

Das System soll:

- Klingelknopf zur Verfügung stellen

- Klingelsignal an Gegenstellen weiterleiten

Klingeln signalisieren

Das System soll das Absetzen eines Klingelsignals ermöglichen. Hierbei muss gewährleistet werden, dass möglichst viele Geräte mit unterschiedlicher Ausstattung und Betriebssystem die Nachricht empfangen können. Anschließend werden die empfangenen Informationen visualisiert. Mittels einer Positionsbestimmung kann die Nachrichtensignalisierung auf in der Nähe befindliche Geräte beschränkt werden.

Das System soll:

- Position des Bewohners ermitteln
- Anhand der Position geeignetes Gerät zur Signalisierung auswählen
- Die Signalisierung auf möglichst vielen Geräten anzeigen, wenn die Position nicht bestimmt werden kann

Klingeln annehmen

Das System soll dem Bewohner die Möglichkeit geben, das Klingeln anzunehmen.

Der Bewohner soll:

- Das Klingeln annehmen können

Das System soll:

- Auf eine Bestätigung des Bewohners warten
- Nach Ablauf einer Wartezeit das Klingeln automatisch ablehnen

Klingeln ablehnen

Das System soll dem Bewohner die Möglichkeit geben, das Klingeln abzulehnen.

Der Bewohner soll:

- Das Klingeln ablehnen können

Das System soll:

- Die Ablehnung des Bewohners an die Türklingel weiterleiten

- Dem Besucher die Ablehnung visualisieren

Aufbau und Abbau einer Audio- und Videoübertragung

Nach Annahme des Klingelsignals auf einem bestimmten Gerät soll nun die spezifisch konfigurierte Übertragung zwischen diesem Gerät und der Türklingel aufgebaut werden. Nach einer Kommunikation findet der Verbindungsabbau statt. Dieser kann von jeder der beteiligten Seite durchgeführt werden und wird entweder mit dem Beenden des Gespräches oder dem Öffnen der Tür in die Wege geleitet.

Der Bewohner soll:

- Die Übertragungsarten wählen können
- Die Kommunikation beenden dürfen
- Die Tür öffnen können

Der Besucher soll:

- Die Kommunikation beenden dürfen

Das System soll:

- Eine audiovisuelle Übertragung zwischen beiden Geräten ermöglichen
- Eine Audioübertragung von beiden Geräten, Videoübertragung nur ausgehend von der Türklingel ermöglichen
- Dem Bewohner bei der Kommunikation durch die Wohnung folgen
- Die Verbindung zwischen den Geräten beenden
- Die Geräte wieder in den Ursprungszustand versetzen

Tür öffnen

Den Anwendern des Systems soll das Ansteuern der Tür ermöglicht werden. Nachdem ein Besucher geklingelt hat, soll dem Bewohner diese Möglichkeit zur Verfügung stehen. Dafür muss er das Klingeln nicht annehmen, sondern er kann die Tür direkt öffnen. Wenn einem Besucher die Berechtigung gegeben wurde, die Tür zu öffnen, soll ihm diese Funktion nach einer Authentifizierung angezeigt werden.

Der Bewohner soll:

- Die Möglichkeit erhalten, die Tür öffnen zu können

Der Besucher soll:

- Nach einer Authentifizierung die Möglichkeit erhalten, die Tür zu öffnen

Das System soll:

- Die Möglichkeit besitzen, das Türschloss anzusteuern, um die Tür zu öffnen

Videonachricht hinterlassen

Für den Fall, dass sich niemand in der Wohnung befindet, sollte das System auch die Möglichkeit bereitstellen, Nachrichten in Form einer kurzen Videobotschaft zu hinterlassen.

Der Besucher soll:

- Eine Videobotschaft für den Bewohner hinterlassen können

Das System soll:

- Einen Knopf zum Starten der Aufnahme bereitstellen
- Einen Knopf zum Beenden der Aufnahme bereitstellen
- Die aufgezeichnete Nachricht an einen persistenten Speicherplatz übertragen

Videonachricht abrufen

Dem Bewohner soll die Möglichkeit gegeben werden, Videobotschaften abzurufen, die ein Besucher für ihn hinterlassen hat.

Der Bewohner soll:

- Die Möglichkeit haben, Videobotschaften abzurufen

Das System soll:

- Einen Knopf zum Abholen der Aufnahme bereitstellen

Textuelle Nachrichten für Besucher hinterlassen

Der Bewohner soll die Option erhalten, Nachrichten für Besucher im Fall seiner Abwesenheit zurückzulassen. Hier wird zwischen Mitteilungen unterschieden, die für jeden oder nur einer bestimmten Person zugänglich sein soll. Nachrichten, die für bestimmte Personen hinterlassen wurden, können nach einer Authentifizierung an der Türklingel abgerufen werden. Die öffentlichen Mitteilungen hingegen sind für jeden zugänglich.

Der Bewohner soll:

- Eine Textnachricht für Besucher hinterlassen können
- Die Möglichkeit erhalten, die Nachricht einer oder mehrerer Personen zuzuordnen zu können

Der Besucher soll:

- Die Nachrichten abrufen können
- Nach einer Authentifizierung Nachrichten abrufen können, die speziell für ihn hinterlassen wurden

Das System soll:

- Dem Bewohner die Funktionalität bereitstellen, Nachrichten zu hinterlegen
- Dem Bewohner die Möglichkeit geben, Nachrichten bestimmten Personen zuzuordnen
- Dem Besucher die Nachrichten zu präsentieren
- Dem Besucher nach einer Authentifizierung die für ihn hinterlassenen Nachrichten anzeigen

Persistente Nachrichtenspeicherung

Dem System wird ein Dienst zur Verfügung gestellt, der es ermöglicht, Nachrichten im Netzwerk abzulegen. Dadurch können die Anwender die Nachrichten von jedem beliebigen Gerät aus abrufen. Dieser Dienst sichert die Persistenz der Daten und schützt diese vor unerlaubtem Zugriff.

Authentifizierung von Personen

Dem System wird ein Dienst zur Verfügung gestellt, der Anhand von Authentifizierungsmechanismen die Identität von Personen verifizieren kann. Dadurch können bestimmte Personen mit gesonderten Rechten ausgestattet werden, die nicht jeder Person zur Verfügung stehen sollen.

3.2.2 Nicht-Funktionale Anforderungen

Die nicht-funktionalen Anforderungen beschreiben, in welcher Qualität die genannten Funktionen umgesetzt werden sollen. Damit besitzen sie eine große Auswirkung auf den Verbrauch von Ressourcen, Entwicklung und Wartung. Zusätzlich dienen diese Anforderungen dazu, die Akzeptanz des Systems gegenüber dem Anwender zu erhöhen ([Buschmann u. a. \(2000\)](#)).

Zuverlässigkeit

Für die Akzeptanz eines System ist die Zuverlässigkeit eine Grundvoraussetzung. Dabei muss das korrekte Verhalten stets sichergestellt sein. Im Fall eines Fehlers muss zusätzlich der Übergang in einen gesicherten Zustand gewährleistet werden. Es wäre fatal, wenn das System durch einen Fehler jeder Person Zutritt zur Wohnung gewähren würde oder im umgekehrten Fall niemanden mehr in die Wohnung lässt.

Aufgrund der Zuverlässigkeit sollte die Installation selten in einen inkorrekten Zustand geraten. Sollte dies doch ein mal vorkommen, muss es dem Bewohner möglich sein, das System wieder in einen korrekten Zustand zu überführen.

Sicherheit

Eine Voraussetzung für den Einsatz dieses Systems ist die Sicherheit. In Bezug auf das System werden hier zwei Aspekte der Sicherheit betrachtet. Zum einen die Sicherheit der Daten, zum anderen die des Wohnungsumfeldes.

Die Sicherheit der abgelegten Daten, spielen eine große Rolle für die Privatsphäre der Bewohner. Hier muss mit modernen Sicherheitsrichtlinien darauf geachtet werden, dass jede Person nur auf die Daten zugreifen kann, die für diese auch bestimmt sind. Möchte ein

Bewohner zum Beispiel eine Nachricht für eine bestimmte Person an der Türklingel hinterlassen, muss sichergestellt sein, dass auch nur diese Person die Nachricht abrufen kann. Das Gleiche gilt auch im umgekehrten Fall.

Ein anderer Aspekt ist die Sicherheit des Wohnungsumfeldes. Das bedeutet, es muss anhand von modernen Authentifizierungsmechanismen und Sicherheitsrichtlinien gewährleistet sein, dass keine Person widerrechtlich Zutritt zur Wohnung erlangen kann.

Erweiterbarkeit

Die Umgebung, in die sich das System integrieren soll, ist einem ständigen Wandel ausgesetzt. Dem Anwender muss es möglich sein, Geräte in das System zu integrieren, ohne diese explizit einstellen zu müssen. Dazu gehören elektronische Geräte, die in einer Wohnung öfters durch modernere ersetzt werden. Des Weiteren sollen mobile Geräte der Besucher automatisch in das System integriert werden, wenn sich diese in der Wohnung befinden. Auf der anderen Seite muss dem Anwender das Entfernen dieser Geräte genau so leicht gemacht werden wie das Einfügen. Um dies zu gewährleisten, ist eine lose Kopplung der Geräte notwendig. Dadurch stehen die Funktionalitäten während des Hinzufügens, Entfernens oder Austauschens der Komponenten weiterhin zur Verfügung.

Zur Erweiterbarkeit gehört nicht nur der Ausbau von Funktionalitäten durch die Hardware, sondern auch die Erweiterung durch Software. Dem Entwickler wird hier durch definierte Schnittstellen ermöglicht, dem System weitere Fähigkeiten hinzuzufügen. Des Weiteren muss es möglich sein, die Zuverlässigkeit, die Sicherheit und den Komfort des Systems nachträglich zu erhöhen.

Privatsphäre

Um das System für den Benutzer möglichst komfortabel zu gestalten und mit einem großen Funktionsumfang zu versehen, müssen private Daten erhoben werden. Anhand dieser Daten kann das System Entscheidungen treffen und aus den Gewohnheiten des Bewohners lernen. Hierbei kann es sich um Positionsdaten handeln, um zu erkennen, ob sich ein Bewohner in einem privaten Bereich befindet. Dazu gehört ebenfalls die Erkennung und Speicherung von Bewegungsprofilen, damit eine stattfindende Kommunikation dem Bewohner durch die Wohnung folgen kann. Aber auch die Nachricht eines Besuchers darf nur von der Person abrufbar sein, für die diese Nachricht bestimmt ist.

Es ist also darauf zu achten, dass das System nur Daten erhebt, die es für die Gewährleistung der Funktionalitäten wirklich benötigt. Des Weiteren muss darauf geachtet werden,

dass keine unbefugte Person oder ein anderes System darauf Zugriff erhält, da sonst eine eventuelle Veröffentlichung der Daten nicht mehr verhindert werden kann.

3.3 Kontextabhängigkeit

In der Umgebung eines Smart Homes, welches sich durch den Ansatz des Ubiquitous Computing durch eine Vielzahl von Computern auszeichnet, erzeugen die zur Verfügung stehenden Dienste unzählige Informationen. Damit diese Informationsflut den Anwender nicht überfordert, ist es essentiell, den aktuellen Kontext zu erfassen, ihn auszuwerten und den Anwendungen im Smart Home zur Verfügung zu stellen.

Die Generierung einer Kontextabhängigkeit unterteilt sich in verschiedene Teilaspekte. Um Anwendungen einen Kontext zur Verfügung zu stellen, müssen diese erst einmal identifiziert werden. Wurden verschiedene Kontexte für eine Anwendung ermittelt, müssen Informationen zusammengetragen werden. Anhand derer kann dann entschieden werden, ob ein Kontext zur Zeit zutrifft.

Die Erkennung und Bekanntgabe des Kontextes wird hierbei nicht durch das System gewährleistet, sondern durch die Umgebung, in der das System integriert wird, bereitgestellt ([Ellenberg \(2010/2011\)](#)).

3.3.1 Identifizierte Kontexte

Für die hier zu konzipierende Türklingelanwendung stellen Kontexte eine erhebliche Unterstützung bei der Bereitstellung von Funktionalitäten dar. Sie bieten die Möglichkeit, dem Anwender nur die Funktionalitäten einzublenden, die er momentan benötigt. Somit wird dieser nicht mit der Auswahl unzähliger Fähigkeiten und Informationen überfordert.

Anhand der entwickelten Szenarien konnten verschiedene Kontexte identifiziert werden, die für eine Türklingelanwendung von Nutzen sind. Diese werden im Folgenden vorgestellt:

- **Bewohner ist zu Hause und verfügbar:**
In diesem Kontext befindet sich der Bewohner zu Hause und ist bereit, Besucher zu empfangen. Durch mögliche Privatsphäreinstellungen, die vom Benutzer getätigt wurden, werden der Anwendung zusätzliche Informationen übermittelt. Dadurch kann diese beispielsweise erkennen, welche Art der Kommunikation aufgebaut werden soll.
- **Bewohner ist zu Hause und nicht verfügbar:**
In diesem Kontext befindet sich der Bewohner zwar zu Hause, möchte aber nicht gestört werden. Wann dies der Fall ist wird vom Bewohner gesteuert. Dieser kann zum

Beispiel einstellen, dass der Kontext aktiv wird, wenn er auf dem Sofa einschläft oder einen Film guckt. Möglich ist aber auch, den Kontext explizit bei dem Klingeln eines Besuchers zu aktivieren.

- Bewohner ist nicht zu Hause: Dieser Kontext wird aktiviert, sobald der Bewohner die Wohnung verlässt. Hierdurch erkennt die Anwendung zum Beispiel das die Funktionalität des Klingelns im Moment nicht angeboten werden muss.

3.3.2 Informationsgewinnung

Nachdem die erforderlichen Kontexte für die Anwendung identifiziert wurden, müssen diese jetzt während der Laufzeit des Systems erkannt werden. Hier bildet die Informationsgewinnung die Basis zur Erkennung eines Kontextes.

Um der Anwendung einen Kontext bereitzustellen, werden eine Vielzahl von Informationen benötigt. Dabei spielt nicht nur die Positionsbestimmung eine Rolle. Schmidt ([Schmidt u. a. \(1999\)](#)) beschreibt in seinem Artikel, dass zur Kontexterkenkung möglichst viele Sensortechniken zum Einsatz kommen müssen. Dazu zählen optische, Bewegungs-, Lokations-, Audio-, Bio- und spezialisierte Sensoren. Die Verknüpfung der einzelnen Daten, welche von den Sensoren bereitgestellt werden, wird als Sensorfusion bezeichnet.

3.3.3 Entscheidungsfindung

Das System sammelt die Informationen, die durch die Sensoren zur Verfügung gestellt werden. Danach wertet dieses die Informationen aus und legt ihre Priorität fest. Anschließend wird anhand spezieller Muster verglichen, ob die gesammelten Informationen einem Kontext zuzuordnen sind. Um allen im Netzwerk vorhandenen Komponenten die gleichen Informationen zur Verfügung stellen zu können, sollte die Kontexterstellung durch eine zentrale Stelle ausgeführt werden. Somit können Konsistenzprobleme vermieden werden ([Voskuhl \(2010/2011\)](#)).

Am Beispiel des Kontextes "Bewohner ist zu Hause und nicht verfügbar" ist ersichtlich, dass eine große Menge an Informationen benötigt wird, um eine Entscheidung zu treffen. Hier spielt nicht nur die Position eine Rolle, sondern auch wie spät es ist, ob es hell oder dunkel ist und in welchem Zustand sich der Bewohner befindet. Dabei ist es nicht nur wichtig, welche Sensoren miteinander kombiniert werden, sondern auch, welche Informationen diese liefern. Wird beispielsweise durch den Lokationssensor ermittelt, dass der Bewohner auf dem Sofa liegt und die Biosensoren mitteilen, dass dieser schläft, würde der Kontext zutreffen. Liegt er aber auf dem Sofa, ist wach und durch einen Audiosensor wird signalisiert, dass er Musik

hört, würde dieser Kontext nicht mehr zutreffen. Hier würde sich das System für den Kontext "Bewohner ist zu Hause und verfügbar" entscheiden.

3.4 Interaktionen

Der Erfolg des Systems hängt aber nicht nur von den Funktionalitäten ab, die es dem Anwender bietet, sondern vielmehr wie er diese bedienen kann. Während die Kommunikationsfähigkeit des Menschen darauf ausgelegt ist, mit anderen Menschen zu kommunizieren, wird diese bei der Mensch-Maschinen-Kommunikation stark eingeschränkt. Das liegt vor allem an den verwendeten Eingabemedien wie Maus und Tastatur. Diese ermöglichen dem Menschen keine natürliche Kommunikation mit dem Computer und erschweren somit eine intuitive Bedienung.

Für das zu erstellende System, welches in ein Smart Home integriert werden soll, müssen also andere Bedienkonzepte verwendet werden. Denkbar wären hier Bedienkonzepte, die sich an einer natürlichen Mensch-zu-Mensch Kommunikation orientieren, beispielsweise eine Touch- oder Kamera-basierte Gestenerkennung sowie eine Sprachsteuerung. Für einen tieferen Einblick in die verschiedenen Konzepte sei hier auf die Arbeiten von ([Rahimi und Vogt, 2008](#)) und ([Witt, 2011](#)) verweisen.

Ein weiteres Kriterium für eine erfolgreiche Mensch-Maschinen-Interaktion stellt die Oberfläche des Systems dar. Um eine Funktion des Systems in Anspruch zu nehmen, sollte der Anwender sich nicht durch unzählige Fenster und Menüs klicken müssen ([Dahm \(2005\)](#)). Viel sinnvoller ist es hier, den Anwender durch den Interaktionsprozess zu führen, welches von der Maschine aber auch eine gewisse Flexibilität und natürliche Intelligenz erfordert.

Mit natürlicher Intelligenz ist in diesem Zusammenhang gemeint, dass die Maschine anhand der vorherigen Arbeitsschritte die Folgenden erahnt. Für den Fall, dass dies nicht möglich ist, muss das System dem Anwender die Möglichkeit geben, in den Prozess einzugreifen und anschließend aus dem Fehler zu lernen.

Die Abbildung 3.5 zeigt die unterschiedlichen Interaktionsgrundlagen von Mensch und Maschine. Dabei besteht der große Unterschied darin, Informationen zu interpretieren und auf diese zu reagieren. Der Mensch nutzt dafür sein natürliches Wissen, welches er sich über Jahre hinweg angeeignet hat. Die Maschine hingegen besitzt so ein Wissen nicht. Sie kann nur der einprogrammierten Logik folgen. Intuitive Reaktionen auf unvorhergesehene Situationen sind dadurch schwerer zu realisieren.

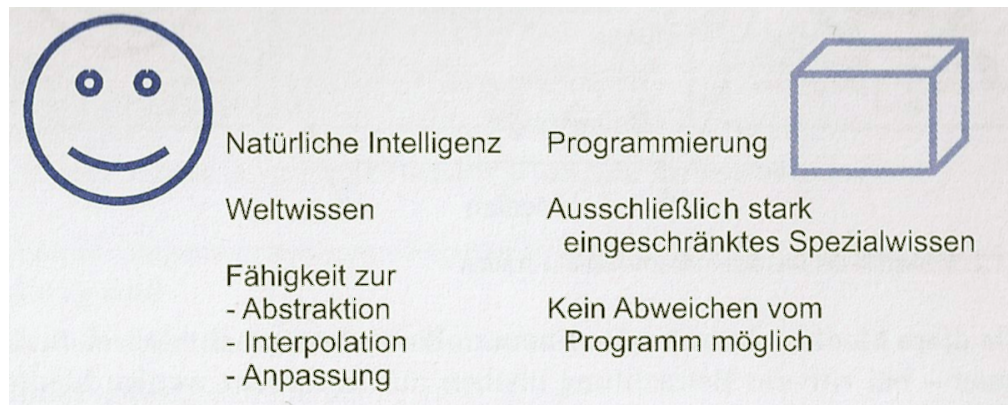


Abbildung 3.5: Unterschiede in den Grundlagen der Interaktion (Dahm (2005), S.20)

Interaktionen der Anwender

Anhand der Szenarien wurden verschiedene Interaktionen mit dem System identifiziert. Auf welche Art und Weise eine Mensch-Maschinen-Interaktion durchgeführt werden kann, hängt dabei von dem Anwender ab. Der Bewohner des Smart Homes erhält hier mehr Interaktionsmöglichkeiten als der Besucher.

Der Besucher interagiert mit dem System ausschließlich über die Türklingel. Da diese durch ein Smartphone oder gegebenenfalls durch einen Tablet-PC repräsentiert wird, bieten sich für den Besucher Touch-basierte Interaktionen an. Das bedeutet, Interaktionen wie das Klingeln oder hinterlassen einer Nachricht werden über Berührungen gesteuert.

Der Bewohner kann hingegen auf wesentlich mehr Interaktionsmöglichkeiten zurückgreifen. Denkbar sind hier Touch-basierte Steuerelemente genauso wie eine Kamera-basierte Gestenerkennung oder eine Sprachsteuerung. Ein Beispiel hierfür ist das Ablehnen eines Besuchers. Nach der Visualisierung des Besuchers sagt der Bewohner einfach „Ich bin beschäftigt“. Die Sprachsteuerung der intelligenten Wohnung erkennt den Befehl und leitet diesen an die Türklingel weiter.

Um den Umgang mit dem System zu erleichtern und den Komfort zu erhöhen, sollten diverse Interaktionen nicht mehr durch den Bewohner ausgeführt werden. Durch die Kontexterkenennung im Smart Home können automatisch Anpassungen im System der Türklingel erfolgen. Dazu gehören beispielsweise die Privatsphäreinstellung. Zusätzlich muss aber noch gewährleistet sein, dass der Bewohner diese Einstellungen auch selber vornehmen kann.

3.5 Machbarkeitsstudie

Anhand der gesammelten Information kann nun ermittelt werden, ob das System realisierbar ist. Hierfür werden bestimmte Anforderungen an Hardware und Software gestellt. Dabei steht hier nicht die mögliche Realisierung aller Funktionalitäten im Vordergrund, die in den Szenarien ermittelt wurden, sondern eine generelle Realisierbarkeit. Dazu werden die Möglichkeiten des mobilen Betriebssystems Android™ im folgenden Abschnitt kurz aufgeführt.

3.5.1 Anforderungen

Prinzipiell stellt ein modernes Smartphone, das mit dem Betriebssystem Android™ ausgestattet ist, alle Eigenschaften zur Verfügung, die benötigt werden, um dieses Projekt zu ermöglichen. Hardwareseitig stehen ein Mikrofon, zwei Kameras, ein Touchdisplay und diverse Schnittstellen für die Kommunikation mit anderen Geräten zur Verfügung.

Softwareseitig können die benötigten Funktionen durch eine Vielzahl von APIs realisiert werden. Dazu zählen die Übertragung und Speicherung von audiovisuellen Daten und die Verbindung zu anderen Komponenten.

Andere mobile Betriebssysteme stellen zum aktuellen Zeitpunkt keine Alternative zu Android™ dar. Hier sind insbesondere Windows Phone 7 und Apples iOS zu nennen. Ersteres bot zum Zeitpunkt der Implementierung keinen Zugriff auf Socket-Schnittstellen, welches essentiell für das Versenden von audiovisuellen Daten ist. Gegen Apple spricht hingegen der kostenpflichtige Developer Account und die geringe Geräteauswahl.

3.5.2 Beispielprojekte

Aufgrund der Open Source Orientierung bildet das mobile Betriebssystem Android™ eine gute Basis. Den Entwicklern wird hier eine Vielzahl von Möglichkeiten geboten. Diese baut Google mit dem neuen Projekt Android@Home weiter aus. Hierdurch soll Android™ nun auch Einzug in den Markt der Hausautomatisierung halten. Zu den ersten Projekten zählen die Ansteuerung von Leuchtmitteln und die Steuerung des Heimkinos durch das Projekt „Tungsten“. Dies ist in dem Onlineartikel von Stephen Shankland und Jan Kaden ([Shankland und Kaden \(2011\)](#))³ nachzulesen.

Daraus ist ersichtlich, dass es sich bei Android™ nicht nur um ein mobiles Betriebssystem für Handys handelt. Es ist vielmehr so Vielseitig, dass es in den verschiedensten Gebieten

³Bei der Erstellung der Arbeit standen zu diesem Themengebiet noch keine wissenschaftlichen Studien zur Verfügung.

zum Einsatz kommt, sei es auf Smartphones, Tablet-PCs oder als Steuerungskomponente für diverse Umgebungen.

3.6 Fazit

In diesem Kapitel wurden Szenarien mit unterschiedlichen Ausprägungen entwickelt und vorgestellt, die durch das hier vorgestellte System möglich werden. Auf Seiten des Bewohners wurden grundlegende Interaktionen identifiziert, wie die Annahme und das Ablehnen einer klingelnden Person. Die Präsentationsmöglichkeiten des Bewohners dem Besucher gegenüber, also ob dieser sich durch eine Live-Videoübertragung dem Besucher auf der Türklingel zeigen möchte oder zum Schutz seiner Privatsphäre nur eine Audioübertragung zulässt. Weiter wurden Interaktionen erkannt, die dem Bewohner die Möglichkeit offerieren, konkrete Personen mit gesonderten Rechten auszustatten.

Dem Besucher stehen hingegen Interaktionen wie das Klingeln für eine spezifische Person oder einer Gruppe zur Verfügung sowie das abrufen von Nachrichten. Nach einer Authentifizierung stehen weitere Interaktionen, wie das Abrufen spezieller Nachrichten oder das Öffnen der Tür, zur Verfügung.

Für die Anwender beider Seiten muss eine intuitive Steuerung entwickelt werden, die einen einfachen Umgang mit den Funktionalitäten gewährleistet. Gerade im Bereich der Wohnungsumgebung mit ihrer heterogenen Gerätelandschaft stellt dies doch einen erheblichen Aufwand dar. Eine Steuerung durch Tastatur und Maus ist für dieses System also ungeeignet, da eine natürliche Steuerung schwer zu realisieren ist. Hier bieten sich Touch-basierte Bedienkonzepte ([Rahimi und Vogt \(2008\)](#)) sowie eine Spracherkennung an.

Hinsichtlich der zu treffenden Designentscheidungen ist aber nicht nur der Funktionsumfang und die daraus resultierenden Anforderungen zu betrachten, sondern auch die vorherrschende Umgebung, in der das System zum Einsatz kommen soll. Im folgenden Kapitel werden diesbezüglich verschiedene Architekturvarianten untersucht und diskutiert.

4 Design

Im vorherigen Kapitel wurden anhand von Szenarien Anforderungen für das System entwickelt und spezifiziert. Auf Basis dieser Anforderungen soll in diesem Kapitel mittels Methoden des Software-Engineerings ein Design für das System entwickelt werden.

4.1 Architektur

Das Einsatzgebiet einer Türklingel ist typischerweise an eine bestimmte Umgebung gebunden, ein Haus oder eine Wohnung. Um dem Anwender den Funktionsumfang der beschriebenen Szenarien anbieten zu können, muss das System in unterschiedlichen Räumen und auf verschiedenen Geräten zur Verfügung stehen. Dadurch entsteht eine heterogene Gerätelandschaft, mit der das System umgehen muss.

Hieraus können mehrere Aspekte abgeleitet werden. Der Erste besagt, dass die Umgebung aus mehreren Komponenten besteht. Der Zweite sagt etwas über die Kommunikation dieser Komponenten aus. Diese müssen auf die eine oder andere Weise autonom zusammenarbeiten und bilden somit die Kernaufgabe eines Verteilten Systems ab. Der letzte Aspekt besagt etwas über die Wirkung des Systems gegenüber dem Anwender aus. Obwohl die gesamte Umgebung aus unterschiedlichen Komponenten und Systemen besteht, soll hier dem Benutzer suggeriert werden, dass er es mit einem einzigen System zu tun hat.

4.1.1 Systemüberblick

Das hier zu entwickelnde Design besteht aus unterschiedlichen Komponenten, die in das bestehende System des Living Place integriert werden sollen. Das System besteht aus der Türklingel die durch ein Smartphone an der Tür repräsentiert wird. Den einzelnen Gegenstellen, die sich innerhalb der Wohnung befinden und einem Nachrichtensystem. Die Abbildung [4.1](#) zeigt die einzelnen Komponenten des Gesamtsystems und die Kommunikationswege untereinander.

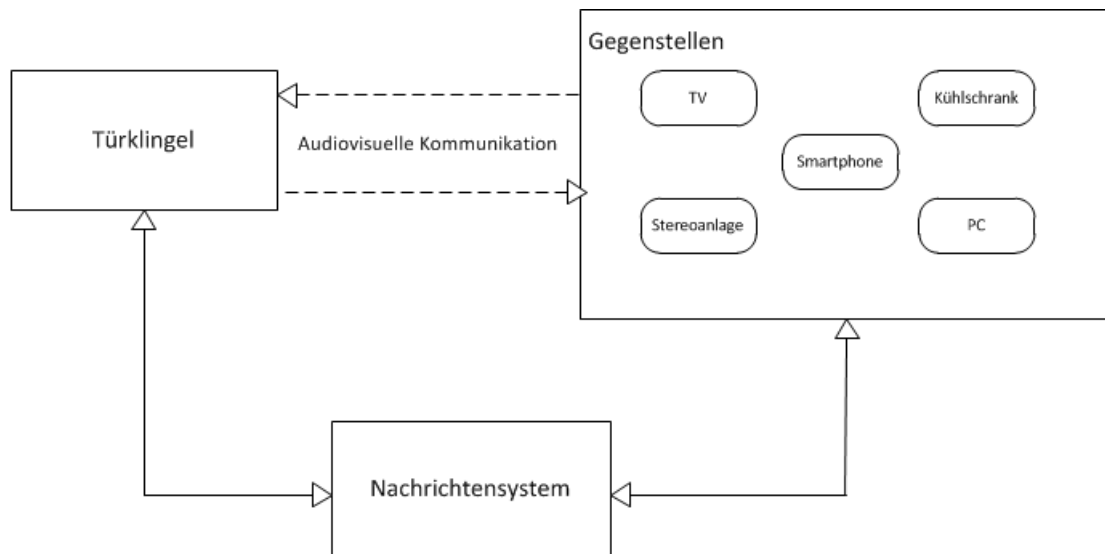


Abbildung 4.1: Systemüberblick

Hierbei gibt es zwei unterschiedliche Arten, wie die einzelnen Komponenten untereinander kommunizieren. Man unterscheidet hier die synchrone von der asynchronen Kommunikation. Die synchrone Kommunikation wird durch die audiovisuelle Verbindung zwischen der Gegenstelle und der Türklingel repräsentiert. Sie bildet ein reales Gespräch zwischen Gesprächspartnern ab. Der Austausch von Informationen und Einstellungen zwischen den Komponenten wird über ein Nachrichtensystem mittels asynchroner Nachrichten ermöglicht. Eine detaillierte Beschreibung der Kommunikationsarten folgt in Kapitel 4.2.

Dienstorientierte Architektur

Im Mittelpunkt dieses Architekturmusters stehen Dienste, die von unterschiedlichen Komponenten innerhalb eines Systems angeboten werden. Sie kapseln Funktionalitäten und Daten und machen diese über wohldefinierte Schnittstellen erreichbar. Im Vergleich zu Komponenten sind Dienste plattformunabhängig und bieten somit eine höhere Interoperabilität, welche die Realisierung Verteilter Anwendungen erleichtert. Eine stark verbreitete Form der Dienstorientierten Architekturen stellt die Service-orientierte Architektur (SOA) dar. Weitere Aspekte dieses Architekturmusters werden in (Schill und Springer (2007)) beschrieben.

Für das System ist damit nicht von Bedeutung, welches Geräte in die Umgebung integriert wird, sondern welche Dienste von diesem Gerät zur Verfügung gestellt werden. Ein Fernseher würde zum Beispiel Dienste zur Wiedergabe von Audio- und Videoinhalten zur Verfügung stellen. Ein Mikrofon hingegen nur den Dienst Audioinhalte aufzuzeichnen. Des Weiteren

können die Dienste auch miteinander kombiniert werden, um dem Anwender komplexere Funktionen zur Verfügung zu stellen. Hierbei werden zwei Arten unterschieden. Zum einen die Orchestrierung, zum anderen die Choreographie.

Die Orchestrierung hat das Ziel, komplexe Dienste anhand der Zusammensetzung einzelner einfacher Dienste zu erzeugen. Dabei präsentiert sich der orchestrierte Dienst dem Anwender gegenüber als ein einziger Dienst, indem er die verwendeten Dienste nach außen hin kapselt. Ein Beispiel hierfür ist der audiovisuelle Dienst. Dieser wirkt nach außen hin wie ein einziger Dienst, im Hintergrund greift dieser aber auf die Dienste verschiedener Geräte zu. Im Gegensatz dazu steht die Choreographie. Hier werden die verschiedenen Dienste zu einem Ablauf zusammengefasst. Es entsteht ein fester Ablauf der einzelnen Dienste. Im Gegensatz zur Orchestrierung sind hier alle verwendenden Dienste für den Anwender sichtbar ([Schill und Springer \(2007\)](#)).

Die Intelligenz der einzelnen Komponenten muss in diesem Architekturmuster sehr stark ausgeprägt sein, da die Verwaltungsaufgaben hier von keiner zentralen Komponente übernommen werden. Dadurch entstehen höhere Kosten, weil normale Komponenten wie Mikrofone oder Lautsprecher zusätzlich noch mit der entsprechenden Intelligenz und der zugehörigen Kommunikationsfähigkeit versehen werden müssen.

Allerdings bietet dieses Muster auch ein hohes Maß an Flexibilität. Durch die lose Kopplung können jederzeit Komponenten hinzugefügt, entfernt oder getauscht werden.

Zentralisierte Architektur

Laut ([Dunkel u. a. \(2008\)](#)) stellt dieses Architekturmodell das grundlegende Modell für Verteilte Systeme dar. Es teilt sich in zwei Partitionen auf, dem Anbieter eines Dienstes, also dem Server und dem Kunden, auch Client genannt.

Die Dienste, die der Server zur Verfügung stellt, kann man im Allgemeinen mit dem Erledigen einer festgelegten Aufgabe gleichsetzen. Dies könnte zum Beispiel eine Berechnung oder das Erfragen eines Datensatzes sein. Die Kommunikation zwischen dem Client und dem Server ist meist eine synchrone Kommunikation. Der Client wartet nach dem Senden einer Anfrage solange, bis er vom Server die Antwort erhält. Das Modell kann aber auch auf einer asynchronen Kommunikation beruhen, bei der der Client nach dem Senden einer Anfrage erst mal weiterarbeitet. Zu einem späterem Zeitpunkt bekommt der Client eine Benachrichtigung vom Server, dass eine Antwort vorliegt.

Für das hier zu erstellende System würde das bedeuten, der Server müsste jede Komponenten im System kennen. Dazu zählen beispielsweise die Adresse und die angebotenen Dienste eines Gerätes, wie auf [Abbildung 4.2](#) dargestellt. Durch die Zentralisierung besitzt dieses Architekturmuster den Vorteil, dass auch normale Komponenten wie Mikrofone verwendet werden können, ohne diese mit spezieller Intelligenz ausstatten zu müssen. Dadurch

können Kosten eingespart werden. Auf der anderen Seite stellt der Server aber auch einen sogenannten „single point of failure“ dar. Produziert dieser einen Fehler, ist das ganze System nicht mehr funktionsfähig. Um die Ausfallsicherheit des Systems zu erhöhen, muss hier mit Redundanzen gearbeitet werden. Weitere Eigenschaften sind unter ([Schill und Springer \(2007\)](#)) zu finden.

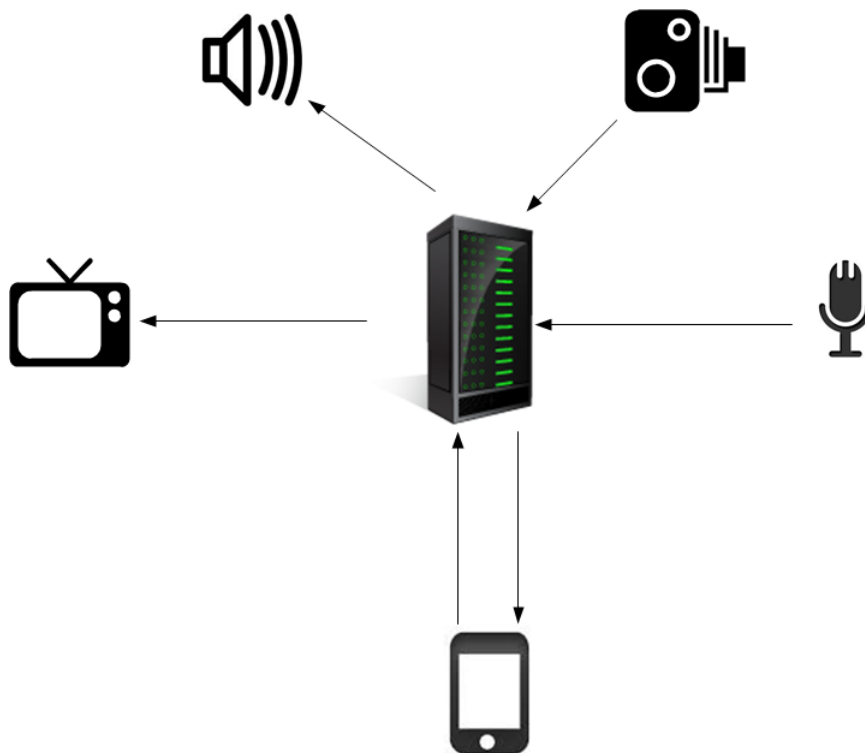


Abbildung 4.2: Beispiel Client-Server Architektur

Eine modernere Ausprägung stellt die 3-Tier Architektur [4.3](#) dar. In Folge dessen würden die Klienten die Präsentationsebene und die Steuerungsebene beinhalten, während das Modell auf unterschiedlichen Servern verteilt wird. Die Logik der Anwendung würde auf einem Applikationsserver Platz finden. Gleichzeitig werden die Daten auf einen Datenbankserver abgelegt.

Durch das Ablegen der Daten in einem Datenbanksystem wird nicht nur die Persistenz erhöht, sondern das System hat zusätzlich noch Zugriff auf Daten, die von anderen Dienst dort abgelegt wurden, zum Beispiel Adressen, Vorlieben oder Berechtigungen einer Person.

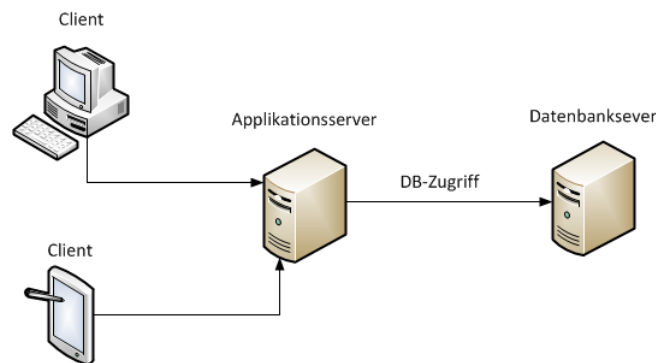


Abbildung 4.3: Beispiel 3-Tier Architektur

Peer-to-Peer Architektur

In einem Peer-to-Peer Netzwerk werden die beteiligten nicht mit einer bestimmten Rolle belegt, wie es zum Beispiel beim der Client-Server Architektur der Fall ist. Jeder Knoten in diesem Netzwerk ist ein gleichgestellter Partner, daher auch der Name Peer(engl.: Ebenbürtige). Ein Peer ist in seiner Rolle nicht beschränkt, er kann der Anbieter eines Dienstes sein oder der Kunde, der einen anderen Dienst in Anspruch nehmen möchte (Dunkel u. a. (2008), Dustdar u. a. (2003)). Peers kommunizieren unmittelbar und direkt miteinander. Dies bedeutet, es finden keine Umwege über Server statt.

Die Abbildung 4.4 soll den prinzipiellen Aufbau eines Peer-to-Peer Systems darstellen. Durch den Wegfall einer zentralen Organisation wird die Flexibilität im Vergleich zu einem Client-Server System stark erhöht. Es können zur Laufzeit Komponenten getauscht, hinzugefügt oder entfernt werden. Zudem existiert in diesem Architekturmuster kein „single point of failure“. Fällt ein Peer aus, ist das System immer noch funktionsfähig. Es muss lediglich ein neuer Peer gesucht werden, der den gleichen Dienst anbietet. Um diesen Funktionsumfang realisieren zu können, müssen die Komponenten mit zusätzlicher Intelligenz ausgestattet werden. Des Weiteren benötigen die Geräte zusätzliche Schnittstellen, um die Kommunikationsfähigkeit zu gewährleisten. Dadurch wird die Anzahl der infrage kommenden Geräte stark reduziert, da die auf dem Markt verfügbaren Gerät meist nicht die erforderlichen Eigenschaften besitzen.

Die gezeigte Darstellung soll die gegenseitige Kommunikation der Komponenten untereinander symbolisieren. Dies ist mit handelsüblichen Komponenten wie Kameras, Mikrofonen oder Lautsprechern nicht machbar. Da diese nicht mit der geeigneten Rechenleistung, Kommunikationsfähigkeit und Intelligenz ausgestattet sind. Daher muss für jede Komponente, die ins Wohnungssystem integriert werden soll, ein passendes Modul erstellt werden. Durch das

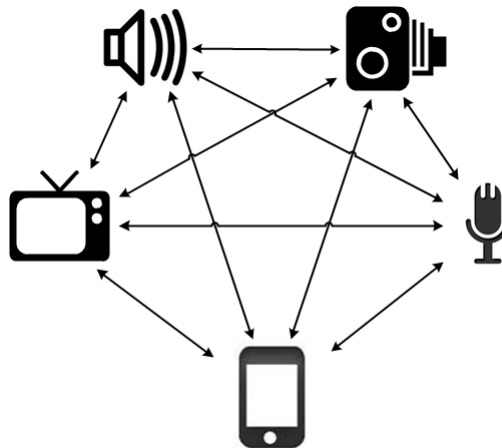


Abbildung 4.4: Beispiel Peer-to-Peer Architektur

Modul werden dem System passenden Schnittstellen zur Verfügung gestellt, um die Fähigkeiten der Geräte verwenden zu können.

Event-Driven Architektur

Das Hauptaugenmerk bei dieser Architektur liegt bei der Erzeugung und Bekanntmachung von Ereignissen. Jedes Ereignis einer Komponente wird in Form einer Nachricht über eine Middleware versendet.

Die Middleware ist dann für die Weiterleitung der Meldung zuständig. Das bedeutet, die Ereignisse werden allen Komponenten im System bekannt gemacht, sofern sie sich für diese Ereignisse registriert haben. Dadurch müssen sich die einzelnen Komponenten in diesem System nicht kennen, es muss lediglich ein Vereinbarung über das Nachrichtenformat existieren. Diese asynchrone Form der Kommunikation ermöglicht eine lose Kopplung der Komponenten ([Dunkel u. a. \(2008\)](#)).

Für das zu erstellende System bietet dieser Ansatz eine gute Basis. Dienste können über eine Nachricht den interessierten Komponenten bekannt gemacht werden, ohne detaillierte Informationen über diese zu kennen, wie [Abbildung 4.5](#) verdeutlicht.

In der Abbildung soll gezeigt werden, wie das Smartphone eine Kommunikation mit und den Komponenten in der Wohnung initiiert, sobald ein Besucher klingelt. Das Smartphone fragt dafür über die Middleware Geräte an, die in der Wohnung Audio- und Videoinhalte wiedergeben können. Die asynchronen Nachrichten werden hier durch die gestrichelten Linien

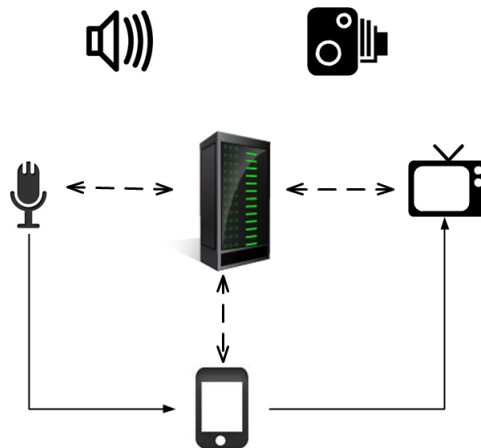


Abbildung 4.5: Beispiel Event-Driven Architektur

symbolisiert. Der Fernseher hat diesen Aufruf empfangen und teilt dem Smartphone die Verbindungseinstellungen mit. Dies fängt darauf hin mit der Übertragung der Daten an, welche hier durch die durchgezogene Linien repräsentiert wird.

Der Bewohner hat nun der Kommunikation zugestimmt und möchte Audioinhalte aus der Wohnung zum Smartphone übertragen. Dazu fragt das Mikrofon nun die Verbindungseinstellungen des Smartphones ab. Nach Erhalt der Einstellung überträgt dies die Audioinhalte aus der Wohnung zur Türklingel und eine Kommunikation zwischen Bewohner und Besucher kann stattfinden.

4.1.2 Abgrenzungen

In Abbildung 4.1 ist der grundsätzliche Systemaufbau abgebildet. Nun ist zu klären, welches Architekturmodell für das System am geeignetsten ist. Hierfür werden die Vor- und Nachteile der verschiedenen Modelle analysiert und gegeneinander abgewogen.

In einer dienstorientierten Architektur, werden Funktionalitäten durch Dienste repräsentiert. Diese befinden sich nicht zentralisiert auf einem Server, wie beim Client-Server Modell, sondern dezentralisiert auf den einzelnen Komponenten des Systems. Hierdurch können Funktionalitäten durch den Austausch, das Hinzufügen oder Entfernen von Komponenten, erweitert oder dezimiert werden. Zusätzlich kann hier eine Diensteredundanz erzeugt werden, da mehrere Geräte die selben Dienste zur Verfügung stellen können. Des Weiteren entsteht durch die Dezentralisierung kein „single point of failure“, wie dies in zentralisierten Architekturen der Fall ist.

Der Nachteil dieses Modells besteht in ihrer aufwendigen Kommunikation. Bevor eine Komponente einen Dienst in Anspruch nehmen kann, muss diese erst einmal prüfen, ob dieser Dienst im System existiert und unter welcher Adresse dieser erreichbar ist. Die Komponenten müssen also Algorithmen besitzen, um andere Komponenten und ihre Dienste zu ermitteln und zu verwalten. Um die Kommunikation zu vereinfachen, wäre der Einsatz eines Verzeichnisdienstes denkbar. Hierdurch wird allerdings wieder ein Art Zentralisierung erreicht, was bei diesem Modell nicht wünschenswert ist.

Beim Client-Server Modell sind zwei Variationen möglich. Zum einen kann die Türklingel an sich als Server fungieren, welches aus sicherheitstechnischen Gründen aber nicht wünschenswert ist. Zum anderen kann ein dedizierter Server eingesetzt werden. Das erste Beispiel ist nicht sinnvoll, da der Server alle Daten enthält, die das System benötigt. Dazu zählen eine Liste der erreichbaren Clients, Konfigurationen des Netzwerks und hinterlassene Nachrichten. Durch die Positionierung des Servers an der Tür wäre das System ein leichtes Ziel für Angriffe und Vandalismus. Aus diesem Grund muss die Türklingel ebenfalls als Client fungieren und im Netzwerk muss sich ein dedizierter Server befinden, der die Aufgaben übernimmt.

Dieses Modell führt zu einer höheren Komplexität und Performanzverlust im Vergleich zum Peer-to-Peer Modell. Da die Türklingel in diesem Fall keine Kenntnis über Eigenschaften des Netzwerkes besitzt, handelt der Server die Kommunikation und den Datenaustausch zwischen den Clients ab. Zusätzlich unterhält er eine Liste aller verfügbaren Clients und muss diese auf dem aktuellsten Stand halten. Durch die zentralisierte Abhandlung der Funktionalitäten entsteht ein „single point of failure“. Dieser kann nur durch eine Redundanz, wie einem zusätzlichen Server, ausgeglichen werden.

Für das Modell spricht die einfach zu realisierende Kommunikation, da alle Funktionalitäten an einem Punkt vereint sind.

Das Peer-to-Peer Modell hingegen bieten die höherer Performanz, da die Daten nicht den Umweg über einen Server gehen müssen, sondern direkt zwischen Client und Türklingel übertragen werden. Allerdings hängt die Performanz auch von den eingesetzten Geräten und ihrer Leistungsfähigkeit ab. Des Weiteren müssen keine Listen über verfügbare Clients geführt werden. Dies verringert die Komplexität des Systems. Informationen und Signalisierung können über einen Broadcast im Netzwerk bekannt gemacht werden.

Der dadurch entstehende Netzwerkverkehr ist zu vernachlässigen, da es sich hier nur um eine geringe Nachrichtenanzahl und -größe handelt. Gegen dieses Modell spricht die benötigte Intelligenz der Komponenten, die bisher in Standardkomponenten, wie beispielsweise Lautsprechern, nicht vorhanden ist. Diese Intelligenz müsste für jede Komponente nachgerüstet werden. Insofern relativiert sich die vorher genannte Komplexitätsverringerng wieder.

Die Event-Driven Architektur stellt hier das geeignetste Modell zur Realisierung dar. Da alle Komponenten erst durch Ereignisse aktiv werden und dadurch Prozesse im System in Gang gesetzt werden. Benötigt eine Komponente einen Dienst, wird dieser über das Verschicken

einer asynchrone Nachricht angefragt. Reagiert eine Komponente auf diese Nachricht, wird eine Verbindung aufgebaut. Auf die Art entsteht eine lose Kopplung. Dies bietet den Vorteil, dass sich die im System vorhandenen Geräte nicht kennen müssen um ihre Dienste miteinander zu kombinieren. Des Weiteren kommt das Modell auch im Living Place zum Einsatz. Durch die Anwendung dieses Modells kann das hier zu realisierende System nahtlos in die Umgebung integriert werden.

Der Nachteil dieses Modells stellt die Kommunikation über die Middleware dar. Fällt diese aus, können die Komponenten keine Anfragen mehr an andere schicken. Um einen Ausfall entgegenzuwirken, muss für entsprechende Redundanzen gesorgt werden.

4.1.3 Komponentenarchitektur

Die Umgebung unterteilt sich in mehrere Komponenten, die den Anwendern den Umgang mit dem System ermöglichen. Der Austausch von Nachrichten zwischen den Geräten wird durch eine Middleware verwaltet. Die Anwendungskomponenten basieren auf dem Model-View-Controller Muster, somit kann die Logik Betriebssystemübergreifend verwendet werden.

4.1.4 Model-View-Controller

Das Model-View-Controller Konzept, kurz MVC, wurde 1979 entwickelt ([Reenskaug \(1979\)](#)). Ziel des Musters ist es, eine interaktive Anwendung in drei Komponenten zu unterteilen. Das Model beinhaltet die Kernfunktionalitäten, die View stellt die Anzeige für den Anwender dar und der Controller ist für die Verarbeitung der Benutzereingaben zuständig. Detailliertere Informationen sind in den folgenden Werken zu finden ([Buschmann u. a. \(2000\)](#), [Dustdar u. a. \(2003\)](#)).

Durch die strikte Trennung von Model und View wird es ermöglicht, das Modell der Türklin- gel ohne große Veränderungen auf die Gegenstelle zu portieren. Hier sind dann lediglich noch Anpassungen der GUI von Nöten. Der grundsätzliche Aufbau wird in der folgenden Abbildung [4.6](#)¹ dargestellt.

Model

Das Model repräsentiert die Daten und die Kernfunktionalitäten der Anwendung. Es werden Schnittstellen für die Steuerungskomponente und der Ansicht zur Verfügung gestellt. Das Model ist unabhängig von den weiteren Komponenten des MVC

¹Es besteht die Möglichkeit einer Erweiterung mittels Observer-Pattern ([Buschmann u. a. \(2000\)](#))

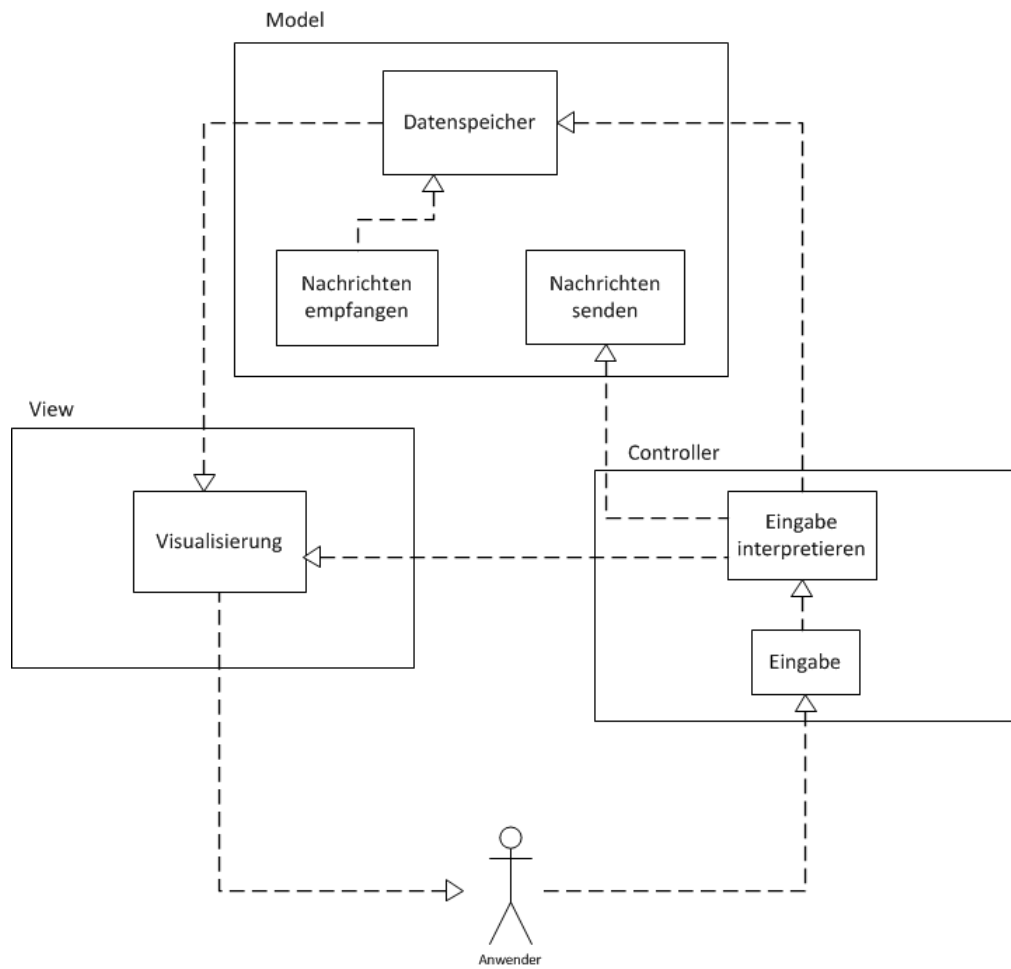


Abbildung 4.6: Komponentenübersicht der MVC Architektur

Das Model übernimmt die Bearbeitung der Daten und beinhaltet die Programmlogik. Das bedeutet, die Komponente ist für das Senden und Empfangen asynchroner Nachrichten zuständig, wie sie auftreten, wenn Audio- und Videodaten versendet oder empfangen werden müssen. Die Signalisierung eines klingenden Besuchers und das Empfangen von Einstellungen. Auch das Senden und Empfangen von Daten und dessen Verarbeitung gehört zu den Aufgaben des Models. Daraus ergeben sich diverse Schnittstellen, die der Präsentation und der Steuerung zur Verfügung gestellt werden müssen. Hat ein Verbindungsaufbau zwischen einer Gegenstelle und der Klingel stattgefunden, empfängt die Klingel permanent Daten. Nach dem Empfang der Daten werden diese der Präsentationsebene bereitgestellt. Über eine entsprechende Schnittstelle kann nun auf diese Daten zugegriffen und der Person vor der Tür präsentiert werden.

Andersherum verhält es sich bei einer Interaktion die, von dem Besucher ausgeht. Möchte eine Person vor der Tür dem Bewohner eine Nachricht hinterlassen, weil dieser gerade nicht zur Verfügung steht, muss die Steuerung die entsprechende Interaktion interpretieren. In Folge dessen wird die Nachrichtenaufzeichnung über eine Schnittstelle zum Model veranlasst. Das Model initialisiert dementsprechend die erforderliche Hardware wie Kamera und Mikrofon und startet die Aufnahme.

Da diese Nachrichten persönliche und vertrauliche Daten enthalten können, spielt dessen Persistenz und Zugriffsschutz eine wichtige Rolle. Zur Gewährleistung der Persistenz wäre es sinnvoll, die Nachrichten nach ihrer Aufzeichnung auf einem im Netzwerk vorhandenen Speicher abzulegen und diesen durch Berechtigungen zu schützen. Damit kann sichergestellt werden, dass keine unberechtigte Person Zugriff auf die Daten erhält. Diese Aspekte werden hier nicht weiter betrachtet, da der Fokus der Arbeit auf der Interaktion mit einer Gegensprechanlage liegt.

View

Die View präsentiert dem Anwender die Informationen des Models. Dafür ruft die View die entsprechenden Daten über die Schnittstellen ab und zeigt diese an. Zu jeder Ansicht gehört eine Steuerungskomponente, welche die Eingabe des Anwenders entgegennimmt und interpretiert.

Die View oder auch Präsentation, bildet nicht nur die Schnittstelle zur Programmfunktionalität ab, sondern sie bildet auch die Schnittstelle zwischen dem Anwender und dem System. Um dem Anwender den Umgang mit dem System zu erleichtern wird eine intuitive Bedienung benötigt. Hierzu gehört beispielsweise, dass nur Funktionalitäten angeboten werden, die derzeit auch durchführbar sind. Ein Beispiel hierfür ist, dem Besucher nur dann das Klingeln zu ermöglichen, wenn der Bewohner zu Hause ist .

Dafür besitzt die Präsentation Zugriff auf Schnittstellen des Modells und der Steuerung. Dabei dienen die Schnittstellen zum Modell der visuellen Repräsentation der Daten, also der Anzeige empfangener Daten und der Anpassung der Oberfläche an den aktuellen Zustand des Systems. Die Schnittstellen zur Steuerung dienen der Erkennung der gewünschten Interaktion des Anwenders. Durch sie wird also die gewünschte Interaktion des Anwenders interpretiert und die entsprechende Logik im Modell aufgerufen.

Controller

Die Steuerung nimmt die Benutzerinteraktionen der View entgegen, wertet diese aus und reagiert dementsprechend. Sie kann eine oder mehrere Präsentationen verwalten. Zu jeder Ansicht existiert ein Modell in dem die Daten hinterlegt sind. Durch die Trennung von Präsentation und Steuerung ist es jederzeit möglich, die Funktionalitäten des Systems zu erweitern, ohne dabei die Oberfläche anpassen zu müssen.

Die Steuerung empfängt die vom Benutzer gewünschte Interaktion. Dies kann, je nach Gerät, durch unterschiedliche Eingabemöglichkeiten erfolgen. Außerhalb der Wohnung, also an der Türklingel, die auf einem Smartphone mit dem Betriebssystem Android™ basiert, besteht das Eingabegerät aus einem Touchdisplay. In der Wohnung hingegen kommen je nach verwendeter Gegenstelle mehrere Eingabegeräte zum Einsatz. Das können Touchdisplays, Tastaturen, Mäuse oder Fernbedienungen sein. Daher ist es sinnvoll, die empfangenen Daten aus der Präsentation unabhängig vom Eingabegerät zu gestalten. Aus diesem Grund besteht die Oberfläche aus Knöpfen, anhand derer der Anwender das System steuern kann. Knöpfe können von jedem Eingabegerät aus angesteuert werden. Würde die Steuerung hingegen mit einer Gestenerkennung realisiert werden, würde die Auswahl von möglichen Geräten, die für dieses System verwendet werden können, eingeschränkt werden. Durch Betätigung eines Knopfes auf der Oberfläche wird eine Schnittstelle in der Steuerung aufgerufen, durch welche die gewünschte Interaktion erkannt wird.

4.2 Kommunikation

Die Kommunikation ist nicht nur ein elementarer Mechanismus bei der Verständigung der Menschen untereinander, sondern auch bei technischen Systemen. Durch den Austausch von Nachrichten wird es Instanzen verteilter Systeme ermöglicht, miteinander zu interagieren und zu kooperieren. Hierbei gibt es zwei grundsätzlich verschiedene Arten des Nachrichtenaustausches, die synchrone und die asynchrone.

4.2.1 Synchroner Kommunikation

Das System besteht aus unterschiedlichen Kommunikationswegen. Die Synchroner Kommunikation ist hierbei eine Abbild der menschlichen Kommunikation, die auf dem Request-Response Prinzip basiert. Ein Kommunikationspartner stellt dem Anderen eine Anfrage (Request) und wartet daraufhin solange, bis eine Antwort (Response) von diesem zurück kommt. Eine direkte Kommunikation zwischen Gesprächspartnern ist also stets synchron.

Eine synchrone Kommunikation findet zwischen dem Datenübertragungsdienst der Türklingel und dem Client statt, sobald die Funktionalität der audiovisuellen Kommunikation in Anspruch genommen wird. Nachdem der Bewohner durch einer Interaktion an einem Gerät in der Wohnung signalisiert hat, dass er mit der Person vor der Tür in Verbindung treten möchte, wird die Verbindung zur Türklingel hergestellt. Anschließend werden Audio- und Videodaten zwischen dem Client und der Türklingel ausgetauscht.

Die hier entstehende Kommunikation ist einer natürlichen Mensch zu Mensch Kommunikation nachempfunden. Die Basis dieser Kommunikation bilden nicht nur Wörter. Gestik, Mimik, Tonfall, Behührungen und die Beziehung der Menschen zueinander spielen ebenso eine Rolle (([Dahm, 2005](#)), ([Watzlawick, 2007](#))). Die Übertragung dieser Faktoren, kann durch die audiovisuelle Übermittlung der Daten synthetisiert werden.

4.2.2 Asynchrone Kommunikation

Eine asynchrone Form der Kommunikation zwischen Menschen entsteht, wenn diese zu unterschiedlichen Zeitpunkten stattfindet. Um diese Art der Kommunikation zu realisieren, bedienen sich Menschen seit jeher technischer Hilfsmittel, wie Briefe oder Anrufbeantworter. Im Gegensatz zum Request-Response Prinzip der synchronen Kommunikation wartet der Kommunikationsteilnehmer hier nicht auf eine Antwort, sondern nutzt die Zeit bis zum Eintreffen der Antwort zum erledigen anderer Aufgaben. Gerade im technischen Bereich bietet das einige Vorteile hinsichtlich der Parallelisierbarkeit und Entkopplung der Komponenten ([Dunkel u. a. \(2008\)](#)).

Alle Interaktionen zwischen den Geräten und der Türklingel, außer der Direkten zwischen Bewohner und Besucher, können über asynchrone Nachrichten abgehandelt werden. Dazu werden über das Publisher-Subscriber Prinzip Nachrichten ausgetauscht. Clientseitig kann das zum einen das Abfragen und Senden von Einstellungen sein, zum anderen das Abholen hinterlassener Nachrichten. Dafür muss in der Umgebung ein Dienst zur Verfügung stehen, der diese Meldungen empfangen und weiterleiten kann.

Damit jedes Gerät dieses Systems die Nachrichten auch erkennen und verarbeiten kann, müssen konkrete Nachrichten und dessen Format definiert werden. Hierfür eignet sich das

JSON Format. Da dies einen sehr geringen Overhead im Vergleich zu XML mit sich bringt und für jede beliebige Programmiersprache geeignete Bibliotheken existieren, die eine Verarbeitung erleichtern. Zusätzlich wird JSON schon von anderen Diensten im Living Place Hamburg verwendet.

Definierte Nachrichten

Die Nachrichten, die nun vorgestellt werden, dienen den Komponenten des Systems zur Erkennung der Interaktionen. Dabei ist jede Meldung einer bestimmten Interaktion zugeordnet. Sie bestehen aus mindestens zwei Feldern. Dazu zählt das "ID"-Feld, anhand der das System eine zutreffende Nachricht erkennt und das "Content"-Feld, in der die gewünschte Interaktion bekannt gemacht wird. Je nach Interaktion können weitere Felder in der Nachricht enthalten sein, die weitere spezifische Eigenschaften bekannt machen.

Nachrichten von der Türklingel:

- Ring: {"ID": "DoorBell", "Content": "Ring", "IP": "...", "Port": "...}
Hier wird den Gegenstellen ein Klingeln signalisiert. Die Nachricht enthält die IP Adresse und den Port der Türklingel, um einen Verbindungsaufbau initialisieren zu können.
- Abort Connection: {"ID": "DoorBell", "Content": "Abort"}
In dieser Mitteilung wird den Gegenstellen die Beendigung der aktuellen Kommunikation mitgeteilt.
- Current Settings: {"ID": "DoorBell", "Content": "current Preferences", ... }
Diese Nachricht enthält alle im System aktuell angewendeten Einstellungen. Sie wird verschickt, wenn eine Gegenstelle die Einstellungen explizit anfordert. Hier sind Informationen enthalten wie IP und Ports der Türklingel und des ActiveMQ, ob Videodaten übertragen werden sollen und ob sich der Bewohner zu Hause befindet.

Nachrichten zur Türklingel:

- Connect to Bell: {"ID": "DoorBell", "Content": "Connect", "Udpport": "...", "Rtpport": "...", "Rtcpport": "...}
Diese Nachricht wird an die Türklingel versendet, wenn der Bewohner mit dem Besucher kommunizieren möchte. Sie enthält den UDP Port zum Empfangen von Bilddaten und die RTP Ports zum Empfangen der Audiodaten.
- Stop Connection: {"ID": "DoorBell", "Content": "Stop"}
Hiermit wird das Beenden der Kommunikation zwischen den Geräten bekannt gemacht. Diese Nachricht hat die selbe Funktion wie die Nachricht "Abort Connection". Allerdings wird hier dem System bekannt gemacht, dass der Bewohner die Verbindung beendet hat. Dies ist für spätere Funktionen wie das Öffnen der Tür wichtig.

- Not Available: {"ID": "DoorBell", "Content": "not Available" }
Möchte der Bewohner zur Zeit nicht gestört werden, wird durch diese Nachricht auf der Klingel eine Interaktion gestartet. Hierbei wird dem Besucher mitgeteilt, dass momentan niemand zur Verfügung steht.
- Set Settings: {"ID": "DoorBell", "Content": "Preference", "key": "...", "value": "... }
Bei dieser Nachricht werden der Klingel Einstellungen übermittelt. Hierbei kann es sich um eine oder mehrere Einstellungen handeln. Voraussetzung hierfür ist, dass der "Key" dem System bekannt ist, ansonsten wird die Einstellung abgelehnt. Gesendet werden diese Einstellungen von einer Gegenstelle, wenn der Anwender Änderungen an den Einstellungen vornimmt. Andere Systeme der Umgebung, wie die Kontextererkennung beispielsweise, haben auch die Möglichkeit, Einstellungen an die Türklingel zu übermitteln.
- Load Messages: {"ID": "DoorBell", "Content": "available Messages", "IP": "...", "Port": "... }
Sollten Nachrichten auf der Klingel hinterlassen worden sein, können sie mit Hilfe dieser Nachricht abgerufen werden. Der Türklingel wird anhand der IP und des Ports mitgeteilt, wohin die Nachrichten transferiert werden sollen.

Nachrichten zwischen Gegenstellen:

- Client react: { "ID": "DoorBell", "Content": "Client react" }
Hiermit wird den Gegenstellen signalisiert, dass der Bewohner das Klingeln an einem Gerät des Systems entgegen genommen hat. Somit können sie wieder in ihren Ausgangszustand zurück kehren.

Definierte Keys

Um Einstellungen vornehmen oder auslesen zu können, müssen Keys definiert werden. Anhand dieser Keys wird es dem System möglich, die übertragenen Werte der entsprechenden Einstellung zuzuordnen. Für die Gewährleistung der Funktionalitäten müssen diverse Keys definiert werden, wie beispielsweise für die Zuordnung von Ports und IPs. Eine Auflistung aller vom System verwendeten Keys würde hier allerdings zu weit führen, daher werden in der folgenden Tabelle 4.1 nur einige exemplarisch dargestellt.

4.3 Dienste

Für die Erzeugung der Funktionalitäten des Systems werden verschiedene Dienste benötigt. Essentiell sind hier Dienste zur Audio- und Videoübertragung. Der in diesen Beispielen gern genannte Kühlschrank mit Display kann beispielsweise einen Dienst zur Videowiedergabe

Key	Datentyp	Beschreibung
at_home	boolean	Dieser Key macht dem System bekannt, ob der Bewohner zu Hause ist
video_on	boolean	Hiermit wird angegeben, ob Videodaten übertragen werden sollen
video_size	String	Definiert die zu übertragende Videogröße, zum Beispiel "640x480"
video_quality	int	Dieser Key definiert die gewählte Qualität der Videoübertragung

Tabelle 4.1: Auszug der Einstellungsmöglichkeiten

bereitstellen. Ein Fernseher stellt hingegen Dienste für Audio- und Videowiedergabe zur Verfügung. Zusätzlich werden Geräte benötigt, die Dienste für die Aufnahme von Audio- und Videoinhalten bereitstellen. Hierzu können simple Geräte wie Webkameras verwendet werden oder auch einzelne Komponenten aus komplexen Geräten wie Tablet-PC oder Smartphones.

Ein Kontaktdienst stellt dem System Informationen bekannter Personen des Bewohners zur Verfügung. Dazu werden nicht nur Adressdaten gesammelt, sondern auch Vorlieben der Person und dessen Berechtigungen. Dafür arbeitet dieser eng mit dem Authentifizierungsdienst zusammen. Da dieser Dienst ohne den Kontaktdienst nicht funktionsfähig ist, ist es sinnvoll ihn in den Kontaktdienst zu integrieren. Hierdurch wird zusätzlich die Komplexität des Systems verringert.

Die an der Klingel aufgenommenen Daten werden also an den Kontaktdienst weitergeleitet, der diese dann durch die Authentifizierungsfunktionalität verifiziert. Die entsprechenden Berechtigungen werden anschließend an die Tür zurückgegeben. Hierbei kann es sich zum Beispiel um eine Berechtigung zum Öffnen der Tür handeln.

Des Weiteren wird ein Ortungsdienst benötigt, der dem System die aktuelle Position des Bewohners mitteilen kann. Die Vorteile einer solchen Funktion liegen zum einen darin, dem Bewohner durch ein in der Nähe befindliches Gerät zu informieren. Zum anderen kann er sich während der Kommunikation mit dem Besucher frei in der Wohnung bewegen. Bewegt sich der Bewohner von dem aktuellen Gerät weg und kommt in die Nähe eines neuen Gerätes, welches die erforderlichen Dienste zur Verfügung stellt, können diese in Anspruch genommen werden.

Eine weitere Funktionalität der Türklingel soll das Hinterlassen von Nachrichten darstellen. Durch die Positionierung der Klingel außerhalb des Wohnbereichs wird ein Angriffspunkt in das System erzeugt. Aus diesem Grund muss ein Dienst zur Verfügung stehen, der die generierten Daten auf einen Netzwerkspeicher transferiert, damit bei einem eventuellen Diebstahl keine Daten verloren gehen oder in falsche Hände gelangen.

Die in einer Wohnung zusätzlich installierten Sensoren und Aktoren bieten dem System

weiter Dienste an, womit der Funktionsumfang erweitert wird. Hierbei werden Informationen über die Position und den aktuellen Zustand des Bewohners gesammelt sowie Daten zur aktuellen Helligkeit des Tageslichtes, Uhrzeit oder Temperatur. Des Weiteren stehen dem System Aktoren zur Ansteuerung des Türschlosses oder der Lichtanlage zur Verfügung. Diese Ansammlung von Sensoren und Aktoren beschreibt Voskuhl als Sensorwolke (Voskuhl (2009)).

Ziel soll die nahtlose Integration in ein bestehendes System sein. Durch die Integration in das System und der bestehenden Dienstvielfalt entstehen Synergieeffekte, die dem Bewohner ein hohes Maß an Funktionsvielfalt generieren und ihn bei den alltäglichen Aufgaben unterstützen.

Gegenstand dieser Arbeit ist es, die Realisierbarkeit der Szenarien anhand der Implementierung einer Gegensprechanlage zu evaluieren. Für diese Funktionalität werden einige Dienste wie zum Beispiel der Authentifizierungsdienst, Ortungsdienst und die Inanspruchnahme von Sensordaten und Aktoren nicht benötigt und sind daher nicht Bestandteil dieser Arbeit. Dienste zur Audio- und Videoübertragung hingegen sind essentiell für eine Gegensprechanlage und müssen somit in die Architektur mit einbezogen werden. Im Hinblick auf die zukünftige Weiterentwicklung ist aber auch darauf zu achten, das System möglichst offen zu gestalten, um die oben genannten Dienste in Anspruch nehmen zu können.

Exklusive Zugriffe

Bei einer Architektur, die auf der Inanspruchnahme unterschiedlicher Dienste von unterschiedlichen Geräten beruht, ist die Zugriffscharakteristika zu betrachten. Hierbei ist zu beachten, dass einige Dienste von mehreren gleichzeitig in Anspruch genommen werden können, während andere ihren Dienst nur für einen einzigen zur Verfügung stellen können. Als Beispiel können hier zwei Dienste herangezogen werden. Auf der einen Seite der Dienst zur Audioaufnahme. Dieser kann seine aufgenommenen Audiodaten mehreren Interessenten im Netzwerk gleichzeitig zur Verfügung stellen. Auf der anderen Seiten haben wir den Dienst der Audiowiedergabe, der beispielsweise durch einen Fernseher zur Verfügung gestellt wird. Dieser kann nur durch einen Interessenten zur Zeit genutzt werden, da sich bei der Wiedergabe mehrerer Audioquellen diese überlagern und sie der Anwender nicht mehr versteht. Daher muss dem System bekannt sein, welche Dienste zur Zeit aktiv sind und nur exklusiv von einem Interessenten genutzt werden können.

Bei dem Start einer Interaktion muss das System wissen, welches dieser Geräte zum aktuellen Zeitpunkt seine Dienste, zum Beispiel das Anzeigen des Live-Videos, zur Verfügung stellen kann. Dafür müssen die einzelnen Komponenten dem System mitteilen, dass sie zur Zeit eine Aufgabe bearbeiten und deshalb nicht zur Verfügung stehen. Denkbar wäre bei dieser Umsetzung auch, dass die Komponenten nicht nur bekannt geben, dass sie eine Aufgabe

bearbeiten, sondern auch welche das ist. Somit wird dem System die Möglichkeit gegeben, zu entscheiden ob eine gerade ausgeführte Aufgabe auch unterbrochen und zu einem späteren Zeitpunkt fortgesetzt werden kann. Als Beispiel einer unterbrechbaren Aufgabe kann das Zeigen eines Spielfilmes auf dem Fernseher herangezogen werden. Im Fall eines Klingelns wird dies auf dem Fernseher visualisiert. Möchte der Bewohner mit der Person vor der Tür sprechen, pausiert die Wiedergabe des Films. Während der Konversation wird dieser dann im Hintergrund aufgezeichnet und kann somit zu einem späteren Zeitpunkt wieder abgespielt werden. Falls in der Wohnung nicht so viele Komponenten zur Verfügung stehen, die zur Darstellung der Klingelsignalisierung geeignet sind, ermöglicht diese Erweiterung die Visualisierung des Klingelns, obwohl alle Komponenten eine Aufgabe bearbeiten.

4.4 Fazit

Hinsichtlich der gewählten Architektur und der identifizierten Dienste zeichnen sich erforderliche Komponenten und ihre Kommunikation untereinander ab. Das dargestellte Komponentendiagramm [4.7](#) stellt diese dar.

Das Diagramm stellt die benötigten Komponenten für eine Einfamilienhaus dar, in dem nur eine Türklingel Verwendung findet. Denbar wäre aber auch eine Realisierung in einem Mehrfamilienhaus durch eine Anpassung der Komponenten. Hierbei würde das Diagramm um mehrere Türklingeln, Ortungsdienste und Kontaktdienste erweitert. Durch das vorher definierte Nachrichtenformat und dessen Inhalt kann das System ohne Probleme erweitert werden. Eine Anpassung des gewählten Designs ist nicht erforderlich.

Die Komponenten, Türklingel und Gegenstellen, bilden die Benutzerschnittstellen ab. Bei bestimmten Geräten wie beispielsweise einem Fernseher beinhaltet dies nicht nur die dargestellte Komponente einer Gegenstelle, sondern auch einen audiovisuellen Dienst. Audiovisuelle Dienste stellen in der Wohnung eingebaute Kameras und Mikrofone dar. Durch den Kontaktdienst werden personenbezogene Daten gespeichert und für die Türklingel und die Gegenstellen bereitgestellt. Zusätzlich beinhaltet dieser einen Authentifizierungsfunktionalität.

Das folgende Sequenzdiagramm [4.8](#) zeigt einen Verbindungsaufbau sowie dessen Abbau. Hieran soll nochmals die Kommunikation der Komponenten untereinander verdeutlicht werden.

Der Verbindungsaufbau wird durch das Klingeln eines Besucher in Gang gesetzt. Da die Türklingel durch den Ortungsdienst ständig mitgeteilt bekommt, wo sich der Bewohner zur Zeit aufhält, kann diese Information in die asynchrone Dienstanfrage eingebaut werden. Eine Gegenstelle, die sich in der Nähe des Bewohners befindet, signalisiert diesem das Klingeln anhand einer Visualisierung. Daraufhin nimmt der Bewohner die Kommunikation an und die

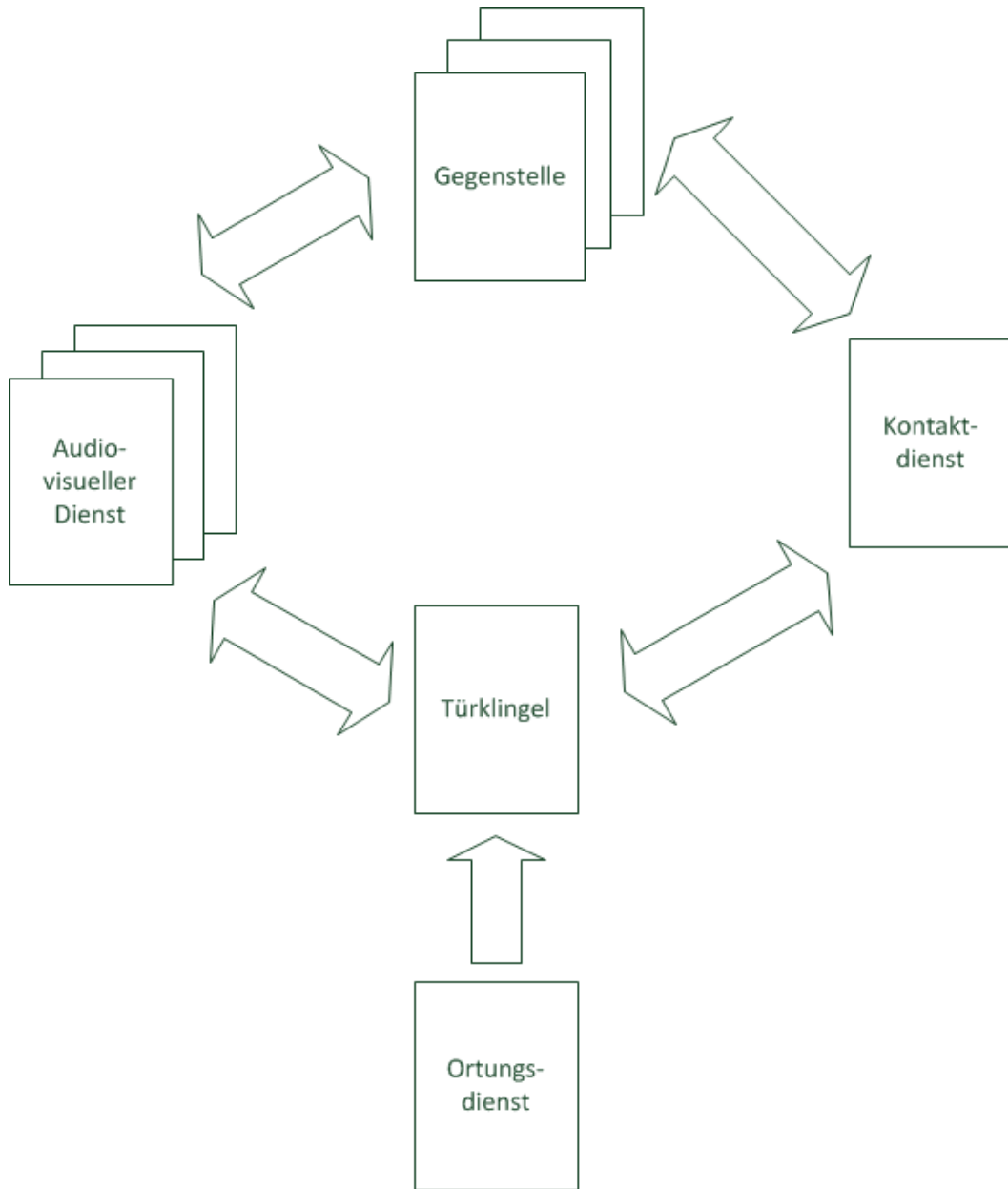


Abbildung 4.7: Komponentendiagramm

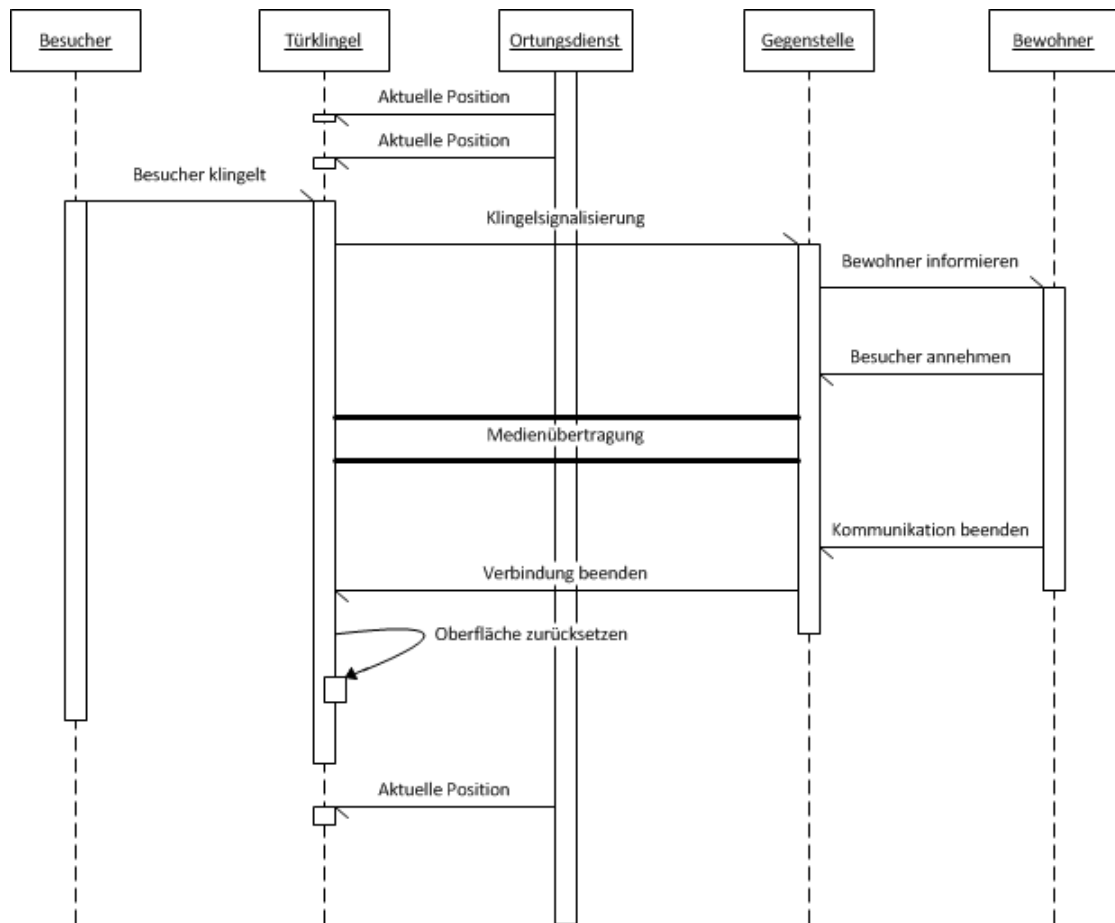


Abbildung 4.8: Sequenzdiagramm: Verbindungsaufbau und Verbindungsabbau

Medienübertragung wird angezeigt.

Zu einem späteren Zeitpunkt beendet der Bewohner die Kommunikation und die Gegenstelle meldet der Türklingel die Beendigung der Medienübertragung. Daraufhin setzt die Türklingel ihr Oberfläche zurück und ist nun für eine neue Interaktion bereit.

5 Realisierung

Das im Jahr 2009 eingerichtete Living Place (2.3) an der Hochschule für Angewandte Wissenschaften in Hamburg stellt eine Umgebung zur Verfügung, in der Studenten den Bereich des Ubiquitous Computing erforschen können. Hierdurch werden Anwendungen entwickelt, die dem Menschen den Alltag erleichtern sollen, ganz nach den Visionen von Mark Weiser (Weiser (1991)). Dazu gehört auch das hier vorgestellte System.

Auf der Basis des im vorherigen Kapitel entwickelten Designs steht nun die Realisierung und Implementierung einer auf der Android™-basierten Türklingel mit der Funktionalität einer Gegensprechanlage im Fokus. Grundlage hierfür ist die Verwendung des Android™-Betriebssystems in der Version 2.3 für den Betrieb auf einer Handyplattform und 3.0 für die Tabletversion.

5.1 Technische Umsetzung

Um eine Gegensprechanlage zu realisieren werden verschiedenste Funktionalitäten benötigt. Dazu zählen die Möglichkeit Ton und Bilder aufzuzeichnen und diese auch wiedergeben zu können. Dafür muss die entsprechende Hardware, also Kamera, Mikrofon, Display und Lautsprecher in einem Gerät vorhanden sein.

5.1.1 Türklingel

Sichtung geeigneter Geräte

Aufgrund des ausgewählten Betriebssystems ergibt sich die Möglichkeit aus einer breiten Palette von Geräten unterschiedlichster Hersteller das geeignetste Gerät auszuwählen. Hätte man zum Beispiel das Betriebssystem der Firma Apple als Implementierungsgrundlage herangezogen, wäre die Auswahl auf ein einziges Modell beschränkt.

Nach einigen Recherchen kristallisierte sich das von Samsung produzierte Nexus S (Abbildung: 5.1) heraus. Auf diesem Smartphone befindet sich noch die Urform des Android™-Systems, ohne dass Anpassungen seitens des Herstellers vorgenommen wurden. Dadurch

ergibt sich ein schneller Zugriff auf Updates. Bei Geräten anderer Hersteller werden die Updates erst geprüft, bevor diese für den Kunden freigegeben werden, um sicherzustellen, dass die durch den Hersteller vorgenommenen Änderungen noch kompatibel zum aktuellen System sind.



Abbildung 5.1: links: Google Nexus S ¹, rechts: Motorola Xoom²

Nach der Markteinführung des Motorola Xoom (Abbildung: 5.1) wurde das System auf den Tablet-PC mit der Android™ Version 3.0 portiert. Das Motorola Xoom bietet gegenüber einer Smartphone-basierenden Gegensprechanlage mehr Rechenleistung und ein wesentlich größeres Touchdisplay. Durch diese Attribute können dem Anwender die Interaktionsmöglichkeiten besser präsentiert werden.

Sichtung geeigneter Funktionen in Android™

Die API von Android™ ermöglicht das Aufzeichnen von Inhalten auf verschiedenste Arten. Es können komplette Videos oder Bildinhalte von Audioinhalten getrennt aufgezeichnet werden.

Für ersteres wird der `MediaRecorder` verwendet. Dieser ermöglicht das Aufzeichnen von

¹Quelle: <http://www.google.de/nexus/#/gallery>

²Quelle: <http://www.motorola.com/Consumers/DE-DE/>

[Consumer-Products-and-Services/Android-Tablets/ci.MOTOROLA-XOOM-with-Wi-Fi-DE-DE.overview](http://www.motorola.com/Consumers/DE-DE/Consumer-Products-and-Services/Android-Tablets/ci.MOTOROLA-XOOM-with-Wi-Fi-DE-DE.overview)

Videos in verschiedenen Formaten und Qualitätsstufen. Für die Realisierung einer Gegenprechanlage benötigt man die Fähigkeit, die aufgezeichneten Daten per Live Stream verschicken zu können. Dies ist mit dem `MediaRecorder` allerdings nicht möglich, da dieser zur Aufzeichnung eines Videos einen festen Speicherplatz voraussetzt. Daraus resultiert eine Zwischenspeicherung der Daten, um diese dann zu verschicken. Durch die Implementierung eines Erzeuger-Verbraucher Konzeptes wäre es denkbar, dieses Problem zu umgehen. Hieraus ergeben sich aber weitere Nachteile, wie eine Verzögerung der Übertragung, da ein Thread erstmals Daten aufnehmen muss, damit diese dann von einem Anderen verschickt werden können. Des Weiteren stellt der `MediaRecorder` keine Funktionalitäten zum Pausieren einer Aufnahme zur Verfügung. Dies wird aber benötigt, um dem Verbraucher-Thread den Zugriff auf die Daten zu ermöglichen. Das bedeutet schlussendlich, man darf nur Videos mit sehr geringer Laufzeit von ca. einer Sekunde aufnehmen. Angenommen eine Kommunikation würde zwei Minuten in Anspruch nehmen, dann würde der `MediaRecorder` in diesem Zeitraum ca. 120 Videos erzeugen. Daraus generiert sich die Problematik, noch eine effiziente Speicherverwaltung entwickeln zu müssen, um die anfallenden Datenmengen verwalten zu können. Zusätzlich stellt der `MediaRecorder` in Android™ 2.3 nur die Funktion bereit, Videos mit der Rückkamera aufzunehmen, welche sich für den Einsatz in einer Türklingen als unpraktisch erweist.

Eine weitere Möglichkeit bietet Android™ mit dem direkten Zugriff auf die Rohdaten der Kamera. Diese werden in einem `ByteArray` abgelegt und können somit ohne Weiteres über ein verbindungsloses Datagramm oder eine Socketverbindung an den Kommunikationspartner weitergeleitet werden. Dadurch werden Verzögerungen beim Versenden der Daten vermieden und es wird keine aufwendige Speicherverwaltung benötigt. Da die Daten nicht zwischengespeichert werden müssen. Allerdings muss man sich hier vor Augen halten, dass nur einzelne Bilder und keine Videosequenz übertragen wird. Das bedeutet, es sind keine Audiodaten vorhanden. Diese müssen gesondert aufgezeichnet und übermittelt werden. Außerdem ist das übertragene Datenvolumen um einiges höher, da die Daten nicht, wie bei der Übertragung eines Videos, vom Codec komprimiert werden können. Für den Einsatz in einem lokalen Netzwerk spielt dies aber eher eine untergeordnete Rolle.

5.1.2 Gegenstellen

Damit das System funktioniert, werden noch Gegenstellen benötigt, mit der die Türklingel interagieren kann. Zur Zeit der Realisierung bot kein Hersteller ein Gerät an, welches die Voraussetzungen für eine Gegenstelle erfüllte. Einige Hersteller von Fernsehgeräten ([Samsung \(2008\)](#)) bieten zwar die Möglichkeit, Multimediadaten von Geräten abzuspielen. Um aber als Gegenstelle zu fungieren, müsste der Fernseher auch aktiv Daten empfangen können. Dies ist mit den vorhandenen Geräten zur Zeit noch nicht realisierbar. Daher wurde eine entsprechende Client-Applikation geschrieben, um das System zu testen.

Die Implementierung der Client-Applikation erfolgte in der Programmiersprache Java, da dies mehrere Vorteile mit sich brachte. Zum einen unterliegt diese Anwendungen ähnlichen Abläufen wie die Türklingel. In Folge dessen konnten große Teile der Programmlogik aus der Türklingel wiederverwendet werden. Durch die Strukturierung nach dem Modell des MVC-Patterns, wie im Design Kapitel 4.1.4 beschrieben, konnte die Logik ohne nennenswerten Aufwand extrahiert werden. Zum anderen werden innerhalb der Wohnung zwei Betriebssysteme verwendet, wobei es sich um Windows 7 und Mac OSX handelt. Durch die Verwendung der Sprache Java kann ein und die selbe Applikation auf beiden Betriebssystemen verwendet werden.

Damit die Software alle erforderlichen Fähigkeiten einer Gegensprechanlage umsetzen kann, wird diese auf einem Mac Mini installiert. Dieser fungiert als Media-PC innerhalb der Wohnung und ist in Folge dessen mit einem Fernseher, der als Anzeigegerät dient, verbunden. Um audiovisuelle Daten zur Türklingel übertragen zu können, wird der Mac Mini zusätzlich noch mit einer Webcam und einem Mikrofon ausgestattet.

Darüber hinaus wird der Rechner noch mit Bibliotheken ausgestattet, um der Client-Applikation den einfachen Zugriff auf Daten der Webcam und des Mikrofons zu ermöglichen. Unter dem Betriebssystem Mac OSX findet die Open Source Bibliothek Processing Verwendung, während unter Windows 7 OpenCV zum Einsatz kommt, die ebenfalls Open Source ist.

5.1.3 Kommunikation

In den vorigen Abschnitten wurden die unterschiedlichen Geräte vorgestellt, mit denen der Anwender interagiert. Im diesem Abschnitt werden nun die Arten der Kommunikation zwischen der Türklingel und den Gegenstellen erläutert. Hierbei handelt es sich zum einen um eine synchrone Form der Kommunikation (5.1.3), zum anderen um eine asynchrone Form (5.1.3).

Die asynchrone Form wird mittels des ActiveMQ realisiert. Hierbei handelt es sich um ein Open Source³ Message Broker - System von der Apache Software Foundation. Die Vermittlung der Nachrichten kann hier auf zwei Arten geschehen, über Topics oder Queues. Topics sind hierbei als eine Art verbindungslose Kommunikation zu betrachten, bei der eine Nachricht nicht mehr zur Verfügung steht, sobald der ActiveMQ diese verteilt hat. Bei der Verteilung achtet dieser zusätzlich nicht darauf, ob die Clients die Nachricht wirklich empfangen haben. Im Gegensatz hierzu steht eine Queue. In ihr wird eine Nachricht solange aufbewahrt, bis ein Client diese Nachricht abrufen kann. Dadurch kann eine Nachricht allerdings nur an einen Empfänger gesendet werden, während bei einem Topic die Nachricht an eine Gruppe von Clients weitergeleitet werden kann.

³Apache 2.0 licensed

Kommunikation zwischen den Geräten

Diese Form der synchronen Kommunikation tritt innerhalb des Systems auf, wenn Bewohner und Besucher miteinander interagieren. Dies findet statt, nachdem die Verbindung über den ActiveMQ mittels asynchroner Nachrichten initialisiert wurde.

Um die Verbindung zwischen den Geräten möglichst Performant zu halten, tauschen die beiden Geräte verbindungslose Pakete aus. Dabei werden die Bilddaten getrennt von den Audiodaten versendet, wodurch eine höhere Unterbrechungsfreiheit erreicht werden kann. Geht beispielsweise ein Paket im Netzwerk verloren, das Bilddaten enthält, bleibt das Bild stehen. Jenes Pakets das die Audiodaten enthält, kommt aber an. Somit sieht der Anwender für eine kurze Zeit ein Standbild, während der Ton aber weiterhin flüssig wahrnehmbar bleibt. Würde man hier beides über ein Paket abwickeln, würde der Anwender bei einem Paketverlust eine kurze Unterbrechung der Kommunikation hinnehmen müssen. Dies kann natürlich auch bei der getrennten Versendung der Daten vorkommen, aber die Wahrscheinlichkeit ist geringer.

Die Erfassung und Bearbeitung der audiovisuellen Daten wurde auf beiden Gerätetypen (Türklingel und Gegenstellen) einheitlich gestaltet. Das versenden der Bilddaten erfolgt über ein verbindungsloses Datagramm-Paket. Dazu werden, wie in 5.1.1 beschrieben, Rohdaten der Kamera ausgelesen und in das JPEG Format konvertiert. Nach der Erfassung der Audiodaten, werden diese über verbindungslose RTP ⁴ Pakete verschickt. Hierfür wurde die Open Source Bibliothek RTP Java Library, kurz "jlibrtp" ⁵, benutzt.

Kommunikation mit dem ActiveMQ

Die Verbindung zum ActiveMQ ist eine Form der asynchronen Kommunikation. Sie wird von den Geräten zur Initialisierung einer Verbindung, Übertragen von Einstellungen oder zum Austausch von Nachrichten verwendet. Hierbei stellt es einen signifikanten Unterschied dar, ob die Nachricht von einer Gegenstelle oder einer Türklingel, also einem AndroidTM-Gerät, versendet wird.

Für eine Gegenstelle, die mit der Programmiersprache Java geschrieben wurde, ist die Kommunikation einfach zu realisieren. Hiefür wird der Java Message Service (JMS) verwendet. Im Rahmen des Living Place Hamburg wurde speziell für diese Kommunikation eine Bibliothek implementiert, der sogenannten LPWrapper. Dieser kapselt die Funktionen von JMS und ermöglicht so das leichte Versenden und Empfangen von asynchronen Nachrichten.

⁴Realtime Transport Protocol

⁵Quelle: <http://sourceforge.net/projects/jlibrtp/> (Version 0.2.2)

Auf einem Android™-Gerät kann dieser aber nicht verwendet werden, da hier nur eine abgespeckte Java Version vorhanden ist, die keine Funktionalitäten von JMS beinhaltet. Daher wurde hier eine Proxy-Applikation (5.1.3) entwickelt, die Nachrichten von Android™-Geräten empfangen soll und diese an den ActiveMQ weiterleitet. Zusätzlich wurde eine Bibliothek (5.1.3) implementiert um auf die Funktionen des Proxys zugreifen zu können. Ziel ist es, zukünftigen Entwicklern den Zugriff auf den Proxy und damit auf den ActiveMQ zu erleichtern. Das Empfangen der Nachrichten wird über einen HTTP-Request abgehandelt. Dafür wird auf dem Android™-Gerät ein Thread gestartet, der die Nachrichten vom ActiveMQ über den Request abfragt und falls Daten vorhanden sind, diese in eine JSON Objekt umwandelt. Daraufhin wird der Inhalt des Objektes untersucht und weiterverarbeitet.

AndroidProxy

Im Rahmen der Realisierung wurde eine Proxy-Applikation entwickelt, um die Kommunikation zwischen einem Android™-Gerät und dem ActiveMQ im Living Place Hamburg zu ermöglichen. Besonders steht hier im Vordergrund, dass es Android™-Geräten nicht möglich ist, Nachrichten an den ActiveMQ zu schicken.

Um dieses Problem zu lösen stellt der AndroidProxy eine Socket-Schnittstelle zur Verfügung, an welche die Android™-Geräte ihre Nachricht schicken können. Die Software empfängt die Nachrichten, liest wichtige Informationen wie den Topicnamen, an den diese weitergeleitet werden soll, aus und übermittelt die Daten an den ActiveMQ. Intern wird dazu der LPWrapper verwendet, der speziell für die Kommunikation im Living Place entwickelt wurde. Diese Software wird nur für das Senden von Mitteilungen benötigt. Die Proxy-Applikation läuft auf dem selben Server wie der ActiveMQ, dadurch werden zusätzliche Netzwerkschnittstellen und erhöhter Netzwerkverkehr vermieden. Die folgende Abbildung 5.2 soll die Kommunikation nochmals verdeutlichen.

AndroidPublisher

Durch den oben beschriebenen AndroidProxy ist es Android™-Geräten nun möglich, Nachrichten an den ActiveMQ zu übertragen. Um zukünftigen Entwicklern den Umgang mit dem AndroidProxy zu erleichtern, wurde die Bibliothek AndroidPublisher entwickelt, welche nach dem Prinzip eines Wrappers arbeitet. Dieser verbirgt das Nachrichtenformat und die Verbindungsart vor dem Entwickler. Somit braucht sich dieser hierüber keine Gedanken mehr zu machen.

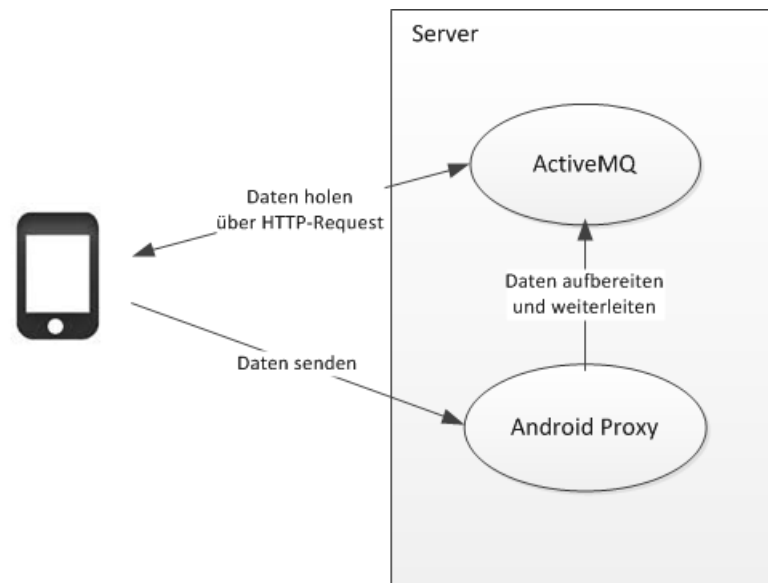


Abbildung 5.2: Kommunikation zwischen Android™ und dem ActiveMQ

Der AndroidPublisher dient also dem einfacheren Umgang mit dem ActiveMQ und dem AndroidProxy. Um den AndroidPublisher zu initialisieren, übergibt der Entwickler einige Informationen wie die IP und den Port des AndroidProxys. Soll nun eine Nachricht versendet werden, wird diese zusammen mit dem Topicnamen über einen Methodenaufruf übergeben. Danach werden die Daten in das richtige Format konvertiert und anschließend versendet.

5.2 Evaluation

Die Realisierung der Anwendung wird in diesem Abschnitt auf ihre Funktionalitäten hin evaluiert. Dazu werden die tatsächlichen Anforderungen und Abläufe bei der Verwendung der Türklingel mit den in den Szenarien geschilderten verglichen.

5.2.1 Erfüllung der Anforderungen

Im dritten Kapitel wurden anhand von Beispielszenarien Anforderungen ermittelt. Hierbei ist zu beachten, dass nicht alle Anforderungen erfüllt werden konnten, da nur Grundfunktionalitäten einer Gegensprechanlage implementiert wurden. Im Folgenden werden funktionale und nicht-funktionale Anforderungen auf ihre Umsetzung hin überprüft.

Funktionale Anforderungen

- Klingeln senden:
Diese Anforderung wurde erfüllt. Der Besuch kann auswählen, für welche Person er klingeln möchte.
- Klingeln signalisieren:
Diese Anforderung wurde zum Teil umgesetzt. Die Ortung einer Person wurde nicht mit ins System integriert. Das Klingeln wird hier auf allen aktiven Gegenstellen visualisiert.
- Klingeln annehmen:
Diese Anforderung wurde erfüllt. Der Bewohner erhält in der Visualisierung die Möglichkeit, das Klingeln anzunehmen.
- Klingeln ablehnen:
Diese Anforderung wurde erfüllt. Der Bewohner erhält in der Visualisierung die Möglichkeit, das Klingeln abzulehnen.
- Aufbau und Abbau einer Audio- und Videoübertragung:
Diese Anforderung wurde erfüllt.
- Tür öffnen:
Diese Anforderung wurde nicht erfüllt. Dem System ist es nicht möglich, die Tür zu öffnen, da keine elektronischen Schlösser vorhanden sind.
- Videonachricht hinterlassen:
Diese Anforderung wurde erfüllt.
- Videonachricht abrufen:
Diese Anforderung wurde erfüllt.
- Textuelle Nachrichten für Besucher hinterlassen:
Diese Anforderung wurde nicht erfüllt. Bei der Abwesenheit des Bewohners wird zwar eine Nachricht angezeigt, diese kann aber nicht verändert werden.
- Persistente Nachrichtenspeicherung:
Diese Anforderung wurde nicht erfüllt, da im Netzwerk kein Dienst zur Datenspeicherung zur Verfügung stand.
- Authentifizierung von Personen:
Diese Anforderung konnte nicht erfüllt werden, da kein entsprechender Dienst zur Verfügung stand.

Nicht-Funktionale Anforderungen

Anhand der erfüllten funktionalen Anforderung konnte die Erstellung einer Gegensprechanlage im Smart Home Umfeld nachgewiesen werden. Nun ist noch die Qualität dieser Anlage durch die nicht-funktionalen Anforderungen zu evaluieren.

Hinsichtlich der Sicherheit des Systems wurde darauf geachtet, dass keine Möglichkeit besteht, in das Innere des Systems einzudringen. Dafür beinhaltet die Oberfläche beispielsweise nur statische Objekte, die von einem Besucher nicht verändert werden können. Des Weiteren kommuniziert das System ausschließlich über vordefinierte Nachrichten. Diese kann ein Angreifer zwar verändern, dabei müssen aber die in Abschnitt 4.2.2 definierten Konventionen eingehalten werden, da die Mitteilungen ansonsten ignoriert werden. Verändert ein Angreifer eine Nachricht und hält dabei die Konventionen ein, besitzt dieser nur die Möglichkeit, Einstellungen zu ändern. Zugriff auf das Innere des Systems oder Benutzerdaten erhält der Angreifer jedoch nicht, da diese nicht über die Nachrichten übertragen werden.

Die Erweiterbarkeit des Systems ist durch die getroffenen Designentscheidungen gewährleistet. Durch das MVC-Pattern können Funktionalitäten hinzugefügt werden, ohne die GUI ändern zu müssen. Hierdurch können neue Dienste einfach in Anspruch genommen werden. Dies gilt auch für den umgekehrten Fall. Möchte der Anwender die GUI anders gestalten, kann diese implementiert werden, ohne die Logik der Anwendung verändern zu müssen.

Die Zuverlässigkeit des System lässt sich anhand der getroffenen Entscheidungen erkennen. Hierzu zählt das Designs des Systems, mit der strikten Trennung der Komponenten und deren Kommunikation untereinander sowie der modulare Aufbau im inneren der Komponenten. Dies gewährleistet den einfachen Austausch von Geräten, zum Beispiel bei einem Defekt oder um den Funktionsumfang zu erhöhen.

Der Schutz der Privatsphäre konnte nur ansatzweise realisiert werden. Der Anwender hat die Möglichkeit, die Art der Übertragung in einem Einstellungsmenü zu wählen. Darin hat er die Wahl, Videodaten mitzusenden oder dies zu unterbinden. Eine ortsgebundene Privatsphäreinstellungen wurde nicht umgesetzt.

5.2.2 Funktionsweise des Prototypen

In diesem Abschnitt sollen die Bedienung und die verwendeten Abläufe des realisierten Prototyps vorgestellt werden. Dabei werden insbesondere die Hauptfunktionalitäten wie das Klingeln und die Art der visuellen Darstellung erläutert.

Nachdem die Anwendung gestartet wurde, wird der Hauptbildschirm angezeigt, der den Ausgangspunkt jeder Interaktion bildet. Der in Abbildung 5.3⁶ gezeigte Startbildschirm, bietet dem Besucher als erstes die Möglichkeit, die Familie auszuwählen, bei der er klingeln möchte. Dabei kann diese Anzeige flexibel gestaltet werden, je nachdem, in welchem Umfeld, also Ein- oder Mehrfamilienhaus, die Türklingel zum Einsatz kommt.

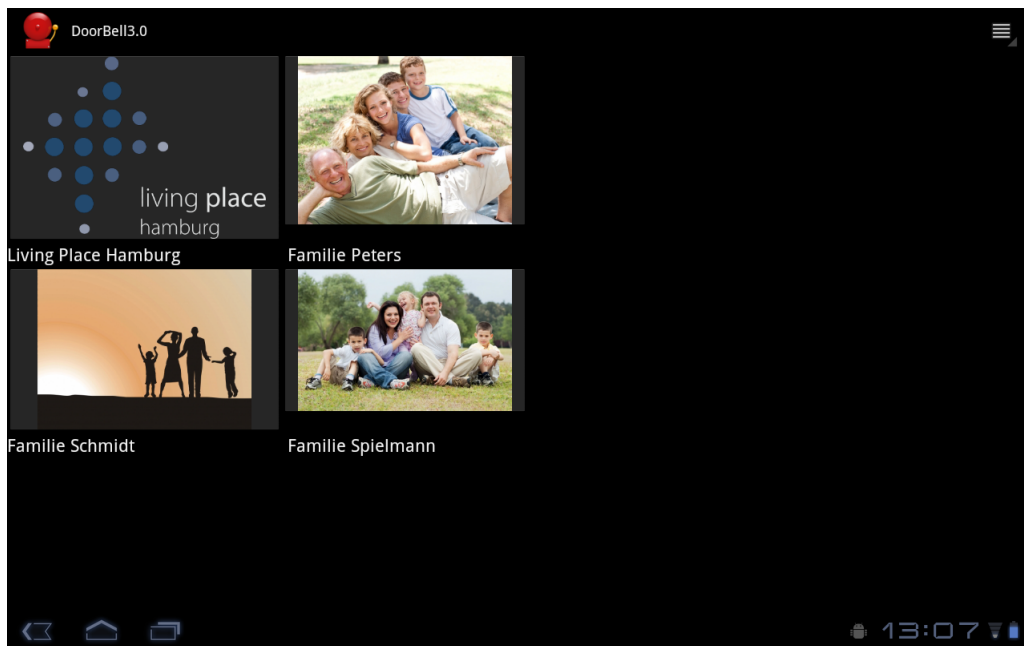


Abbildung 5.3: Hauptbildschirm der Türklingel

Wählt der Besucher eine der angezeigten Familien aus, werden auf der rechten Seite des Bildschirms die in der Wohnung lebenden Personen angezeigt (5.4). Welche Personen hier aufgeführt werden, kann der Bewohner frei konfigurieren.

Nachdem der Besucher einen Eintrag aus der Liste ausgewählt hat, sendet die Türklingel eine Nachricht an den Nachrichtendienst und erfragt somit vorhandene Ressourcen. Geräte, die sich für dieses Ereignis beim Nachrichtendienst registriert haben, empfangen diese Nachricht und signalisieren dem Bewohner, dass eine Person geklingelt hat. Dieser kann nun entscheiden, auf welchem Gerät er das Klingeln annimmt. Hat der Bewohner ein Gerät ausgewählt, liest die Client-Applikation daraufhin die erforderlichen Verbindungsdaten aus der empfangenen Nachricht aus und baut mit diesen eine Verbindung zur Türklingel auf.

⁶Die gezeigten Familienportraits in den Abbildungen 5.3 und 5.4 wurden von Cecelia / FreeDigitalPhotos.net zur Verfügung gestellt

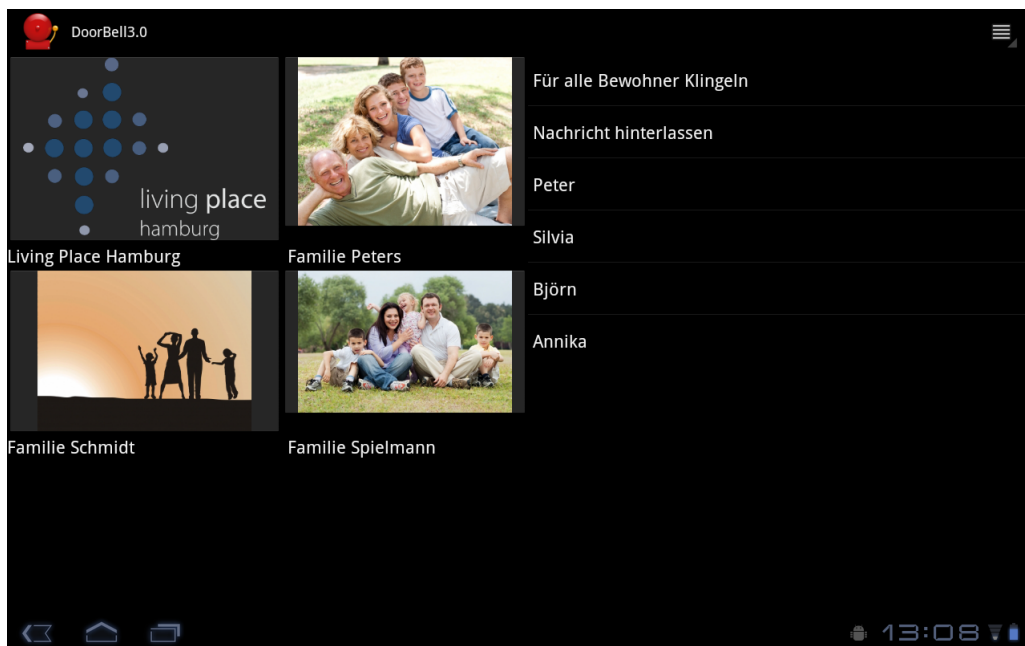


Abbildung 5.4: Auswahlmöglichkeiten der Türklingel

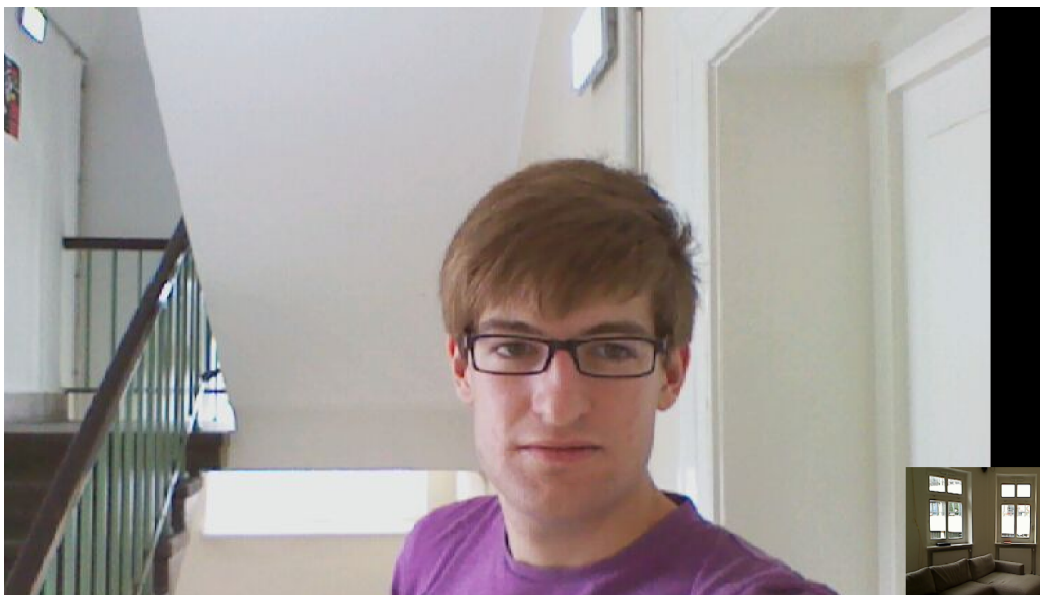


Abbildung 5.5: Darstellung der audiovisuellen Übertragung auf der Gegenstelle

Nachdem die Verbindung aufgebaut wurde, werden die von der Türklingel empfangenen Daten angezeigt, wie in Abbildung 5.5 dargestellt ist.

Dabei teilt sich die Anzeige in zwei Bereiche auf. Der Hauptbereich zeigt den Besucher, während der zweite Bereich anzeigt, welche Bilddaten zur Türklingel gesendet werden. Diese werden in der Anzeige unten rechts eingeblendet.

Nachdem der Bewohner gesehen hat, wer vor der Tür steht, kann dieser entscheiden, ob er das Klingeln annehmen möchte und somit ein Gespräch zustande kommt oder ob er dies ablehnt. Wird das Klingeln abgelehnt, sendet die Client-Applikation eine Nachricht an die Türklingel, was zur Folge hat, dass die Verbindung beendet wird. Der Besucher wird daraufhin über die Abwesenheit des Bewohners benachrichtigt.

Nimmt der Bewohner jedoch das Klingeln an, werden die aus der Wohnung aufgezeichneten Inhalte auf der Türklingel angezeigt, wie in Abbildung 5.6 zu sehen ist.



Abbildung 5.6: Darstellung der audiovisuellen Übertragung auf der Türklingel

Daraufhin können Bewohner und Besucher miteinander kommunizieren. Beendet eine der beiden Parteien die Kommunikation, versendet der Initiator eine Nachricht, um dies der anderen Komponente bekannt zu machen. Anschließend versetzen sich beide Applikationen wieder in ihren Ausgangspunkt und warten auf eine neue Interaktion.

5.2.3 Erkenntnisse aus der Evaluation

Die Evaluierung hat gezeigt, dass die Erstellung einer Gegensprechanlage machbar ist. Dies wird aus den umgesetzten Anforderungen und der Arbeitsweise des Prototypen deutlich. Des Weiteren hat sich gezeigt, dass die im Design beschriebene Architektur umgesetzt werden konnte. Durch den Einsatz des MVC-Entwurfsmusters wird es zusätzlich ermöglicht, weitere Interaktionen in das System zu integrieren.

Die Problemfelder, die sich bei der Erstellung des Systems ergaben, konnten zufriedenstellend gelöst werden. Hier sind insbesondere die Live-Übertragung von Videoinhalten (5.1.1) und die Kommunikation mit dem ActiveMQ (5.1.3) zu nennen.

Auf dieser Grundlage können nun weitere Entwicklungen stattfinden, in denen weitere Interaktionsmöglichkeiten erforscht und die Funktionalitäten ausgebaut werden können. Diese kann beispielsweise durch das Einbinden eines Ortungsdienstes oder die Implementierung eines Kontaktdienstes geschehen.

Ebenfalls zeigte die Evaluation, dass die bisherige Realisierung nur einen kleinen Teil der Funktionalitäten bereitstellt. Es fehlen noch die Ansteuerungsmöglichkeiten für Hardware, um beispielsweise dem System das Öffnen der Tür zu ermöglichen sowie die Kontextsensitivität. Erst durch diese Punkte stellt das System für den Anwender einen Mehrwert dar.

Ebenfalls hat die Evaluation gezeigt, dass dies System schon in heutigen Wohnungsumgebungen zum Einsatz kommen kann. Im Gegensatz zu Smart Home Umgebungen hat die Türklingel allerdings keinen Zugriff auf gewisse Dienste, wie beispielsweise einen Ortungsdienst. Daher kann die Türklingel hier nicht alle Funktionalitäten zu Verfügung stellen.

6 Schluss

6.1 Zusammenfassung

Im Rahmen dieser Arbeit sollten Interaktionsmöglichkeiten und dessen Realisierung untersucht werden. Basis hierfür bildete das mobile Betriebssystem Android™. Die Wahl fiel auf dieses mobile System, da dieses im Gegensatz zu den anderen Betriebssystemen einige Vorteile besitzt. Dazu gehören unter anderem die große Gerätevielfalt und die Open Source Plattform.

Im Verlauf der Arbeit stellte sich jedoch heraus, dass Android™ nicht alle Funktionalitäten zu Verfügung stellt, die für die Realisierung benötigt wurden. Aus diesem Grund wurde externe Bibliotheken eingebunden und einige Funktionen zusätzlich implementiert.

Am Anfang dieser Arbeit wurden drei Szenarien (3.1) analysiert, durch die verschiedene Mensch-Maschinen-Interaktionen identifiziert werden konnten. Das Spektrum dieser Szenarien reicht hier von einfachen Funktionalitäten, die auch durch konventionelle Gegensprechanlagen abgebildet werden, bis hin zu wesentliche komplexeren Funktionen. Aus dieses Szenarien gingen verschiedene funktionale und nicht-funktionale Anforderungen hervor, die das System später abbilden sollte.

Nachdem die verschiedenen Anforderungen und Eigenschaften an das System ermittelt wurden, musste eine passende Architektur gefunden werden. Um dies zu erreichen, wurden zu erst verschieden Architekturmuster (4.1.1) miteinander verglichen. Hierfür wurden die Vor- und Nachteile der verschiedenen Architekturen im Hinblick auf das zu realisierende System bewertet (4.1.2). Dabei kristallisierte sich die Event-Driven Architektur heraus.

Nach der Wahl der Systemarchitektur wurde eine Komponentenarchitektur konzipiert, die sich auf das Prinzip des MCV Patterns stützt. Da die Komponenten auf unterschiedlichen Geräten realisiert werden sollten, konnte hiermit eine einfache Portierung sichergestellt werden. Zusätzlich wurde die Kommunikation der Komponenten untereinander untersucht und ein festes Nachrichtenformat wurde definiert.

Auf Basis der getroffenen Designentscheidungen wurde das System umgesetzt. Um das System später auch testen zu können, wurde nicht nur die Türklingel (5.1.1) implementiert, sondern auch eine Client-Applikation, die als Gegenstelle (5.1.2) fungiert. Dabei wurden einige Grundfunktionalitäten implementiert, die für die Verwendung einer Gegensprechanlage

Vorraussetzung sind. Einige Funktionalitäten, die hierfür benötigt wurden, waren bereits in Android™ vorhanden. Andere wurden durch externe Bibliotheken hinzugefügt oder zusätzlich implementiert.

Aus den Erfahrungen, die in der Evaluation gesammelt wurden, ergaben sich einige Erkenntnisse, die der Weiterentwicklung des Systems dienen können. Diese werden im folgenden Ausblick aufgeführt.

6.2 Ausblick

Die Arbeit hat gezeigt, dass eine intelligente Türklingel ein hohes Potential besitzt. Aber auch, dass eine gewisse Infrastruktur in einer Wohnung vorhanden sein muss, um die Funktionalitäten nutzbar zu machen. Bis Wohnungen so eine Infrastruktur besitzen, wird es allerdings noch einige Zeit dauern. Bis dahin wäre die Realisierung eine abgespeckten Version aber durchaus denkbar.

Essentiell für die Akzeptanz des Systems wird die Sicherheit sein. Das Hauptaugenmerk liegt hier vor allem in der Authentifizierung von Personen. Einige Möglichkeiten hierfür wurden in dem Szenario "Benutzerauthentifizierung" (3.1.2) geschildert. Um die Sicherheit für die Zukunft zu erhöhen, wäre es denkbar, Personen durch das Erstellen und den Austausch elektronischer Zertifikate zu verifizieren. Aber auch der neue elektronische Personalausweis (Bundesministerium) mit seiner Ausweisfunktionalität könnte in Zukunft für die Verifizierung von Personen verwendet werden.

Ein weiterer essentieller Punkt stellen zukünftige Bedienkonzepte dar. Gerade im Smart Home Bereich sind aktuelle Konzepte, basierend auf einer Tastatur- und Maussteuerung, eher ungeeignet. Hier würde der Einsatz einer Sprachsteuerung oder einer Gestenerkennung eine wesentlich intuitivere Bedienung ermöglichen. Denkbar ist hier auch eine Kombination dieser beiden Techniken.

Ob Android™-Geräten der Einzug ins Heim und Haus gelingen wird oder nicht wird die Zeit zeigen. Einiges spricht allerdings dafür. Zum einen werden die Geräte in Zukunft immer leistungsfähiger, kleiner und kostengünstiger. Zum anderen werden durch das Projekt Android@Home, das auf der Developer Conference Google I/O 2011 (Google (2011c)) vorgestellt wurde, einige neue APIs entstehen. Hierdurch wird es Entwicklern erleichtert, spezielle Anwendungen für Smart Home Umgebungen zu konzipieren und zu implementieren. Zusätzlich werden im Laufe der Zeit Komponenten auf den Markt kommen, die sich über Android™-Geräte steuern lassen, wobei hier schon ein Anfang durch die Leuchtmittel der Firma Lighting Science gemacht wurde.

Inwiefern aus diesen Vorraussetzungen nützliche Funktionen entstehen, die dem Anwender einen Mehrwert bringen, bleibt abzuwarten. Des Weiteren wird sich zeigen, wie Aspekte der

Sicherheit, Schutz der Privatsphäre und Stabilität der Systeme mit in die Entwicklung einfließen. Mit diesen Fragestellungen werden sich mit Sicherheit in der nahen Zukunft weitere wissenschaftliche Arbeiten beschäftigen.

Literaturverzeichnis

- [Augusto 2007] AUGUSTO, Juan C.: Ambient Intelligence: the Confluence of Ubiquitous/Pervasive Computing and Artificial Intelligence. (2007). – URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.147.553&rep=rep1&type=pdf>. – abgerufen am 07.07.2011
- [BMBF] BMBF, Bundesministerium für Bildung und F.: *Technik für ein selbstbestimmtes Leben*. – URL <http://www.aal-deutschland.de/>. – abgerufen am 21.07.2011
- [Bregman] BREGMAN, Mark: *The Convenience of Small Devices: How Pervasive Computing Will Personalize E-Business*. – URL http://domino.watson.ibm.com/comm/wwwr_thinkresearch.nsf/pages/bergman398.html. – abgerufen am 06.07.2011
- [Bundesministerium] BUNDESMINISTERIUM, Das Bundesministerium des I.: *Der neue Ausweis*. 2010 - 2011. – URL http://www.personalausweisportal.de/DE/Der_Neue_Ausweis/der-neue-ausweis_node.html. – abgerufen am 10.07.2011
- [Buschmann u. a. 2000] BUSCHMANN, Frank ; MEUNIER, Regine ; ROHNERT, Hans ; SOMMERLAD, Peter ; STAL, Michael: *Pattern-orientierte Software-Architektur - Ein Pattern-System*. 1. korr. Nachdruck. Addison Wesley Verlag, 2000. – URL <http://amazon.com/o/ASIN/3827312825/>. – ISBN 9783827312822
- [Dahm 2005] DAHM, Markus: *Grundlagen der Mensch-Computer-Interaktion*. Pearson Studium, 12 2005. – 368 S. – URL <http://amazon.com/o/ASIN/3827371759/>. – ISBN 9783827371751
- [Dey 2001] DEY, Anind K.: Understanding and Using Context. In: *Personal Ubiquitous Comput.* 5 (2001), January, S. 4–7. – URL <http://dx.doi.org/10.1007/s007790170019>. – ISSN 1617-4909
- [Dunkel u. a. 2008] DUNKEL, Jürgen ; EBERHART, Andreas ; FISCHER, Stefan ; KLEINER, Carsten ; KOSCHEL, Arne: *Systemarchitekturen für Verteilte Anwendungen*. Hanser Fachbuchverlag, 2008. – URL <http://amazon.com/o/ASIN/3446413219/>. – ISBN 9783446413214

- [Dustdar u. a. 2003] DUSTDAR, Schahram ; GALL, Harald ; HAUSWIRTH, Manfred: *Software-Architekturen für Verteilte Systeme: Prinzipien, Bausteine und Standardarchitekturen für moderne Software (Xpert.press) (German Edition)*. 1. Springer, 7 2003. – URL <http://amazon.com/o/ASIN/3540430881/>. – ISBN 9783540430889
- [Ellenberg 2010/2011] ELLENBERG, Jens: Klassifizierung von Kontext in einer intelligenten Wohnung. (2010/2011), S. 14. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-seminar/ellenberg/bericht.pdf>. – abgerufen am 26.07.2011
- [Google 2011a] GOOGLE: *Android Compatibility*. 2011. – URL <http://source.android.com/compatibility/index.html>. – abgerufen am 08.07.2011
- [Google 2011b] GOOGLE: *Android Licenses*. 2011. – URL <http://source.android.com/source/licenses.html>. – abgerufen am 07.07.2011
- [Google 2011c] GOOGLE: *Google I/O 2011 - Developer Conference*. 2011. – URL <http://www.google.com/events/io/2011/index-live.html>. – abgerufen am 11.07.2011
- [Google 2011d] GOOGLE: *Philosophy and Goals*. 2011. – URL <http://source.android.com/about/philosophy.html>. – abgerufen am 07.07.2011
- [Krumm 2009] KRUMM, John (Hrsg.): *Ubiquitous Computing Fundamentals*. Chapman and Hall/CRC, 9 2009. – URL <http://amazon.com/o/ASIN/1420093606/>. – ISBN 9781420093605
- [Meier 2010] MEIER, Reto: *Professional Android 2 Application Development (Wrox Programmer to Programmer)*. 2. Wrox, 3 2010. – URL <http://amazon.com/o/ASIN/0470565527/>. – ISBN 9780470565520
- [Otto und Voskuhl 2010/2011] OTTO, Kjell ; VOSKUHL, Sören: Weiterentwicklung der Architektur des Living Place Hamburg. (2010/2011), S. 22. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/otto-voskuhl.pdf>. – abgerufen am 26.07.2011
- [Rahimi und Vogt 2008] RAHIMI, Mohammadali ; VOGT, Matthias: *Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008
- [Rahimi und Vogt 2009/2010] RAHIMI, Mohammadali ; VOGT, Matthias: Aufbau des Living Place Hamburg. (2009/2010). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-proj/rahimi-vogt.pdf>. – abgerufen am 09.06.2011

- [Reenskaug 1979] REENSKAUG, Trygve: *MODELS - VIEWS - CONTROLLERS*. Dezember 1979. – URL <http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf>. – abgerufen am 24.05.2011
- [Rösch 2011] RÖSCH, Sebastian: *Audiovisuelle Kommunikation in Smart Home Umgebungen am Beispiel eines Raumklangtelefons*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2011. – (unveröffentl.)
- [Samsung 2008] SAMSUNG: Samsung's Guide to DLNA. (2008). – URL http://www.samsung.com/us/pdf/dlna_guide.pdf. – abgerufen am 26.07.2011
- [Savov 2011] SAVOV, Vlad: *Visualized: Android activations mapped geographically, chronologically, breathtakingly (video)*. 24.02.2011. Feb 2011. – URL <http://www.engadget.com/2011/02/24/visualized-android-activations-mapped-geographically-chronolog/>. – abgerufen 07.07.2011
- [Schill und Springer 2007] SCHILL, Alexander ; SPRINGER, Thomas: *Verteilte Systeme: Grundlagen und Basistechnologien (eXamen.press) (German Edition)*. 1. Springer, 3 2007. – URL <http://amazon.com/o/ASIN/3540205683/>. – ISBN 9783540205685
- [Schmidt u. a. 1999] SCHMIDT, Albrecht ; BEIGL, Michael ; GELLERSEN, Hans-W.: There is more to context than location. In: *Computers & Graphics* 23 (1999), Nr. 6, S. 893 – 901. – URL <http://www.sciencedirect.com/science/article/pii/S009784939900120X>. – ISSN 0097-8493
- [Shankland und Kaden 2011] SHANKLAND, Stephen ; KADEN, Jan: *Google stellt Android@Home vor*. Mai 2011. – URL http://www.zdnet.de/news/mobile_wirtschaft_google_stellt_android_home_vor_story-39002365-41552871-1.htm. – abgerufen am 11.05.2011
- [TAB 2009] TAB, Oliver Raabe Daniel J. Koch Peter Georgieff Peter N.: *Zukunftsreport - Ubiquitäres Computing / Büro für Technologiefolgen-Abschätzungen beim Deutschen Bundestag*. Mai 2009. – Forschungsbericht
- [Tanenbaum und van Steen 2007] TANENBAUM, Andrew S. ; STEEN, Maarten van: *Verteilte Systeme - Prinzipien und Paradigmen*. Pearson Education Deutschland, 2007. – URL <http://amazon.com/o/ASIN/3827372933/>. – ISBN 9783827372932
- [Voskuhl 2009] VOSKUHL, Sören: *Bereitstellung einer Sensorwolke*. (2009). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/Voskuhl/bericht.pdf>. – abgerufen am 09.06.2011

- [Voskuhl 2010/2011] VOSKUHL, Sören: Entwicklung einer Architektur für Context-Aware Systeme. (2010/2011). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-seminar/voskuhl/bericht.pdf>. – abgerufen am 09.06.2011
- [Watzlawick 2007] WATZLAWICK, Paul: *Menschliche Kommunikation - Formen Störungen Paradoxien*. 11., unveränderte Auflage. Hans Huber Verlag, 2007
- [Weiser 1991] WEISER, Mark: *The Computer for the 21st Century*. 09 1991. – URL <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>. – abgerufen 11.04.2011
- [Witt 2011] WITT, Kristoffer: *Kontextabhängige multimodale Interaktion mit Schwerpunkt Spracherkennung im Smart-Home Umfeld*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2011

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 1. August 2011

Ort, Datum

Unterschrift