



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Sobin Ghose

**Konzeption und Evaluation eines interaktiven
Badezimmerspiegels**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Sobin Ghose

**Konzeption und Evaluation eines interaktiven
Badezimmerspiegels**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

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

Eingereicht am: 18. Februar 2014

Sobin Ghose

Thema der Arbeit

Konzeption und Evaluation eines interaktiven Badezimmerspiegels

Stichworte

Smart Mirror, Spiegel-Display, Mensch-Computer-Interaktion, HCI, Disappearing Computing, Gesten, 3D-Gestensteuerung

Kurzzusammenfassung

In dieser Arbeit wurde ein Smart Mirror beispielhaft für das Badezimmer konzipiert, entwickelt und evaluiert. Dazu wurden die Anforderungen an ein solches System analysiert und durch Soft- und Hardwarelösungen umgesetzt. Die Interaktion mit dem System wurde durch Gesten- und Sprachsteuerung, wie auch durch ein Spiegel-Display umgesetzt. Anschließend wurde das System für eine Evaluation untersucht.

Sobin Ghose

Title of the paper

Conception and evaluation of an interactive bathroom-mirror

Keywords

Smart Mirror, Mirror Display, Human-Computer-Interaction, HCI, Disappearing Computing, Gesture, 3D-Gesture-Controll

Abstract

In this paper, a smart mirror for the bathroom was designed, developed and evaluated. The requirements for such a system were analyzed and implemented through software and hardware solutions. The interaction with the system was implemented via gesture and voice control, as well as through a mirror display. Then, the system was examined in an evaluation.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Gliederung der Arbeit	2
2	Analyse	3
2.1	Szenario	5
2.1.1	Der morgendliche Aufenthalt im Badezimmer einer intelligenten Wohnung der Zukunft	5
2.2	Aufgabenstellung	6
2.2.1	Allgemein	6
2.2.2	Funktionale Anforderungen	7
2.2.3	Interaktionsrepertoire	9
2.2.4	Rahmenbedingungen	9
2.2.5	Anwendungsfälle	11
2.2.6	Zielsetzung	14
2.3	HCI - Human Computer Interaction	15
2.3.1	Allgemein	15
2.3.2	HCI am Smart Mirror	15
2.4	Smart Home	17
2.4.1	Allgemein	17
2.4.2	Smart Mirror im Smart Home	17
2.5	AAL - Ambient Assisted Living	17
2.6	Labor Living Place an der HAW Hamburg	18
2.6.1	Allgemein	18
2.6.2	Infrastruktur des Living Place	20
2.7	Verwandte Arbeiten und Produkte	21
2.7.1	Verwandte Arbeiten an der HAW	21
2.7.2	Weitere verwandte Arbeiten	23

2.7.3	Verwandte Produkte	27
2.8	Laborumgebung	29
2.8.1	Living Place als Umgebung des Smart Mirrors	29
2.8.2	Darstellung	30
2.8.3	Steuerung	31
2.9	Gesten	32
2.9.1	Kulturelle Interpretation	33
2.9.2	Abgrenzung Motion Tracking und Gestenerkennung	33
2.9.3	Kategorisierung von manipulativen Gesten	34
2.9.4	2D-Gesten	35
2.9.5	3D-Gesten	35
2.10	Spracherkennung	37
2.10.1	Spracherkennung vor dem Smart Mirror	38
2.10.2	Start- Endproblematik bei Sprachsteuerung	38
2.10.3	Befehlinterpretation	39
3	Design und Realisierung	40
3.1	Gesamtsystem Konzept	40
3.2	Living-Place-Architektur	41
3.2.1	Konzept	41
3.2.2	Kommunikation	42
3.3	Authentifizierung	42
3.3.1	Wissen	43
3.3.2	Besitz	44
3.3.3	Boimetrie	44
3.3.4	Fazit	46
3.4	Software-Architektur	46
3.4.1	Architektur und Entwurfsmuster	47
3.4.2	Kommunikation	50
3.4.3	Dienste	51
3.4.4	Kopplung	52
3.5	Software und Frameworks	52
3.5.1	Microsoft Speech SDK	52
3.5.2	Kinect SDK	52
3.5.3	WPF - Windows Presentation Foundation	53

3.5.4	Leap Motion SDK	53
3.5.5	ActiveMQ - Message Broker	53
3.5.6	JSON - JavaScript Object Notation	54
3.5.7	OpenWeatherMap API	54
3.5.8	IMAP E-Mail Bibliothek	54
3.5.9	Google Data APIs	54
3.6	Technische Architektur	54
3.6.1	Hardware	55
3.6.2	Interpretation	59
4	Evaluation	68
4.1	Technische Machbarkeit	68
4.1.1	Testumgebung	70
4.1.2	Alphatests	72
4.2	Usability	72
4.2.1	Gewonnene Erkenntnisse	72
4.3	Fazit	76
5	Schluss	77
5.1	Zusammenfassung	77
5.2	Ausblick	78
6	Anhang	80
6.1	A Grammatic Speech Recognition	80
6.2	B Aufbau der Voice Commands	95
6.3	C Interaction Stream	96

1 Einleitung

1.1 Motivation

Im heutigen Zeitalter, in dem 3D-Gesten zur Steuerung von Spielkonsolen genutzt werden und fast jeder eine Vorstellung von dem Begriff "Smart Home" hat, verändern diese technischen Entwicklungen unseren Alltag und unsere Ansprüche an Geräte und Technik. Im Bereich der Haustechnik ist es inzwischen Standard, dass Lampen über Bewegungssensorik, Heizungen per Zeitpläne und Temperatur gesteuert werden, wie auch dass sich die Waschmaschine erst abends, wenn der Strom günstiger ist, automatisch anstellt oder der autonome Rasenmäher und Staubsauger uns Teile der Hausarbeit abnehmen. Diese Geräte können meist schon per Smartphone oder Computer konfiguriert und die Einstellungen nach den Bedürfnissen des Benutzers angepasst werden. Dabei interagiert der Nutzer nicht direkt mit dem Gerät, sondern mit dem Smartphone, Tablet oder Computer. Direkte Interaktionen mit den Geräten finden meist über Fernbedienungen oder Knöpfe am Gerät statt.

Hinzu kommt der aktuelle Trend des "Internet der Dinge", durch den Produkte wie Smart TVs, Kühlschränke und Pflanzentöpfe mit Internetschnittelle auf den Markt kommen. Der Personal Computer verschwindet aus unserem Denken und wird durch Smart Things (Intelligente Dinge/Geräte) ersetzt.

Deshalb wäre eine intuitive Steuerung durch Sprache und Gesten, wie wir sie teilweise durch moderne Konsolenspiele kennen, für Smart Things wünschenswert. Also eine Interaktion mit den intelligenten Dingen selbst, wie wir sie mit Menschen oder auch Tieren gewöhnt sind.

Diese Arbeit soll einen ersten Überblick über den aktuellen Stand der Technik der Sprach- und Gestensteuerung liefern. Am Beispiel eines Smart Mirrors im Badezimmer soll untersucht werden, ob Interaktionen über diese Technologien bereits möglich sind. Dabei soll die Steuerung des Systems dem Nutzer leichtfallen, und nicht erst antrainiert werden müssen.

Dafür werden konkrete kommunikative Gesten und eine freie Wahl der Formulierung der Befehle für die Bedienung entwickelt und getestet. Darüber hinaus soll das System im For-

schungslabor installiert werden, um weiteren Arbeiten und Untersuchungen zur Verfügung zu stehen.

1.2 Gliederung der Arbeit

In der Analyse (2) wird anfangs auf die aktuellen gesellschaftlichen Veränderungen durch moderne Technologien eingegangen. Daraufhin wird ein Szenario (2.1) entworfen auf dessen Basis die Aufgabenstellung (2.2) entwickelt wird. Im Anschluss wird auf die Kernthemen dieser Arbeit sowie auf verwandte Bereiche wie Human Computer Interaction (2.3), Smart Home (2.4) und AAL - Ambient Assisted Living (2.5) eingegangen. Daraufhin werden das Forschungslabor für das Smart Home der HAW (2.6) und die damit verwandten Arbeiten und Produkte (2.7) vorgestellt. Nach dem Aufzeigen der konkreten Laborumgebung und des Modells für den Smart Mirror (2.8) wird allgemein auf die Gesten- und die Spracherkennung eingegangen.

Im Kapitel 3 Design und Realisierung wird erst auf das Gesamtkonzept des Systems eingegangen und dann die für dieses Projekt wichtigen Eigenschaften und Schnittstellen des Living Places (3.2) erläutert. Die verschiedenen Anforderungen und Möglichkeiten der Authentifizierung eines solchen Systems werden im Anschluss ausgeführt. Desweiteren wird die Soft- und Hardwarearchitektur erläutert und ein Überblick auf die konkret verwendete Soft- und Hardware gegeben.

Das Evaluationskapitel (4) beginnt mit der Erläuterung der technischen Machbarkeit des Systems (4.1). Im Anschluss wird auf die Usability Aspekte, wie z.B. die Ergebnisse der Alpha-tests des Systems (4.2) genauer eingegangen.

Das Ende bildet der Schluss (5), in dem eine Zusammenfassung der Arbeit, ein Ausblick auf zukünftige Entwicklungen und aktuelle Fragestellungen zum Thema der Arbeit zu finden sind.

2 Analyse

Durch die Durchdringung der digitalen Welt in unserem Leben, werden die zeitlichen Räume unseres Denkens kleiner und die Anforderungen nach Flexibilität, Mobilität und Anpassungsfähigkeit an uns immer größer. Um diesen Erwartungen gerecht zu werden, nutzen wir moderne Technologien. So geben nach der Life Studie 85,8% aller Deutschen an:

“Der Umgang mit digitalen Medien (Internet, Computer, Mobiltelefon etc.) ist für mich ein selbstverständlicher und wichtiger Bestandteil meines täglichen Lebens.”

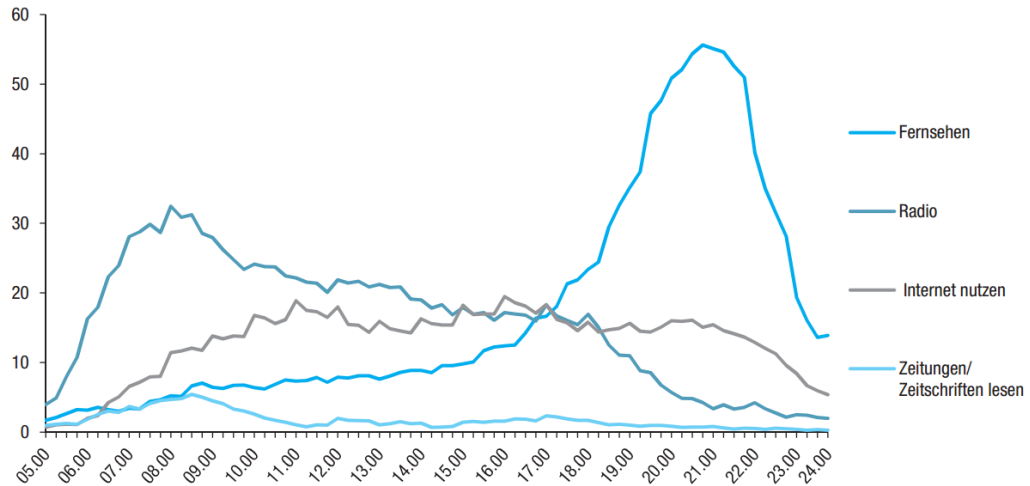
c/o Telekom AG (2013)

Zudem gaben 64% der Befragten an, dass durch technische Geräte und dem Internet sich ihre Lebensqualität erhöhen würde.

Beispielhaft hierfür ist, dass viele Menschen sich morgens schon aktuelle Informationen über das Wetter, die Verkehrslage, Nachrichten und Termine via Notebooks, Tablets oder ihren Smartphones verschaffen, bevor Sie das Haus verlassen. Einige dieser Informationen nehmen unter hohem Zeitdruck an Wichtigkeit zu, wie z.B. die aktuelle Verkehrslage oder eine Terminverschiebung. Dabei wird oftmals die Beschaffung genau dieser Daten morgens ausgelassen.

In einer zukünftigen, intelligenten und technologischen Umgebung sollten die Informationen den Menschen direkt zur Verfügung gestellt werden, ohne dass die Nutzer dafür einen großen Aufwand betreiben müssen. Ein Nachweis für die Nachfrage solcher Systeme ist die hohe Nutzung des Radios zwischen 5:00 und 10:00 Uhr morgens in Deutschland nach der ARD/ZDF-Onlinestudie 2013. Das Radio stellt Informationen bereit, welche nebenläufig von den Zuhörern aufgenommen werden können (vgl. [van Eimeren und Frees \(2013\)](#)).

Abb. 11 Mediennutzung der Onliner im Tagesverlauf "gestern" 2013
in %



Basis: Deutschspr. Erwachsene ab 14 Jahren (n=1 389).

Quelle: ARD/ZDF-Onlinestudie 2013.

Abbildung 2.1: ARD ZDF Onlinestudie 2013 (van Eimeren und Frees (2013))

Diese allgemeinen Informationen, die durch das Radio vermittelt werden, und weitere personenbezogene Daten, wie E-Mails und Termine müssen für den Nutzer intuitiv und einfach abrufbar sein, da besonders unter Zeitdruck andere Tätigkeiten im Fokus liegen. In diesem Kontext können Smart Environment und Disappearing Computing Elemente den Nutzer unterstützen.

Ein aktuelles Beispiel hierfür ist die Google Search App Erweiterung Google Now. Sie dient als persönlicher Assistent und lässt Informationen situationsbedingt erscheinen, wie z.B die Route zum nächsten Termin oder das Wetter von der aktuellen Position.

Für die Untersuchung der Möglichkeiten solcher Systeme wird in dieser Arbeit beispielhaft der Aufenthalt morgens im Badezimmer in einem Smart Home unter Einsatz eines Disappearing Computers herangezogen.

2.1 Szenario

Im folgenden Abschnitt wird ein Szenario entworfen, welches verschiedene Anforderungen der Informationsorganisation anhand eines morgendlichen Aufenthaltes im Badezimmer einer zukünftigen, intelligenten Wohnung aufzeigt. Im Mittelpunkt liegt dabei die nebenläufige Steuerung eines Systems, über welches dem Bewohner Informationen zugänglich gemacht werden, die ihm beim weiteren Verlauf des Tages unterstützen können. Dabei steht eine Interaktion mit dem System über Gesten und Sprachbefehlen im Vordergrund.

Dieses Szenario beschreibt verschiedene Anwendungsfälle, die maßgeblich in die Aufgabenstellung dieser Arbeit einfließen.

2.1.1 Der morgendliche Aufenthalt im Badezimmer einer intelligenten Wohnung der Zukunft

Nach einer erholsamen Nacht wird Herr X durch einen simulierten Sonnenaufgang in seinem Schlafzimmer behutsam geweckt. Im Anschluss einer erfrischenden Dusche beginnt Herr X mit der morgendlichen Körperpflege vor dem Badezimmerspiegel.

Durch den Ausruf "Spiegel, das Display aktivieren" erscheinen an den seitlichen Rändern des Badezimmerspiegels Informationen, wie eine Liste von E-Mails und den aktuellen Terminen sowie der Wetterbericht. In der Mitte des Spiegel-Displays ist noch ausreichend Platz für sein eigenes Spiegelbild. Während Herr X sich abtrocknet, verschafft er sich durch den Satz "Spiegel, die E-Mail-Übersicht öffnen" ein Bild über seine neusten E-Mails. Mit Anweisungen wie "Nächste E-Mail", "Letzte E-Mail" oder "Markiere die siebte E-Mail" navigiert er durch die, über dem gesamten Spiegel angezeigte, E-Mail-Übersicht. Da ihm ein Kollege eine E-Mail geschickt hat, lässt er sich mit der Anweisung "E-Mail öffnen" alle Details der E-Mail anzeigen. Nach dem Abtrocknen zieht sich Herr X seinen Bademantel über und fängt an sich seine Haare zu föhnen, nachdem er die Sprachsteuerung deaktiviert hat, die aufgrund des lauten Föhns nicht funktionieren würde. Während des Föhns schließt er durch eine Geste mit der Hand seine E-Mails und scrollt mit weiteren Gesten durch die Liste seiner Termine. In der neuen E-Mail des Kollegen wurde ein Termin für den späten Vormittag abgesagt. Diesen Termin öffnet Herr X mit seiner freien Hand und löscht ihn aus dem Kalender.

Während des Kämmens überprüft Herr X das Wetter für den Tag. Dazu aktiviert er die Sprachsteuerung und sagt "Wetter-Detailansicht anzeigen". Als er sieht, dass es inzwischen recht warm werden soll, kommt ihm die Idee, dass man am Wochenende grillen könnte. Durch den Befehl "Wetter-Übersicht öffnen" guckt er sich das Wetter für Samstag an. Mit einer einfachen

Geste schließt er die Wetter-Übersicht und öffnet den Kalender, um zu überprüfen, ob es schon andere Pläne für Samstag gibt.

Erfreut über seine neuen Pläne beendet er die Anwendung durch den Befehl "Spiegel, Display ausschalten" und startet in den Tag.

2.2 Aufgabenstellung

2.2.1 Allgemein

In diesem Kapitel werden die konkreten Fragestellungen und die daraus resultierenden Aufgabenstellungen aufgezeigt.

Das unter "Der morgendliche Aufenthalt im Badezimmer einer intelligenten Wohnung der Zukunft" (2.1) vorgestellte Szenario liefert Beispiele für verschiedene Anwendungsfälle. Der Spiegel bietet sich als Display im Badezimmer des Smart Homes an, da jedes Badezimmer einen Spiegel besitzt und die gleichzeitige Nutzung als Spiegel und Display vorteilhaft genutzt werden kann. Zudem werden durch das versteckte Display Grundsätze des Disappearing Computing mit aufgenommen.

Weitere Informationen zum Spiegel-Display wird es in dem Abschnitt Darstellung (2.8.2) geben.

Allgemein lassen sich folgende Fragestellungen formulieren:

- Welche Anforderungen müssen an ein solches System gestellt werden.
- Welche Techniken können zum Anzeigen der Informationen im Badezimmer genutzt werden.
- Welche Sensoren werden für die Steuerung benötigt.
- Welche Möglichkeiten der Interaktion muss das System anbieten.
- Gibt es bereits vergleichbare Arbeiten und Projekte.
- Welche speziellen Problematiken müssen bei der Umsetzung bedacht werden.
- Welche konkreten Anwendungsfälle lassen sich für einen Smart Mirror erstellen.
- Wie muss ein entsprechendes Test-Setting in einem Smart Home aufgebaut sein.
- Welche Sicherheitsaspekte bezüglich der personenbezogenen Daten müssen beachtet werden.

Auf diese Fragestellungen wird in der Aufgabenstellung und den weiteren Kapiteln der Analyse eingegangen und mögliche Lösungen dazu werden im Kapitel Design und Realisierung (3) vorgestellt.

2.2.2 Funktionale Anforderungen

Im Kontext des Informationszuflusses und der Steuerung eines Systems im Badezimmer einer intelligenten Wohnung gibt es eine Vielzahl von möglichen Anforderungen. Daher wird im Rahmen dieser Arbeit lediglich auf die im Szenario aufgezeigten Anforderungen eingegangen.

- Modus Wahl - Der Nutzer kann zwischen verschiedenen Modi wechseln:
 - Spiegel
 - Display
 - Mischform (Hybrid-Modus)
- Programme - Der Nutzer hat Zugriff auf beispielhafte Anwendungen:
 - E-Mails
 - Kalender
- Allgemeine Informationen - Dem Nutzer werden allgemeine Informationen bereitgestellt:
 - aktuelles Wetter und Wettervorhersagen
 - Uhrzeit
 - Datum
- Bedienung - Der Nutzer hat verschiedene Steuerungsmöglichkeiten:
 - per Gesten
 - per Sprachsteuerung

Modus-Wahl Durch die Modus-Wahl kann der Nutzer entscheiden, ob er den Smart Mirror als Spiegel oder als Display nutzen will. Die Mischform hingegen ermöglicht ihm, neben seinem Spiegelbild am Rand des Spiegels zusätzliche Teasers zu seinen E-Mails und Terminen zu sehen. Zudem hat er einen Überblick über die aktuellen Informationen.

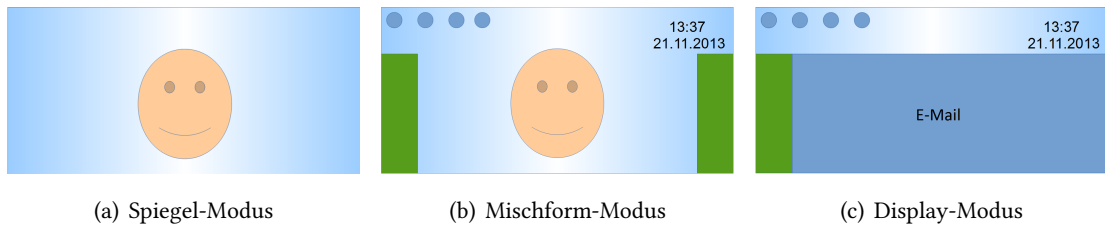


Abbildung 2.2: Vergleich der Modi

Programme Als Beispiel für Anwendungen von Programmen wurden ein E-Mail-Client und eine Kalender-Anwendung gewählt, wobei es verschiedene Ansichten für die jeweilige Anwendung gibt. Im Modus der Mischform sollen lediglich Teasers der einzelnen E-Mails und Termine angezeigt werden. Im Display-Modus kann der Nutzer eine Übersicht aller Elemente oder eine Detail-Ansicht eines Elements öffnen.

Die Neuerstellung oder Bearbeitung von Elementen ist im Badezimmer ungeeignet, da die Eingabe von Text keine Nebentätigkeit ist. Lediglich eine Option zum Löschen von Elementen wie E-Mails und Terminen wird dem Nutzer bereitgestellt.

Allgemeine Informationen Die aktuellen Informationen werden bereits in der Mischform angezeigt, nur das Wetter bietet noch zwei weitere Optionen für eine Detailansicht des Tages oder eine Wochenübersicht der Wettervorhersage.

Bedienung Wie im Szenario dargestellt, soll es dem Nutzer möglich sein, per Sprach- und Gestensteuerung mit dem Smart Mirror zu interagieren. Dabei müssen sich die beiden Arten der Steuerungen auch deaktivieren lassen.

Zudem sollte sichergestellt werden, dass der Smart Mirror auch nur durch nur eine der beiden Steuerungsmöglichkeiten vollständig bedient werden kann. Als Ausnahme dieser Anforderung steht das Aktivieren der jeweiligen Steuerungsart, wenn diese bereits deaktiviert ist.

Die Bedienung über Gesten sollte mit einer Hand umgesetzt werden können, da die zweite Hand frei für andere Tätigkeiten sein muss. Die Sprachsteuerung sollte eine Vielzahl von Satzstrukturen und Wörtern erkennen und dadurch für den Nutzer frei und einfach gestaltet sein.

2.2.3 Interaktionsrepertoire

Für eine Bedienung eines Smart Mirrors im Badezimmer wurden für diese Arbeit im Modell folgende Interaktionsarten festgelegt:

Markieren

Selektierbarkeit von Buttons, E-Mails und Terminen muss gewährleistet werden, um dem System mitzuteilen, auf welches Element sich eine Aktion bezieht. Zudem kann über eine Markierung im User Interface dem Nutzer eine visuelle Rückmeldung gegeben werden, wie das System die letzte Interaktion des Nutzers interpretiert hat.

Aktivieren

Markierte Elemente müssen aktivierbar sein, um eine Auswahl bestätigen zu können. Dies wird in grafischen Oberflächen meist durch einen doppelten Maus-Klick umgesetzt, bei einem gestengesteuerten System müssen hier für entsprechende Analogien gefunden werden. Durch das "Klicken" wird dem System mitgeteilt, dass das markierte Element aktiviert werden soll.

Bildlauf (Scrolling)

Da die Größe der Oberfläche begrenzt ist, muss der Nutzer die Möglichkeit haben, Listen und lange Texte zu scrollen. Bei grafischen Oberflächen wird dies durch Klicken auf die Scrollbar oder durch das Drehen des Mousrads bewerkstelligt. Bei Touch Displays wird zum Scrollen oftmals eine gradlinige Wisch-Geste verwendet.

2.2.4 Rahmenbedingungen

Bei der Realisierung eines Smart Mirrors im Badezimmer sind folgende Rahmenbedingungen zu beachten:

Position Der Smart Mirror wird über das Waschbecken als Badezimmerspiegel angebracht. Die Problematik der Wasserbeständigkeit des Systems wird in dieser Arbeit außen vor gelassen.

Interaktionsbereich Der Nutzer kann sich nur in bestimmten Winkeln im Spiegel sehen. Zudem steht der Nutzer im Normalfall maximal 1,5 Meter weit vom Spiegel entfernt. Daher müssen die Sensoren nur einen beschränkten festen Bereich abdecken (siehe Abbildung 2.2.4).

Berührungslose Steuerung Wie im Abschnitt Bedienung im Kapitel Funktionale Anforderungen (2.2.2) beschrieben, wird die Steuerung per Gesten- und Sprachbefehlen umgesetzt. Maus, Tastatur und Fernbedienungen sind als Eingabegeräte im Badezimmer ungeeignet. Außerdem wäre die Umsetzung der Disappearing Computing Aspekte eines Smart Homes (2.4) mit diesen Geräten nicht möglich.

Ein Mirror mit Touch Display, wie er im Projekt Interavtive Mirror (2.7.3) genutzt wurde, würde unangenehme Wasser- und Fettflecken auf dem Spiegel hinterlassen. Daher eignet sich diese Bedienung ebenfalls nicht für das Badezimmer.

Einpersonenhaushalt In dieser Arbeit wird von einem Einpersonenhaushalt ausgegangen. Auf eine Untersuchung der Nutzung des Systems von verschiedenen Benutzern (Mandantenfähigkeit) wird daher im Rahmen dieser Arbeit verzichtet. Weitere Details dazu folgen im Kapitel Living Place (2.6).

Authentifizierung Da mehrere Personen, wie zum Beispiel Gäste, das Badezimmer nutzen, muss sichergestellt werden, dass die privaten Daten des Bewohners nicht jedem Nutzer des Badezimmers zur Verfügung stehen.

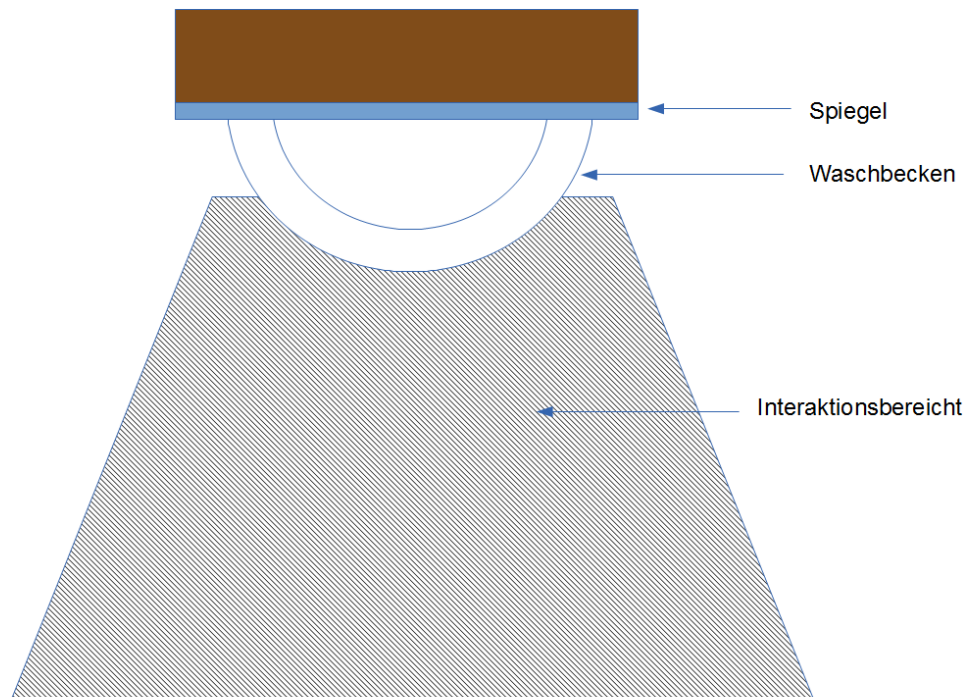


Abbildung 2.3: Interaktionsbereich

2.2.5 Anwendungsfälle

Im Folgenden wird auf einige beispielhafte Anwendungsfälle eingegangen. Abbildung 2.4 ist eine allgemeine Darstellung für alle Anwendungsfälle.

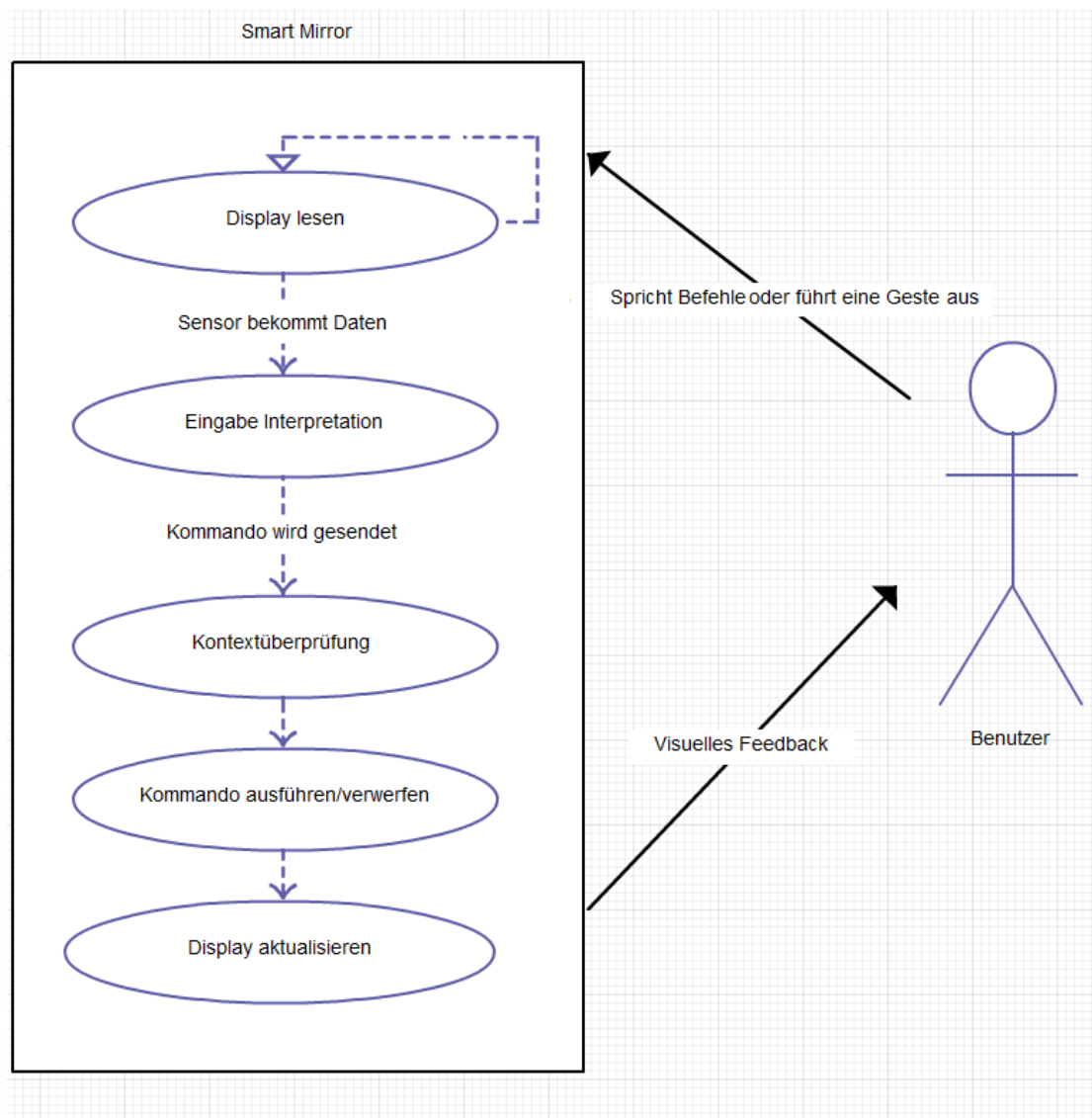


Abbildung 2.4: Allgemeine Darstellung des Use Cases

In allen Anwendungsfällen befindet sich der Bewohner (A) des Smart Homes in seinem Badezimmer vor dem Smart Mirror (S).

E-Mails lesen

Beschreibung Der Nutzer möchte eine ausgewählte E-Mail über die Gestensteuerung zum Lesen öffnen.

Vorbedingungen Der Nutzer befindet sich im Interaktionsbereich des Smart Mirrors und das System befindet sich im Hybrid-Modus.

Verwendete Anwendungsfälle Keine

Ergebnis Der Smart Mirror zeigt im Display-Modus die Details der ausgewählten E-Mail an.

Standardablauf Der Nutzer (A) hebt eine Hand um den Smart Mirror (S) zu steuern. S erkennt die Hand als aktive Steuerungshand und gibt A eine visuelle Rückmeldung durch das Erscheinen des Cursors, welcher sich der Bewegung der Hand von A anpasst. A navigiert den Cursor auf den Button der vorbestimmten E-Mail in der Liste. Nun führt A eine Klickgeste durch, in dem er die Hand nach vorne schiebt, S erkennt diese und öffnet die Detailansicht der E-Mail im Display-Modus.

E-Mails löschen

Beschreibung Der Nutzer möchte eine geöffnete E-Mail über die deaktivierte Sprachsteuerung löschen.

Vorbedingungen Der Smart Mirror zeigt im Display-Modus die Details der zu löschenden E-Mail an.

Verwendete Anwendungsfälle E-Mail lesen

Ergebnis Die Sprachsteuerung ist aktiviert und der Smart Mirror zeigt im Hybrid-Modus die aktualisierte E-Mail Liste an.

Standardablauf Der Nutzer (A) navigiert den Cursor auf den Button der Sprachsteuerung am oberen Rand des Smart Mirrors (S). Nun führt A eine Klickgeste durch, S erkennt diese und aktiviert die Sprachsteuerung und teilt A durch eine Änderung des Buttons den Erfolg der Aktivierung mit. A spricht laut "Die E-Mail löschen" aus. S erkennt den Sprachbefehl und führt die Aktion für die geöffnete E-Mail aus. Als Rückmeldung zeigt S oben ein Fenster mit der Information, welcher Befehl erkannt und ausgeführt wurde.

Wettervorhersage lesen

Beschreibung Der Nutzer möchte mit der Gestensteuerung zur Wettervorhersage für den Samstag in 10 Tagen navigieren.

Vorbedingungen Der Smart Mirror ist im Spiegel-Modus und die Gestensteuerung ist deaktiviert.

Verwendete Anwendungsfälle Keine

Ergebnis Die Gestensteuerung wird aktiviert und der Smart Mirror zeigt im Display-Modus das Wetter für den Samstag in 10 Tagen an.

Standardablauf Der Nutzer (A) aktiviert den Hybrid-Modus des Smart Mirrors (S) durch den Sprachbefehl "Das Display aktivieren". S erkennt den Sprachbefehl und startet im Hybrid-Modus das System. A sagt: "Gestensteuerung aktivieren", daraufhin aktiviert S die Gestensteuerung und ändert das Symbol für den Status der Gestensteuerung als Rückmeldung für A. A hebt eine Hand, um S zu steuern. S erkennt die Hand als aktive Steuerungshand und gibt A eine visuelle Rückmeldung durch das Erscheinen des Cursors, welcher sich der Bewegung der Hand von A anpasst. A navigiert den Cursor auf die Wetteranzeige. Nun führt A eine Klickgeste durch, S erkennt diese und öffnet die Wetter-Detailansicht im Display-Modus. A navigiert nun auf den Button "Wochenübersicht" und führt nochmals eine Klickgeste durch. S öffnet daraufhin die Wochenübersicht der Wettervorhersage. A navigiert auf die Liste der Tage und greift mit der Hand zu. Diese Greifgeste wird von S als Start der Scrollgeste erkannt und die folgenden horizontalen Bewegungen der geschlossenen Hand von A werden von S als Scrollen interpretiert. A bewegt die Hand nach links bis auf dem Display der gewünschte Tag zu sehen ist. A öffnet die Hand, S erkennt dies als Ende der Scrollgeste an. Nun kann A das Wetter für den Samstag in 10 Tagen lesen.

2.2.6 Zielsetzung

Ziel dieser Arbeit ist die Konzipierung und Untersuchung der Interaktionsmöglichkeiten eines Smart Mirrors im Badezimmer in einer Smart-Home-Umgebung. Das System soll den Bewohnern einer zukünftigen Wohnung die Informationsaufnahme während des Badezimmersaufenthaltes ermöglichen. Dies soll so intuitiv und nahtlos geschehen, dass der Nutzer die Steuerung beiläufig neben den gewöhnlichen Tätigkeiten vor dem Badezimmerspiegel ausführen kann.

Der Fokus der Arbeit liegt daher auf der Konzeption, Entwicklung und Evaluation der Ansteuerung des Systems über 3D-Gesten und Sprachbefehle.

Ebenfalls sollte das System, wenn es erfolgreich umgesetzt werden kann, in den Living Place (2.6) einfließen und für weitere Arbeiten und Untersuchungen zur Verfügung stehen. Dies gilt nicht nur für die Hardware, wie das Spiegel-Display, sondern auch für die Dienste der Sensordatenerfassung und der Dateninterpretation. Auf Grund dessen ist eine lose Koppelung in der Architektur des Gesamtsystems umzusetzen.

Als weiteres Feature könnten evtl. einzelne Services wie eine Wetterdienst-Webschnittstelle dem Living Place zur Verfügung gestellt werden.

2.3 HCI - Human Computer Interaction

2.3.1 Allgemein

Human Computer Interaction (Deutsch: Mensch-Computer-Interaktion) ist ein Teilgebiet der Informatik, welches sich mit der Interaktion zwischen den Benutzern und Computersystemen auseinandersetzt. HCI beschreibt ebenfalls Möglichkeiten der Schnittstellen zwischen Computern und den Menschen. Daher ist HCI ein hochgradig interdisziplinärer Bereich, in dem Informatiker, (Medien-) Psychologen, Designer, Soziologen und Anthropologen zusammenarbeiten.

Ziel ist es, die Nutzer durch möglichst intuitive und einfache Technologien zu unterstützen. Dabei spielt die Usability eines Systems eine große Rolle.

Allgemein bekannt sind Schnittstellen wie Maus, Tastatur, und Touchscreens vom Menschen zum Computer. 3D-Gesten und Sprachinteraktionen gibt es bisher nur in speziellen Gebieten wie Gaming oder die Sprachwahl beim telefonischen Support. Schnittstellen vom Computer zum Menschen sind für gewöhnlich Displays, Lautsprecher, Licht- und Vibrationssignale.

2.3.2 HCI am Smart Mirror

Da die grundsätzlich genutzten Eingabegeräte weder intuitiv, noch für die Nutzung im Badezimmer geeignet sind, werden in dieser Arbeit die Interaktionen von 3D-Gesten und Sprachbefehlen anhand eines computergestützten Badezimmerspiegels untersucht. Hierbei werden

verschiedene natürliche Eingabemöglichkeiten bereitgestellt und mit einem Spiegel-Display als Ausgabemedium untersucht. Der User soll den Spiegel wie einen gewöhnlichen Badezimmerspiegel nutzen können, zusätzlich jedoch die Möglichkeit haben, sich bei Bedarf weitere Informationen über das Display anzeigen zu lassen. Die Navigation sollte möglichst nahtlos (Seamless Interaction) über freie Sprachbefehle und 3D-Gesten erfolgen.

Da der Nutzer im Normalfall vor dem Badezimmerspiegel verschiedene Tätigkeiten ausführt, muss die Bedienung so leicht, natürlich und intuitiv sein, dass die Bedienung beiläufig getätigt werden kann.

Die Spracherkennung soll auf der ersten Interpretationsebene durch eine Grammatik mit einer großen Auswahl von Satzstrukturen umgesetzt werden. Dies soll den Nutzer eine freie und natürliche Bedienung des Smart Mirrors per Sprache ermöglichen.

Beispiele zum Öffnen der dreizehnten E-Mail:

1. "Öffne die dreizehnte E-Mail"
2. "Die dreizehnte E-Mail öffnen"
3. "Die dreizehnte E-Mail anzeigen"
4. "Wechsel zur dreizehnten E-Mail"
5. "Zeige die dreizehnte E-Mail"
6. ...

Wenn eine E-Mail bereits markiert ist, sollte es reichen "Öffne die E-Mail" oder "Die E-Mail anzeigen" zu sagen, da das System aus dem Kontext wissen müsste, welche E-Mail gemeint ist. Eine detaillierte Beschreibung der Anforderung einer solchen Spracherkennung befindet sich im Kapitel Spracherkennung (2.10) der Analyse.

Die Steuerung durch Gestenerkennung soll dem Nutzer ebenfalls leichtfallen. Für diesen Zweck werden manipulative Gesten (vgl. Boetzer u. a. (2008)) genutzt, sowie bewegungsverfolgende Gesten für die Navigation und kontinuierliche Gesten zum Auswählen und Scrollen verwendet. Auf weitere Details wird im Kapitel 3D-Gesten (2.9.5) eingegangen.

2.4 Smart Home

2.4.1 Allgemein

Unter dem Begriff Smart Home wird intelligentes Wohnen verstanden, welches das Ziel hat, die Wohn- und Lebensqualität für die Bewohner zu erhöhen. Dazu werden computergesteuerte Automatisierungstechniken von Geräten und Mechaniken eingesetzt. Haustechnik, Haushaltsgeräte und Unterhaltungstechnik werden vernetzt und angesteuert, um dies zu ermöglichen.

Da die Nutzung neuer Technologien wie Smartphones und soziale Medien kulturelle und soziale Veränderungen in modernen Gesellschaften bewirken, ist die Integration von neuen vernetzten Technologien im Wohnkontext ein sinnvoller Schritt.

Weitere Informatikbereiche wie Seamless Interaction und Disappearing Computing nehmen auf den Smart-Home-Sektor großen Einfluss, da die Systeme nahtlos in die Abläufe der Nutzer integriert werden sollen und diese vom Nutzer nicht primär über einen Personal Computer gesteuert werden, sondern über NUI (Natural User Interface) oder automatisiert und intelligent geregelt werden.

2.4.2 Smart Mirror im Smart Home

Der Smart Mirror kann im Rahmen eines Smart Homes den Bewohner Informationen und Benachrichtigungen über seine Wohnung, seiner sozialen Interaktionen und sein zeitliches Management geben. In dieser Arbeit wird daher ein Prototyp eines solchen Smart Mirrors für ein Badezimmer in einem Smart Home konzipiert und auf die Bedienbarkeit hin getestet. Dabei liegen Problemstellungen wie die Wasserbeständigkeit des Spiegels, oder die Steuerung anderer Systeme des Smart Homes nicht im Fokus dieser Arbeit.

2.5 AAL - Ambient Assisted Living

Ambient Assisted Living (AAL, Deutsch: Altersgerechte Assistenzsysteme für ein selbstbestimmtes Leben) bezeichnet die Entwicklung von Technologien für die nahezu unbemerkte Unterstützung von Menschen mit Behinderungen oder altersbedingten Bewegungseinschränkungen. Dazu werden, unter Einsatz moderner Technologien, intelligente Assistenzsysteme entwickelt, um die Lebensqualität des Nutzers zu verbessern. Diese sollen ihm helfen, länger in seiner gewohnten Umgebung leben zu können.

Das Bundesministerium für Bildung und Forschung erläutert AAL folgendermaßen:

"Unter 'Ambient Assisted Living' (AAL) werden Konzepte, Produkte und Dienstleistungen verstanden, die neue Technologien und soziales Umfeld miteinander verbinden und verbessern mit dem Ziel, die Lebensqualität für Menschen in allen Lebensabschnitten, vor allem im Alter, zu erhöhen. Übersetzen könnte man AAL am besten mit 'Altersgerechte Assistenzsysteme für ein gesundes und unabhängiges Leben'. Damit wird auch schon skizziert, dass AAL in erster Linie etwas mit dem Individuum in seiner direkten Umwelt zu tun hat."

Innovation + Technik GmbH (2013)

Die Unterstützung sollte situationsabhängig und unaufdringlich auf den Nutzer zugeschnitten sein. Um sich den Bedürfnissen des Nutzers besser anzupassen, ist es sinnvoll verschiedene Systeme miteinander zu vernetzen, und somit die Umgebung des Nutzers besser zu erfassen. AAL ist ein verwandter Bereich zu Smart Home, da die eingesetzten Techniken und Aufgabenstellungen oftmals ähnlich sind (vgl. **Müller (2013)** und **Berndt u. a. (2009)**).

2.6 Labor Living Place an der HAW Hamburg

2.6.1 Allgemein

Das Departement Informatik hat zur Untersuchung für zukünftiges Leben im urbanen Raum seit 2009 ein eigenes Forschungs- und Usability Labor, das Living Place Hamburg. In dem ca. 140 m² großen Studio-Appartment werden Untersuchungen in Bereichen wie Seamless Interaction, Ubicuitous Computing, Dissappearing Computing, Context Awareness, Tangible Interfaces und Companion Systems im Kontext von Smart Living und Ambient Assisted Living im digitalen Zeitalter durchgeführt.



Abbildung 2.5: Living Place Hamburg

Die Loftwohnung, die komplett mit Küche, Bad, Schlaf- und Wohnbereich ausgestattet ist, wurde in Zusammenarbeit von Medientechnikern, Designern und Informatikern der HAW Hamburg konzipiert und umgesetzt. Wie in [Abbildung 2.6](#) zusehen ist, gibt es einen Bereich von ca. 130 m², in dem ein Kontrollraum, eine Werkstatt (südlicher Entwicklungsraum) und zwei Entwicklungsbüros zur Verfügung stehen. Über den Kontrollraum lassen sich Sensoren und Aktoren überwachen und ansteuern. Durch die Vorbelegung als Hausmeisterwohnung ist die Infrastruktur der Wohnung ideal für Untersuchungen im realen Wohnungskontext, da ein temporäres Leben in der Wohnung für Tests und Evaluationszwecke möglich ist.

Auf die Sensor- und Aktorenvelfalt des Living Places wird an dieser Stelle nicht weiter eingegangen, da erst durch den Smart Mirror erste Sensoren und Aktoren im Badezimmer eingebaut werden. Weitere Informationen dazu gibt es im Kapitel “Living Place als Umgebung des Smart Mirrors” ([2.8.1](#)).

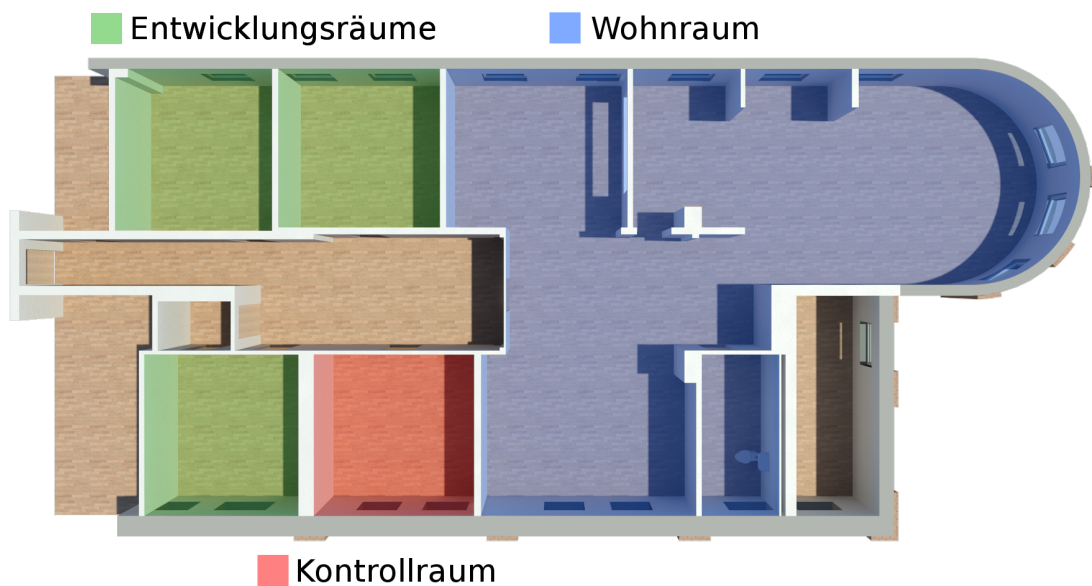


Abbildung 2.6: Übersicht des Living Place Hamburg

Fachübergreifend forschen Informatiker, Designer, Soziologen, Medientechniker und Künstler unter der Leitung von Kai von Luck und Gunter Klemke an der Wohnung von Morgen. Finanziert wird das Projekt maßgeblich durch die Finanzbehörde und die Behörde für Wissenschaft und Forschung, wie auch von einigen Unternehmen.

Weitere Informationen zum Living Place sind unter <http://livingplace.informatik.haw-hamburg.de/blog/> zu finden.

2.6.2 Infrastruktur des Living Place

Die Infrastruktur ist nach **Rahimi und Vogt (2011)** "Seamless Interaction - Natürliche Interaktionen in Smart Living Umgebungen" in drei Teile gegliedert:

- Kommunikation bzw. Vermittlung
- Sensoren
- Aktoren

Die Kommunikation wird von einem Messagebroker, einem ActiveMQ realisiert, über den die Daten von Sensoren abrufbar und Aktoren ansteuerbar sind. Als Format der Messages, die über den ActiveMQ gesendet werden, wurde JSON gewählt.

Über diese Kommunikationsstruktur werden die verschiedenen Dienste des Smart Mirrors mit den Sensoren, der Kinect und dem Leap Motion Controller sowie den Aktor, dem Spiegel-Display, kommunizieren.

Eine detaillierte Beschreibung der Kommunikation befindet sich im Kapitel Design und Realisierung (3.2.2).

2.7 Verwandte Arbeiten und Produkte

2.7.1 Verwandte Arbeiten an der HAW

Im Rahmen des Living Places und anderen Laboren der HAW Hamburg wird bereits seit mehreren Jahren die Mensch-Computer-Interaktion durch neue Technologien untersucht. Im Folgenden werden einige dieser Arbeiten vorgestellt. Außerdem wird kurz auf Überschneidungen mit und Abgrenzungen zu dieser Bachelorarbeit eingegangen.

J. Boetzer - Bewegungs- und gestenbasierte Applikationssteuerung auf Basis eines Motion Trackers

In der Bachelorarbeit von **Boetzer (2008)** wurde untersucht, in wieweit sich verschiedene Anwendungen durch Gesten bedienen lassen. Dazu wurden Usability-Tests durchgeführt, um zu testen, welche Gesten bzw. Bewegungen intuitiv bedienbar und performant für bestimmte Aktionen innerhalb von Spielen und Applikationen sind. Zudem wird auf die verschiedenen Arten von Gesten, wie auch Problemfelder, beispielsweise die Start-End-Problematik von 3D-Gesten, eingegangen.

Problematiken wie das Start-End-Problem und die Gestenerkennung im 3D-Raum sind ebenfalls im Kontext eines Smart Mirrors im Badezimmer eines Smart Homes zu lösen. Die in der Arbeit von **Boetzer (2008)** vorgestellte Lösung des Start-End-Problems lässt sich leider nicht auf den Smart Mirror übertragen, weil der Ansatz darauf aufbaut, dass jede Bewegung als Anfang einer Geste interpretiert wird, und eine Bewegungspause das Ende einer Geste darstellt. Da aber der Nutzer vor dem Smart Mirror auch andere Tätigkeiten und Bewegungen ausführt, müssen andere Lösungen für dieses Problem gefunden werden.

Olaf Potratz - Ein System zur physikbasierten Interpretation von Gesten im 3D-Raum

In der Bachelorarbeit von [Potratz \(2011\)](#) wurden die Möglichkeiten von der Interpretation von 3D-Gesten in einem physikbasierten System untersucht. Die Gestenerkennung wurde mittels des ARTtrackers umgesetzt.

Die untersuchten Gesten der verwandten Arbeit sind: Schieben, Greifen, Heben und Loslassen als zustandsändernde Anweisung für die Physik-Engine, um Manipulationen im Raum durchzuführen.

Die Programmstruktur und Architektur angelegt an dem MVC-Entwurfsmuster, erweitert durch eine Observer-Komponente, ist ebenfalls vorteilhaft für die Softwarearchitektur des Smart Mirrors, da in der Arbeit von [Potratz \(2011\)](#) einen ActiveMQ als Messagebroker genutzt wird und es sich ebenfalls um ein gekapseltes System handelt, welches mit 3D-Gesten gesteuert werden kann. Das ARTtracker-System ist jedoch durch die Nutzung von Markern, die der Nutzer in oder an der Hand tragen muss, für den Einsatz im Badezimmer ungeeignet.

Johann Heitsch - Ein Framework zur Erkennung von dreidimensionalen Gesten

In der Masterarbeit von [Heitsch \(2008\)](#) wurde ein Framework auf Basis einer Gestenerkennungsmaschine realisiert. Außerdem wird ein Überblick über die verfügbaren Motion-Tracking-Technologien wie unter anderem dem Flystick und der WiiMote gegeben.

Die in der Arbeit von Herrn Heitsch eingesetzten Motion Tracker sind inzwischen teilweise veraltet und benötigen zusätzliche Controller/Gamepads/Fernbedienungen, welche für die Badezimmer-Nutzung ungeeignet sind.

Die Start-Stop-Erkennung einer 3D-Geste wird über einen Knopf des WiiMote Controllers ausgelöst. Dieser Ansatz kann daher nicht für den Smart Mirror genutzt werden.

Arne Bernin - Einsatz von 3D-Kameras zur Interpretation von räumlichen Gesten im Smart Home Kontext

In der Masterarbeit von [Bernin \(2011\)](#) wurden die Einsatzmöglichkeiten von 3D-Kameras für die Gestenerkennung im Kontext eines Smart Homes untersucht. Die Problematik der Latenzzeiten für die Verarbeitung wird untersucht und verschiedene Verfahren für die Gestenerkennung mittels 3D-Kameras werden vorgestellt. Dabei werden unter anderem die verschiedenen Latenzzeiten und optische Verfahren des 3D-Trackings verglichen und untersucht.

Während der Schwerpunkt der Arbeit von [Bernin \(2011\)](#) auf einem technischen Vergleich zwischen verschiedenen 3D-Kameras liegt, wird in dieser Arbeit der Umfang des SDKs und die Größe der Entwickler Community eine vorrangige Rolle spielen, da eine konkrete Anwendung entwickelt werden soll. Zudem sind inzwischen weitere Tracking-Sensoren auf dem Markt verfügbar.

Kristoffer Witt - Kontextabhängige multimodale Interaktion mit Schwerpunkt Spracherkennung im Smart-Home Umfeld

In der Masterarbeit von [Witt \(2011\)](#) wurden die Möglichkeiten einer Sprachsteuerung in einer Smart-Home-Umgebung untersucht. Zudem wurde zur Veranschaulichung des Konzepts ein Prototyp gebaut. Außerdem wird gezeigt, wie sich die Robustheit von Spracherkennung durch Grammatiken und dem Kontext des Befehls verbessern lassen.

In der Arbeit von [Witt \(2011\)](#) kam bereits die MS Speech API zum Einsatz für eine Spracherkennung eines Systems im Smart-Home-Bereich, bei dem Grammatiken mit Semantik benutzt wurden, um eine möglichst offene und freie Sprache als Befehlseingabe umsetzen zu können. Dieser Ansatz kann ebenfalls für die Sprachsteuerung des Smart Mirrors übernommen werden.

2.7.2 Weitere verwandte Arbeiten

Weltweit gibt es eine ganze Reihe von Arbeiten, die sich mit Motion Tracking und 3D-Gesten auseinandergesetzt haben. Wichtige Werke sind unter anderen "Finger mouse: A freehand pointing interface" von T.A. Mysliwiec, "Understanding Motion Capture for Computer Animation and Video Games." von Menache und "Center for Gesture and Speech Research" von McNeill.

Im Folgendem liegt der Fokus daher auf Arbeiten, die sich mit Smart Mirrors oder Mirror Displays auseinandersetzen.

Magic Mirror Table with Social-Emotion Awareness for Smart Home

In dem Projekt von [Yu u. a. \(2012\)](#) wurde getestet, ob ein interaktiver Spiegel im Smart Home Kontext die Emotionen der Benutzer erkennen und beeinflussen kann. Als Analogie wurde der Zauberspiegel der Königen aus dem Märchen "Schneewittchen" verwendet. Der Spiegeltisch besteht ebenfalls aus einem Einweg Spiegel, der vor einen LCD montiert wurde. Über eine Gesichtserkennung wird der User authentifiziert. Das System erkennt die Emotion aus Kamerabildern. Bei einem traurigen Nutzer werden aufheiternde Wörter auf dem Spiegel-Display

angezeigt und vorgelesen. Zusätzlich wird die Lieblingsmusik des Nutzers im Hintergrund abgespielt.

Das System kann zwischen vier folgenden Modi wechseln:

1. Mirror Mode - Der Bildschirm ist ausgeschaltet. Daher verhält sich der Magic Mirror wie ein Spiegel. Über die Kamera werden Fotos geschossen, über die die Emotionen des Users analysiert werden. Je nach Ergebnis der Analyse kann der Magic Mirror in den Alleviation Mode wechseln.
2. Alleviation Mode - Eine Textnachricht wird auf dem Spiegel angezeigt. Außerdem wird eine Sprachnachricht und Musik über die Lautsprecher abgespielt.
3. Reminder Mode - In diesem Modus kann der Nutzer seinen Kalender einsehen und bearbeiten.
4. Silence Mode - In den Silence Mode geht das System, wenn der Nutzer außerhalb der Reichweite des Spiegels ist. Der Bildschirm und die Kamera werden ausgeschaltet. Sollte der Bewegungsmelder eine Bewegung erkennen, wechselt das System in den Mirror Mode.

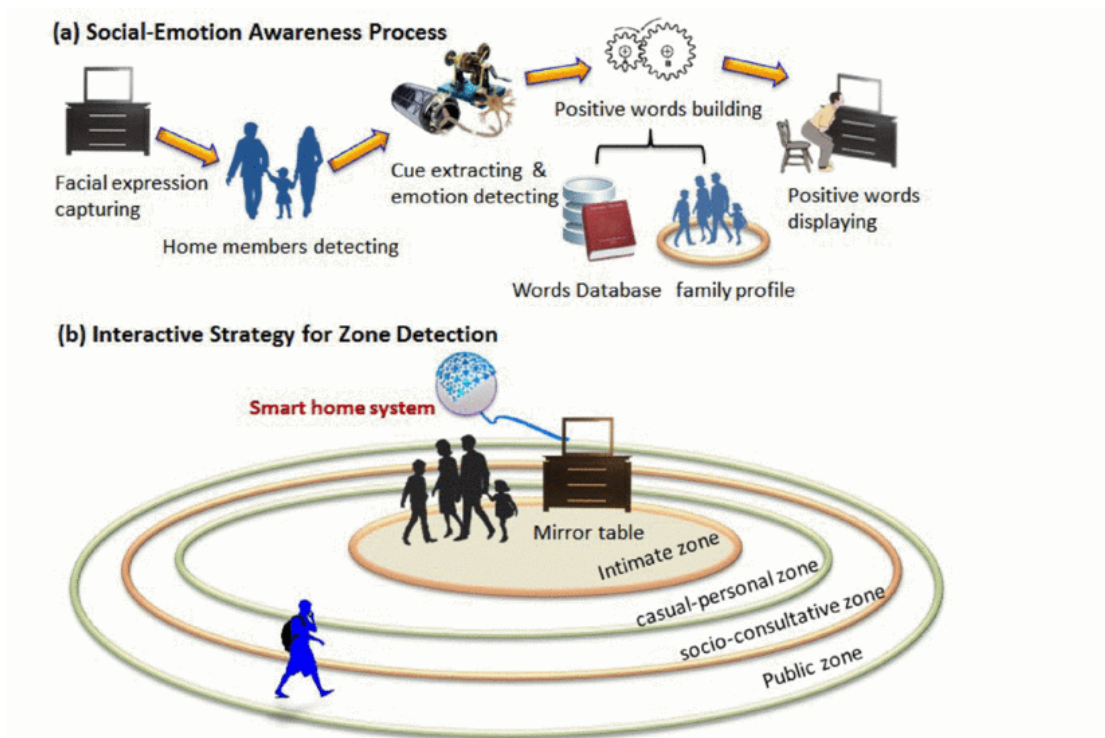


Abbildung 2.7: The Social-Emotion Awareness Process for Magic Mirror Table (a) and the Interactive Strategy for Zone Detection (b) [Yu u. a. (2012)]

Um die emotionalen Auswirkungen des Systems auf den Nutzer zu untersuchen, wurden die Herzfrequenzen gemessen. Im evaluierten Testszenario konnte der Magic Mirror traurige Nutzer aufheitern.

Im Gegensatz zu dem in dieser Arbeit konstruierten Smart Mirror kann das Spiegel-Display lediglich an oder aus sein. Dadurch ist es nicht möglich einen Teil des Displays als Spiegel und den anderen Teil als Display zu verwenden, wie in Abbildung 2.2 beschrieben. Die Interaktion über die Emotion ist außerdem auf einer anderen Ebene als die Steuerung eines Informationssystems über Gesten oder Spracherkennung.

Der Ansatz die Emotionen mit in die Steuerung einzubeziehen, könnte im Smart-Living-/Smart-Environment-Kontext jedoch zukünftig eine wichtige Rolle spielen. Das System könnte darüber Informationen filtern, die dem Nutzer angezeigt werden, oder die aktuelle Emotion als Information an ein Companion-System weiterleiten. Solche Ansätze werden bereits im Kontext von Affective Computing, Emotionale Modellierung und HCI an der HAW durch Arbeiten

wie “Emotionale Modellierung in Mensch-Maschine-Interaktionen” von Müller (2013) oder “A Framework Concept for Emotion Enriched Interfaces” von Bernin (2012) untersucht (vgl. Yu u. a. (2012)).

Augmented Rendering of Makeup Features in a Smart Interactive Mirror System for Decision Support in Cosmetic Products Selection

In diesem Projekt wurde ein “smart interactive mirror” (SIM) entwickelt und untersucht, welcher den Benutzer bei der Auswahl von Kosmetikartikeln unterstützen soll. Wie auch beim Magic Mirror Table wird ein halbdurchlässiger Spiegel vor einem Bildschirm montiert. Durch IR Sensoren, RFIDs und eine Kamera wird die Gesichtsposition und der vom User ausgewählte Kosmetikartikel und deren Position berechnet.

Die Arbeit von Rahman u. a. (2010) ist ein gutes Beispiel für das Einsetzen von Augmented Reality im Smart-Home-/Smart-Mirror-Bereich, da der Controller wie in der Realumgebung genutzt wird, und das Ergebnis ebenfalls im Spiegel zu sehen ist.

Der Augmented-Reality-Ansatz wird in dieser Arbeit nicht weiter betrachtet, da im Rahmen dieser Untersuchung die Bedienbarkeit der konkreten Anwendung im Vordergrund liegt.

Smart Mirror For Ambient Home Environment

In dieser Studie von Hossain u. a. (2007) wurde anhand eines simulierten Mirror Displays eine serviceorientierte Architektur für einen Mirror UI getestet.

Das System bietet folgende Dienste:

- Nachrichten/RSS feeds
- Wetterinformationen
- Multimedia-Service
- Lichtkontrolle (an/aus)
- Arzneischränk (auf/zu)
- Sicherheitskamera

Zusätzlich wurde eine Authentifikation per Gesichtserkennung umgesetzt. Über einen Web-Service als Message Bus konnte der Spiegel auf das Multimedia Center, die Authentifikation und das Internet zugreifen.

Die Studie von [Hossain u. a. \(2007\)](#) unterscheidet sich in einigen Punkten von dieser Arbeit, da der Fokus auf der Authentifikation und der Darstellung von personenbezogenen Informationen lag. Es wurde ein einfacher flacher Touchscreen benutzt und zur Simulation eines Spiegels ein Livestream auf dem Bildschirm angezeigt und kein Spiegel-Display wie in dieser Arbeit beim Smart Mirror eingesetzt.

2.7.3 Verwandte Produkte

Cybertecture Mirror

Der Cybertecture Mirror ist ein Mirror Display, den man seit Anfang 2011 bestellen kann. Mit einer Breite von 50,7 cm, einer Höhe von 81,5 cm und einer Tiefe von 8 cm soll der Spiegel über dem Waschbecken im Badezimmer aufgehängt werden. Das Spiegel-Display wird mit einer wasserdichten Fernbedienung oder einem Smartphone bedient. Ein "Health" Sensor ist ebenfalls im Lieferumfang enthalten. Über Ethernet oder WLAN wird der Spiegel mit dem Internet verbunden.

Folgende Funktionen soll der Spiegel erfüllen:

- Health - Der Sensor kann das Gewicht messen, und berechnet den BMI, den Fett-, Knochen- und Wasseranteil des Benutzers. Diese Daten werden dann in einer Statistik ausgewertet, die auf dem Spiegel-Display angezeigt werden können. Außerdem kann man sich die Daten über eine Weboberfläche ansehen. Die Vision ist, damit die telemedizinischen Technik voranzutreiben, die eine Behandlung trotz räumliche Trennung von Arzt und Patient ermöglichen.
- Entertainment - Über das Internet kann der User sich Video-Streams angucken. Zwei interne Lautsprecher geben den Sound wieder.
- Information - Uhrzeit, Wettervorhersage und soziale Medien wie Facebook oder Twitter können eingebunden werden. Die Benachrichtigungen werden im systemeigenen "Feed" angezeigt.
- Exercise - Vor dem Spiegeln können verschiedene Fitnessübungen durchgeführt werden. Diese werden durch Anweisungen und Hilfslinien im Spiegelbild unterstützt.

Die nahtlose Interaktion wird durch das Bedienen einer Fernbedienung stark beeinträchtigt und eignet sich daher nicht für eine Verwendung beim Smart Mirror. Der Vorteil einer solchen

Fernbedienung ist jedoch, dass keine weiteren Sensoren zur Ansteuerung benötigt werden. Das System ist als Blackbox konzipiert, lässt sich jedoch zusätzlich durch Android Apps ansteuern (vgl. [Cybertecture \(Mirror\) Ltd \(2010\)](#)).

Interactive Mirror

Der Interactive Mirror wurde von Lit Studios und Interference inc entwickelt und ist ein Touch-Mirror Display. Bei diesem Projekt ging es um eine tiefere Untersuchung von HCI. Durch das Benutzen eines Touchscreens, dessen Berührung dann als Maus-Aktionen interpretiert wird, ist das System dynamischer als der Cybertecture Mirror und intuitiver zu bedienen. Auf den Fotos und im Video des Interactive Mirrors werden lediglich einige Flash Applikationen vorgeführt, wie das Malen auf dem Bildschirm und das Spielen von einem kleinen Weltraumspiel. Außerdem konnte man bei den Benutzern beobachten, dass beim Malen selbständig Objekte auf dem Spiegelbild nachgemalt oder erweitert wurden. Der Benutzer erweitert selbständig und intuitiv seine visuelle Realität. Hierbei sieht man beispielhaft, dass Augmented Reality als Bestandteil bei Spiegel-Displays intuitiv durch HCI eingesetzt werden kann.

Im dem Projekt wurde ebenfalls ein Spiegel-Display gebaut, welches zusätzlich Touch als Eingabe unterstützt. Auf die konkreten Ergebnisse der Untersuchungen im HCI Bereich wurde leider nicht im Blog eingegangen. Durch die Interaktion mit dem Spiegel-Display durch Touch-Gesten ist das System als Badezimmerspiegel ungeeignet, da Schmutz und Fettflecke auf einem Badezimmerspiegel stark stören (vgl. [Kasal \(2010\)](#)).

Magic Mirror

Der Magic Mirror von [MRS Systems \(2012\)](#) ist eine virtuelle Anprobe von Kleidungsstücken in einem Bekleidungsgeschäft. Im Spiegel sieht der Nutzer sich selbst, das ausgewählte Kleidungsstück überdeckt den Nutzer als ob er es tragen würde. Bei Bewegungen oder Drehungen wird das virtuelle Kleidungsstück dem Körper angepasst, wie es sich real verhalten würde. Durch eine Kamera können Fotos aufgenommen und diese direkt auf Facebook geteilt werden. Zum nächsten Kleidungsstück gelangt der Nutzer durch eine Wisch-Geste. Ein Verkäufer unterstützt den Nutzer mit einem Tablet PC. Der Nutzer kann über das System direkt die Farbe des Kleidungsstück ändern oder bestellen.

Der Augmented-Reality-Ansatz der virtuellen Kleidungsanprobe scheint reizvoll für viele Kunden zu sein. Außerdem können die Nutzer durch 3D-Gesten Fotos schießen und somit durch den Spiegel mit der Außenwelt interagieren.

Die symbolisch-manipulativen Gesten für die Steuerung scheinen den Benutzern leicht zu fallen, da die Gesten ähnlich gestaltet sind wie bei einem gewöhnlichen Touchscreen. Die Methaphern für diese Aktionen existieren bereits wie z.B. die Wisch-Geste. Für das "Klicken" eines Elements wird die Hand einfach ruhig über ein entsprechendes Element gehalten.

2.8 Laborumgebung

In diesem Kapitel wird auf die Laborumgebung und im Anschluss darauf auf die Planung des Modells des Smart Mirrors eingegangen. Als Laborumgebung an der HAW wurde das Badezimmer des Living Places (2.6) gewählt.



Abbildung 2.8: Badezimmer des Living Places

2.8.1 Living Place als Umgebung des Smart Mirrors

Das Living Place eignet sich durch seine Beschaffenheit als Labor für Smart Home ideal für die Entwicklung und Evaluation des Smart Mirrors. Der Smart Mirror ist das erste Projekt,

welches später Bestandteil des Badezimmers des Living Places an der HAW wird.

Es werden verschiedene Bedienungsmöglichkeiten für das Smart-Mirror-System implementiert und getestet, wie 3D-Gesten und Sprachbefehle. Da die Living-Place-Wohnung als Loftwohnung konzipiert ist und der Aufenthalt im Badezimmer für gewöhnlich einzeln wahrgenommen wird, werden Problemstellungen, die aus der gleichzeitigen Benutzung von mehreren Personen entstehen, nur im Ansatz erwähnt, aber nicht weitergehend untersucht. Weitere Details dazu wurden in der Aufgabenstellung unter Rahmenbedingungen (2.2.4) bereits erläutert.

Da verschiedene Personen das Badezimmer benutzen, müssen die vertraulichen Daten wie E-Mails und Kalendereinträge geschützt werden. Auf die verschiedenen Möglichkeiten der Authentifikation wird im Kapitel 3.3 eingegangen. Desweiteren ist anzumerken, dass der Video-Stream der Kinect Kamera deaktiviert ist und die Daten der IR-Kamera und der Skelett-Erfassung intern verarbeitet werden. Eine genauere Erläuterung dazu befindet sich im Kapitel Interaction Stream (3.6.1). Lediglich die Interaktionsdaten werden dem Sensornetzwerk des Living Places ohne Verbindung zum Internet bereitgestellt.

Bei der Vorstellung des Living Places während der “Hamburger Nacht des Wissens 2013” konnte der Smart Mirror bereits vorgestellt, und erste Alphatests durchgeführt werden. Der Testaufbau und die Ergebnisse dieser Tests sind in dem Kapitel Design Evaluation Alphatest (4.1.2) zu finden. Da das Badezimmer sich auf der Südseite des Gebäudes befindet, und das Badezimmerfenster durch die Sichtschutzbeschaffenheit weniger IR-Strahlung der Sonne ins Zimmer lässt, sind die lokalen Bedingungen im Living Place für IR-basierte Sensoren optimal.

2.8.2 Darstellung

Die Darstellung des Smart Mirrors besteht aus einem LCD, vor dem ein spezieller Einwegspiegel montiert wurde. Durch die Beschaffenheit eines Einwegspiegels sieht der Nutzer einen gewöhnlichen Spiegel, wenn der Monitor ausgeschaltet ist oder ein schwarzes Bild angezeigt wird. Sobald der Monitor Licht bzw. Farben anzeigt, werden diese im Spiegel sichtbar.

Durch diesen Effekt muss bei der Erstellung der grafischen Oberfläche darauf geachtet werden, dass keine schwarzen oder dunklen Elemente verwendet werden, da diese die “Farbe” des Spiegelhintergrundes im Bezug des Blickwinkels des Nutzers haben, welche sich schon durch kleine Bewegungen des Kopfes stark verändern.

Als Display wurde ein 27 Zoll Monitor verwendet, welcher sich durch einen guten Schwarzwert und hohe Helligkeitswerte auszeichnet.



Abbildung 2.9: Smart Mirror

2.8.3 Steuerung

Für diese Arbeit wurden die “Kinect für Windows” von Microsoft (2.10(a)) und der “Leap Motion” Controller (2.10(b)) als Sensorik gewählt. Die Geräte zeichnen sich durch umfangreiche SDKs und aktive Communities aus. Die Kinect besitzt für die Sprachsteuerung Mikrofone, eine Kamera und eine Infrarot Kamera zum Motion Tracking.

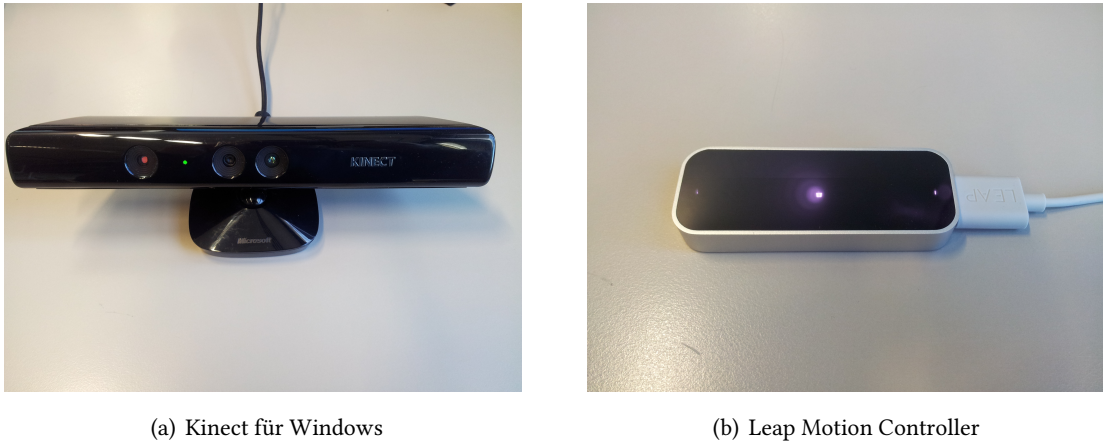


Abbildung 2.10: Steuerungssensorik

Die Kinect benötigt einen Mindestabstand von 40 cm, wobei laut Hersteller der optimale Bereich erst ab 80 cm beginnt. Da die Hände des Nutzers durchaus näher als 80 cm am Spiegel sind, wurde der Leap Motion Controller für das Motion Tracking im Nahbereich ausgewählt.

2.9 Gesten

Kommunikative Gesten können lautsprachersetzend oder lautsprachbegleitend sein. Da bei der Beschreibung der Bedienung (2.2.2) im Abschnitt Funktionale Anforderungen gefordert wurde, dass das System jeweils nur mit einer der beiden Steuerungsarten bedient werden kann, wird nur auf die lautsprachersetzenden kommunikativen Gesten eingegangen.

Gesten können ebenso zur nonverbalen Kommunikation zwischen Menschen oder auch mit Tieren genutzt werden. Besonders sind folgende Gestentypen, nach Klassifizierung von McNeill (1992), im Rahmen einer Steuerung von Computersystemen zu beachten:

- Zeigegesten (Deiktische Gesten) - Die einfache Zeigegeste ist eine der ersten Gesten, die von Kindern gelernt wird, meist durch das einfache Zeigen mit dem Zeigefinger auf ein Objekt. Wobei Erwachsene zudem auf imaginäre Objekte oder Orte zeigen können.
- Ikonische Gesten - Ikonische Gesten sind Gesten welche die Wirklichkeit in einer beliebigen Form abbilden. Mit diesen Gesten werden z.B. Handlungen nachgeahmt, Objekte dargestellt oder im Raum angeordnet.
- Metaphorische Gesten - Diese Gesten stellen abstrakte Bilder da. Sie sind den ikonischen Gesten sehr ähnlich und werden meist benutzt um imaginäre Objekte abstrakt zu

manipulieren, wie z.B das Halten einer davor durch eine ikonische Geste gezeichneten Box.

2.9.1 Kulturelle Interpretation

Jedoch ist zu beachten, dass eine Geste auch aus mehreren Typen (z.B Zeigegeste und ikonische Geste) bestehen kann. Trotzdem können diese Gesten für Menschen aus einem Land oder dem selben Kulturkreis oftmals richtig interpretiert werden. Viele Gesten werden von Menschen aus unterschiedlichen Ländern oder Kulturkreisen unterschiedlich genutzt. [Cassell \(1998\)](#) z.B. zeigte auf, dass Gesten wie das amerikanische "V" für Victory (Sieg), bei dem aus Zeigefinger und Mittelfinger ein "V" geformt wird, in Großbritannien eine Geste der Verachtung an die gerichtete Person darstellt, sollte die Handinnenseite zum Ausführenden zeigen. Auch das Verneinen durch das Schütteln mit dem Kopf kann leicht mit dem Hin und Her wiegen des Kopfes in Indien, Pakistan und Bulgarien verwechselt werden, welche dort als Zustimmung gedeutet wird.

Daher ist bei einer Gestensteuerung die interkulturell zum Einsatz kommt darauf zu achten, dass entweder lediglich Gesten verwendet werden, die keine Mehrfachbedeutung oder negative Interpretation zulassen, oder dass für unterschiedliche Regionen verschiedene Gesten dieselbe Aktion auslöst.

2.9.2 Abgrenzung Motion Tracking und Gestenerkennung

Motion Tracking Beim Motion Tracking oder auch Motion Capture werden Verfahren und Techniken zur Erfassung von Bewegungen benutzt. Bei der Klassifizierung von Gesten (2.11) nach [Pavlovic u. a. \(1997\)](#) sind Bewegungen der Hände und Arme die Voraussetzung für Gesten zur HCI. Um die Bewegungen zu erfassen, werden optische Verfahren wie Kameras oder nicht optische Verfahren wie magnetische oder akustische Tracking Techniken verwendet. Desweiteren werden auch Motion-Tracking-Verfahren entwickelt, um Gesichtszüge oder Bewegungen des gesamten Körpers zu erfassen.

Für das Motion Tracking wird wie in Abschnitt Steuerung (2.8.3) beschrieben, die Kinect für Windows und der Leap Motion Controller genutzt.

Detaillierte Informationen zu allgemeinen Motion Tracking des Living-Place-Labors der HAW Hamburg finden man unter anderem in der Bachelorarbeit von [Potratz \(2011\)](#) oder [Boetzer \(2008\)](#). Weitere Informationen zum Gesichts-Motion-Tracking findet man u.a. in der Masterarbeit von [Müller \(2013\)](#).

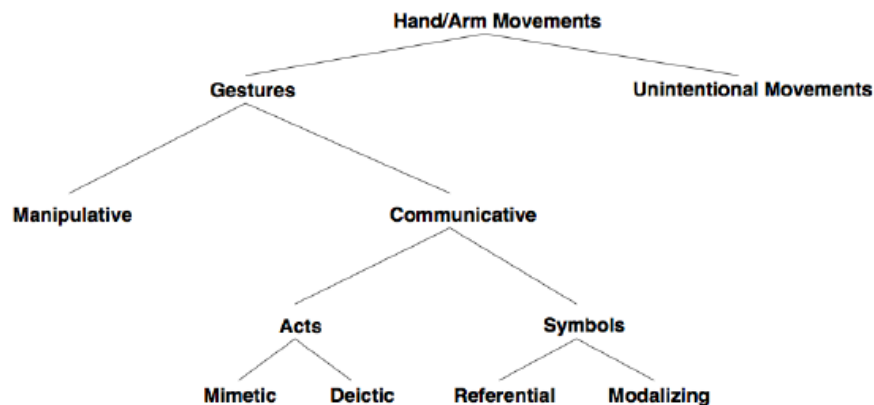


Abbildung 2.11: Klassifizierung von Gesten [Pavlovic u. a. (1997)]

Kamerabasierte Gestenerkennung Die Gestenerkennung nimmt die erste Unterscheidung zwischen “Gesture” und “Unintentional Movements” vor (vgl. Abbildung 2.11). Dafür wird auf die Daten des optischen Motion Trackings zurückgegriffen und durch Algorithmen und mathematischen Methoden die Erkennung einer Geste definiert.

Hierbei ist u.a. besonders auf die Start- Endproblematik zu achten, auf die noch im Abschnitt Start- Endproblematik (2.9.5) eingegangen wird.

2.9.3 Kategorisierung von manipulativen Gesten

Eine weitere Kategorisierung von manipulativen Gesten wurden von einer Arbeitsgruppe auf dem WIWITA (2008) durchgeführt. Sie unterteilt die manipulativen Gesten in bewegungsverfolgende, kontinuierliche und symbolische-manipulative Gesten (vgl. Belke (2000)).

- **Bewegungsverfolgende Gesten** - Sie entsprechen am ehesten Manipulationen in der realen Welt. Zudem wird dem Nutzer ermöglicht, in realistischer Weise abstrakte Objekte zu manipulieren, wie das Rotieren oder Fallenlassen von Objekten. Da die meisten Gesten, die sich aus der realen Welt übertragen lassen, bewegungsverfolgende Gesten sind, sind diese äußerst intuitiv.
- **Kontinuierliche Gesten** - Diese Gesten haben zwar keine Entsprechung in der realen Welt, sind jedoch metaphorisch stark an Aktionen oder Bewegungen in der realen Welt angelegt. Durch diese Metaphern sind sie ebenfalls leicht zu erlernen und zu merken.

- Symbolisch-manipulative Gesten - Für diese Gesten besteht ebenfalls keine Entsprechung in der realen Welt und somit erfordern sie einen gewissen Grad von Abstraktion. Nutzer müssen einen geringfügigen Aufwand in Kauf nehmen, um diese Gesten zu erlernen. Symbolisch-manipulative Gesten können jedoch oft sinnvoll eingesetzt werden, da es für die Manipulation in einer virtuellen Welt nicht unbedingt entsprechende Analogien in der realen Welt geben muss.

2.9.4 2D-Gesten

2D-Gesten wie die Bedienung eines Touchscreen oder die Mausgesten bei Opera [Opera Software ASA (2012)] sind zwar grundsätzlich für diese Arbeit nicht näher zu betrachten, jedoch sind dem User aus diesem Kontext bereits einige erlernte kontinuierliche und symbolisch-manipulative Gesten bekannt und ermöglichen dadurch ein leichteres Erlernen der Entsprechung als 3D-Gesten. Allerdings ist dies nicht immer möglich. Ein Beispiel hierfür sind die Pinch-to-Zoom-Gesten, bei denen zwei Finger auf einen Touchscreen gelegt werden und durch das Auseinanderziehen der Finger eine Vergrößerung und durch das Zusammenziehen der Finger eine Verkleinerung des Objekts vorgenommen wird. Die gleichen Gesten können mit zwei Händen als 3D-Gesten nicht durchgeführt werden, da z.B. bei einer starken Vergrößerung die Geste mehrmals hintereinander ausgeführt werden müsste. Während bei der 2D-Geste das Ende durch das Heben der Finger vom Touchscreen signalisiert wird, und das Zusammenziehen in der Luft der Finger als Vorbereitung für ein weiteres ausführen der Geste ist, würde das Zusammenziehen der Hände als 3D-Geste bereits als Verkleinerung interpretiert werden.

2.9.5 3D-Gesten

Gegenüber den 2D-Gesten kommt bei 3D-Gesten, zu der Höhe, der Breite und der Zeit noch die Tiefe als weitere Dimension hinzu. Daher ist die Ausführung von 3D-Gesten schwerer korrekt auszuführen. Zudem fehlt oftmals der Rückkanal, der bei 2D-Gesten visuell umgesetzt wird. Wie Boetzer (2008) in seiner Arbeit beschrieben hat, sollten deswegen 3D-Gesten eine hohe Toleranz aufweisen. Desweiteren ist es hilfreich, wenn sich die Gesten nur auf einer Ebene bewegen.

Ein weiteres Problem welches Boetzer (2008) aufzeigt ist, dass die unterschiedlichen Körpergrößen der Nutzer mit bei der Entwicklung von Gesten berücksichtigt werden müssen.

Der aktuelle starke Zuwachs von Patenten für neu entwickelte Steuerungsgesten kann eben-

falls ein Indiz für die Komplexität und den Aufwand für die Entwicklung methodischer neuer Gesten sein.

Start- Endproblematik

Bei der Entwicklung von Gesten ist das “Start- Endproblem” oder auch das “Midas touch problem”^[1] ein zentrales Problem. Dabei gilt es die Unterscheidung von den unbeabsichtigten Bewegungen von den intentionalen Gesten zu differenzieren (vgl. Abbildung 2.11).

Die Schwierigkeit besteht darin den Start- und den Endpunkt einer Geste durch ein Computersystem zu erkennen, da ein Mensch, auch bei dem Versuch still zu stehen, nie völlige Bewegungslosigkeit erreicht.

Dieses Problem wird bei Touchscreen z.B. dadurch gelöst, dass die Geste startet sobald der Finger den Touchscreen berührt und mit dem Hochheben des Fingers vom Touchscreen endet. Auch das Drücken bzw. Loslassen eines Knopfes, wie bei der Nutzung des WiiMote Controllers, kann den Start- und Endpunkt einer Geste angeben.

Bei anderen Anwendungen kann auch eine Ruhephase zwischen zwei Gesten als Ende und Start einer Geste dienen. Dabei müssen jedoch die geringen Bewegungen des Nutzers beim Stillstand als Ruhephase erkannt werden.

Leider sind diese Ansätze für die nebenläufige Nutzung im Badezimmer ungeeignet, da der Nutzer ständig andere Bewegungen durchführt und zudem die Hände nicht durchgehend frei hat. Das Drücken eines Knopfes wie bei dem WiiMote Contoller wäre evtl. durch einen Schalter, welchen man mit dem Fuß betätigt, umzusetzen. Dies würde jedoch den Interaktionsbereich des Nutzers stark verkleinern.

Im Kapitel Design und Realisierung (3.6.2) wird auf die in diesem Projekt gewählte Lösung des Start- Endproblems eingegangen.

Steuerungsgesten

In den folgenden Paragraphen werden die benötigten Steuerungsgesten für das System dargestellt. Sie dienen zur Umsetzung des in Abschnitt 2.2.3 beschriebenen Interaktionsrepertoire.

¹Über König Midas, der bis 695 v. Chr. Herrscher von Phrygien war, gab es mehrere antike mythische Anekdoten von seine Gier und Dummheit. So soll er sich die Gabe gewünscht haben, dass alles was er berührt zu Gold wird. Dies wurde schnell zu einem Fluch statt einem Segen, da er auch die Nahrung, die er zu sich nehmen wollte oder Verwandte, die er anfasste, zu Gold wurden.

Bewegen Durch die Bewegung der Hand soll der Nutzer ein Steuerelement fokussieren können, um dieses zu markieren. Hierzu soll eine frei wählbare Hand als Analogie zur Maus dienen und die Veränderung der Position der Hand, die Position des Cursors anpassen.

Klicken Für das Klicken bewegt der Nutzer seine Hand nach vorne, als wenn er einen an einer Wand befestigten Knopf eindrücken würde. Dabei muss der Cursor auf einem Steuerelement positioniert sein, welches dann durch die Klickgeste aktiviert wird.

Scrollen (Bildlauf)

Greifgeste Eine Liste kann durch eine Greifgeste gescrollt werden. Das Schließen der Hand signalisiert den Start der Geste. Die Hand kann nun bei horizontalen Listen nach links und rechts bewegt werden und bei vertikalen Listen nach oben und unten, um die Scroll-Richtung vorzugeben. Durch das Öffnen der Hand wird die Geste beendet.

Ein "Nachscrollen" kann wie bei Touchscreen Scroll-Gesten erfolgen, wenn die geschlossene Hand über die Liste hinaus gezogen und erst dann geöffnet wird.

Kreisgeste Eine weitere Möglichkeit des Scrollen ist die Kreisgeste. Diese symbolisch-manipulative Geste ist seit dem iPod von Apple einer breiten Masse von Nutzern bekannt. Durch das Kreisen eines Fingers im Uhrzeigersinn scrollt eine vertikale Liste nach unten und eine horizontale Liste nach rechts. Durch das Kreisen gegen den Uhrzeigersinn scrollt man dementsprechend in die andere Richtung. Die Geste startet sobald sich der Finger im Interaktionsraum befindet bzw. der Cursor sich auf einem scrollbaren Element befindet.

2.10 Spracherkennung

Bei der Spracherkennung wird durch ein computergestütztes System versucht, die gesprochenen Wörter eines Menschen in Text umzuwandeln (Transkription). Dabei kommen meist Mikrophone für die Abnahme der Sprache zum Einsatz. Um die Erkennungsgenauigkeit zu verbessern, werden inzwischen zusätzlich die Lippen der sprechenden Person mit einer Kamera aufgenommen und eine Auswertung der Lippenbewegung vorgenommen.

Die Spracherkennung bietet sich hervorragend als Steuerung des Smart Mirrors an, da Hände und Arme für die eigentlichen Tätigkeiten im Badezimmer benutzt werden können.

Wie **Pfister und Kaufmann (2008)** beschrieben haben, kann man Spracherkennung in folgende Systemklassen unterteilen:

- Einzelworterkenner - Hierbei werden einzelne Wörter oder kurze Kommandos mit Pause gesprochen.
- Keyword-Spotter - Hier werden ebenfalls einzelne Wörter oder kurze Kommandos gesprochen, wobei diese in einem beliebigen Kontext vorkommen.
- Verbundworterkenner - Bei dem Verbundworterkenner werden Sequenzen aus fließend gesprochenen Wörtern aus einem kleinen Vokabular erkannt.
- Kontinuierlicher Spracherkenner - Vollständige fließend gesprochene Sätze werden von einem kontinuierlichen Spracherkenner erkannt.

2.10.1 Spracherkennung vor dem Smart Mirror

Witt (2011) stellte in seiner Masterarbeit fest, dass die Spracherkennung lediglich unter kontrollierten Bedingungen gut funktioniert. Dabei ging er unter anderem auf folgende Parameter ein:

- Mikrofontyp und der Abstand zum Sprecher - Diese Parameter ändern sich bei dem Smart Mirror nur geringfügig, da immer das Mikrofon am Spiegel benutzt wird, und der Nutzer grundsätzlich den Spiegel nur steuert wenn er sich im Spiegel sieht. Daher ist der Abstand zum Spiegel auf den Interaktionsbereich begrenzt (vgl. Abbildung 2.2.4).
- Unterschiedliche Sprecher - Der Smart Mirror wird primär von dem Bewohner des Apartments gesteuert.
- Lärmbedingungen - Die Lärmbedingungen können im Badezimmer durch die Nutzung eines Föhns, des Wasserhahns oder einer elektrischen Zahnbürste schlecht sein. In diesem Fall sollte der Nutzer auf die Gestensteuerung zurückgreifen.

2.10.2 Start- Endproblematik bei Sprachsteuerung

Wie Witt (2011) in seiner Masterarbeit beschrieben hat, werden grundsätzlich alle Sprachäußerungen vor dem Mikrofon analysiert und auf Befehle abgebildet. Dies führt wie bei der Gestensteuerung zum Start- Endproblem für den Fall, dass eine Person nicht den Spiegel anspricht. In einem Badezimmer einer Ein-Personen-Wohnung ist dies zwar unwahrscheinlich, könnte jedoch durch ein Telefonat, das Sprechen zu einer zweiten Person in der Wohnung, bei Selbstgesprächen oder einem laufenden Radio eintreten.

Da Push-To-Talk-Ansätze ähnlich wie das Knopfdrücken bei dem WiiMote Controller für die

Gestensteuerung, für dieses Szenario nicht geeignet sind, müssen Ansätze wie die Sprachanalyse und das Einsetzen einer Mindestwahrscheinlichkeit für den geäußerten Sprachbefehl, wie auch eine Überprüfung der Selektion und Semantik, ob eine Interpretation der Sprache für die Grammatik vorliegt, umgesetzt werden.

2.10.3 Befehlinterpretation

Die Transkription von der gesprochenen Sprache zum Text ist lediglich die erste Ebene. Eine weiterführende Untersuchung des Textes ist nötig, um konkrete Befehle zu erstellen. So sollte das System durch die Aussagen:

“Öffne die dritte E-mail” und “Die dritte E-Mail öffnen”

die gleichen standardisierten Kommandos für das System erzeugen.

Zudem müssen die gebildeten Kommandos aufgrund des aktuellen Anwendungskontextes überprüft und interpretiert werden. So kann ein Sprachbefehl wie:

“Nach unten scrollen”

auf eine Liste von E-Mails oder eine Liste von Kalendereinträgen wie auch auf das Runterscrollen der offenen Detailinformationen einer E-Mail bezogen sein. Dabei sollte davon ausgegangen werden, dass das zuletzt markierte Element gemeint ist, wenn der Sprachbefehl nicht direkt ein anderes Steuerelement beinhaltet.

3 Design und Realisierung

Im Design- und Realisierungskapitel wird auf die Konzepte, die Architektur, die verwendeten Frameworks, sowie auf die Designentscheidungen für das Smart-Mirror-System eingegangen. Dabei wird erst ein Überblick über den aktuellen Aufbau der Test- und Entwicklungsumgebung gegeben und daraufhin wird auf die gewählte Software-Architektur eingegangen. Am Ende folgt eine Aufstellung der verwendeten Software und Frameworks zur Realisierung, sowie die Hardware Architektur und die konkrete Umsetzung des Smart Mirrors.

3.1 Gesamtsystem Konzept

Das Smart-Mirror-System wird im Living Place als Testumgebung im Badezimmer installiert. Dabei nutzt das System die Kommunikationsstruktur der Testumgebung. Physikalisch besteht das System aus zwei Sensoren für die Steuerung, dem Spiegel-Display für die Darstellung, sowie aus einem leistungsfähigen Kleinrechner, über dem lokal die Anwendungen ausgeführt werden. Durch die Aufteilung der Anwendungen in fünf Bestandteile ist das Auslagern auf Remote-Systeme teilweise möglich.

Wie bereits in der Analyse im Abschnitt 2.8.3 beschrieben, wird die Kinect für Windows und der Leap Motion Controller als Sensorik für die Steuerung verwendet. Das Spiegel-Display wurde ebenfalls in der Analyse im Abschnitt 2.8.2 vorgestellt. Dieses wird per HDMI an die GUI-Anwendung angeschlossen und dient damit als Darstellung bzw. Output des Systems. Die fünf Anwendungen haben folgende Funktionen:

- Konfiguration und Start-Anwendung - Über diese Konsolen-Anwendung konfiguriert und startet der Nutzer die weiteren Anwendungen.
- Grafische Nutzeroberfläche (GUI) - Die grafische Oberfläche ist für die Darstellung der Inhalte auf dem Spiegel-Display zuständig. Sie gibt dem Nutzer eine visuelle Rückmeldung über seine Interaktionen und führt eine Überprüfung der Befehle anhand des aktuellen Anwendungskontextes durch.

- Sprachinterpretier - Diese Konsolen-Anwendung verfügt über einen Befehlssatz. Die Daten der Spracherkennung werden zu eindeutigen Befehlkommandos umgewandelt und an die GUI geschickt.
- Leap-Motion-Adapter - Der Leap-Motion-Adapter nimmt die Daten der Leap Motion an, und führt eine Gestenerkennung durch. Diese Gesten werden in ein vordefiniertes Format umgewandelt und versendet.
- Kinect-Adapter - Der Kinect-Adapter führt die äquivalenten Arbeiten des Leap-Motion-Adapters für die Kinect durch. Zusätzlich findet hier die Datenverarbeitung des Mikrofons der Kinect statt. Auf Grundlage einer Grammatik wird eine Spracherkennung durchgeführt. Diese Daten werden dem Sprachinterpretier zur Verfügung gestellt.

Das System hat zudem eine Anbindung durch IMAP zu einem E-Mail-Konto, ein Zugriff auf einen Online-Kalender, wie auch einen Zugang zu einem Wetter-Service.

3.2 Living-Place-Architektur

Das Living Place der HAW Hamburg wurde bereits im Kapitel Living Place (2.6) vorgestellt. Im Folgenden wird konkreter auf das gesamte Konzept und die Kommunikationsstruktur des Smart Homes eingegangen, da diese Vorgaben für das Smart-Mirror-System geben.

3.2.1 Konzept

Als architektonisches Muster des Softwaresystems des Living Places wurde das Blackboard-System (vgl. [Ellenberg u. a. \(2011\)](#)) gewählt. Über das Blackboard können verschiedene Dienste Daten anbieten oder abonnieren, dabei kommunizieren und kennen aber die verschiedenen Dienste untereinander nicht. Eine Gruppe von Experten, die in Zusammenarbeit ein Problem lösen, kann als Metapher für das Blackboard betrachtet werden.

Bei Ablage oder Änderungen der Daten, welche die Dienste dem Blackboard zur Verfügung stellen, werden die anderen Dienste, die Interesse an diesen Daten haben, darüber informiert. Im Gegensatz zu Datenbanken müssen beim Blackboard die Daten nicht angefragt werden, da die interessierten Dienste direkt über Veränderungen informiert werden.

Die verschiedenen Komponenten des Smart Mirrors werden als einzelne Dienste implementiert, die untereinander über das Blackboard kommunizieren. Im Folgenden wird detaillierter auf die konkrete Kommunikation eingegangen.

3.2.2 Kommunikation

Die Kommunikation wird über ein Message-Broker-System realisiert. Dieses System ist der zentrale Kommunikationsknoten zwischen Nachrichten-Sender und -Empfänger. Durch den Einsatz des Message-Broker-Systems müssen Sender und Empfänger sich nicht kennen, um Nachrichten auszutauschen.

Konkret wird ein ActiveMQ eingesetzt, der die Kommunikation über zwei unterschiedliche Kommunikationsmechanismen anbietet.

- Producer-Consumer - Dieses Pattern bildet eine *Point-to-Point* Queue ab. Diese wird in der Abbildung 3.1(a) durch die Nachricht A verdeutlicht. Sollte es mehrere Consumers für einen Queue geben, bekommt nur einer die Nachricht.
- Publisher-Subscriber - Dieses Pattern bildet ein *one-to-many* Topic ab. In Abbildung 3.1(b) ist zu sehen, dass die Nachricht A an alle Subscribers ausgeliefert wird.

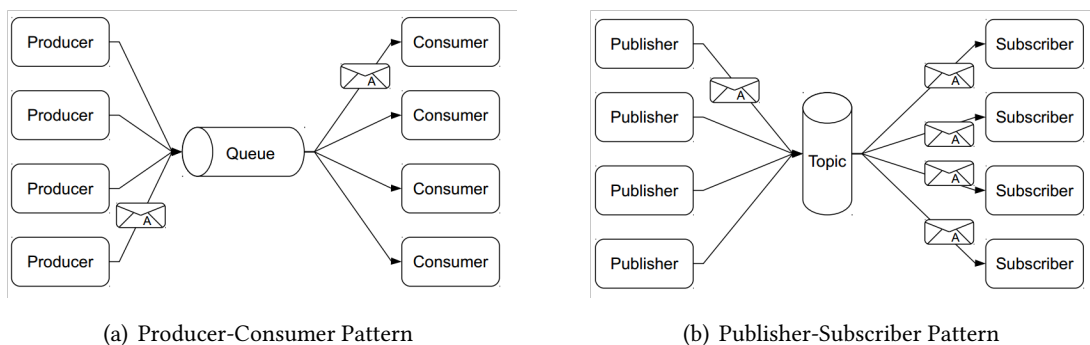


Abbildung 3.1: Kommunikationsmechanismen [Otto (2013)]

Da die Sensor-Daten des Smart Mirrors grundsätzlich durch eine Queue vorgehalten werden müssen, werden die Komponenten des Smart Mirrors ihre Kommunikation durch den Producer-Consumer-Mechanismus vom ActiveMQ des Living Places umgesetzt.

Die Persistenz der Daten wird im Living Place durch eine Datenbank, welche durch einen ActiveMQ Wrapper angebunden ist, umgesetzt. Konkret werden eine MongoDB als Datenbank und JavaScript Object Notation (JSON) als Nachrichtenformat für den ActiveMQ gewählt.

3.3 Authentifizierung

Da in der Analyse im Abschnitt 2.2.4 die Notwendigkeit von Sicherheitsvorkehrungen der persönlichen Daten aufgezeigt wurde, werden an dieser Stelle die verschiedenen Möglichkeiten

der Authentifizierung näher untersucht.

Der Smart Mirror soll ohne Tastatur steuerbar sein, deshalb müssen hier andere Ansätze als die Passwordeingabe auf ihre Tauglichkeit geprüft werden.

Grundsätzlich gibt es drei verschiedene Möglichkeiten der Authentifizierung, die u.a. auch **Poguntke (2007)** anführt:

- Wissen - Passwörter, PIN, Antwort auf Sicherheitsfragen, etc.
- Besitz - RFID Karte, TAN, SIM Karte im mTAN-Verfahren, physikalischer Schlüssel, etc.
- Körperliche Merkmale (Biometrie) - Fingerabdruck, Gesichtserkennung, Handschrift, Iriserkennung, etc.

In der Praxis werden häufig zwei verschiedene Typen zusammen für die Authentifizierung genutzt um die Sicherheit zu erhöhen. Ein Beispiel hierfür ist das Geldabheben mit einer EC-Karte. Der Nutzer muss als Besitz die Karte bei sich haben und das Wissen über die PIN besitzen, um Geld abzuheben.

Diese unterschiedlichen Authentifizierungstypen werden im folgenden Abschnitt kurz erläutert und es werden konkrete Methodiken für den Smart Mirror aufgezeigt.

3.3.1 Wissen

Das meist verwendete Mittel für eine Authentifizierung durch Wissen im IT-Bereich ist das Kennwort. Wie bereits im Abschnitt Rahmenbedingungen (2.2.4) erläutert, sind Passwörter für die Authentifizierung im Badezimmer ungeeignet. Zudem besteht die Gefahr, dass der User das Wissen vergisst. Dafür müssten dann entsprechende Mechanismen umgesetzt werden, damit der Nutzer sich auch auf eine andere Art und Weise authentifizieren kann.

Gesten als Authentifizierung

Der Nutzer kann eine spezielle Geste durchführen, um sich zu authentifizieren. Bei Smartphones hat sich z.B. das Nutzen eines Musters über 3x3 oder 4x4 Punkten etabliert. Den gleichen Ansatz kann man auch mit 3D-Gesten umsetzen, wobei die Verwendung der dritten Dimension weit umfangreichere Möglichkeiten als 2D-Muster bietet.

Da ein Nutzer im Badezimmer grundsätzlich allein ist, sollte die Geheimhaltung der Gesten kein großes Risiko darstellen.

Spracherkennung

Bei der Authentifizierung durch eine Spracherkennung muss der Nutzer einen bestimmten Satz sprechen, um das System zu entsperren. Dabei kommt es lediglich auf die gesprochenen Wörter an, und nicht wie bei der biometrischen Stimmerkennung (oder auch Sprechererkennung) auf die Stimme.

Diese Art der Authentifizierung hat den Nachteil, dass weitere Personen in der Wohnung auch bei verschlossener Tür den geheimen Satz hören könnten. Zudem können schon leichte Hintergrundgeräusche die Falschrückweisungsrate enorm erhöhen.

3.3.2 Besitz

Die Verwendung von Besitz ist allgemein durch physikalische Schlüssel oder EC-Karten zum Authentifizieren bekannt. Da der Smart Mirror über keine Sensorik verfügt, durch die der Besitz eines Objekts erkannt werden kann, ist die Authentifizierung durch Besitz für diese konkrete Konfiguration ungeeignet.

Allerdings ist eine Authentifizierung durch die Nutzung von NFC (Near Field Communication) oder Bluetooth des Handys oder die Verwendung von RTLS (Real Time Location Systems) wie UbiSense im Living Place möglich. Dies würde jedoch voraussetzen, dass der Bewohner immer ein Gerät bei sich trägt, wenn er ins Badezimmer geht. Dies ist jedoch grundsätzlich eher ungewöhnlich, da ein Bewohner nicht immer sein Smartphone bei sich trägt, wenn er sich in seiner Wohnung aufhält.

Allgemein kann Besitz leicht geklaut werden oder verloren gehen. Daher muss eine Backup-Authentifizierung implementiert werden, welche im IT-Bereich oftmals über Wissen realisiert wird.

3.3.3 Biometrie

Biometrische Verfahren reichen von der Fingerabdruck-Erkennung bis zur Herzrhythmuserkennung mittels Elektrokardiogrammen (vgl. [Pluta \(2013\)](#)). Die Sensorik des Smart Mirrors bietet eine Kamera, zwei Distanz-Kameras sowie Mikrofone an. Der Vorteil von biometrischen Authentifikationsverfahren liegt darin, dass der Nutzer weder spezielles Wissen benötigt, noch einen bestimmten Gegenstand bei sich tragen muss. Daher sind solche Verfahren für ein Badezimmer optimal geeignet.

Stimme zur Authentifizierung

Gegenüber der Spracherkennung kommt es bei der Stimmerkennung nicht auf den Satz, sondern auf die Stimme des Nutzers an. Die Stimme eines Menschen ist ebenso wie ein Fingerabdruck einzigartig.

Das Sicherheitsrisiko bei der Spracherkennung, dass weitere Personen in der Wohnung bei der Authentifizierung zuhören, ist bei der Stimmerkennung nicht gegeben. Lediglich das Aufnehmen der Stimme muss als Risiko betrachtet werden. Zudem ist die Stimmerkennung ebenfalls anfällig gegenüber störenden Hintergrundgeräuschen. So könnte ein laufender Wasserhahn oder eine aktive Lüftung zu einer hohen Falschrückweisungsrate führen. Außerdem können die Veränderungen der Stimme durch eine Erkältung oder Heiserkeit ebenfalls zu Falschrückweisungen führen.

Gesichtserkennung zur Authentifizierung

Wie z.B. in der bereits vorgestellten Studie (2.7.2) des Smart Mirror For Ambient Home Environment von [Hossain u. a. \(2007\)](#) beschrieben, ist eine Authentifizierung per Gesichtserkennung möglich. Die Gesichtserkennung wird schon seit mehreren Jahren bei Notebooks über die Webcam angewendet. Sie bietet den Vorteil, dass der Nutzer keine Aktion ausführen muss, sondern lediglich aus dem Interaktionsbereich (2.2.4) auf bzw. in den Spiegel gucken muss. Da die Kinect eine Kamera besitzt, sollte die Einbindung einer Gesichtserkennung für die Authentifizierung ohne großen Aufwand durchführbar sein. Auch die Lichtverhältnisse im Badezimmer könnte man dem Spiegel so anpassen, dass die Kamera eine gute Aufnahme erstellen kann. Durch die Nutzung der Distanzkamera kann zusätzlich sichergestellt werden, dass es sich um ein echtes Gesicht und nicht um ein Foto handelt.

Handgeometrie zur Authentifizierung

Ähnlich wie bei der Gesichtsgeometrie ist auch die Handgeometrie bei jedem Menschen einmalig. Wie in der Arbeit von [Gruher und Stoll \(2001\)](#) erläutert wird, ist das System günstig und genießt eine hohe Akzeptanz. Die meisten Verfahren zur Abmessung der Hand erfordern jedoch eine Ablage der Hand und eine Ansicht von oben und den Seiten. Dies macht das System für das Badezimmer ungeeignet. Zudem bietet die Leap Motion zur Zeit noch keinen Zugriff auf die Raw-Daten, aus denen man ein konkretes Handmodell errechnen könnte.

3.3.4 Fazit

Allgemein scheinen biometrische Verfahren für die Authentifizierung am geeignetsten für die Anwendung im Badezimmer zu sein. Bei der im Smart Mirror verwendeten Sensorik liegt die Nutzung der Stimme oder des Gesichts als Authentifizierung nahe. Allerdings ist die Gesichtserkennung aufgrund der Hintergrundgeräusche vorzuziehen. Sollte der Nutzer keine Erfassung seiner biometrischen Daten wünschen, kann auf ein gestenorientiertes System, wie z.B. die Mustererkennung zurückgegriffen werden.

3.4 Software-Architektur

Im Folgenden wird auf die Architektur der Software eingegangen. Dabei ist zu beachten, dass die Testumgebung des Living Places die Blackboard-Architektur nutzt. Da der Smart Mirror als ein weiteres System dem Living Place hinzugefügt werden soll, ist dies unbedingt beim Design zu beachten.

Daher werden verschiedene Interpretationslevel des Smart-Mirror-Systems auf dem Blackboard zur Verfügung gestellt. Wie aus Abbildung 3.2 zu sehen ist, soll das konkret bedeuten, dass die Sensor Daten an das Blackboard liefert. Daraufhin werden diese Daten interpretiert und wieder auf das Blackboard geschrieben. Nun bekommt die Benutzeroberfläche die Daten und wertet diese bezüglich des aktuellen Anwendungskontextes aus. Die konkret ausgeführten Befehle können erneut auf das Blackboard geschrieben werden.

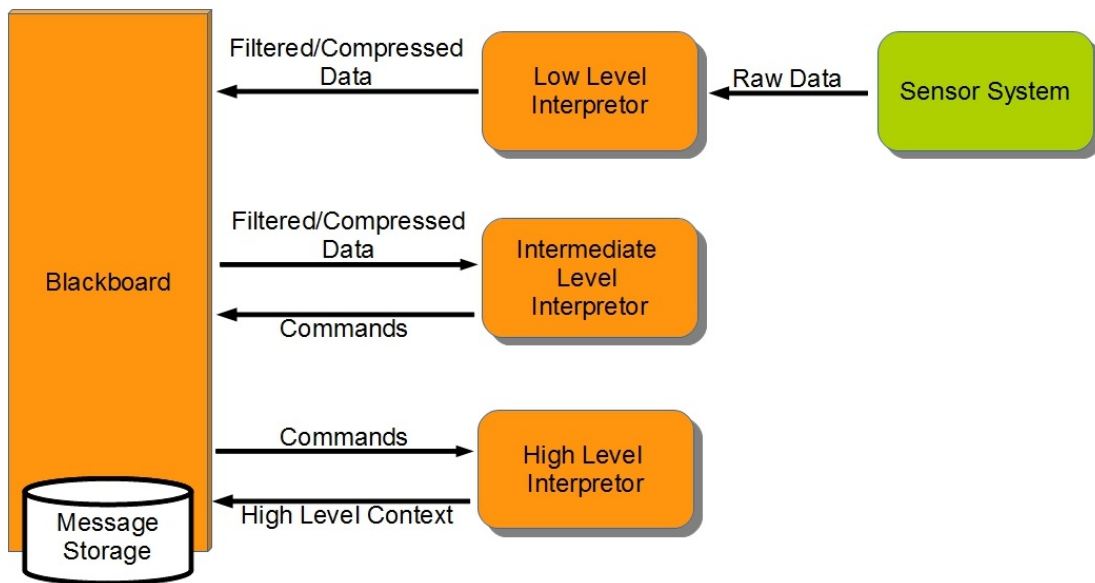


Abbildung 3.2: Blackboard - Interpretationslevel (vgl. [Ellenberg u. a. \(2011\)](#))

3.4.1 Architektur und Entwurfsmuster

MVC - Model-View-Controller

Das Model-View-Controller-Modell wurde erstmals 1979 für Benutzeroberflächen in Smalltalk von Reenskaug konzipiert. Das Entwurfsmuster strukturiert ein Software-System in drei verschiedene Teile, deren Funktionalitäten voneinander abgegrenzt werden sollen. Das MVC-Modell überträgt die Trennung von Eingabe, Verarbeitung und Ausgabe auf GUI-basierende Systeme. Da die drei Komponenten getrennt implementiert werden, sind diese leichter austauschbar und zudem lassen sich Fehler besser lokalisieren (vgl. [Kaufmann \(2011\)](#)).

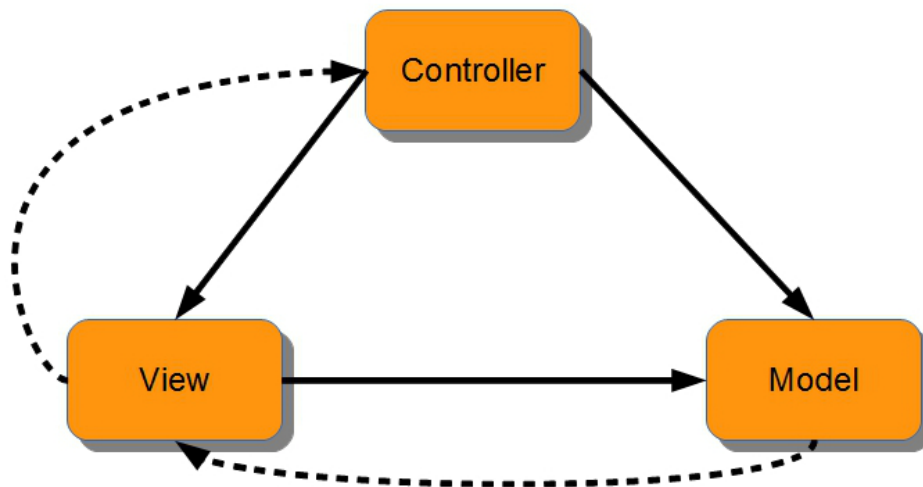


Abbildung 3.3: Model-View-Controller-Modell

Die durchgezogenen Linien symbolisieren direkte Assoziationen, während die gestrichelten Linien indirekte Assoziationen darstellen (z.B. über einen Observer).

View Die View ist für die Präsentation der Daten des Modells zuständig. Sie ist zudem die Benutzerschnittstelle des Systems und kennt aufgrund dessen ebenfalls den Controller.

Controller Der Controller nimmt die Benutzereingaben von einer View oder mehreren Views entgegen und agiert entsprechend. Er informiert das Model über Datenänderungen, welche der Nutzer über die View eingegeben hat.

Model Im Model werden die Daten für die Anwendung gehalten und verwaltet und es enthält ggf. die Komponentenlogik. Das Model liefert auf Anfragen die Daten und führt Änderungen der Daten durch.

MVC Observer Wie u.a. [Potratz \(2011\)](#) beschreibt, gibt es eine MVC-Variante, welche eine aktive Model-Komponente enthält. Diese wird benötigt falls Änderungen am Model durchgeführt werden, ohne dass die Controller-Komponente beteiligt ist. In diesem Fall muss die View direkt vom Model benachrichtigt werden. Dies kann durch das Verwenden des Observer-

Patterns bewerkstelligt werden. Ein schematischer Aufbau eines MVCs mit Observer ist in Abbildung 3.3 zu sehen.

MVVM - Model-View-ViewModel

Die Model-View-ViewModel-Variante vom MVC wurde 1995 von Microsoft für User-Interfaces veröffentlicht. Sie ist eine Spezialisierung des Presentation Model von Fowler (2004). Dabei wird eine Datenbindung (Data Binding) verwendet, welche eine Trennung von Markup und Logik für die Darstellung ermöglicht. Dies führt zu einer losen Kopplung zwischen dem Model und der View.

Programmstruktur

Die Umsetzung der Programmstruktur orientiert sich lediglich an dem MVC-Konzept. Die Systemarchitektur ist daher keine konkrete Umsetzung des MVC-Modells. Die Observer-Variante des MVC-Modells wird benötigt, da das Model ebenfalls Wetterdaten, E-Mails und Kalendereinträge eigenständig updatet, ohne dass der Controller dieses Updates veranlasst. Zudem unterscheidet sich die Umsetzung des Smart-Mirror-Systems vom MVC-Modell dadurch, dass die Daten, die der Controller verarbeitet, nicht von der View bereitgestellt werden, sondern durch die an den Adaptern angeschlossene Sensorik. Daher kann die View lediglich die Daten der Anwendungen (2.2.2) über den Controller und die Daten im Model manipulieren.

Controller Im Controller werden die Bewegungsdaten des Users aufgenommen und es wird eine Low Level Interpretation dieser Daten vorgenommen. Dies dient der Komprimierung der Daten, um das Versenden über das Netzwerk zu ermöglichen. Die Daten werden daraufhin an den Datenpuffer des Models gesendet. Die beiden Adapter sind jeweils eigenständige Applikationen und benutzen ein einheitliches Datenformat trotz unterschiedlicher Sensorik.

Model Im Model werden zum einen die Daten der Spracherkennung interpretiert und weiter an die View geleitet, zum anderen werden hier die externen Daten für die E-Mails, den Kalender und den Wetter-Service geladen und manipuliert.

Der Sprachinterpretier ist als eigene Applikation gekapselt. Die anderen Dienste sind lediglich als gekapselte Komponenten umgesetzt.

View Die View besteht physikalisch aus dem Spiegel-Display, welche dem Nutzer die Informationen präsentiert. Sie beinhaltet zusätzlich einen Beobachter auf das Model, um an die Datenänderungen, die nicht vom Nutzer über den Controller initialisiert wurden, zu erhalten.

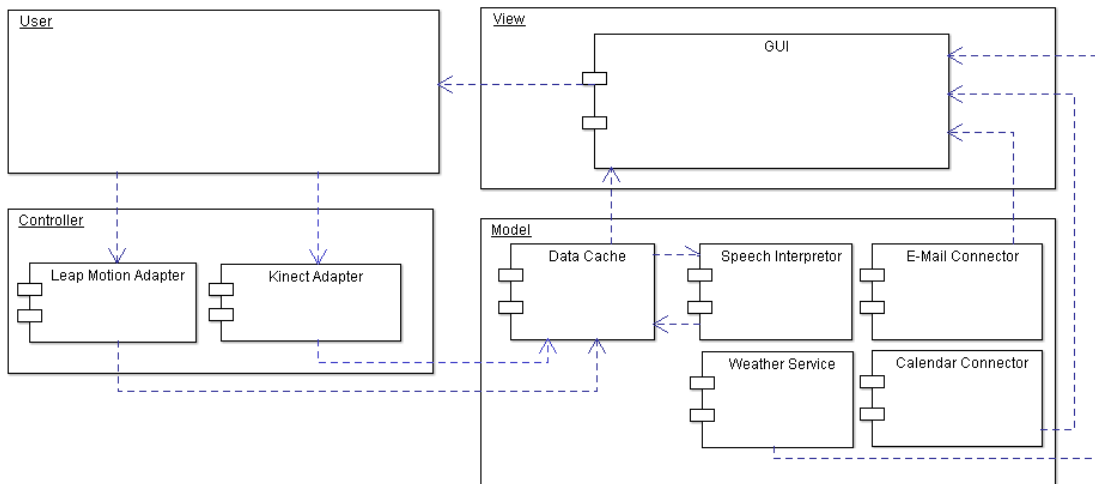


Abbildung 3.4: Komponentendiagramm

MVVM Intern ist die View und ein Teil des Modells mit dem MVVM-Konzept umgesetzt, da für die Darstellung einige Wrapper für die Daten wie E-Mails und Termine implementiert werden müssen. Dies ist nötig, da das Model die Daten teilweise extern im Internet speichert. Da dies bei Manipulationen im Model zu längeren Wartezeiten führen kann, werden erst die Wrapper angepasst, wodurch die View sofort aktualisiert wird. Im Anschluss erfolgt die Anpassung der persistenten Daten.

3.4.2 Kommunikation

Grundsätzlich lässt sich zwischen synchroner Kommunikation wie einem Gespräch und asynchroner Kommunikation, wie z.B. Briefe unterscheiden. In der Informatik wird bei der synchronen Kommunikation nach dem request-response-Verfahren (Anfrage-Antwort-Verfahren) gearbeitet. Eine Seite sendet eine Anfrage und wartet auf eine Rückmeldung, während bei der asynchronen Kommunikation nicht davon ausgegangen wird, dass sofort eine Rückmeldung kommt. Deswegen wird auch nicht blockierend auf diese Rückmeldung gewartet.

Wie im Abschnitt 3.5 noch beschrieben wird, wird die Kommunikationsinfrastruktur des Living Places für den Smart Mirror genutzt. Der Nachrichtenaustausch zwischen den verschiedenen Komponenten des Smart Mirrors geschieht mittels asynchroner Kommunikation und dem

Producer-Consumer-Prinzip (vgl. Abbildung 3.1(a)).

Die folgende Abbildung skizziert den konkreten Informationsaustausch zwischen den Komponenten.

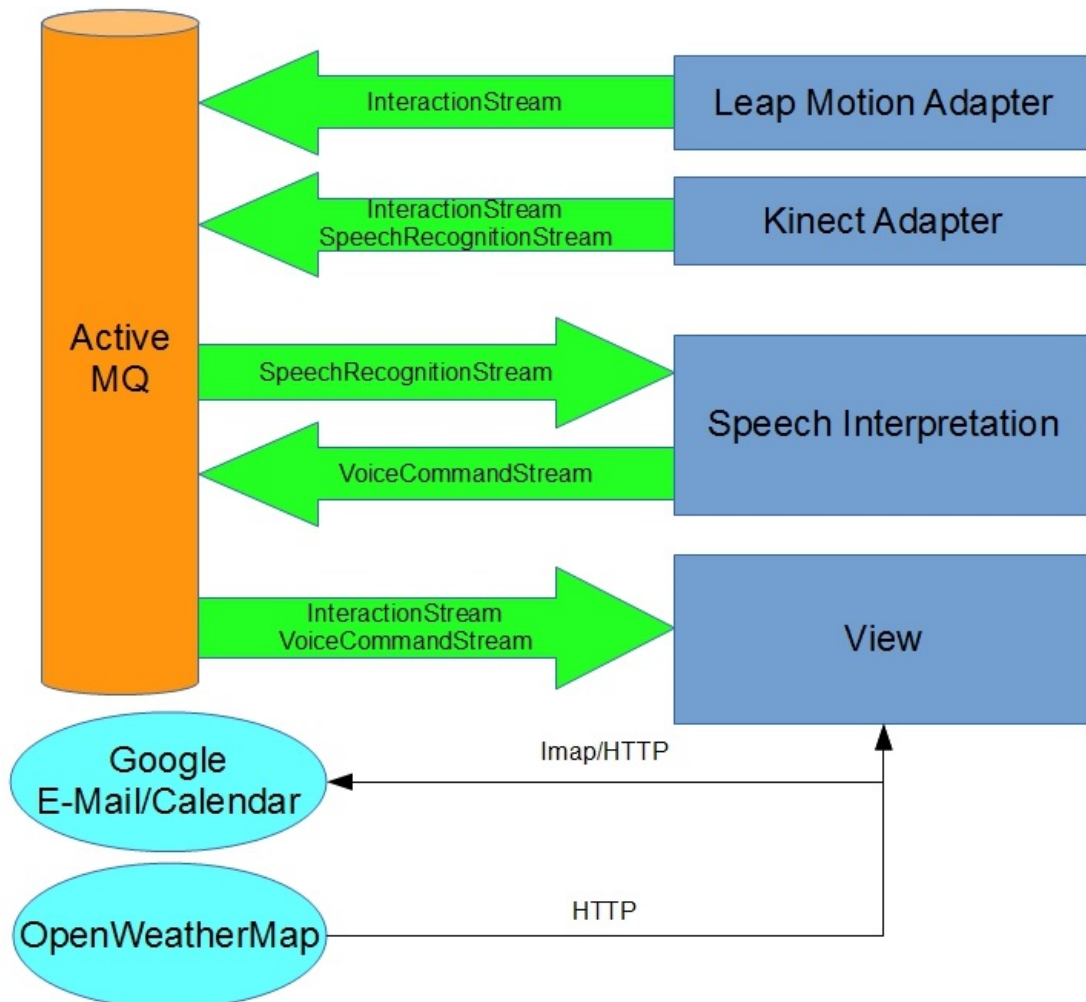


Abbildung 3.5: Kommunikation zwischen den Komponenten

3.4.3 Dienste

Anwendungen wie der E-Mail Client des Modells kommunizieren über IMAP mit dem Server. Ein auf TCP arbeitendes Protokoll wie IMAP lässt sich nicht über ActiveMQ Messages ausführen und muss daher anders integriert werden. Da die Implementierung der Dienste in dieser Arbeit nicht im Mittelpunkt steht, werden diese vorerst nicht als selbstständige Dienste umgesetzt.

Durch die Kapselung der Klassen sollte die Entwicklung von selbstständig laufenden Diensten später leicht ausgelagert werden können.

3.4.4 Kopplung

Der Begriff “Kopplung” beschreibt in der Informatik die Verknüpfung von verschiedenen Softwaremodulen, Anwendungen oder Systemen. Aus einer starken Kopplung resultieren starke Abhängigkeiten, was wiederum einen großen Aufwand bei Änderung einzelner Teilsysteme für das gesamte System bedeutet.

Durch das Einsetzen eines ActiveMQs und einer Abwandlung der MVC-Architektur hat das System eine lose Kopplung, da es wenige Kommunikationsschnittstellen gibt. So lässt sich einfach durch einen weiteren Adapter ein neuer Sensor in das System einarbeiten.

Die in Abbildung 3.5 gezeigten Module sind jeweils als eigenständige Applikationen entworfen. Trotzdem besteht eine gewisse Abhängigkeit, da der Speech Interpreter alle Wörter der Grammatik kennen muss. Ebenso muss die View die Voice Commands des Speech Interpreters auswerten können. Diese Schnittstellen benötigen eine saubere Definition und bilden, wie auch das Nachrichtenformat eine gewisse Abhängigkeit der Module untereinander.

3.5 Software und Frameworks

In diesem Abschnitt wird auf die eingesetzten APIs und Frameworks näher eingegangen.

3.5.1 Microsoft Speech SDK

Das Microsoft Speech SDK stellt Programmier-Schnittstellen für die Spracherkennung und Sprachsynthese bereit. Konkret wird Version 11.0 beim Smart-Mirror-System verwendet und lediglich die Funktionen der Spracherkennung genutzt. Durch das rekursive Überprüfen verschiedener Regeln einer Grammatik wird die Regel gewählt, welche eine Wortkonstellation beinhaltet, die die größte Übereinstimmung mit dem Gesprochenen hat.

Da die Grammatik in XML definiert werden kann, ist sie leicht austausch- und erweiterbar.

3.5.2 Kinect SDK

Das Kinect SDK von Microsoft liefert eine Vielzahl an vorbereiteten Datenstrukturen für das Motion Trackings sowie weitere Funktionen. Für das Smart-Mirror-System wird lediglich der Depth Stream, der Skeleton Stream, der Audio Stream und der Interaction Stream genutzt. Auf Grundlage des Motion Trackings durch den Depth Stream und den Skeleton Stream wird eine

Gestenerkennung durchgeführt, deren Ergebnisse im Interaction Stream ausgegeben werden.

Zu Beginn der Arbeit wurde Version 1.7 genutzt. Später erfolgte die Veröffentlichung und damit die Migration des Systems auf Version 1.8. Da die Implementation zu diesem Zeitpunkt bereits größtenteils abgeschlossen war, wurden keine neuen Features aus der neuen SDK Version genutzt.

Kinect Toolkit

Das Kinect Toolkit liefert allgemeine Erweiterungen, Codebeispiele sowie Vorlagen für die GUI, aufbauend auf dem Kinect SDK. Unter anderem werden die Interaction Bibliothek und Vorlagen des Control Toolkit für die grafische Oberfläche benutzt.

3.5.3 WPF - Windows Presentation Foundation

WPF wurde von Microsoft als Framework für grafische Oberflächen für C# entwickelt. Dabei wird das Konzept des Model-View-ViewModel (vgl. Abschnitt 3.4.1) genutzt. Das zuvor erwähnte Control Toolkit bietet verschiedene Elemente für die Nutzung unter WPF an.

3.5.4 Leap Motion SDK

Das SDK wird in der aktuellen Version 1.0.9.8319 genutzt. Es liefert unter anderem ein Motion Tracking der Finger und eine Erkennung von vier einfachen Gesten, auf die später in Abschnitt 3.6.1 im Detail eingegangen wird.

3.5.5 ActiveMQ - Message Broker

Als Message Brocker wurde der ActiveMQ von **The Apache Software Foundation** genutzt. Da bereits im Kapitel 3.2.2 der Living-Place-Architektur (3.2) detailliert auf die allgemeine Funktionsweise und die Begründung der Nutzung des ActiveMQs eingegangen wurde, wird an dieser Stelle lediglich auf die konkret verwendeten Frameworks für die Implementierung eingegangen.

Für die Implementierung wurde die Apache NMS DLL in Version 1.6.0.3061 für C# und die Apache NMS for ActiveMQ Class Library in Version 1.6.0.3072 verwendet. Mithilfe dieser Bibliotheken wurden zwei Klassen, ein Message Producer und ein Message Consumer entwickelt. Diese werden jeweils von den einzelnen Modulen für die Kommunikation untereinander eingesetzt.

3.5.6 JSON - JavaScript Object Notation

Da JSON das vorgegebene Nachrichtenformat des Living Places ist, wird an dieser Stelle das kompakte Datenformat näher betrachtet.

JSON soll zum Zweck des Datenaustauschs zwischen Anwendungen für Menschen wie auch für Maschinen eine einfache Textform aufweisen. Zudem hat das JSON-Format einen geringeren Overhead als XML und ist daher besser zum Datenaustausch geeignet. JSON ist gegenüber von XML ein reines Datenaustauschformat, während XML eine Auszeichnungssprache ist, und damit vielseitigere Möglichkeiten als JSON anbietet.

Da die Kinect und die Leap Motion ihre Daten mit mehr als 50 Frames per Second senden, musste für C# eine schnelle Bibliothek gefunden werden. JSON.NET von [Newton-King \(2013\)](#) wurde aufgrund der guten Leistung und der umfangreichen Dokumentation gewählt. Konkret wurde Version 4.5 der Newtonsoft.Json verwendet.

3.5.7 OpenWeatherMap API

Die OpenWeatherMap API ist ein freier Online-Service über den das aktuelle Wetter, eine Wettervorhersage, wie auch verschiedene Wetterkarten abgerufen werden können.

Weitere Informationen finden Sie auf der Homepage: <http://openweathermap.org/API>.

3.5.8 IMAP E-Mail Bibliothek

Für den E-Mail Client wurden die ActiveUp.Net.Imap in Version 5.0.345.364 und die ActiveUp.Net.Common Version 1.0 verwendet. Mit diese Bibliothek wurde der Mail Client entwickelt, welcher per IMAP die E-Mails abrufen, oder Such- und Löschanfragen aufgibt.

3.5.9 Google Data APIs

Für die Kalender-Anbindung wurde die Google.GData.Calendar API in Version 2.2 verwendet. Mit dieser API wurde die Schnittstelle zwischen dem Smart-Mirror-System und dem Online-Kalender, welcher bei Google gehostet ist, umgesetzt.

3.6 Technische Architektur

In diesem Kapitel soll ein Überblick über die Datenstrukturen der verwendeten Hardware gegeben werden. Die Abbildung 3.6 zeigt, wie die Datenübertragung physikalisch umgesetzt wird.

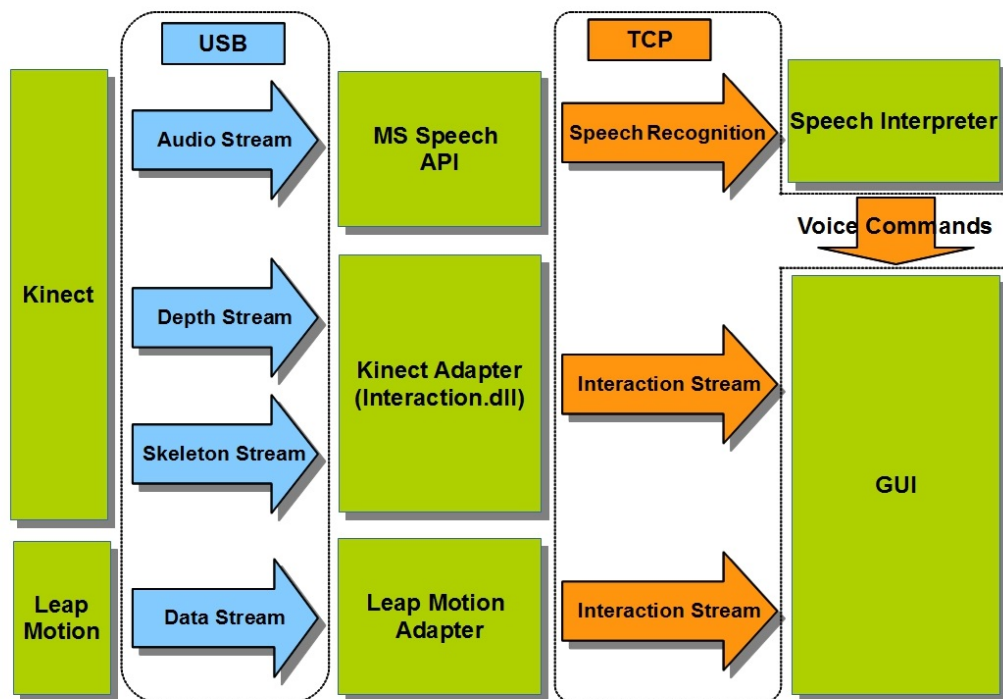


Abbildung 3.6: Standards für die Kommunikation

3.6.1 Hardware

Kinect

Die Kinect (siehe Abbildung 2.10(a)) wird sowohl für die Gestensteuerung als auch für die Sprachsteuerung genutzt. Im Folgenden wird explizit auf die bereitgestellten Streams eingegangen.

Auf den Color Stream wird nicht im Detail eingegangen, da er für das TestszENARIO dieser Arbeit nicht benötigt wird.

Depth Stream Der Depth Stream liefert bei einer Auflösung von bis zu 640x480 Pixel pro Frame ein Millimeter genauen Wert der Entfernung zum ersten Objekt vor der Kinect. Der Depth Stream gehört genauso wie der Skeleton Stream zum bereitgestellten Motion Tracking des SDKs, dieser liefert jedoch noch keine Gestenerkennung.

Skeleton Stream Der Skeleton Stream liefert Frames, die jeweils pro Skelett 20 Joints beinhalten (siehe Abbildung 3.7). Im Tracking Mode Seated für Personen bei denen nur der Torso und der Kopf zu sehen ist, werden nur die Joints überhalb des “Spine” Joints geliefert.

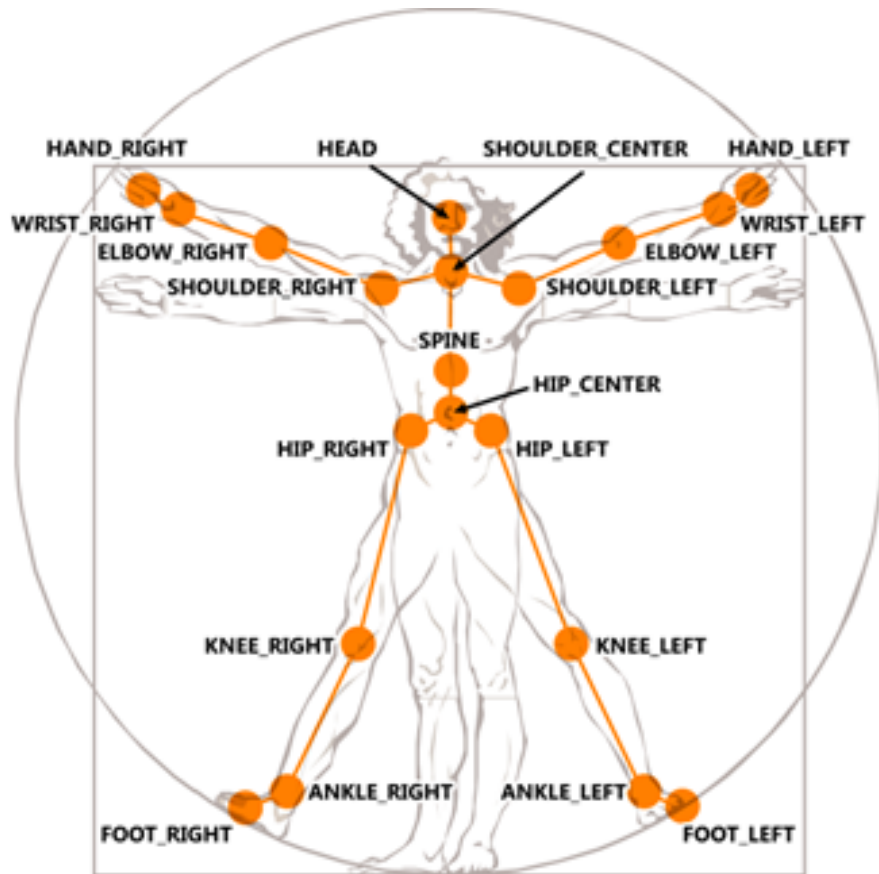


Abbildung 3.7: Skeleton Stream Joints [Microsoft Corporation (2014)]

Interaction Stream Der Interaction Stream wird von der Interaction DLL aus dem Toolkit bereitgestellt. Dieser DLL muss der Skeleton Stream wie auch der Depth Stream übergeben werden. Der Interaction Stream liefert für jedes gefundene Skelett die aktive Hand, die Positionen der Hände, wie auch eine Interpretation von Gesten. Dabei wird eine “Press”-Geste erkannt wie auch das Schließen (Greifen) und Öffnen einer Hand.

Microsoft löst ein Teil der Probleme der 3D-Gestensteuerung (vgl. Abschnitt 2.9.5) durch die physikalische Interaktionszone (PhIZ). Diese bestimmt den Bereich, in dem Bewegungen der Hand als Gesten interpretiert werden. Diese PhIZ dient zudem dazu, dass für die Steuerung

eines Cursors die Hand lediglich in einem angenehmen Bereich bewegt werden kann. Auch das Problem der unterschiedlichen Körpergrößen wird durch die PhIZ gelöst.

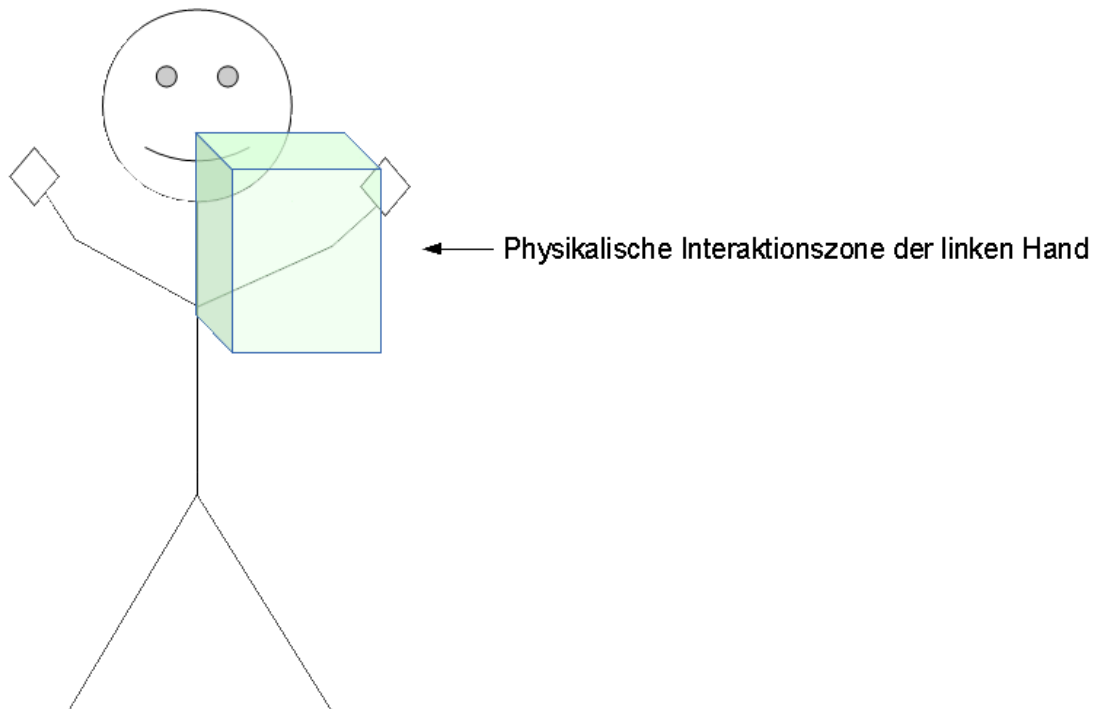


Abbildung 3.8: Skizze der PhIZ

Die PhIZ ist für die rechte und linke Hand spiegelverkehrt. Leider gibt es von Microsoft keine detailliere Beschreibung der PhIZ. Bei der linken Hand beginnt sie auf der Horizontalen am Mittelpunkt des Körpers und geht circa bis zum seitlich ausgestreckten linken Ellenbogen. Vertikal beginnt sie ebenfalls circa beim "Spine" Joint und geht ca. bis zur Höhe des "Head" Joints (siehe Abbildung 3.7).

Im Anhang C (6.3) befindet sich eine Auflistung von Elementen und der Aufbau des Interaction Streams.

Audio Stream Der Audio Stream nutzt das PCM-Format für die Encodierung bei 16000 Hz und 16 Bits pro Sekunde.

Über vier interne Mikrofone liefert die Kinect nicht nur einen Audio Stream, sondern auch den Winkel der Sound-Quelle zur Kinect. Dadurch ist eine Zuweisung der Sprache auf ein

bestimmtes Skelett von mehreren möglich. Da das Testszenario lediglich von einer Person vor dem Smart Mirror ausgeht, wird auf diese Funktion nicht weiter eingegangen.

Leap Motion

Der Leap Motion Controller von Leap Motion Inc. ist ein kleiner USB Sensor durch den über 3D-Gesten ein Computer gesteuert werden kann. Der Sensor tastet einen Bereich von bis zu einen Meter in die Tiefe ab und liefert mit bis zu 300 fps einen Data Stream.

Data Stream Der Data Stream liefert pro Frame eine Liste aller gefundenen Hände und "Pointable Objects" (z.B. ein Stift). Die Hand enthält wiederum die registrierten Finger.



Abbildung 3.9: Hand Modell des Leap Motion Controller

Zudem liefert ein Frame ebenfalls erkannte Gesten:

Circle Durch das Im-Kreis-Bewegen eines Fingers oder Stiftes wird diese Geste erkannt. Sie liefert ebenfalls eine der Richtungen mit: im Uhrzeigersinn oder gegen den Uhrzeigersinn.

Screen Tabs Durch das nach vorne Bewegen eines Fingers wird diese Geste erkannt. Sie ist äquivalent zu der Pressgeste der Kinect, wobei die Kinect zusätzlich eine Prozentzahl, die angibt wie weit bereits “gedrückt” wurde, mitliefert.

Key Tab Durch das Runterdrücken eines Finger wie bei einer Klaviertaste wird diese Geste erkannt.

Swipe Die Swipe-Geste wird durch eine schnelle lineare Bewegung eines Fingers oder eines Stiftes erkannt. Allerdings wird besonders bei dieser Geste die Problematik des Start-End-Problems (vgl. Abschnitt 2.9.5) deutlich, da dem User nicht intuitiv bewusst ist, bei welcher Geschwindigkeit der Controller die Bewegung als Swipe-Geste oder als normale Bewegungsverfolgung interpretiert.

Spiegel-Display

Das Spiegel-Display besteht aus zwei Komponenten, dem 27-Zoll LCD und einem speziellen Einwegspiegel von 596 x 335 mm. Der Spiegel wurde über der Display-Fläche fixiert und am Rand mit einen Holzrahmen verkleidet (siehe Abbildung 2.9). Das Display hat eine Auflösung von 1920x1080 Pixel und somit ein Bildverhältnis von 16:9, und wird von der GUI über HDMI angesteuert.

3.6.2 Interpretation

Im Folgenden wird erläutert, wie die Frameworks konfiguriert und die bereitgestellten Daten interpretiert werden.

Kinect Adapter

Der Kinect Adapter setzt auf dem SDK von Microsoft auf und wurde wie das gesamte Projekt in C# umgesetzt. Der Adapter konfiguriert und initialisiert den Depth-, Skeleton- und Audio-Stream. Eine strikte Kapselung von Motion-Tracking-Daten und Audiodaten ist leider nicht möglich, da der Sensor lediglich von einer Anwendung angesteuert werden kann.

Motion Tracking und Gestenerkennung Wie bereits im vorangegangenen Hardwareteil zur Kinect (siehe Abschnitt 3.6.1) beschrieben, werden der Depth und Skeleton Stream an die Interaktionsbibliothek weitergeleitet. Eine Trennung dieser Gesteninterpretation in eine andere Anwendung ist aus den folgenden zwei Gründen nicht möglich:

- Geschlossener Source Code: Das SDK liegt lediglich als compilierte DLL vor. Die enthaltenen Klassen sind nach außen hin geschützt. Daher ist eine manuelle Erstellung einer Instanz dieser Klassen nicht möglich. Folglich kann ein Objekt einer solchen Klasse nicht aus dem JSON-Format deserialisiert werden. Dies ist jedoch nur in dem Fall kritisch, wenn die Daten, wie in diesem Projekt, weiter mit der geschlossenen Interaktionsbibliothek der Kinect verarbeitet werden sollen.
- Datenmenge: Die Übertragung des Depth Stream mit einer Auflösung von 640x480 Pixel hat die zur Verfügung stehende Hardware unter Verwendung eines lokalen Messagebroker-Systems überlastet.

Diese Probleme werden in dem linken Bild von Abbildung 3.10 verdeutlicht.

Die interpretierten Daten des Interaction Streams können von der Größe der Datenmenge über den Messagebroker gesendet werden. Jedoch ist der Konstruktor des Interaction Stream ebenfalls geschützt und der Source Code geschlossen (siehe Abbildung 3.10 Mitte). Da das Toolkit als Source Code bereitgestellt wird, kann dieses Problem durch das Einsetzen eines Wrappers und einer Schnittstellenerweiterung des Toolkits gelöst werden, wie es in der rechten Darstellung der Datenübertragung in Abbildung 3.10 zu sehen ist.

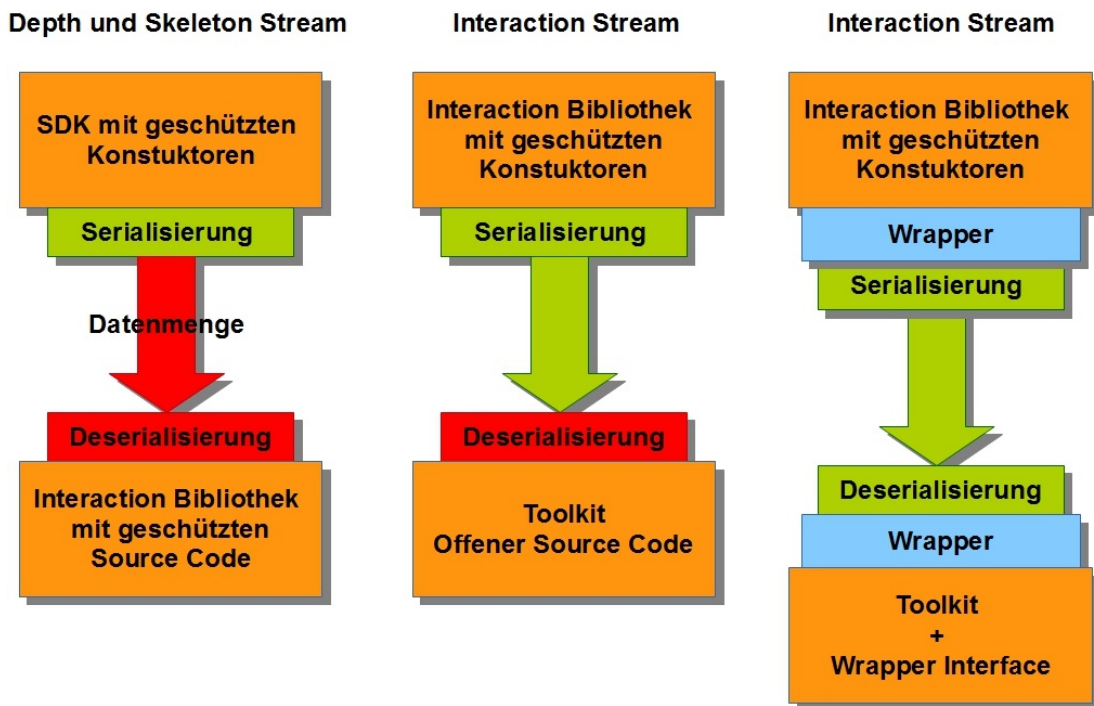


Abbildung 3.10: Darstellung der Datenübertragung der Streams

Durch diese Lösung ist es möglich die Interaktionsdaten über ein Netzwerk zu streamen und die Toolkits der Kinect von Microsoft weiter zu nutzen.

Anmerkung: Diese Designentscheidung wurde mit den Möglichkeiten des SDK 1.7 für die Kinect getroffen. Das SDK 1.8, welches später veröffentlicht wurde, bietet eine Netzwerkschnittstelle per Websockets an. Diese würde jedoch nicht in das Konzept des Messagebroker-Systems der aktuellen Testumgebung passen.

Speech Recognition Aufgrund des Messagebroker-Systems und der Datenmenge wird auf dem Audio Stream im Kinect Adapter eine ersten Low Level Interpretation durchgeführt. Diese Spracherkennung, welche mittels der MS Speech API umgesetzt wurde, rechnet auf Grundlage rekursiv verschachtelter Regeln einer umfangreichen Grammatik die Regel raus, welche eine passende Syntax bereitstellt. Die konkret erkannte Wortfolge und die Höhe der Konfidenz des Ergebnisses werden weiter zum Speech Interpreter gesendet.

Die Konfidenz gibt nicht die Übereinstimmung der Sprache zur Regel wieder, sondern das Zutreffen einer Regel in Abgrenzung zu den anderen Regeln. Sollten Wörter oder Sätze gesprochen werden, die nicht in der Grammatik enthalten sind, kann dennoch eine Regel eine hohe

Konfidenz bekommen, da das Zutreffen für andere Regeln extrem unwahrscheinlich ist.

Die Grammatik ist mittels SRGS (Speech Recognition Grammar Specification) in einer XML-Datei definiert. Dies ermöglicht ein schnelles Austauschen und Erweitern der Grammatik. Die Grammatik, welche in Anhang A (6.1) einzusehen ist, soll dem Nutzer eine möglichst freie Sprachsteuerung des Smart Mirrors ermöglichen. Daher werden eine Vielzahl verschiedener Satzbau-Möglichkeiten unterstützt, welche im deutschen Sprachgebrauch möglich sind. Zudem können verschiedene Wörter benutzt werden. Statt "Öffnen" kann der Nutzer z.B. auch "Anzeigen" oder "Zeige ... an" sagen. Dies wurde durch die Bildung von zehn Regeln in der Grammatik umgesetzt, welche ebenfalls im Anhang A (6.1) aufgelistet sind.

Zum Navigieren kann der Nutzer in seinem Sprachbefehl einen Index nennen (z.B. "Öffne die dritte E-Mail") oder durch "Weiter", "Zurück", "Nächste E-Mail öffnen" zum nächsten oder vorigen Element gelangen.

Dies soll dem Nutzer die Steuerung erleichtern, da er gebrauchssprachliche Formulierungen nutzen kann, und nicht nur einen bestimmten Satzbau bzw. ein bestimmtes Key-Word für eine Aktion benutzen muss. Die Spracherkennung wird daher durch einen Verbundworterkenner realisiert.

Leap Motion Adapter

Das Leap Motion SDK liefert eine Vielzahl an "Pointable Objects" (erkannte Finger und Stifte), wie auch ein Handmodell. Während das Kinect SDK schon für die Hände Informationen über die Aktivitätszustand der Hand mitliefert, muss beim Leap Motion ein anderes Kriterium gefunden werden, um zu entscheiden, welches "Pointable Object" zur Steuerung genutzt wird. Eine Möglichkeit ist es den Zeigefinger zu wählen, da der Mensch meist diesen Finger benutzt, um auf Objekte zu zeigen, oder dies der primäre Finger für die Steuerung auf großen Touchscreens ist. Dieser Ansatz hat jedoch den Nachteil, dass der Nutzer nicht mit einem "Pointable Object" wie einer Zahnbürste oder etwas ähnlichen den Spiegel steuern kann. Zudem ist die Qualität der Bestimmung eines Fingers stark vom Winkel der Hand zum Leap Motion Controller abhängig, sodass nicht gewährleistet ist, dass der Zeigefinger immer als dieser vom SDK erkannt wird.

Stattdessen wird das "Pointable Object" genutzt, welches am weitesten vom Nutzer auf der Z-Achse entfernt ist. Somit kann der Nutzer einen beliebigen Finger, ein "Pointable Object" oder die ganze Hand zum Steuern nutzen.

Interaktionsbereich Der Interaktionsbereich der Leap Motion weist eine ähnlichen Größenordnung wie die PhIZ der Kinect (3.8) auf. Dies führt dazu, dass die Bewegungen mit dem Finger unnötig groß sind. Durch die sehr hohe Auflösung der Leap Motion reicht ein Interaktionsbereich von 24cm x 26cm x 10cm, der über den Adapter konfiguriert werden kann. Dies ermöglicht dem Nutzer nur durch Bewegungen des Handgelenks und des Fingers mit dem Cursor von einem Bildschirmrand zum anderen zu gelangen. Dies hat zudem den Vorteil, dass nicht alle Bewegungen im Sensorbereich der Leap Motion als Gesten interpretiert werden. Dadurch könnte der Nutzer die Größe und die Position des virtuellen Steuerungsbereichs durch eine Konfigurierung anpassen. Aufgrund dessen kann die Körpergröße und die Präferenzen der Hand (linke oder rechte) ausgeglichen werden. Die Entwicklung von 3D-Gesten für verschiedene Körpergrößen ist wie bereits in der Analyse (2.9.5) beschrieben, ein Grundproblem und kann wie eben erläutert ansatzweise gelöst werden.

Gesten

Screen Touch Die mitgelieferte Scree-Touch-Geste (3.6.1) konnte nicht verwendet werden, da der konfigurierte virtuelle Steuerungsbereich kleiner ist als der von der Sensorik abgetastete Gesamtbereich. Zudem beinhaltet ein Attribut des Interaction Streams einen prozentualen Wert, welcher beschreibt, zu wie viel Prozent gerade "gedrückt" wird. Sollte sich das "Pointable Object" am Anfang des Interaktionsbereiches befinden, beträgt dieser Wert 0% und am Ende des Interaktionsbereiches auf der Z-Achse 100%. Diese Berechnung sollte von der aktuellen Konfiguration des virtuellen Interaktionsbereiches abhängig sein. Daher wurde die Pressgeste im Adapter neu berechnet.

$$\left(\frac{z-s}{e-s}\right) = x \begin{cases} \text{falls } x \geq 1 : \text{Press} \\ \text{sonst} : \overline{\text{Press}} \end{cases}$$

z = Aktueller Wert auf der Z-Achse

s = Konfigurierter Start des virtuellen Interaktionbereichs auf der Z-Achse

e = Konfiguriertes Ende des virtuellen Interaktionbereichs auf der Z-Achse

Durch die Berechnung $\frac{z-s}{e-s}$ erhält man den prozentualen Anteil der aktuellen Fingerposition (z) im virtuellen Interaktionsbereich $e - s$. Sollte dieser Wert größer gleich 1 sein, wird die Bewegung als Pressgeste erkannt.

Circle Das Erkennen der Greifgeste einer Hand wird vom Leap Motion SDK nicht erkannt. Als Alternative wurde daher die Circle Geste gewählt. Diese wurde bereits zuvor beim Leap Motion SDK (3.6.1) erläutert.

Speech Interpreter

Der Speech Interpreter formt durch eine "Key-Word"-Analyse aus den Sätzen, welche durch den Speech Recognizer (3.6.2) mit einem Verbundworderkennung erkannt wurden, in konkrete Befehle um. Dabei wird eine Generalisierung mehrerer Wörter (z.B. "öffnen" und "anzeigen") auf eine Entität durchgeführt. Andere Konjunktionen und Formen von Wörtern werden ebenfalls generalisiert (z.B. "nächster", "nächste", "nächsten"). Artikel und andere Hilfsörter fließen nicht in den Befehl ein und werden daher verworfen.

Ein Befehl kann folgende Attribute beinhalten:

- Action - Die Action gibt an welche Aktion ausgeführt werden soll.
- Application - Die Application definiert die Zielanwendung des Befehls.
- View - Die View definiert die Zielansicht der Application des Befehls.
- Orientation - Durch die Orientation kann die Richtung des Scrollens bestimmt werden.
- System - Durch die System-Attribute können die Bedienungsarten und das Display de- oder aktiviert werden.
- Index - Über den Index kann der Nutzer ein bestimmtes Element ansprechen.
- Day - Gibt an ob ein bestimmter Wochentag genannt wurde.
- Name - Gibt Auskunft darüber, ob das System namentlich angesprochen wurde.

Eine detaillierte Auflistung der einzelnen Zustände der Attribute befindet sich im Anhang B (6.2).

Ein Befehl muss nicht alle Attribute beinhalten. Eine "Action" oder eine "Orientation" alleine kann schon für einen sinnvollen Befehl ausreichen, wenn die Anwendung, an die der Befehl gerichtet ist, aus dem aktuellen Kontext den Befehl eindeutig einem Element zuordnen kann.

Kontext-Interpretation

Die Kontext-Interpretation ist in zwei Teile aufgeteilt. Beide sprechen jeweils einen Methodensatz an, welcher die Daten des Models abrufen oder verändern kann.

Gestensteuerung Durch das Erweitern des Toolkits von Microsoft um die Wrapper Schnittstelle können die bereitgestellten Kontrollelemente für die GUI genutzt werden. Die Kontext-Interpretation wird daher größtenteils direkt vom Toolkit durchgeführt. Lediglich das Scrollen der Leap Motion muss gesondert behandelt werden, da diese Aktion gegenüber der Greifgeste nicht im Toolkit für die Kinect vorgesehen ist. Das Konzept der beiden Gesten ist ebenfalls unterschiedlich, da das Scrollen per Greifen eine Start- und eine Endgeste besitzt. Alle Bewegungen zwischen diesen beiden Gesten werden als Scroll-Befehl interpretiert. Dem gegenüber liefert das Leap Motion SDK innerhalb eines einzelnen Frames die Information, dass eine Circle-Geste erkannt wurde. Über den Beginn der Circle-Geste wird das System nicht informiert. Somit müssen mehrere Circle-Gesten hintereinander durchgeführt werden, um eine größere Liste durchzuscrollen. Daher wird aus mehreren Circle-Gesten des Leap Motion SDKs eine neue Geste.

Diese neue abstrakte Scroll-Geste beginnt durch das Kreisen des Fingers in eine Richtung und endet mit dem Aufhören des Kreisens. Dadurch wird die Circle-Geste des SDKs auf die in der Analyse angesprochene Kreisgeste (2.9.5) überführt.

Lösung der Start-Endproblematik von Gesten Die in der Analyse beschriebene Start-Endproblematik (2.9.5) wird durch drei Eigenschaften bzw. Vorkehrungen bewältigt:

- Deaktivierung der Gestensteuerung - Bei Bedarf kann der Nutzer die Gestensteuerung deaktivieren, um falsch positive Ergebnisse bei der Gestenerkennung zu verhindern.
- Interaktionsbereich - Es werden nur Gesten innerhalb des Interaktionsbereichs erkannt. Dieser lässt sich bei der Verwendung des Leap Motion Controllers vom Nutzer konfigurieren und daher in eine für die Steuerung vorteilhafte Position bringen.
- Kontextsensitive Überprüfung einer Geste - Gesten haben lediglich Auswirkungen auf das System, wenn diese auf einem Steuerelement ausgeführt wird. Daher würde z.B. eine falsch positiv erkannte Pressgeste in der Mitte des Smart Mirrors im Hybrid-Modus keine Auswirkungen auf die Daten oder die Ansicht haben, also ein "Klick" ins Leere sein.

Daher wird der Nutzer auf drei Ebenen unterstützt. Durch die Deaktivierung kann der Nutzer steuern, wann das System Gesten erkennt. Durch den Interaktionsbereich steuert der Nutzer in welchem Bereich Bewegungen als Gesten erkannt werden. Die kontextsensitive Überprüfung filtert einen Teil der falsch positiv erkannten Gesten aus.

Sprachsteuerung Die Kontext-Interpretation der Sprachsteuerung überprüft als Erstes die Konfidenz des Befehls. Sollte diese zu gering sein, wird der User aufgefordert den Befehl zu wiederholen. Daraufhin wird überprüft, ob der Befehl eine bestimmte Application beinhaltet. Sollte der Befehl keine Application enthalten, wird davon ausgegangen, dass der Befehl auf die aktive Anwendung bezogen ist.

Im nächsten Schritt wird überprüft, ob eine der möglichen Ansichten explizit angesprochen wird. Wie bei der Application wird als Default die aktive View gewählt, wenn keine andere im Befehl enthalten ist.

Die verschiedenen Ansichten können wiederum die Actions ausführen, sollten diese zulässig sein. In der Auswertung der Action werden noch die anderen Attribute, wie die Orientation oder der Index einbezogen.

Durch diese Vorgehensweise kann der User innerhalb eines Befehls komplette Anweisungen wie "Spiegel, lösche die dreizehnte E-Mail aus der E-Mail Liste" geben. Einfache Befehle wie "Lösche die E-Mail" können ebenfalls umgesetzt werden, sollte eine E-Mail in der Liste markiert oder in der Detailansicht geöffnet sein.

Da der Speech Recognizer dauerhaft nach Wörtern lauscht, würde das System auch bei leichten Nebengeräuschen den Nutzer auffordern, den Befehl zu wiederholen. Um dies zu verhindern, wird über das Attribut "Name" des Sprachbefehls überprüft, ob der Nutzer das System konkret anspricht.

Sollte der Nutzer jedoch eine Reihe von Befehlen per Sprachsteuerung geben wollen, ist das dauernde namentliche Ansprechen des Smart Mirrors störend. Deswegen kann der Nutzer über den Befehl "Hallo, Spiegel" dem System mitteilen, dass alle folgende Befehle ohne die namentliche Benennung des Spiegels gesprochen werden. Sollte der Nutzer wieder zur Standardeinstellung der Sprachsteuerung zurückkehren wollen, muss er den Befehl "Tschüss Spiegel" sprechen. Durch diese Umsetzung kann sich die Sprachsteuerung in einen der folgenden drei Zuständen befinden.

1. Standard - In diesem Zustand muss der Nutzer den Spiegel namentlich ansprechen, damit der Befehl auf den Kontext interpretiert wird. Bei Hintergrundgeräuschen wie einem Föhn oder Straßengeräuschen, welche als "Sprache" erkannt werden, wird der Nutzer nicht aufgefordert den Befehl zu wiederholen.
2. Frei - In diesem Zustand wird für jeden Befehl eine Kontextinterpretation ausgeführt, unabhängig davon, ob das System namentlich angesprochen wurde oder nicht.

3. Deaktiviert - Kein Befehl wird interpretiert. Daher ist die Aktivierung nur noch per Gestensteuerung möglich.

4 Evaluation

Im Evaluationskapitel wird auf die technische Machbarkeit, die Usability der Steuerung sowie die Überprüfung der Umsetzung der Anforderungen eingegangen.

4.1 Technische Machbarkeit

Die technische Machbarkeit eines interaktiven Badezimmerspiegels wurde durch diese Arbeit gezeigt. Die in der Analyse beschriebene Darstellung (2.8.2) und die funktionalen Anforderungen wie u.a. die Mischform (Hybrid-Modus) der Moduswahl (2.2.2) konnten ebenfalls vollständig umgesetzt werden.

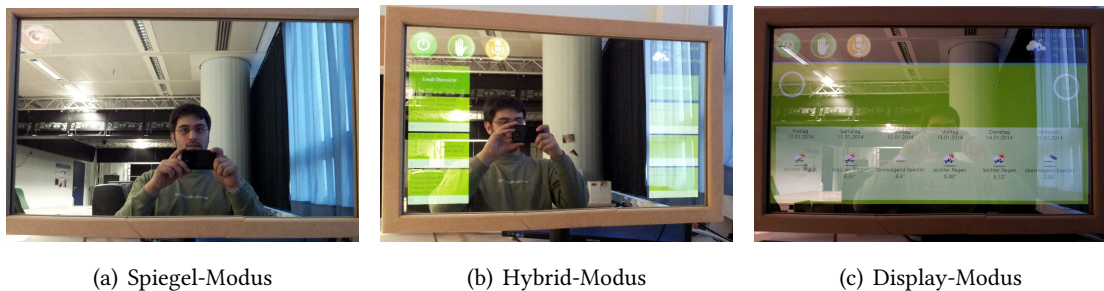


Abbildung 4.1: Modus Wahl

Ebenso konnte das geforderte Interaktionsrepertoire (2.2.3) konzipiert, entwickelt und in Alpha-tests (4.1.2) getestet werden. Die Ergebnisse dieser Tests werden später im Abschnitt Usability (4.2) näher erläutert.

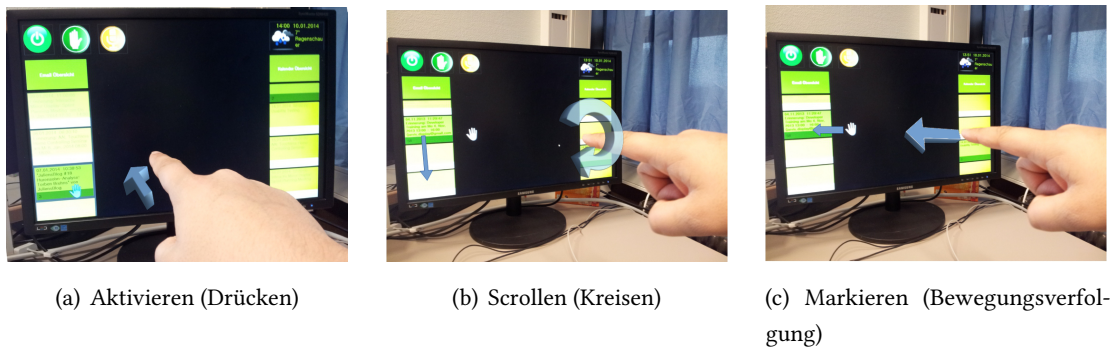


Abbildung 4.2: Leap Motion Gesten

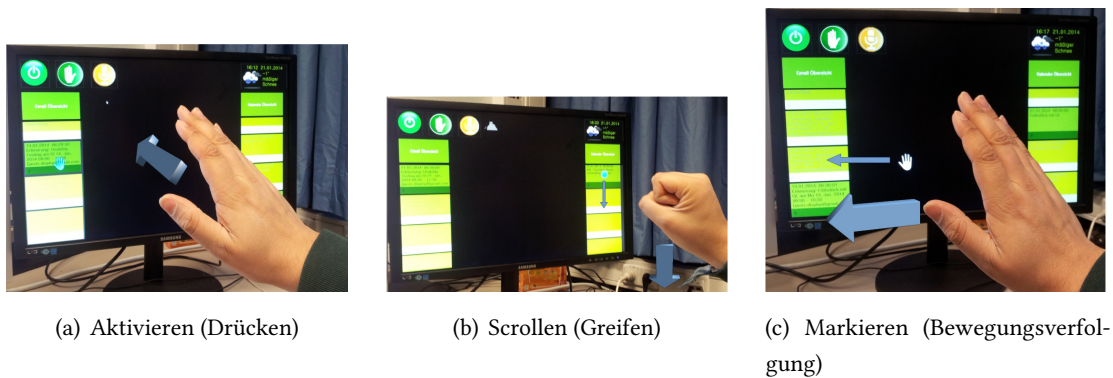


Abbildung 4.3: Kinect Gesten

Die in der Analyse erläuterten Rahmenbedingungen (2.2.4), wie die Umsetzung des Interaktionsbereichs und die berührungslose Steuerung konnten ebenfalls konzipiert und umgesetzt werden. Da die Authentifizierung nicht zu den Kernanforderungen gehört, wurde sie im Design (3.3) lediglich konzipiert, da eine Implementierung im zeitliche Rahmen des Projekts nicht umzusetzen war.

Die Konzipierung und Implementation des Systems hat sich an den geforderten Anwendungsfällen (2.2.5) orientiert. Daher können diese mittels des Smart Mirrors ausgeführt werden. Auf die allgemeine Aufgabenstellung sowie die Zielsetzung wird ebenfalls im Abschnitt Usability (4.2) näher eingegangen.

4.1.1 Testumgebung

Das Smart-Mirror-System nutzt dasselbe Messagebroker-System (3.2), welches im Living Place genutzt wird. Daher ist eine Überführung des Smart-Mirror-Systems ins Living Place ohne großen Aufwand möglich.



Abbildung 4.4: Living Place als Testumgebung

Weitergehend wird auf den konkreten Spiegelaufbau eingegangen.

Spiegel Aufbau

Das Spiegel-Display wird auf Augenhöhe montiert. Direkt über das Spiegel-Display wird die Kinect mit einem Neigungswinkel von ca. 20° nach unten eingestellt und fixiert. Der Leap Motion Controller wird auf dem Waschbecken in Höhe von ca. 80 cm montiert, wobei unterschiedliche Höhen durch die Konfiguration des virtuellen Interaktionsbereiches ausgeglichen werden können.

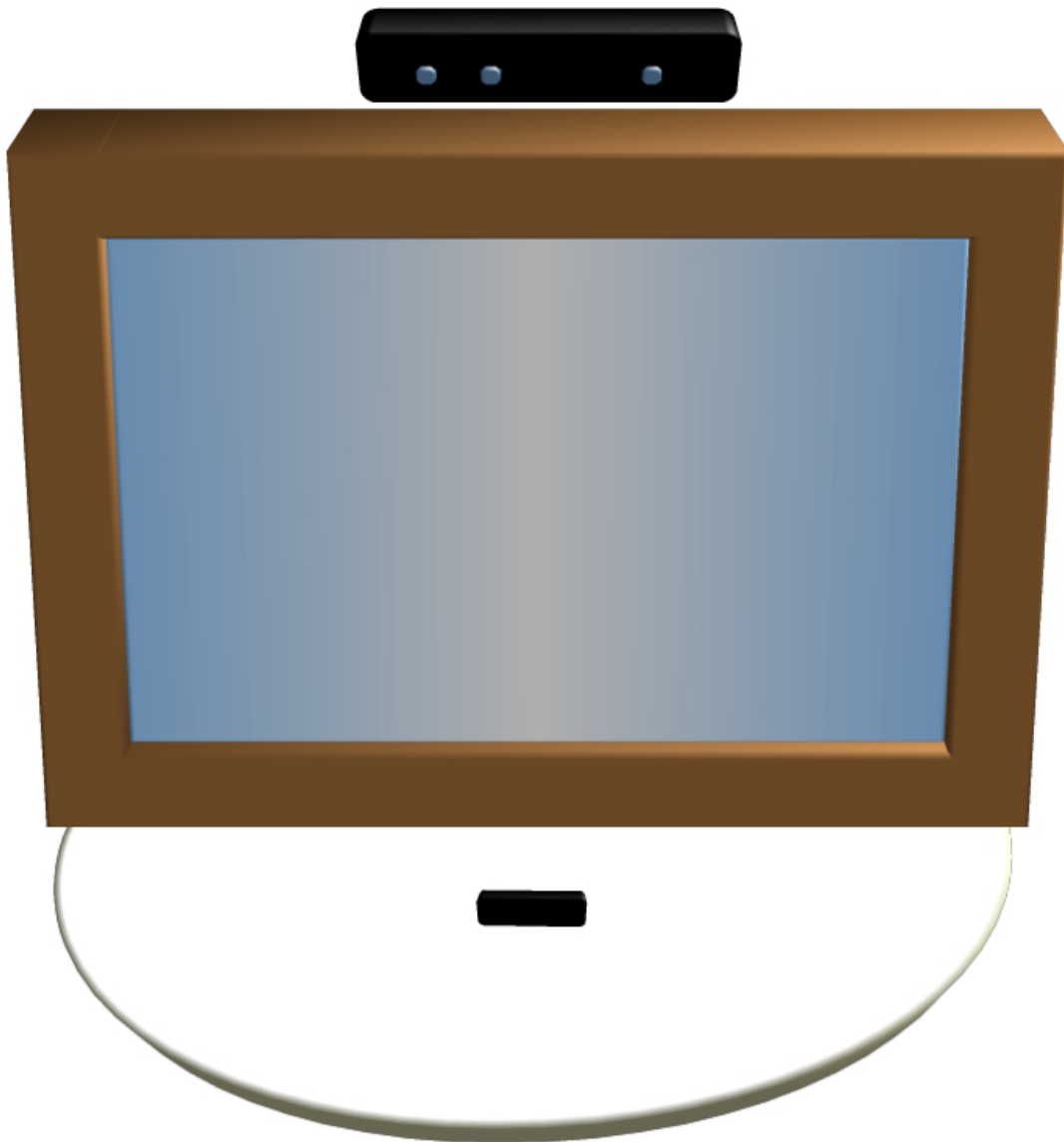


Abbildung 4.5: Aufbau des Spiegel-Display und der Sensorik (oben Kinect, mitte Spiegel-Display, unten Leap Motion)

In den Tests hat sich gezeigt, dass für eine gute Lesbarkeit der Inhalte auf dem Spiegel-Display, der Hintergrund, welcher im Spiegel von der Testperson zu sehen ist, keine Lampen, stark reflektierte oder stark angestrahlte Objekte aufweisen sollte.

4.1.2 Alphatests

Es wurden verschiedene Alphatests im Entwicklungsbüro und dem Living Place durchgeführt. Unter anderem bei Vorstellungen des Smart Mirrors bei der “Hamburger Nacht des Wissens 2013” <http://nachtdeswissens.hamburg.de/> und dem “World Usability Day 2013” <http://worldusabilityday.org>.

Bei den Alphatests im Living Place konnte die Usability des Systems bei Nutzern ohne eine Einführung in die Steuerung betrachtet werden, bei den Tests im Entwicklungsbüro mit einer kleinen Einführung. Daher können bereits erste Schlussfolgerungen über den Grad der Intuitivität der Steuerung gefasst werden. Diese werden ebenfalls im folgenden Abschnitt Usability (4.2) erläutert.

4.2 Usability

Die Usability wurde in den verschiedenen Entwicklungsstadien durch mehrere Alphatests (4.1.2) untersucht. Daher konnten teilweise erste Systemschwächen ausgebessert werden, sodass bei der Nacht des Wissens ein solides System getestet werden konnte.

4.2.1 Gewonnene Erkenntnisse

Es zeigte sich, dass die Steuerung per Gesten mit großem Interesse aufgenommen worden ist. Das konkrete Szenario für die Eignung im Badezimmer wurde von einigen Personen infrage gestellt. Dies lag jedoch nicht an der technischen Umsetzung des Smart Mirrors, sondern an der grundsätzlichen Fragestellung, ob ein Informationssystem im Badezimmer gewünscht wird. Zudem konnte beobachtet werden, dass besonders junge Menschen einen guten Zugang zur Gestensteuerung fanden. Dies wird unter anderem darauf zurückzuführen sein, dass die Gestensteuerung über Computer und Konsolenspiele bereits dieser Generation bekannt ist. Im Folgenden wird konkret auf einige Probleme der Sprach- und Gestensteuerung eingegangen.

Sprachsteuerung

Die Sprachsteuerung hat bei geübten Nutzern eine hohe Erkennungsrate, nur wenige Wiederholungsaufforderungen und kaum falsche Erkennungen. Dafür muss der Nutzer jedoch laut, langsam und deutlich reden. Die im Folgenden genannten Probleme haben sich bei der Nutzung der Microsoft Speech API (3.5.1) in der Testumgebung (4.1.1) gezeigt.

Zudem ist wichtig, dass der Nutzer nach Abbrechen eines Sprachbefehls solange wartet, bis das System bereit für eine neue Spracheingabe ist. Dies ist für den Nutzer ungewohnt, da bei

der Mensch-zu-Mensch Kommunikation ein falsch begonnener Satz meist durch das schnelle Fortführen des berichtigen Satzes richtig gestellt wird. Wie ein Alphatest gezeigt hat, würde eine Anzeige, welche dem Nutzer eine Rückmeldung gibt, ob das System bereit für einen neuen Sprachbefehl ist, die Usability des Systems erhöhen.

Sprachwahl und Aussprache Die Sprachsteuerung wurde während der Entwicklung von Englisch auf Deutsch umgestellt, da eine richtige und deutliche Aussprache der Wörter ungenügend wichtig für die Verlässlichkeit der Spracherkennung ist. Dies zeigte sich ebenfalls bei nach Deutschland immigrierten Mitmenschen. Trotz einer korrekten Grammatik und einer für Menschen sehr gut verständlichen Aussprache, hatte das System Probleme die Sätze dieser Personen korrekt zu erkennen.

Hall und Hintergrundgeräusche Hintergrundgeräusche können das System dazu veranlassen, die Spracherkennung zu starten. Sollte der Nutzer während einer solchen Falscherfassung beginnen eine neuen Befehl zu geben, kann das System diesen nicht richtig interpretieren, ähnlich wie bei der zuvor beschriebenen Berichtigung eines Satzes.

Hier müsste überprüft werden, welche Filtermechanismen das Ergebnis verbessern können.

Namentliches Ansprechen Wie bereits im Kapitel Design und Realisierung (3.6.2) erläutert wurde, ist das namentliche Ansprechen des Spiegels zum einen nötig, damit das System den Nutzer nicht bei jedem Geräusch auffordert den Befehl zu wiederholen. Zum anderen ist es störend bei einer Reihe von Befehlen, die nacheinander gesprochen werden.

Die ebenfalls im Abschnitt 3.6.2 erläuterten Zustände der Sprachsteuerung bieten dafür eine Lösung.

Psychologische Aspekte Bei den Alphatests zeichnete sich ab, dass einige Probanden eine Hemmung haben, mit einem Spiegel bzw. Computer zu sprechen. Zudem sprechen einige Nutzer in unkorrekter deutscher Grammatik die Befehle aus, was eventuell auf die Verunsicherung, ob der Spiegel sie verstehen kann, zurückzuführen ist, da diese Testpersonen sonst grundsätzlich ihre Sätze grammatikalisch korrekt formulieren.

Gestensteuerung

Grundsätzlich konnte durch die Alphatests gezeigt werden, dass die Nutzer nach einer kleinen Einweisung und kurzer Eingewöhnungsphase das System schnell und präzise per Gestensteuerung bedienen konnten. Dies wurde zusätzlich durch die Gestaltung der GUI mit großen

Elementen gefördert. Im Folgenden wird auf einige Probleme eingegangen, welche sich teilweise besonders bei der ersten Nutzung ohne Einführung gezeigt haben.

Latenz Probleme Bei der Remote-Steuerung des Smart Mirrors durch die Kinect ergab sich eine störende Latenz. Da sich die Latenz bei der Remote-Nutzung per Leap Motion in einem nicht störenden Bereich befindet, wird davon ausgegangen, dass die starke Latenz bei der Remote-Nutzung der Kinect durch die Berechnung der Gestenerkennung der Interaction Bibliothek in Verbindung mit der Auslastung des Message Brockers, auf der zur Verfügung stehenden Hardware, verursacht wird.

Für Testzwecke wurde daher das Motiontracking und die Gestenerkennung des Kinect Adapters direkt an die grafische Oberfläche gebaut. Dies führte zu einer starken Verringerung der Latenzzeiten.

Durch die neuen Features in der SDK Version 1.8 könnte eine Remote-Steuerung mit geringeren Latenzzeiten umgesetzt werden. Doch wie bereits bei der Beschreibung des Kinect Adapters (3.6.2) angemerkt worden ist, passt dieses Vorgehen nicht zur Kommunikationsstruktur des Living Places.

Interaktionszone Die Probanden hatten teilweise Probleme, den Interaktionsbereich zu finden. Dies kann auf drei Gründe zurückgeführt werden:

- **Initialisierungszeit der Kinect:** Wenn ein neuer Nutzer den Interaktionsbereich der Kinect betritt, benötigt das System Zeit, um den neuen Nutzer zu erkennen. Einige Nutzer sind davon ausgegangen, dass sie das System sofort steuern können, sobald sie vor dem Smart Mirror stehen. Die Erkennung des Nutzers durch die Kinect kann jedoch einige Zeit in Anspruch nehmen (ca. 10 Sekunden) und teilweise erst durch eine Bewegung des Nutzers erfolgen.
Dies führte bei einigen Nutzern zu Verunsicherungen. Daher sollte den Nutzern ein Feedback über den aktuellen Status der Sensorerfassung gegeben werden.
- **Sensoren-Vielfalt:** Gelegentlich war für den Nutzer nicht klar erkenntlich, welcher der beiden Sensoren aktuell aktiv ist. Daher haben die Nutzer teilweise versucht, das System über Gesten im Interaktionsbereich der Kinect zu steuern, wobei die Leap Motion als Steuerungseinheit aktiviert war. Ein Hinweis, welcher Sensor aktiv ist, sollte ebenfalls dem Nutzer durch die grafische Oberfläche gegeben werden.
- **Unsichtbarer virtueller Interaktionsbereich der Leap Motion:** Der konfigurierbare virtuelle Interaktionsbereich der Leap Motion ist mit der Standardgröße von 24cm x 24cm

x 10cm recht klein. Da es keine Markierungen gab, war es für einige Nutzer schwer, diesen Interaktionsbereich zu finden. Eine initiale Festlegung der Größe und Position durch den Nutzer selbst könnte hier dem Nutzer helfen. Auch das Nutzen von zusätzlichen Hilfsmitteln wie ein intelligentes Maßband oder das Einblenden der virtuellen Interaktionszone im Spiegelbild kann den Nutzer bei der Konfiguration und Nutzung diesbezüglich unterstützen.

Kinect Bei Nutzung der Kinect kommt es gelegentlich zu großen Fehlern des Motion Trackings, welche sich durch schnelles Hin- und Herspringen des Cursors äußern. Eine genaue Untersuchung dieser Problematik war im Rahmen dieser Arbeit nicht möglich. Jedoch konnten drei mögliche Fehlerquellen identifiziert werden:

- Störende Infrarotstrahlung - Durch Sonneneinstrahlung im Interaktionsbereich bzw. auf den Sensor, und Wechselwirkungen zwischen dem Leap Motion Controller und der Kinect.
- Skeletterkennung/Primärer Nutzer - Springende Wahl des primären Nutzers bei fehlerhafter Skeletterkennung.
- Hand am Körper in der PhIZ - Fehlerhafte Berechnung der Position der Hand in der PhIZ, sollte diese zu nah am Körper sein.

Eine Verbesserung dieser Probleme könnte durch die bessere Sensorik der Kinect 2 geschehen, welche im März 2014 erscheinen soll, oder auch ein Software Update für die Filterung störender Infrarotstrahlung, wie es durch die Leap Motion Software durchgeführt wird.

Erschöpfung des Arms durch längere Nutzung Ein weiteres Problem, welches die Usability des Systems einschränkt, ist das bei längerer Bedienung des Systems der Arm ermüdet. Dies tritt bei der Nutzung der Kinect schneller auf, da die Bewegungen größer sind und die Hände weiter vom Körper entfernt sind. Dabei ist jedoch zu beachten, dass der Smart Mirror kein System ist, welches vom Nutzer ununterbrochen über einen längeren Zeitraum gesteuert wird.

Gesamtkonzept der Steuerung

Die freie Wahl der Steuerungsmöglichkeiten des Systems ermöglicht die Nutzung während verschiedenen Tätigkeiten im Badezimmer. Durch das relativ kleine Interaktionsrepertoire (2.2.3) kann das gesamte System mit beiden Steuerungsarten genutzt werden. Jedoch zeigte

sich bereits in dieser Arbeit, dass die Sprachsteuerung grundsätzlich mächtiger ist, als die Gestensteuerung. Dies wird am Smart Mirror durch Befehle wie “Scrolle zur fünfundzwanzigsten E-Mail” deutlich. Dieser Befehl lässt sich durch die mit den von den SDKs zur Verfügung gestellten Gesten nicht benutzerfreundlich umsetzen. Der einfache, nicht so mächtige Sprachbefehl “Scrolle runter” ist äquivalent zur Scroll-Geste der Gestensteuerung. Weitere denkbare Sprachbefehle wie “Öffne die E-Mail von Hans” lassen sich nicht durch Gestensteuerung umsetzen, da das gesamte Gesten-Repertoire gegenüber dem Wörterrepertoire eines Menschen um ein Vielfaches kleiner ist. Lediglich das Erlernen einer Gebärdensprache würde dem Nutzer ermöglichen, durch Gesten ein ähnlich mächtiges Interaktionrepertoire zu haben.

4.3 Fazit

Zusammenfassend konnte diese Arbeit erfolgreich die technische Machbarkeit eines Smart Mirrors für das Badezimmer zeigen. Eine ausreichend gute Usability der beiläufigen Steuerung ist nach einer kurzen Trainingsphase des Nutzers ebenfalls gewährleistet. Die aufgezeigten Probleme konnten teilweise bereits gelöst werden oder lassen sich mit bestehenden Technologien und durch geringfügigen Systemanpassungen lösen. Die Fragen der allgemeinen Aufgabenstellung (2.2.1) konnten durch die Analyse sowie durch das Design und die Tests beantwortet werden. Die funktionalen Anforderungen (2.2.2) konnten erfüllt werden (vgl. 4.1), des Weiteren konnten die formulierten Ziele des Arbeit (2.2.6) erreicht werden.

5 Schluss

5.1 Zusammenfassung

In dieser Arbeit wurde untersucht, wie ein Smart Mirror für das Badezimmer im Kontext eines Smart Homes konzipiert und auf welche Art mit diesem interagiert werden kann. Auf Grundlage der in der Analyse erarbeiteten Rahmenbedingungen (2.2.4) und Anforderungen (2.2.2) wurde im Kapitel Design und Realisierung (3) die Software- und Hardwarearchitektur geplant und umgesetzt.

Anhand erster Alphatests (4.1.2) konnte eine Evaluation (4) des Systems durchgeführt werden. Der Smart Mirror wird in Kürze im Living Place (2.5) installiert und steht damit als fester Bestandteil des Forschungslabors weiteren Projekten und Arbeiten zur Verfügung.

Insbesondere die interaktive Steuerung über Gesten und Sprache wurde in dieser Arbeit primär untersucht, da diese als sinnvolle Steuerungsmedien gewählt wurden (2.2.2). Nach der Entwicklung eines Interaktionsrepertoires in der Analyse (2.2.3) wurde dies durch das Design für die Sprachsteuerung wie auch die Gestensteuerung umgesetzt. Konkret wurde für die Sprachsteuerung eine Grammatik (3.6.2) erarbeitet sowie ein Befehls- (3.6.2) und Kontextinterpret (3.6.2). Für die Gestensteuerung wurde im Design- und Realisierungskapitel eine Übersicht der durch die SDKs zur Verfügung stehenden Gesten erstellt (Leap Motion 3.6.1, Kinect 3.6.1) und pro Sensor ein Adapter (Leap Motion 3.6.2, Kinect 3.6.2) entwickelt. Diese Adapter setzen eine Überführung der Sensordaten und bereitgestellten Gesten zu den geforderten Interaktionsrepertoire (2.2.3) um.

Die Befehle werden durch den Kontext-Interpreter (3.6.2) aufgrund des aktuellen Anwendungskontextes der Benutzeroberfläche auf ihre mögliche Durchführung hin überprüft.

In der Evaluation (4) wurde gezeigt, dass die technische Machbarkeit (4.1) des Smart Mirrors erfolgreich bewiesen werden konnte. Mittels der durchgeführten Alphatests (4.1.2) konnten erste Rückschlüsse über die Usability (4.2) des System gezogen werden. Die gewonnenen Erkenntnisse (4.2.1) gehen konkreter auf die Probleme und Lösungen des Systems unter Usability-Aspekten

ein.

Im Fazit der Evaluation (4.3) konnte der Schluss gezogen werden, dass das System den Anforderungen entspricht und das Projekt erfolgreich abgeschlossen werden konnte.

5.2 Ausblick

Durch die Entwicklung des Consumer-Marketes im Bereich der 3D-Gestensteuerung der letzten Jahre ist zu erwarten, dass hier in naher Zukunft ein großes Wachstum stattfinden wird. So bietet zum Beispiel HP als erster Notebook-Hersteller seit Ende 2013 ein Gerät mit integriertem Leap Motion Sensor an [Heise Online (2013), HP Computers YouTube Channel (2013)].

Die Sprachsteuerung gewinnt seit der Einführung von Siri und der Einbindung von Sprachbefehlen in Google Now ebenfalls eine immer größere Marktrelevanz. Daher wird die Verbindung beider Steuerungsarten in Zukunft wahrscheinlich unseren Umgang mit Computern und computerunterstützten Produkten wie Smart Things stark beeinflussen.

Für das Smart-Mirror-Projekt an der HAW wäre evtl. eine weitere Erforschung handfreier Gesten wie die Steuerung über Füße, Mimik oder Kopfbewegungen möglich. Eine genaue Untersuchung im Bezug auf Interactive Design unter Einbezug anderer Fachrichtungen konnte im Rahmen dieser Arbeit nicht umgesetzt werden.

Eine Verbesserung des Systems könnte durch die Nutzung der im März erscheinenden Kinect 2 möglich sein, wie auch die Nutzung leistungsfähiger Hardware.

Der Smart Mirror zeigt u.a. wie Disappearing Computing in einem Smart Home am Beispiel eines Badezimmerspiegels umgesetzt werden kann. Es ist davon auszugehen, dass sich der momentane Trend des Smart Homes durch intelligente Haussteuerung hin zu kompletten Companion Systems entwickeln wird. Dabei wird deutlich, dass es sich inzwischen nicht mehr um Mensch-Computer-Interaktion (HCI), sondern um Mensch-Maschinen-Interaktion handelt, da für den Menschen keine Computer im eigentlichen Sinne zu erfassen sind.

Die Forschung, wie z.B. durch Baudisch u. a. (2010) von Imaginary Interfaces, deren Interaktionsbereich gegenüber dem Smart Mirror nicht auf eine kleine statische Fläche beschränkt ist, kann künftig ebenso eine starke Rolle spielen, sollten diese Ansätze weiter verfolgt werden. Die erste Basis für solche Systeme wird bereits durch "Internet of Things", also intelligente Geräte geschaffen. Die vielfältigen Entwicklungen bereiten den Weg für Companion Systems, welche uns zukünftig unterstützen könnten, vor. Das Bundesministerium für Bildung und Forschung hat ebenfalls bereits Ende 2012 die Entwicklung folgendermaßen formuliert:

“2.1 Multimodale Schnittstellen für eine Gesellschaft der Vielfältigkeit

Aktuelle technologische Entwicklungen, insbesondere in der Sensorik, ermöglichen eine präzise Wahrnehmung der Umgebung, der Intention oder des kognitiven oder emotionalen Zustands des Nutzers. Dadurch entstehen technische Lösungen, die maßgeschneidert auf den jeweiligen Kontext und den individuellen Nutzer reagieren können und ihm so ein besseres Nutzungserlebnis und eine angemessene Unterstützung bieten. Sie können also einer großen Vielfalt von Lebensbedingungen, Fähigkeiten und Anforderungen verschiedenster Nutzer gerecht werden. Systeme mit diesen Eigenschaften sollten außerdem möglichst unauffällig in die Umgebung oder in einen technischen Gegenstand integriert werden, sodass sie im Alltag nicht sichtbar für den Nutzer sind. Dies ist beispielsweise für Vorhaben von Bedeutung, die durch den Einsatz solcher Technologien die Bedienbarkeit von Alltagstechnik verbessern. ...”

Bundesministerium für Bildung und Forschung (2012)

Vorreiter für Multimodale Schnittstellen und dabei insbesondere für die 3D-Gestensteuerung ist die Spieleindustrie. Durch das Bereitstellen von Geräten wie der Kinect von Microsoft wächst die jüngere Generation bereits mit Interaktionen per 3D-Gesten auf. Dabei gilt es zu erforschen, ob inzwischen Imaginary Interfaces (vgl. **Baudisch u. a. (2010)**) ohne visuelles Feedback von gestensteuerungsaffinen Nutzern für den produktiven Einsatz bei unkritischen Systemen möglich sind.

Da es bereits erste Smart-Mirror-Produkte gibt, kann davon ausgegangen werden, dass zukünftig weitere Produkte dieser Art entwickelt werden. Dabei stellt sich die Frage in wieweit Erkenntnisse aus Bereichen wie HCI und Seamless Interaktion in die Produkte mit einfließen, da für einen großen Teil der Nutzer eine leichte Bedienung zur Annahme des Produktes wesentlich beiträgt. Ebenso muss genauer erforscht werden, wie stark die Aspekte des Disappearing Computing für die Annahme neuartiger Geräte eine Rolle spielen, da die intuitive Erwartungshaltung der Bedienung von Haustechnik sich von der Bedienung eines klassischen Computers stark unterscheidet.

6 Anhang

6.1 A Grammatical Speech Recognition

Vorgegebene Satzbaumöglichkeiten durch Regeln:

1. [Name] Verb [Artikel] ([Nummer] Norm/Norm Nummer) [(Richtung/Tag)]
2. [Name] [Artikel] Norm [(Tag/Nummer)] Verb
3. [Name] [Artikel] [Nummer] Norm Verb
4. [Name] [Artikel] Nummer Norm [Verb]
5. [Name] Verb [Präposition] [Richtung Präposition] [Artikel] (Nummer Norm/Norm/ Norm Nummer)
6. [Name] Verb Präposition Richtung
7. [Name] Verb Adverb Präposition [Artikel] (Nummer Norm/ Norm Nummer)
8. [Name] [Adverb] Präposition [Artikel] (Nummer Norm/Norm Tag/Norm Nummer) [Verb]
9. [Name] Präposition (Richtung/Norm) Verb
10. [Name] (Verb [[Artikel] Norm] Adverb/Adverb Verb)
11. ("Hallo"/"Tschüß") Name

Vollständige XML Datei der Grammtik:

```
1  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2  xmlns:sapi=
3  "http://schemas.microsoft.com/Speech/2002/06/SRGSExtensions"
4  xsi:schemaLocation="http://www.w3.org/2001/06/grammar
5  http://www.w3.org/TR/speech-grammar/grammar.xsd"
6  xml:lang="de-DE" tag-format="semantics/1.0" version="1.0"
7  mode="voice" root="main">
8
9  <rule id="main" scope="public">
10   <tag>out.index=0;</tag>
11   <item repeat="0-1">
12     <ruleref uri="#namen" />
13   </item>
14   <one-of>
15     <item>
16       <ruleref uri="#ProgrammKommando1"/>
17     </item>
18     <item>
19       <ruleref uri="#ProgrammKommando2"/>
20     </item>
```

```

21     <item >
22         < ruleref uri="#ProgrammKommando3"/>
23     </item >
24     <item >
25         < ruleref uri="#ProgrammKommando4"/>
26     </item >
27     <item >
28         < ruleref uri="#ProgrammKommando5"/>
29     </item >
30     <item >
31         < ruleref uri="#ProgrammKommando6"/>
32     </item >
33     <item >
34         < ruleref uri="#ProgrammKommando7"/>
35     </item >
36     <item >
37         < ruleref uri="#ProgrammKommando8"/>
38     </item >
39     <item >
40         < ruleref uri="#ProgrammKommando9"/>
41     </item >
42     <item >
43         < ruleref uri="#ProgrammKommando10"/>
44     </item >
45     <item >
46         < ruleref uri="#ProgrammKommando11"/>
47     </item >
48 </one-of>
49 <tag>out.index=rules.latest();</tag>
50 </rule >
51
52
53 <rule id="ProgrammKommando1" scope="public">
54     <tag>out=0;</tag >
55     <item >
56         < ruleref uri="#verb" />
57     </item >
58     <item repeat="0-1">
59         < ruleref uri="#artikel" />
60     </item >
61     <one-of>
62         <item >
63             <item repeat="0-1">
64                 < ruleref uri="#nummer" />
65                 <tag>out=rules.nummer;</tag >
66             </item >
67             < ruleref uri="#programme" />
68         </item >
69         <item >
70             < ruleref uri="#programme" />
71             < ruleref uri="#nummer" />
72             <tag>out=rules.nummer;</tag >
73         </item >
74     </one-of>
75     <one-of>
76         <item repeat="0-1">
77             < ruleref uri="#richtung" />
78         </item >
79         <item repeat="0-1">
80             < ruleref uri="#tag" />
81         </item >
82     </one-of>
83 </rule >
84
85 <rule id="ProgrammKommando2" scope="public">
86     <tag>out=0;</tag >
87     <item repeat="0-1">
88         < ruleref uri="#artikel" />

```

```
89     </item >
90     <item >
91         <ruleref uri="#programme" />
92     </item >
93     <one-of >
94         <item repeat="0-1">
95             <ruleref uri="#nummer" />
96             <tag>out=rules.nummer;</tag >
97         </item >
98         <item repeat="0-1">
99             <ruleref uri="#tag" />
100        </item >
101    </one-of >
102    <item >
103        <ruleref uri="#verb" />
104    </item >
105 </rule >
106
107 <rule id="ProgrammKommando3" scope="public">
108     <tag>out=0;</tag >
109     <item repeat="0-1">
110         <ruleref uri="#artikel" />
111     </item >
112     <item repeat="0-1">
113         <ruleref uri="#nummer" />
114         <tag>out=rules.nummer;</tag >
115     </item >
116     <item >
117         <ruleref uri="#programme" />
118     </item >
119     <item >
120         <ruleref uri="#verb" />
121     </item >
122
123 </rule >
124
125 <rule id="ProgrammKommando4" scope="public">
126     <item repeat="0-1">
127         <ruleref uri="#artikel" />
128     </item >
129     <item >
130         <ruleref uri="#nummer" />
131         <tag>out=rules.nummer;</tag >
132     </item >
133     <item >
134         <ruleref uri="#programme" />
135     </item >
136     <item repeat="0-1">
137         <ruleref uri="#verb" />
138     </item >
139 </rule >
140
141 <rule id="ProgrammKommando5" scope="public">
142     <tag>out=0;</tag >
143     <item >
144         <ruleref uri="#verb" />
145     </item >
146     <item repeat="0-1">
147         <ruleref uri="#präposition" />
148     </item >
149     <item repeat="0-1">
150         <ruleref uri="#richtung" />
151         <ruleref uri="#präposition" />
152     </item >
153     <item repeat="0-1">
154         <ruleref uri="#artikel" />
155     </item >
156 <one-of >
```



```
157 <item repeat="0-1">
158 <ruleref uri="#nummer" />
159 <tag>out=rules.nummer;</tag>
160 <ruleref uri="#programme" />
161 </item>
162 <item>
163 <ruleref uri="#programme" />
164 </item>
165 <item>
166 <ruleref uri="#programme" />
167 <ruleref uri="#nummer" />
168 <tag>out=rules.nummer;</tag>
169 </item>
170 </one-of>
171 </rule>
172
173 <rule id="ProgrammKommando6" scope="public">
174 <item>
175 <ruleref uri="#verb" />
176 </item>
177 <item>
178 <ruleref uri="#präposition" />
179 </item>
180 <item>
181 <ruleref uri="#richtung" />
182 </item>
183 <tag>out=0;</tag>
184 </rule>
185
186 <rule id="ProgrammKommando7" scope="public">
187 <item>
188 <ruleref uri="#verb" />
189 </item>
190 <item>
191 <ruleref uri="#adverb" />
192 </item>
193 <item>
194 <ruleref uri="#präposition" />
195 </item>
196 <item repeat="0-1">
197 <ruleref uri="#artikel" />
198 </item>
199 <one-of>
200 <item>
201 <ruleref uri="#nummer" />
202 <tag>out=rules.nummer;</tag>
203 <ruleref uri="#programme" />
204 </item>
205 <item>
206 <ruleref uri="#programme" />
207 <ruleref uri="#nummer" />
208 <tag>out=rules.nummer;</tag>
209 </item>
210 </one-of>
211 </rule>
212
213
214 <rule id="ProgrammKommando8" scope="public">
215 <tag>out=0;</tag>
216 <item repeat="0-1">
217 <ruleref uri="#adverb" />
218 </item>
219 <item>
220 <ruleref uri="#präposition" />
221 </item>
222 <item repeat="0-1">
223 <ruleref uri="#artikel" />
224 </item>
```

```
225 <one-of>
226 <item>
227 <ruleref uri="#programme" />
228 </item>
229 <item>
230 <ruleref uri="#nummer" />
231 <tag>out=rules.nummer;</tag>
232 <ruleref uri="#programme" />
233 </item>
234 <item>
235 <ruleref uri="#programme" />
236 <ruleref uri="#nummer" />
237 <tag>out=rules.nummer;</tag>
238 </item>
239 <item>
240 <ruleref uri="#programme" />
241 <ruleref uri="#tag" />
242 </item>
243 </one-of>
244 <item repeat="0-1">
245 <ruleref uri="#verb" />
246 </item>
247 </rule>
248
249 <rule id="ProgrammKommando9" scope="public">
250 <tag>out=0;</tag>
251 <item>
252 <ruleref uri="#präposition" />
253 </item>
254 <one-of>
255 <item>
256 <ruleref uri="#richtung" />
257 </item>
258 <item>
259 <ruleref uri="#programme" />
260 </item>
261 </one-of>
262 <item>
263 <ruleref uri="#verb" />
264 </item>
265 </rule>
266
267 <rule id="ProgrammKommando10" scope="public">
268 <tag>out=0;</tag>
269 <one-of>
270 <item>
271 <ruleref uri="#verb" />
272 <ruleref uri="#adverb" />
273 </item>
274 <item>
275 <ruleref uri="#adverb" />
276 <ruleref uri="#verb" />
277 </item>
278 <item>
279 <ruleref uri="#verb" />
280 <item repeat="0-1">
281 <ruleref uri="#artikel" />
282 </item>
283 <ruleref uri="#programme"/>
284 <ruleref uri="#adverb" />
285 </item>
286 </one-of>
287 </rule>
288
289 <rule id="ProgrammKommando11" scope="public">
290 <tag>out=1;</tag>
291 <one-of>
292 <item>
```

```
293     Hallo
294   </item >
295 <item >
296   Tschüs
297 </item >
298 </one-of >
299 <item >
300   < ruleref uri="#namen" />
301 </item >
302 </rule >
303
304
305
306 <rule id="namen" scope="public">
307   <one-of>
308     <item>Spiegel </item >
309     <item>Computer </item >
310     <item>Jarvis </item >
311   </one-of>
312 </rule >
313
314 <rule id="richtung" scope="public">
315   <item repeat="0-1">nach </item >
316   <item >
317     <one-of>
318       <item>oben </item >
319       <item>unten </item >
320       <item>rechts </item >
321       <item>links </item >
322     </one-of>
323   </item >
324 </rule >
325
326 <rule id="adverb" scope="public">
327   <item >
328     <one-of>
329       <item>weiter </item >
330       <item>zurück </item >
331       <item>runter </item >
332       <item>hoch </item >
333     </one-of>
334   </item >
335 </rule >
336
337
338 <rule id="verb" scope="public">
339   <one-of>
340     <item>aktiviere </item >
341     <item>aktivieren </item >
342     <item>deaktivieren </item >
343     <item>deaktiviere </item >
344     <item>starte </item >
345     <item>starten </item >
346     <item>schlieÙe </item >
347     <item>schließen </item >
348     <item>beende </item >
349     <item>beenden </item >
350     <item>öffne </item >
351     <item>öffnen </item >
352     <item>wechsel </item >
353     <item>markiere </item >
354     <item>markieren </item >
355     <item>wähle </item >
356     <item>wählen </item >
357     <item>auswählen </item >
358     <item>lösche </item >
359     <item>löschen </item >
360     <item>blättere </item >
```

```

361 <item>scroll </item>
362 <item>scrollen </item>
363 <item>blättern </item>
364 <item>zeige </item>
365 <item>zeigen </item>
366 <item>anzeigen </item>
367 <item>an </item>
368 <item>aus </item>
369 <item>anschalten </item>
370 <item>ausschalten </item>
371 </one-of>
372 </rule>
373
374 <rule id="programme" scope="public">
375 <one-of>
376 <item>
377 <item>E Mails </item>
378 <item repeat="0-1">
379 <ruleref uri="#addon"/>
380 </item>
381
382 </item>
383 <item>
384 <item>E Mail </item>
385 <item repeat="0-1">
386 <ruleref uri="#addon"/>
387 </item>
388
389 </item>
390 <item>
391 <item>Kalender </item>
392 <item repeat="0-1">
393 <ruleref uri="#addon"/>
394 </item>
395
396 </item>
397 <item>
398 <item>Termin </item>
399 <item repeat="0-1">
400 <ruleref uri="#addon"/>
401 </item>
402 </item>
403 <item>
404 <item>Nachrichten </item>
405 <item repeat="0-1">
406 <ruleref uri="#addon"/>
407 </item>
408 </item>
409 <item>
410 <item>Nachricht </item>
411 </item>
412 <item>
413 <item>Wetter </item>
414 <item repeat="0-1">
415 <ruleref uri="#addon"/>
416 </item>
417
418 </item>
419 <item>Tag </item>
420 <item>
421 <item>Woche </item>
422 <item repeat="0-1">
423 <ruleref uri="#addon"/>
424 </item>
425 </item>
426 <item>Hilfe </item>
427 <item>Startbildschirm </item>
428 <item>Bildschirm </item>

```

```
429 <item>Display </item>
430 <item>Handsteuerung </item>
431 <item>Gestenerkennung </item>
432 <item>Sprachbefehlerkennung </item>
433 <item>Voice Commands</item>
434 <item>System </item>
435 <item>Programm </item>
436 </one-of>
437 </rule>
438
439 <rule id="addon" scope="public">
440 <one-of>
441 <item>Detailansicht </item>
442 <item>Übersicht </item>
443 <item>Liste </item>
444 </one-of>
445 </rule>
446
447 <rule id="präposition" scope="public">
448 <one-of>
449 <item>nach </item>
450 <item>zu </item>
451 <item>zur </item>
452 <item>zum </item>
453 <item>in </item>
454 <item>im </item>
455 </one-of>
456 </rule>
457
458
459
460 <rule id="bestaetigung" scope="public">
461 <one-of>
462 <item>ja </item>
463 <item>nein </item>
464 <item>ok </item>
465 </one-of>
466 </rule>
467
468 <rule id="artikel" scope="public">
469 <one-of>
470 <item>der </item>
471 <item>das </item>
472 <item>die </item>
473 <item>dem </item>
474 <item>den </item>
475 </one-of>
476 </rule>
477
478 <rule id="nummer" scope="public">
479 <tag>out=0;</tag>
480 <one-of>
481 <item>
482 <ruleref uri="#adjektivSubjekt" />
483 </item>
484 <item>
485 <ruleref uri="#adjektivObjekt" />
486 </item>
487 <item>
488 <ruleref uri="#digit" />
489 </item>
490 <item>
491 <ruleref uri="#adjektive" />
492 </item>
493 </one-of>
494 <tag>out=rules.latest();</tag>
495 </rule>
496
```

6 Anhang

```
497 <rule id="adjektive" scope="public">
498   <one-of>
499     <item>nächste </item>
500     <item>nächten </item>
501     <item>vorige </item>
502     <item>vorigen </item>
503     <item>letzte </item>
504     <item>letzten </item>
505   </one-of>
506   <tag>out=0;</tag>
507 </rule>
508
509 <rule id="adjektivSubjekt" scope="public">
510   <one-of>
511     <item>
512       erste <tag>out = "1"; </tag>
513     </item>
514     <item>
515       zweite <tag>out = "2"; </tag>
516     </item>
517     <item>
518       dritte <tag>out = "3"; </tag>
519     </item>
520     <item>
521       vierte <tag>out = "4"; </tag>
522     </item>
523     <item>
524       fünfte <tag>out = "5"; </tag>
525     </item>
526     <item>
527       sechste <tag>out = "6"; </tag>
528     </item>
529     <item>
530       siebte <tag>out = "7"; </tag>
531     </item>
532     <item>
533       achte <tag>out = "8"; </tag>
534     </item>
535     <item>
536       neunte <tag>out = "9"; </tag>
537     </item>
538     <item>
539       zehnte <tag>out = "10"; </tag>
540     </item>
541     <item>
542       elfte <tag>out = "11"; </tag>
543     </item>
544     <item>
545       zwölfte <tag>out = "12"; </tag>
546     </item>
547     <item>
548       dreizehnte <tag>out = "13"; </tag>
549     </item>
550     <item>
551       vierzehnte <tag>out = "14"; </tag>
552     </item>
553     <item>
554       fünfzehnte <tag>out = "15"; </tag>
555     </item>
556     <item>
557       sechzehnte <tag>out = "16"; </tag>
558     </item>
559     <item>
560       siebzehnte <tag>out = "17"; </tag>
561     </item>
562     <item>
563       achtzehnte <tag>out = "18"; </tag>
564     </item>
```

```
565 <item >
566     neunzehnte <tag>out = "19"; </tag >
567 </item >
568 <item >
569     zwanzigte <tag>out = "20"; </tag >
570 </item >
571 <item >
572     einundzwanzigte <tag>out = "21"; </tag >
573 </item >
574 <item >
575     zweiundzwanzigte <tag>out = "22"; </tag >
576 </item >
577 <item >
578     dreiundzwanzigte <tag>out = "23"; </tag >
579 </item >
580 <item >
581     vierundzwanzigte <tag>out = "24"; </tag >
582 </item >
583 <item >
584     fünfundzwanzigte <tag>out = "25"; </tag >
585 </item >
586 <item >
587     sechsundzwanzigte <tag>out = "26"; </tag >
588 </item >
589 <item >
590     siebenundzwanzigte <tag>out = "27"; </tag >
591 </item >
592 <item >
593     achtundzwanzigte <tag>out = "28"; </tag >
594 </item >
595 <item >
596     neunundzwanzigte <tag>out = "29"; </tag >
597 </item >
598 <item >
599     dreißigte <tag>out = "30"; </tag >
600 </item >
601 <item >
602     einsunddreißigte <tag>out = "31"; </tag >
603 </item >
604 <item >
605     zweiunddreißigte <tag>out = "32"; </tag >
606 </item >
607 <item >
608     dreiunddreißigte <tag>out = "33"; </tag >
609 </item >
610 <item >
611     vierunddreißigte <tag>out = "34"; </tag >
612 </item >
613 <item >
614     fünfunddreißigte <tag>out = "35"; </tag >
615 </item >
616 <item >
617     sechsunddreißigte <tag>out = "36"; </tag >
618 </item >
619 <item >
620     siebenunddreißigte <tag>out = "37"; </tag >
621 </item >
622 <item >
623     achtunddreißigte <tag>out = "38"; </tag >
624 </item >
625 <item >
626     neununddreißigte <tag>out = "39"; </tag >
627 </item >
628 <item >
629     vierzigte <tag>out = "40"; </tag >
630 </item >
631 <item >
632     einundvierzigte <tag>out = "41"; </tag >
```

```
633     </item>
634     <item>
635         zweiundvierzigte <tag>out = "42"; </tag>
636     </item>
637     <item>
638         dreiundvierzigte <tag>out = "43"; </tag>
639     </item>
640     <item>
641         vierundvierzigte <tag>out = "44"; </tag>
642     </item>
643     <item>
644         fünfundvierzigte <tag>out = "45"; </tag>
645     </item>
646     <item>
647         sechsundvierzigte <tag>out = "46"; </tag>
648     </item>
649     <item>
650         siebenundvierzigte <tag>out = "47"; </tag>
651     </item>
652     <item>
653         achtundvierzigte <tag>out = "48"; </tag>
654     </item>
655     <item>
656         neunundvierzigte <tag>out = "49"; </tag>
657     </item>
658     <item>
659         fünfzigte <tag>out = "50"; </tag>
660     </item>
661 </one-of>
662 </rule>
663
664 <rule id="adjektivObjekt" scope="public">
665     <one-of>
666         <item>
667             ersten <tag>out = "1"; </tag>
668         </item>
669         <item>
670             zweiten <tag>out = "2"; </tag>
671         </item>
672         <item>
673             dritten <tag>out = "3"; </tag>
674         </item>
675         <item>
676             vierten <tag>out = "4"; </tag>
677         </item>
678         <item>
679             fünften <tag>out = "5"; </tag>
680         </item>
681         <item>
682             sechsten <tag>out = "6"; </tag>
683         </item>
684         <item>
685             siebten <tag>out = "7"; </tag>
686         </item>
687         <item>
688             achten <tag>out = "8"; </tag>
689         </item>
690         <item>
691             neunten <tag>out = "9"; </tag>
692         </item>
693         <item>
694             zehnten <tag>out = "10"; </tag>
695         </item>
696         <item>
697             elften <tag>out = "11"; </tag>
698         </item>
699         <item>
700             zwölfnten <tag>out = "12"; </tag>
```



```
701 </item>
702 <item>
703   dreizehnten <tag>out = "13"; </tag>
704 </item>
705 <item>
706   vierzehnten <tag>out = "14"; </tag>
707 </item>
708 <item>
709   fünfzehnten <tag>out = "15"; </tag>
710 </item>
711 <item>
712   sechszehnten <tag>out = "16"; </tag>
713 </item>
714 <item>
715   siebzehnten <tag>out = "17"; </tag>
716 </item>
717 <item>
718   achtzehnten <tag>out = "18"; </tag>
719 </item>
720 <item>
721   neunzehnten <tag>out = "19"; </tag>
722 </item>
723 <item>
724   zwanzigten <tag>out = "20"; </tag>
725 </item>
726 <item>
727   einundzwanzigten <tag>out = "21"; </tag>
728 </item>
729 <item>
730   zweiundzwanzigten <tag>out = "22"; </tag>
731 </item>
732 <item>
733   dreiundzwanzigten <tag>out = "23"; </tag>
734 </item>
735 <item>
736   vierundzwanzigten <tag>out = "24"; </tag>
737 </item>
738 <item>
739   fünfundzwanzigten <tag>out = "25"; </tag>
740 </item>
741 <item>
742   sechszwanzigten <tag>out = "26"; </tag>
743 </item>
744 <item>
745   siebenundzwanzigten <tag>out = "27"; </tag>
746 </item>
747 <item>
748   achtundzwanzigten <tag>out = "28"; </tag>
749 </item>
750 <item>
751   neunundzwanzigten <tag>out = "29"; </tag>
752 </item>
753 <item>
754   dreißigten <tag>out = "30"; </tag>
755 </item>
756 <item>
757   einsunddreißigten <tag>out = "31"; </tag>
758 </item>
759 <item>
760   zweiunddreißigten <tag>out = "32"; </tag>
761 </item>
762 <item>
763   dreiunddreißigten <tag>out = "33"; </tag>
764 </item>
765 <item>
766   vierunddreißigten <tag>out = "34"; </tag>
767 </item>
768 <item>
```

```
769     fünfunddreißigten <tag>out = "35"; </tag>
770   </item>
771 </item>
772     sechsdreißigten <tag>out = "36"; </tag>
773 </item>
774 </item>
775     siebenunddreißigten <tag>out = "37"; </tag>
776 </item>
777 </item>
778     achtunddreißigten <tag>out = "38"; </tag>
779 </item>
780 </item>
781     neununddreißigten <tag>out = "39"; </tag>
782 </item>
783 </item>
784     vierzigten <tag>out = "40"; </tag>
785 </item>
786 </item>
787     einundvierzigten <tag>out = "41"; </tag>
788 </item>
789 </item>
790     zweiundvierzigten <tag>out = "42"; </tag>
791 </item>
792 </item>
793     dreiundvierzigten <tag>out = "43"; </tag>
794 </item>
795 </item>
796     vierundvierzigten <tag>out = "44"; </tag>
797 </item>
798 </item>
799     fünfundvierzigten <tag>out = "45"; </tag>
800 </item>
801 </item>
802     sechsvierzigten <tag>out = "46"; </tag>
803 </item>
804 </item>
805     siebenundvierzigten <tag>out = "47"; </tag>
806 </item>
807 </item>
808     achtundvierzigten <tag>out = "48"; </tag>
809 </item>
810 </item>
811     neunundvierzigten <tag>out = "49"; </tag>
812 </item>
813 </item>
814     fünfzigten <tag>out = "50"; </tag>
815 </item>
816 </one-of>
817 </rule>
818
819 <rule id="digit" scope="public">
820   <item repeat="0-1">nummer</item>
821 </one-of>
822 </item>
823   eins <tag>out = "1"; </tag>
824 </item>
825 </item>
826   zwei <tag>out = "2"; </tag>
827 </item>
828 </item>
829   drei <tag>out = "3"; </tag>
830 </item>
831 </item>
832   vier <tag>out = "4"; </tag>
833 </item>
834 </item>
835   fünf <tag>out = "5"; </tag>
836 </item>
```

```
837 <item>
838     sechs<tag>out = "6"; </tag>
839 </item>
840 <item>
841     sieben<tag>out = "7"; </tag>
842 </item>
843 <item>
844     acht<tag>out = "8"; </tag>
845 </item>
846 <item>
847     neun<tag>out = "9"; </tag>
848 </item>
849 <item>
850     zehn<tag>out = "10"; </tag>
851 </item>
852 <item>
853     elf<tag>out = "11"; </tag>
854 </item>
855 <item>
856     zwölf<tag>out = "12"; </tag>
857 </item>
858 <item>
859     dreizehn<tag>out = "13"; </tag>
860 </item>
861 <item>
862     vierzehn<tag>out = "14"; </tag>
863 </item>
864 <item>
865     fünfzehn<tag>out = "15"; </tag>
866 </item>
867 <item>
868     sechszehn<tag>out = "16"; </tag>
869 </item>
870 <item>
871     siebzehn<tag>out = "17"; </tag>
872 </item>
873 <item>
874     achtzehn<tag>out = "18"; </tag>
875 </item>
876 <item>
877     neunzehn<tag>out = "19"; </tag>
878 </item>
879 <item>
880     zwanzig<tag>out = "20"; </tag>
881 </item>
882 <item>
883     einundzwanzig<tag>out = "21"; </tag>
884 </item>
885 <item>
886     zweiundzwanzig<tag>out = "22"; </tag>
887 </item>
888 <item>
889     dreiundzwanzig<tag>out = "23"; </tag>
890 </item>
891 <item>
892     vierundzwanzig<tag>out = "24"; </tag>
893 </item>
894 <item>
895     fünfundzwanzig<tag>out = "25"; </tag>
896 </item>
897 <item>
898     sechszundzwanzig<tag>out = "26"; </tag>
899 </item>
900 <item>
901     siebenundzwanzig<tag>out = "27"; </tag>
902 </item>
903 <item>
904     achtundzwanzig<tag>out = "28"; </tag>
```

```
905 </item>
906 <item>
907   neunundzwanzig<tag>out = "29"; </tag>
908 </item>
909 <item>
910   dreißig<tag>out = "30"; </tag>
911 </item>
912 <item>
913   einsunddreißig<tag>out = "31"; </tag>
914 </item>
915 <item>
916   zweiunddreißig<tag>out = "32"; </tag>
917 </item>
918 <item>
919   dreiunddreißig<tag>out = "33"; </tag>
920 </item>
921 <item>
922   vierunddreißig<tag>out = "34"; </tag>
923 </item>
924 <item>
925   fünfunddreißig<tag>out = "35"; </tag>
926 </item>
927 <item>
928   sechsunddreißig<tag>out = "36"; </tag>
929 </item>
930 <item>
931   siebenunddreißig<tag>out = "37"; </tag>
932 </item>
933 <item>
934   achtunddreißig<tag>out = "38"; </tag>
935 </item>
936 <item>
937   neununddreißig<tag>out = "39"; </tag>
938 </item>
939 <item>
940   vierzig<tag>out = "40"; </tag>
941 </item>
942 <item>
943   einundvierzig<tag>out = "41"; </tag>
944 </item>
945 <item>
946   zweiundvierzig<tag>out = "42"; </tag>
947 </item>
948 <item>
949   dreiundvierzig<tag>out = "43"; </tag>
950 </item>
951 <item>
952   vierundvierzig<tag>out = "44"; </tag>
953 </item>
954 <item>
955   fünfundvierzig<tag>out = "45"; </tag>
956 </item>
957 <item>
958   sechsundvierzig<tag>out = "46"; </tag>
959 </item>
960 <item>
961   siebenundvierzig<tag>out = "47"; </tag>
962 </item>
963 <item>
964   achtundvierzig<tag>out = "48"; </tag>
965 </item>
966 <item>
967   neunundvierzig<tag>out = "49"; </tag>
968 </item>
969 <item>
970   fünfzigten<tag>out = "50"; </tag>
971 </item>
972 </one-of>
```

```
973 </rule >
974
975 <rule id="tag" scope="public">
976   <one-of>
977     <item>für </item>
978     <item>am</item>
979     <item>vom</item>
980   </one-of>
981   <item repeat="0-1">den </item>
982   <one-of>
983     <item>Montag </item>
984     <item>Dienstag </item>
985     <item>Mittwoch </item>
986     <item>Donnerstag </item>
987     <item>Freitag </item>
988     <item>Samstag </item>
989     <item>Sonntag </item>
990   </one-of>
991 </rule >
992 </grammar>
```

6.2 B Aufbau der Voice Commands

```
1  public class VoiceCommand
2  {
3      public action action;
4      public orientation orientation;
5      public application application;
6      public view view;
7      public double Confidence;
8      public int index;
9      public bool Name = false;
10
11     public VoiceCommand(action action , orientation orientation ,
12     application application , view view , int nummer ,
13     double confidence)
14     {
15         this.action = action;
16         this.orientation = orientation;
17         this.application = application;
18         this.view = view;
19         this.index = nummer;
20         this.Confidence = confidence;
21     }
22 }
23 public enum action
24 {
25     None    = 0,
26     Yes     = 1,
27     No      = 2,
28     Select  = 3,
29     Delete  = 4,
30     Scroll  = 5,
31     Next    = 6,
32     Last    = 7
33 }
34 public enum orientation
35 {
36     None    = 0,
37     Right   = 1,
38     Left    = 2,
39     Up      = 3,
40     Down    = 4
```

```
41 }
42 public enum application
43 {
44     None           = 0,
45     Email          = 1,
46     Calender       = 2,
47     News           = 3,
48     Weather        = 4,
49     Display        = 5,
50     HandControl    = 6,
51     VoiceControl   = 7,
52     System         = 8,
53     Help           = 9,
54     Event          = 10,
55     Week           = 11,
56     Day            = 12
57 }
58 public enum day
59 {
60     None           = 0,
61     Monday         = 1,
62     Tuesday        = 2,
63     Wednesday      = 3,
64     Thursday       = 4,
65     Friday         = 5,
66     Saturday       = 6,
67     Sunday         = 7
68 }
69 public enum view
70 {
71     None           = 0,
72     detail         = 1,
73     general        = 2,
74     list           = 3
75 }
```

6.3 C Interaction Stream

Es folgt eine Darstellung des Interaction Streams als C# Klasse wie die von Microsoft erstellten dazugehörigen Enums. Der Interaction Stream hält eine Liste von den erkannten Skeletten. Diese haben eine ID und eine Liste mit jeweils zwei HandPointer, wenn beide Hände erkannt werden.

```
1 public class InteractionStream
2 {
3     List<Skeleton> Skeletons;
4 }
5
6 public class Skeleton
7 {
8     int SkeletonTrackingId { get; set; }
9     List<HandPointer> HandPointers;
10 }
11
12 public class HandPointer
13 {
14     bool IsPrimaryForUser { get; set; }
15     bool IsTracked { get; set; }
16     bool IsActive { get; set; }
```

```
17     bool IsInteractive { get; set; }
18     bool IsPressed { get; set; }
19     double X { get; set; }
20     double Y { get; set; }
21     double PressExtent { get; set; }
22     double RawX { get; set; }
23     double RawY { get; set; }
24     double RawZ { get; set; }
25     int HandType { get; set; }
26     int HandEventType { get; set; }
27 }
28
29 namespace Microsoft.Kinect.Toolkit.Controls
30 {
31     /// <summary>
32     /// Used to denominate an event triggered by some hand of some user.
33     /// </summary>
34     public enum HandEventType
35     {
36         /// <summary>
37         /// No event has been triggered.
38         /// </summary>
39         None = 0,
40
41         /// <summary>
42         /// A hand has been gripped.
43         /// </summary>
44         Grip = 1,
45
46         /// <summary>
47         /// A hand's grip has been released.
48         /// </summary>
49         GripRelease = 2
50     }
51 }
52
53 namespace Microsoft.Kinect.Toolkit.Controls
54 {
55     /// <summary>
56     /// Used to denominate a specific hand of a user.
57     /// </summary>
58     public enum HandType
59     {
60         /// <summary>
61         /// Refers to neither right nor left hand.
62         /// </summary>
63         None = 0,
64
65         /// <summary>
66         /// Refers to left hand.
67         /// </summary>
68         Left = 1,
69
70         /// <summary>
71         /// Refers to right hand.
72         /// </summary>
73         Right = 2
74     }
75 }
```

Literaturverzeichnis

- [Baudisch u. a. 2010] BAUDISCH, Patrick ; GUSTAFSON, Sean ; BIERWIRTH, Daniel: *Imaginary Interfaces: Spatial Interaction with Empty Hands and without Visual Feedback*. 2010. – Erreichbar online unter <http://www.seangustafson.com/static/papers/2010-UIST-Gustafson-ImaginaryInterfaces.pdf>; besucht am 08.02.2014.
- [Belke 2000] BELKE, Michael: *Kommunikation: Das Miteinander von Mensch und Maschine*. Website. 2000. – Erreichbar online unter http://www.techfak.uni-bielefeld.de/ags/wbski/lehre/digiSA/Kommunikation/Ausarbeitungen/michael_belke.pdf; besucht am 03.12.2013.
- [Berndt u. a. 2009] BERNDT, E. ; WICHERT, R. ; SCHULZE, E. ; OESTERREICH, D. ; BÖHM, U. ; GOTHE, H. ; FREYTAG, A. ; DAROSZEWSKA, A. ; STORZ, P. ; MEYER, S. ; DIERKS., C.: *Schlussbericht, Marktpotenziale, Entwicklungschancen, gesellschaftliche, gesundheitliche und ökonomische Effekte der zukünftigen Nutzung von Ambient Assisted Living (AAL)-Technologien*. Website. 2009. – Erreichbar online unter http://www.aal.fraunhofer.de/publications/urn_nbn_de_0011-n-1024464.pdf; besucht am 02.11.2013.
- [Bernin 2011] BERNIN, Arne: *Einsatz von 3D-Kameras zur Interpretation von räumlichen Gesten im Smart Home Kontext*, Masterarbeit, 2011. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/bernin.pdf>; besucht am 10.09.2013.
- [Bernin 2012] BERNIN, Arne: *A Framework Concept for Emotion Enriched Interfaces*. 2012. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/ICEC2012-3.pdf>; besucht am 6.10.2013.
- [Boetzer u. a. 2008] BOETZER, J. ; RAHIMI, M. Mohammad A. ; VOGT, Matthias ; WENDT, P. ; K.V.LUCK: *Gestenbasierte Interaktion mit Hilfe von Multitouch und Motiontracking*. 2008. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/WIWITA2008.pdf>; besucht am 10.09.2013.

- [Boetzer 2008] BOETZER, Joachim: *Bewegungs- und gestenbasierte Applikationssteuerung auf Basis eines Motion Trackers*. Website. 2008. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/boetzer.pdf>; besucht am 23.10.2013.
- [Bundesministerium für Bildung und Forschung 2012] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG: *Bekanntmachung von Richtlinien zur Förderung von Forschung und Entwicklung auf dem Gebiet "Technik stellt sich auf den Menschen ein - innovative Schnittstellen zwischen Mensch und Technik"*. 2012. – Erreichbar online unter <http://www.bmbf.de/foerderungen/20972.php>; besucht am 03.02.2014.
- [Cassell 1998] CASSELL, Justin: *A Framework For Gesture Generation And Interpretation*. In: *Computer Vision in Human-Maschine Interaction*. Cambridge University Press, 1998
- [Cybertecture (Mirror) Ltd 2010] CYBERTECTURE (MIRROR) LTD, James L.: *Cybertecture Mirror*. Website. 2010. – Erreichbar online unter <http://www.cybertecturemirror.com/main.php>; besucht am 7.10.2013.
- [van Eimeren und Frees 2013] EIMEREN, Birgit van ; FREES, Beate: *Rasanter Anstieg des Internetkonsums - Onliner fast drei Stunden täglich im Netz*. (2013). – Erreichbar online unter http://www.ard-zdf-onlinestudie.de/fileadmin/Onlinestudie/PDF/Eimeren_Frees.pdf; besucht am 22.11.2013.
- [Ellenberg u. a. 2011] ELLENBERG, Jens ; KARSTAEDT, Bastian ; VOSKUH, Sören ; LUCK, Kai von ; WENDHOLT, Birgit: *An Environment for Context-Aware Applications in Smart Homes*. 2011. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/IPIN2011.pdf>; besucht am 7.09.2013.
- [Fowler 2004] FOWLER, Martin: *Presentation Model*. 2004. – Erreichbar online unter <http://martinfowler.com/eaDev/PresentationModel.html>; besucht am 18.12.2013.
- [Gruler und Stoll 2001] GRULER, Alexander ; STOLL, Dietmar: *Identifikation durch Handgeometrie*. 2001. – Erreichbar online unter <http://www.informatik.uni-ulm.de/ni/Lehre/WS01/HS-Biometrische-Systeme/ausarbeitungen/Handgeometrie.PDF>; besucht am 13.12.2013.
- [Heise Online 2013] HEISE ONLINE: *HP integriert 3D-Gestensteuerung ins Notebook*. (2013). – Erreichbar online unter <http://www.heise.de/newsticker/meldung/>

- [HP-integriert-3D-Gestensteuerung-ins-Notebook-1962126.html](#); besucht am 20.09.2013.
- [Heitsch 2008] HEITSCH, Johann: *Ein Framework zur Erkennung von dreidimensionalen Gesten*. 2008. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/heitsch.pdf>; besucht am 24.10.2013.
- [Hossain u. a. 2007] HOSSAIN, M.A. ; ATREY, P.K. ; SADDIK, A.E.: Smart mirror for ambient home environment. In: *Intelligent Environments, 2007. IE 07. 3rd IET International Conference on*, 2007, S. 589–596. – ISSN 0537-9989
- [HP Computers YouTube Channel 2013] HP COMPUTERS YOUTUBE CHANNEL: *HP ENVY17 Leap Motion Special Edition*. 2013. – Erreichbar online unter <https://www.youtube.com/watch?v=aWlX2yxcmHo>; besucht am 20.09.2013.
- [Innovation + Technik GmbH 2013] INNOVATION + TECHNIK GMBH, (VDI/VDE-IT): *Innovationen für ein selbstbestimmtes Leben*. 2013. – Erreichbar online unter <http://www.aal-deutschland.de>; besucht am 10.11.2013.
- [Kasal 2010] KASAL, Alpay: *Interactive Mirror*. Website. 2010. – Erreichbar online unter <http://blog.litstudios.com/>; besucht am 07.10.2013.
- [Kaufmann 2011] KAUFMANN, Benjamin: *Das Model-View-Controller Modell*. 2011. – Erreichbar online unter <http://fara.cs.uni-potsdam.de/~kaufmann/tuts/mvc.pdf>; besucht am 15.12.2013.
- [McNeill 1992] MCNEILL, David: *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, 1992
- [Microsoft Corporation 2014] MICROSOFT CORPORATION: *Microsoft Skeleton Stream Joints*. 2014. – Erreichbar online unter <http://msdn.microsoft.com/en-us/library/hh438998.aspx>; besucht am 04.01.2014.
- [Müller 2013] MÜLLER, Larissa: *Emotionale Modellierung in Mensch-Maschine-Interaktionen*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2013. – Erreichbar online unter users.informatik.haw-hamburg.de/~ubicomp/papers.html; besucht am 02.11.2013.
- [MRS Systems 2012] MRS SYSTEMS, (UK) L.: *Magic Mirror *Virtual Shopping Rail - Digital Signage - Social Driven Shopping*. Website. 2012. – Erreichbar online unter <http://www.mrsmobile.co.uk/magic-mirror/>; besucht am 11.10.2013.

- [Newton-King 2013] NEWTON-KING, James: *json.NET*. 2013. – Erreichbar online unter <http://james.newtonking.com/json>; besucht am 10.10.2013.
- [Opera Software ASA 2012] OPERA SOFTWARE ASA: *Mausgesten*. Website. 2012. – Erreichbar online unter <http://help.opera.com/Windows/12.10/de/gestures.html>; besucht am 10.12.2013.
- [Otto 2013] OTTO, Kjell: *Aktuelle Entwicklungskonzepte zur Projektintegration in einem Smart Home anhand von Maven, OSGi und Drools Fusion*. 2013. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/otto.pdf>; besucht am 07.10.2013.
- [Pavlovic u. a. 1997] PAVLOVIC, V.I. ; SHARMA, R. ; HUANG, T.S.: Visual interpretation of hand gestures for human-computer interaction: a review. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19 (1997), Nr. 7, S. 677–695. – ISSN 0162-8828
- [Pfister und Kaufmann 2008] PFISTER, Beat ; KAUFMANN, Tobias: *Sprachverarbeitung - Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer, 2008
- [Pluta 2013] PLUTA, Werner: *Herzrhythmus aktiviert Smartphone*. 2013. – Erreichbar online unter <http://www.golem.de/news/nymi-herzrhythmus-aktiviert-smartphone-1309-101344.html>; besucht am 15.11.2013.
- [Poguntke 2007] POGUNTKE, W.: *Basiswissen IT-Sicherheit: das Wichtigste für den Schutz von Systemen und Daten*. W3L-Verlag, 2007 (IT lernen). – Erreichbar online unter http://books.google.de/books?id=DOqlH_1PwFEC; besucht am 07.12.2013.. – ISBN 9783937137650
- [Potratz 2011] POTRATZ, Olaf: *Ein System zur physikbasierten Interpretation von Gesten im 3D-Raum*. Website. 2011. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/boetzer.pdf>; besucht am 20.8.2013.
- [Rahimi und Vogt 2011] RAHIMI, Mohammad A. ; VOGT, Matthias: *Seamless Interaction - Natürliche Interaktionen in Smart Living Umgebungen*, Master-Thesis, 2011. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/rahimi-vogt.pdf>; besucht am 28.09.2013.

- [Rahman u. a. 2010] RAHMAN, A.S.M.M. ; TRAN, T.T. ; HOSSAIN, S.K.A. ; EL SADDIK, A.: Augmented Rendering of Makeup Features in a Smart Interactive Mirror System for Decision Support in Cosmetic Products Selection. In: *Distributed Simulation and Real Time Applications (DS-RT), 2010 IEEE/ACM 14th International Symposium on*, 2010, S. 203–206. – ISSN 1550-6525
- [c/o Telekom AG 2013] TELEKOM AG c/o: LIFE - Digitales Leben. (2013). – Erreichbar online unter http://www.studie-life.de/wp-content/uploads/2011/11/studie-LIFE_digitales-leben.pdf; besucht am 22.11.2013.
- [The Apache Software Foundation] THE APACHE SOFTWARE FOUNDATION: *ActiveMQ Homepage*. – Erreichbar online unter <http://activemq.apache.org/>; besucht am 12.12.2013.
- [Witt 2011] WITT, Kristoffer: *Kontextabhängige multimodale Interaktion mit Schwerpunkt Spracherkennung im Smart-Home Umfeld*, Diplomarbeit, 2011. – Erreichbar online unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/boetzer.pdf>; besucht am 10.7.2013.
- [WIWITA 2008] WIWITA: *6. Wismarer Wirtschaftsinformatik-Tage*. Website. 2008. – Erreichbar online unter <http://www.wi.hs-wismar.de/~cleve/wiwita.html>; besucht am 05.12.2013.
- [Yu u. a. 2012] YU, Yuan-Chih ; YOU, S.D. ; TSAI, Dwen-Ren: Magic mirror table for social-emotion alleviation in the smart home. In: *Consumer Electronics, IEEE Transactions on* 58 (2012), Nr. 1, S. 126–131. – ISSN 0098-3063

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 18. Februar 2014

Sobin Ghose