Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Kristoffer Witt

Kontextabhängige multimodale Interaktion mit Schwerpunkt Spracherkennung im Smart-Home Umfeld

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Kristoffer A. Witt

Thema der Masterarbeit

Kontextabhängige multimodale Interaktion mit Schwerpunkt Spracherkennung im Smart-Home Umfeld

Stichworte

Smart-Home Umgebung,Ambient Intelligence, Spracherkennung, Multimodale Interaktion, Kontext, Architektur, Ubiquitous Computing

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit der Entwicklung einer Architektur für die Nutzung multimodaler Interaktion in Smart-Home Umgebungen. Das Hauptaugenmerk liegt hierbei auf der Nutzbarmachung der Eingabemodalität Sprache. Als Beweis für die Umsetzbarkeit der Architektur wird ein Prototypisches System implementiert und evaluiert (Proof-Of-Concept).

Kristoffer A. Witt

Title of the Masterthesis

Contextsensitive multimodal interaction in a Smart-Home surrounding, with emphasis to speech recognition

Keywords

Smart-Home, ambient intelligence, speech recognition, multimodal interaction, context awareness, architecture, Ubiquitous Computing

Abstract

In this thesis the development of an architecture for multimodal interaction in smart home surroundings is presented. The focus of this work lies on how speechrecognition can be used as an input modality. As a proof of concept a prototypical application is implemented and evaluated.

Inhaltsverzeichnis

1. Einleitung

Immer mehr kleine nicht mehr als solche zu erkennende Computer halten Einzug in unseren täglichen Alltag: Die Ära des Ubiquitous Computing (Weiser, 1999).

Verknüpft man alle diese Computer, Sensoren und Eingabegeräte in einem Netzwerk und fügt eine steuernde „logisch“ denkende Instanz hinzu so erhält man eine Ambient Intelligence. Eine Intelligente Umgebung (Aghajan u. a., 2009, S. 226).

Angewendet auf den Ort wo wir einen Hauptteil unseres Lebens verbringen, in unseren Häusern und Wohnungen, ergibt sich die Idee des Smart-Homes. Einer Heimstätte die sich passiv wie auch aktiv unseren Bedürfnissen anpasst und uns bei unserem täglichen Leben assistiert.
Wie interagiert man nun mit einem solchen System? So wenig wie nötig aber so natürlich wie möglich! Was ist die natürlichste Form der Kommunikation? Lautsprache (Aghajan u. a., 2009, S. 226).

Seit mehr als 50 Jahren (Pfister und Kaufmann, 2008b, S.283) forschen Hochschulen und Institute an der perfekten Umwandlung von Lautsprache in Text. Trotzdem sind wir heute noch weit weg von der universell einsetzbaren Spracherkennung (vgl. O’Shaughnessy (2008), Baker u. a. (2009), Rabiner (2003)). Insbesondere in Umgebungen mit unkontrollierbaren Lärmbedingungen nimmt die Leistung aktueller Spracherkennungs Software rapide ab. In diese Kategorie fallen leider auch die meisten Smart-Home Umgebungen.

Wenn man sich mit der sprachlichen Interaktion von Mensch und Mensch bzw. in diesem Fall Mensch und Maschine beschäftigt, kommt man um einen Begriff nicht herum „Kontext“. Ohne Kontext lassen bleiben viele Äußerungen vieldeutig. Menschen lassen bei der Interpretation von diesen mehrdeutigen Sätzen ihre Beobachtungen einfließen (z.B. wie verhält sich der Sprecher, wen schaut er an, worauf zeigt er gerade, wovon hat mein Gegenüber gerade gesprochen) und greifen auf ihre Erfahrungen zurück. Meist reicht dies aus, um die Semantik eindeutig zu erfassen. Für künstliche Intelligenzen ist das nicht ganz so einfach. Um die Wahrnehmung eines Menschen mehr schlecht als recht zu kopieren wären eine Vielzahl verschiedener Sensoren notwendig. Und die die unzähligen Werte, die diese liefern müssten auch erstmal so kombiniert werden, dass sie einen Sinn ergeben.

Sprachliche Interaktion für sich ist sehr mächtig. Trotzdem nutzen wir Menschen untereinander noch weit mehr um uns unserem Gegenüber mitzuteilen. Wir gestikulieren mit den Händen, neigen unseren Kopf in eine bestimmte Richtung oder zeigen mit unser Körperhaltung unserer aktuelles Befinden. Überträgt man dieses Verhalten in die Kommunikation mit einem Computer erhält man multimodale Interaktion. Klassischer Weise wird ein Computer über Maus und Tastatur bedient. In der ubiquitären Smart-Home Umgebung ist diese Art der Bedienung allerdings nur selten Sinnvoll. Die Restriktionen die sie einem auferlegt, sei es die

Immobilität aufgrund kurzer Kabel/Reichweiten oder die Festlegung auf graphisches Feedback, stehen im Kontrast zu der Freiheit die wir in unseren eigenen vier Wänden eigentlich genießen wollen. Ein Zeugnis dieses Wunsches nach Unabhängigkeit sind die unzähligen für nahezu alle Geräte vorhandenen Fernbedienungen. Multimodale Interaktion bezeichnet das Interagieren über die volle Bandbreite unserer Sinne. Sie bietet den perfekten Modus für die effiziente und natürliche Kommunikation mit einer Smart-Home Umgebung.

1.1. Zielsetzung

Das Ziel dieser Arbeit ist die Entwicklung einer System-Architektur für Smart-Home Umgebungen. Diese Architektur soll die einfache Einbindung von Sprachsteuerung in allen Aspekten der Interaktion mit dem Smart-Home ermöglichen. Multimodalität soll durch Kombination der Spracherkennung mit Anderen, von der Art der Anwendung abhängigen Eingabemethoden erreicht werden. Zum Beispiel die Verarbeitung von Zeigegesten in Kombination mit Spracheingaben.
Um die Integration neuer Anwendungen in der Smart-Home Domäne möglichst simpel zu gestalten soll ein möglichst transparentes und einfach erweiterbares System geschaffen werden. Das entstehende System soll so konzipiert werden, dass es die in Smart-Home Umgebung häufig auftretenden Kontextänderungen, wie das Wechseln des Raumes oder das Eingehen eines Telefonats, verarbeiten kann.

Zu Demonstrationszwecken und um die Umsetzbarkeit der Architektur zu zeigen, soll eine Prototypische Anwendung entwickelt werden. Die Funktionalität des Prototyps orientiert sich dabei an einem Szenario (siehe Kapitel 3.1). Das Szenario stellt exemplarisch die Aspekte und Problematiken der Interaktion in einer Smart-Home Umgebung vor und dient als Schablone für die Evaluierung der Architektur.

1.2. Gliederung

Die Arbeit gliedert sich in sechs Hauptbestandteile. Den ersten Teil bildet die Einleitung. Diese besteht aus der Einführung in die Thematiken die dieser Arbeit zu Grunde liegen, der Motivation die zur Erstellung dieser Arbeit geführt hat und der Zielsetzung die verfolgt wird.

Der zweite Teil erläutert die Grundlagen die für das Verständnis der Arbeit von Bedeutung sind. Es werden die grundlegenden Technologien, Begriffe und Konzepte erörtert die im weiteren Verlauf Verwendung finden.

Die analytische Aufarbeitung hinsichtlich der Umsetzung des angestrebten Ziels befindet sich im dritten Teil. Den Einstieg bildet ein Szenario das typische Aspekte der Interaktion

in einer Smart-Home Umgebung beinhaltet. Aus diesem Anwendungsfall, der Untersuchung vergleichbarer Arbeiten und der Zielsetzung werden erst die funktionalen und nicht funktionalen Anforderungen abgeleitet und abschließend bezüglich möglicher Problematiken untersucht.

Im vierten Teil erfolgt die Beschreibung der Umsetzung der Ergebnisse aus Teil drei. Die aus den Anforderungen abgeleiteten Designentscheidungen werden vorgestellt und anhand ihrer Funktionalität in System-Komponenten eingeteilt. Anhand von UML-Diagrammen wird das interne Zusammenspiel und die auftretenden Abläufe des Systems erläutert.

Im Mittelpunkt des sechsten Abschnitts stehen, neben den Implementierungs- bzw. Realisierungsdetails, die Resultate und die Vorgehensweise der Evaluierung des Projektes. Einleitend wird die Umgebung in der die Architektur eingesetzt werden soll beschrieben. Es werden verschiedene Einschränkungen festgelegt die den Umfang der Implementierung reduzieren. Anschließend werden der technische Aufbau der Umgebung und die Implementierungswerkzeuge beschrieben. Den Abschluss bilden Erläuterungen zur Umsetzung des Szenarios, verschiedener Hilfsprogramme sowie das Ergebnis der Evaluation.

Der letzte Teil der Arbeit setzt sich zusammen aus einer Reflektion über die Erfüllung der Zielsetzung sowie dem Ausblick auf noch offene Punkte und mögliche Ansätze für Erweiterungen.

2. Grundlagen

Das Kapitel Grundlagen beschreibt alle für das Verständnis der Arbeit wichtigen Grundlagen und Lehrmeinungen. Es dient der Verdeutlichung der grundlegenden Technologien und Konzepte die in dieser Arbeit eingesetzt werden.

2.1. Ubiquitous Computing

Ubiquitous Computing (Ubicomp) ist ein von Marc Weiser erdachter Begriff, der die Allgegenwärtigkeit (engl. ubiquity) von Computern beschreibt. Roy Want, ein Mitarbeiter Weisers Want (2011), nennt Ubicomp das Dritte Zeitalter des Computings (Krumm, 2009, S. 2).

Bis heute sind bereits zwei prägende Zeitabschnitte durchlaufen worden: die zentralistisch geprägte Struktur des Mainframe Computings und die Ära der Personal Computer (siehe Abbildung 1). Begünstigt durch die fortschreitende Miniaturisierung und die einhergehende kostengünstigere Produktion von Komponenten befinden wir uns heute im Zeitalter des Ubiquitous Computing. Computer sind überall, in Lichtschaltern, Autos, Kreditkarten und einer Vielzahl anderer Geräte. Sie sind in vielen Fällen „unsichtbar“.

In seinem Artikel „The Computer for the 21st Century“ schreibt Weiser über die Nutzung von Computern:

> Whenever people learn something sufficiently well, they cease to be aware of it. [...] only when things disappear in this way are we freed to use them without thinking and so to focus beyond them on new goals.(Weiser, 1999, S.1)

Im übertragenen Sinne also, wenn der Computer als einzeln verkörpertes Gerät verschwunden ist, man also nicht mehr bewusst mit ihm interagiert, dann erst konzentriert man sich auf neue Ziele. Wir können uns also erst jetzt, wo Computer kaum noch als solche zu erkennen sind, auf neue Ziele konzentrieren.

2.1.1. Begrifflichkeiten

Neben Ubicomp haben sich in den letzten Jahren verschiedene weitere Begriffe etabliert, die ähnliche oder gleiche Zusammenhänge beschreiben. Mike Kuniavsky fasst diese wie folgt zusammen:

Ubiquitous computing Ubicomp beschreibt die Nutzung von Informationsverarbeitung und Netzwerkkommunikation in Alltägliche Umgebungen, um Dienste, Informationen und Kommunikation ständig verfügbar zu halten.

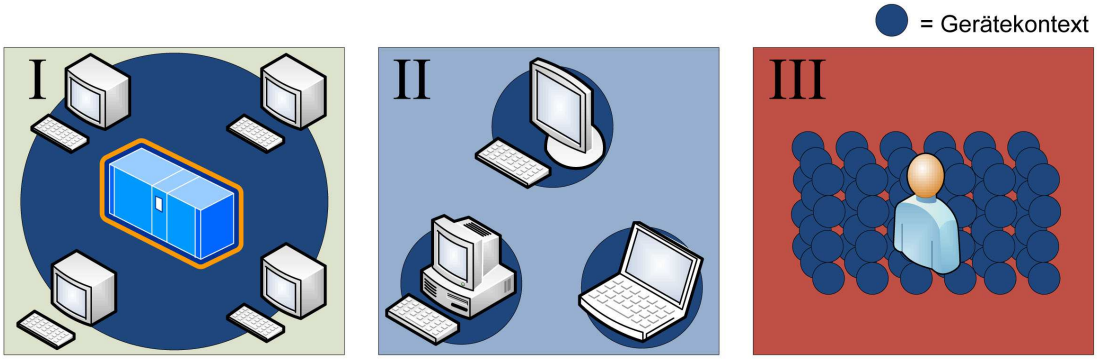

Abbildung 1: Computing Zeitalter

Physical computing Physical computing bezeichnet die Interaktion mit Computern durch physikalische Objekte (Tangible Computing) anstatt der Nutzung von den Typischen Eingabegeräten wie Maus und Tastatur.

Pervasive computing Pervasive computing bezieht sich auf die Prävalenz dieser neuen Art digitaler Technologie.

Ambient Intelligence Bezeichnet die nicht explizit wahrnehmbare Integration algorithmischer Schlussfolgerungen (Logig bzw. Künstliche Intelligenz) in die Funktionalität von Menschen erschaffenener Räume.

Internet of Things Begrifflichkeit die eine Analogie schafft zwischen der Art wie digitale Informationen im Internet organisiert sind und der Verbindung von digital identifizierbaren physikalischen Objekten.

Frei übersetzt aus (Kuniavsky, 2010, Seite 5ff.).

Ubicomp und Pervasive Computing Roy Wants geht in seiner Einleitung zum Buch „Ubiquitous Computing Fundamentals“ (Krumm (2009)) näher auf die Entwicklung der Begrifflichkeiten Ubicomp und Pervasive Computing ein. Er beschreibt sie wie folgt:

> In fact, many texts today describe pervasive and ubiquitous as the same thing.
> ...
> any unique position described by either party (Ubicomp (Xerox Parc) bzw. Pervasive Computing (IBM)) has been slowly integrated into the shared vision and by the mid-2000s any publications that set out to describe this topic presented fundamentally the same position.

(Krumm, 2009, S. 11)

Es wird deutlich, dass die beiden Begrifflichkeiten durch ihre Wortwahl zwar verschiedene Aspekte in den Vordergrund stellen aber die semantischen Schnittmengen so umfangreich sind, dass sich eine synonyme Nutzung anbietet.

Ubicomp/Pervasive Computing und Ambient Intelligence Laut Juan Carlos Augusto sollte Ubiquitous Computing nicht mit Ambient Intelligence gleichgesetzt werden. Er begründet dies wie folgt:

> Some authors equate „Ubiquitous Computing“ and „Pervasive Computing“ with „Ambient Intelligence“. Here we argue that Ubiquitous/Pervasive systems are different as they emphasize the physical presence and availability of resources and miss a key element: the explicit requirement of „Intelligence“

(Augusto, 2004, S.4)

Das bedeutet also die Terme Ubicomp und Pervasive Computing beziehen sich nur auf die Verfügbarkeit von Ressourcen und lassen den Kernaspekt Intelligenz außen vor. Erst durch Erweiterung um deduktive Verarbeitung von Sensorwerten entsteht also aus Ubicomp- und Pervasive Computing-System eine Ambient Intelligence.

Im Zusammenhang mit dieser Arbeit lässt sich also feststellen, dass wegen der angestrebten Nutzung von Kontext- und Sensorinformationen die Anwendung in den Bereich Ambient Intelligence System fällt.

2.2. Verteilte Systeme

Die Grundlage für Ubiquitous Computing und alle seine Spielarten bildet das Forschungsgebiet der Verteilten Systeme. Tanenbaum et. al definieren ein verteiltes System (distributed system) in ihrem Standardwerk „Distributed Systems - Principles and Paradigms“ wie folgt:

> A distributed system is a collection of independent computers that appears to its users as a single coherent system.(Tanenbaum und Steen, 2006, Seite 2)

Ein verteiltes System ist also definiert als Sammlung von unabhängigen (aber verbundenen) Computern, die sich nach Außen hin (dem Anwender gegenüber), als ein zusammenhängendes System präsentieren.

Damit der Anwender bzw. der Entwickler das Netzwerk verschiedener Komponenten als ein Gesamtsystem wahrnehmen kann, wird die sogenannte Transparenz benötigt. Der Begriff

Transparenz kommt vom Lateinischen „transparens“ und bedeutet soviel wie durchscheinend. Er bezieht sich darauf, dass zwar bestimmte das System verbindende Teile vorhanden sind, aber eben für den Anwender oder Entwickler nicht sichtbar (durchsichtig, transparent). Tanenbaum et. al unterscheiden dabei verschiedene Arten der Transparenz (siehe Tabelle 1).

Transparenz	Beschreibung
Zugriff (Access)	Versteckt Unterschiede in der Repräsentation der Daten und wie auf sie zugegriffen wird.
Ort (Location)	Versteckt den Ort einer Ressource.
Migration	Versteckt den möglichen Transfer einer Ressource an einen anderen Ort.
Relocation	Versteckt den möglichen Transfer einer Ressource während sie genutzt wird.
Replication	Versteckt die Tatsache das eine Ressource repliziert wurde.
Gleichzeitigkeit (Concurrency)	Versteckt das Faktum das eine Ressource von unterschiedlichen Anwendern gleichzeitig genutzt wird.
Fehlerfall (Failure)	Versteckt den Fehlerfall einer Ressource

Tabelle 1: Transparenztypen in verteilten Systemen, frei übersetzt aus (Tanenbaum und Steen, 2006, Seite 5)

Für de Umsetzung dieser Transparenzen in einer Softwarearchitektur existieren verschiedene Ansätze. Im Folgenden werden Beispielhaft zwei dieser Ansätze, die für die Realisierung dieser Arbeit von Bedeutung sind, vorgestellt.

Für eine umfassende Einführung in das Gebiet der verteilten Systeme sei auf das Standardwerk von Tanenbaum und Steen (2006) verwiesen.

2.2.1. Service Oriented Architecture

Als Service-orientierte Architekturen (SOA) bezeichnet man eine konzeptuelle, abstrakte Software-Architektur, die es ermöglichen soll, verschiedenartige Dienste, über ein Netzwerk, Anzubieten, zu Suchen und zu Nutzen.

Diese Auftrennung der Funktionalität einer normalerweise monolithischen Anwendung ermöglicht es, hohe Raten von Wiederverwendung einzelner Dienste zu erreichen. Das bedeutet das ein Service/Dienst der eine bestimmte Aufgabe erfüllt mit wenig Aufwand auch in anderen Anwendungen seinen Einsatz finden kann. Für weitere Informationen siehe Melzer (2008).

Webservices Als Webservice wird ein Ausprägung der Service Oriented Architecture bezeichnet deren Funktionalität über das HTTP Protokoll zur Verfügung gestellt wird. Das World Wide Web Konsortium (W3) definiert Webservices wie folgt:

> A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. WebService (2004)

Also als Softwaresystem das Interaktion von Maschine zu Maschine über ein Netzwerk unterstützt. Das System besitzt eine Schnittstelle die in einem Maschinenlesbaren Format (Web Service Description Language, WSDL) deklariert wird. Kommuniziert wird mit dem System über ein XML-Dateiformat das über das HTTP-Protokoll übertragen wird.

WebServices stellen also Geschäftsprozesslogik bereit die von unterschiedlichen Anwendern über das Internet genutzt werden kann. Für eine umfangreichere Betrachtung sei auf Melzer (2008) verwiesen.

2.2.2. Blackboard Architektur

Die Blackboard Architektur beschreibt eine Problemlösungsstrategie die auf dem Zusammenwirken unterschiedlicher Experten zur Lösung von Teilproblemen aufbaut. Die einzelnen Experten bearbeiten Ihr jeweiliges Fachgebiet (Teilproblem) und liefern ihre Ergebnisse an ein zentrales „Blackboard“ (Tafel) zurück. Durch Auswertung dieser Ergebnisse können dann weitere Schlüsse gezogen werden.

Das Blackboard stellt also ein zentrales Kommunikationsmedium dar. Es entkoppelt die Experten und macht sie funktional unabhängig und dadurch leicht austauschbar.

Besonders in der inhomogenen Welt der Smart-Homes in der eine Vielzahl unterschiedlicher Komponenten eingesetzt wird, bietet sich die Blackboard-Architektur für verschiedene Aspekte an. Die zentralisierte und normalisierte Kommunikation ermöglicht eine Überwachung des Gesamtsystems. Es können, ohne großen Aufwand, weitere auf den Ergebnissen der anderen Komponenten arbeitende Module eingefügt werden. Die Dynamik des Systems und die Kommunikation der Komponenten untereinander wird dadurch nicht belastet. Wichtig hierbei ist die Zugriffssteuerung dar bei aufeinander aufbauenden Prozessen schnell Inkonsistenzen entstehen können. Eine Instanz der Blackboard Architektur die auch im Bereich von Smart-Home Umgebungen eingesetzt wird ist zum Beispiel der iRos Event Heap (siehe Johanson (2003))

Für nähere Informationen über das Blackboard Architektur Muster sei auf Hayes-Roth (1985) oder Corkill (1991) verwiesen.

2.3. Smart-Home

Als Smart-Home wird ein „intelligentes“ Habitat bezeichnet, dessen Intelligenz sich von der Fähigkeit ableitet mit den Bewohnern in distinguierter Weise zu interagieren und sie zu unterstützen.

Die Hauptkomponenten die die Funktion des Smart-Homes bestimmen sind Sensorik, Analyse und Aktorik. Ein Smart-Home ist über die aktuellen Vorgänge im Gebäude informiert (Sensorik), leitet daraus Informationen und Wissen ab (Analyse) und (re)agiert anhand dieser Daten um den Anwender zu unterstützen (Aktorik).

Bezeichnung	Beispiel
Museum	Erkennung der Besucherposition vor Ausstellungsstück und der Nationalität anhand der Sprache. Abspielen von gerichteten (über Richt-Lautsprecher), lokalisierten Audioinformationen (über das betrachtete Ausstellungsstück).
Flughafen	Unterstützung des Fluggastes bei der Navigation auf großen Flughäfen anhand von Anzeigetafeln und dem Wissen über den Flugverlauf des Reisenden. „Folgen Sie den grünen Pfeilen zu Ihrem Abflugschalter“.

Tabelle 2: Beispiele für Smart-Spaces

Ein Smart-Home gehört konzeptionell zu den Smart-Spaces die eine Untergruppe des Bereichs Ambient Intelligence (AmI) sind (Vgl. (Augusto, 2004, S.4)). Smart-Spaces ist die generelle Bezeichnung einer „intelligenten“ (s.o.) Umgebung. Während der Begriff „Home“ eine Haus-/Wohnungssituation bezeichnet, umfasst die Smart-Spaces Einteilung alle Räumlichkeiten mit intelligenter Benutzerunterstützung.

2.3.1. Einordnung

Hinsichtlich der Einordnung einer Smart-Homes Umgebung als verteiltes System lässt sich feststellen, dass die oben gegebene Definition (2.2) eindeutig zutrifft. Eine Smart-Home Umgebung besteht aus einer Vielzahl unterschiedlicher, autonomer Geräte. Der Anwender interagiert mit diesen Geräte so als wären Sie Teil eines Gesamtsystems. Zum Beispiel in dem er oder sie davon ausgeht, an jedem Terminal Zugriff auf alle Daten zu haben. Oder über unterschiedliche Peripheriegeräte das Gesamtsystem steuern zu können. Eine eingeschränkte

ortsabhängige Verfügbarkeit von Ressourcen (zum Beispiel Zugriff auf Hauselektronik) wäre dem Anwender nur schwer zu vermitteln und würde die Akzeptanz des Gesamtsystems mindern.

2.4. Kontext

Insbesondere in Umgebungen in denen viele verschiedene Interaktionsmodi zur Verfügung stehen ist Kontext von erheblicher Bedeutung. Die Zuordnung welche Eingaben für welche Anwendungen bestimmt sind oder das berücksichtigen der Befindlichkeit des Bedieners bei Ausgaben (zum Beispiel die Entscheidung ob Akustisch oder Visuell) sind Abhängig vom Kontextinformationen.

2.4.1. Definition

Dey et. al definieren Kontext wie folgt:

> Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.(Dey, 2001, S. 3)

Kontextinformationen sind also diejenigen Informationen, die die Situation einer Entität charakterisieren. Entität bedeutet in diesem Zusammenhang eine Person, ein Ort oder Objekt das an der Interaktion eines Nutzers mit einer Anwendung beteiligt ist.

Im Kontext dieser Arbeit ergibt sich eine Beispielhafte Zuordnung wie folgt:

Entität Ein Bewohner der Smart-Home Umgebung.

Anwendung Steuerung der Smart-Home Umgebung über eine Benutzeroberfläche (z.B. regulieren der Beleuchtung).

Kontext Position des Bewohners und die darin verfügbaren Leuchtmittel.

Terry Winograd verfeinert diese relativ allgemeine Definition in dem er den Begriff des „Settings“ einführt:

> The user of a computer system is always situated in some *setting of people, places, and things* (including computers), regardless of which aspects of that setting are used as *context* in communication.(Winograd, 2001, S. 5)

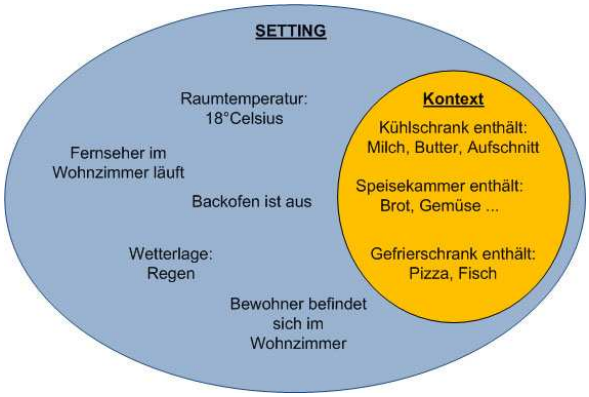

Abbildung 2: Setting/Kontext Beispiel: Szenario Planung des Abendessens

Ein Kontext ist also eine Teilmenge eines Settings. Er bezeichnet nur die Ereignisse und Faktoren die für die Interagierenden Komponenten von Bedeutung sind. In Abbildung 2 wird Beispielhaft dargestellt welche Faktoren zum Setting und welche zum Kontext gehören.

2.4.2. Kontextbewusste Anwendungen

Anhand von kontextbezogenen Informationen lassen sich für den Anwender Mehrwerte generieren. Anwendungen die auf diese Informationen zurückgreifen werden als kontextbewusste (context-aware) Anwendungen bezeichnet. Anhand ihrer Funktionsweise lassen sie sich unterschiedlichen Kategorien zuordnen:

Präsentation Auswertung der Kontextinformationen zur Anpassung der Darstellungsweise von Informationen.

Automatische Ausführung Aus Sensorinformationen abgeleiteter Kontext führt zur impliziten Ausführung einer Aktion.

Kontext-Kennzeichnung Verarbeitung von Kontextinformationen zur Erstellung einer Historie für spätere Referenzierung.

Zur Verdeutlichung enthält Tabelle 3 verschiedene Beispiele.

2.4.3. Deixis

Deixis leitet sich aus dem Griechischen ab und bedeutet „zeigen“. Der Begriff stammt aus der Linguistik. Er bezeichnet sprachliche Ausdrücke, deren Semantik sich nur aus dem Kontext erschließt. Die Deixis unterteilt sich in verschiedene Unterbegriffe anhand deren eine Einordnung des Kontexttyps ermöglicht wird.

Kategorie	Stimulus	Kontextart	Information	Systemreaktion
Präsentation	Eingang einer Nachricht	Applikation	Anwender spielt Videospiel	Ausgabe der Nachricht über Sprachsynthese da visuelle Ausgabe ablenken würde
Automatische Ausführung	Anwender wechselt Position	Position	Anwender bewegt sich von Küchentisch zum Wohnzimmertisch	Aktuell laufende Musikwiedergabe in Küche beenden und im Wohnzimmer fortsetzen
Kontext-Kennzeichnung	Anwender betrachtet Webseite	Benutzerpräferenz	Webseite wird mindestens täglich aufgerufen	System kennzeichnet Webseite als Benutzerpräferenz

Tabelle 3: Beispiel für Kontextverarbeitung

Diskursdeixis Referenziert Ausdrücke aus Texten oder in einem Gespräch, zum Beispiel „das“ in folgendem Gesprächsfragment: „Erinnerst Du noch das WM-Finale 1990? *Das* war ein Spiel!“

Raum-/ Lokaldeixis Verweis auf Ortsbeschreibungen, wie „hier“, „dort“ oder „dahin“.

Personaldeixis Personalpronomen die sich auf Gesprächsteilnehmer oder außenstehende beziehen. „Hast *Du* Morgenabend schon was vor?“

Zeit-/ Temporaldeixis Zeitliche Referenzen wie: „jetzt“, „heute“ oder „gestern“

Zum Teil entnommen aus (Mehling, 2010, S.52ff.).

2.4.4. Einordnung

Für das in dieser Arbeit angestrebte System sind Kontextinformationen von essentieller Bedeutung. In vielen Bereichen der Interaktion mit einer Smart-Home Umgebung müssen Kontextinformationen ausgewertet werde um ein optimales Ergebnis zu erzielen.

Neben der Auflösung von deiktischen Begriffen, hilft ein Kontext auch bei der disambiguierung mehrdeutiger Spracheingaben. In diesem Fall hilft es die Situation eines Anwenders zu kennen, also wo befindet er sich bzw. was tut er gerade.

Anstatt der Aposteriori Korrelation von Kontext und Spracheingabe bietet sich eine auch eine Apriori Nutzung an. Das heißt sobald sich der Kontext eines Anwenders ändert, erfolgt eine Anpassung der Spracherkenner-Grammatiken an die neu entstandene Situation. Dies ermöglicht eine feingranulare Steuerung des zu erkennenden Wortschatzes und vermeidet

so Vieldeutigkeit. Außerdem verhilft ein geringer Wortschatz zu höherer Robustheit der Erkennungsleistung.

Insbesondere helfen Kontextinformationen auch bei der Optimierung des Verhaltens hinsichtlich der Benutzbarkeit des Systems. Ein Anwender der gerade mit dem Zubereiten von Speisen beschäftigt ist, sollte nicht vom System gezwungen werden Eingaben per Tastatur oder anderem Peripheriegerät zu tätigen und dieses Gerät dadurch stark zu verschmutzen. Sowohl bei Ein- wie auch bei Ausgaben sollte der Kontext des Anwenders berücksichtigt werden.

2.5. Multimodale Interaktion

Der Begriff der Multimodalen Interaktion beschreibt das Interagieren von zwei oder mehr Parteien[1] über unterschiedliche Modalitäten. Eine Modalität ist hierbei abgeleitet von den menschlichen Sinnen. Für eine Übersicht siehe Tabelle 4.

Sinn	Modalität
Hören	Auditiv
Fühlen	Taktil
Riechen	Olfaktorisch
Schmecken	Gustatorisch
Sehen	Visuell

Tabelle 4: Sinne und Modalitäten

Das Nutzen multipler Modalitäten ermöglicht es eine für den Menschen intuitivere Benutzerschnittstelle zu schaffen (siehe Reithinger und Blocher (2003)).

Wahlster begründet die Nutzung von multimodaler Interaktion wie folgt:

> Since there are large individual differences in ability and preference to use different modalities, a multimodal dialogue system permits diverse user groups to exercise control over how they interact with application systems. Especially for mobile tasks, multimodal dialogue systems permit the modality choice and switching that is needed during the changing situational conditions. (Wahlster, 2006, S.3)

Wie auch Reithinger und Blocher (2003) beschreibt er also den Hauptnutzen von Multimodalität in der Anpassung an die bevorzugten Interaktionsparameter eines Benutzers. Zusätzlich

[1] Im Rahmen dieser Arbeit beschränkt sich die Interaktion auf die Kommunikation von Anwender und System

beschreibt Wahlster aber auch den Vorteil der vielfältigeren Interaktion bei Wechsel der Umgebungsbedingungen. Steht eine bestimmte Modalität nicht mehr zur Verfügung, zum Beispiel das Auditive (wegen Umgebungslärm), kann auf eine andere Modalität zurückgegriffen werden.

Wie bereits in Abschnitt Ubiquitous Computing (2.1) beschrieben ist die Mensch-Computer-Interaktion per Maus und Tastatur in Ihrer Reichweite stark eingeschränkt. Die Aufgabe einer Smart-Home Umgebung zeichnet sich jedoch unter anderem darin aus, den Anwender zu unterstützen und nicht ihn zu restriktieren. Daher ist es Sinnvoll die Steuerung mit anderen dynamischeren Modalitäten durchzuführen. Diese Modalitäten umfassen Sprache, Gesten, Berührung oder Augenbewegungen. Jede dieser Eingabemethoden bietet dabei ihre Vor- und Nachteile. Dadurch das zu jeder Zeit verschiedene Modalitäten zur Verfügung stehen, können jedoch die Nachteile minimiert werden.

Die Nutzung der Modalitäten muss dabei nicht exklusiv sein. Implizit vorhanden ist die Möglichkeit der Kombination verschiedener Modalitäten. Zum Beispiel der Einsatz eines Sprachbefehls statt des Durchklickens einer Menüstruktur um direkt zum Ziel zu gelangen und dort dann weitere Eingaben mit Maus und Tastatur zu tätigen.

2.6. Spracherkennung

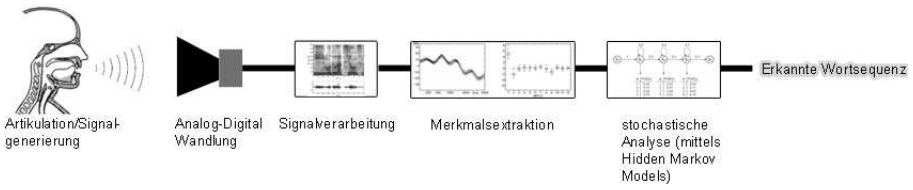

Abbildung 3: High Level Modell des Spracherkennungsablaufs

Spracherkennung versucht mittels eines Computerprogramms Sprachsignale, zum Beispiel von einem Mikrophon aufgezeichnet, in Textform zu überführen (Transkription). Der typische Ablauf ist in Abbildung 3 dargestellt. Dabei gibt es verschiedene Arten von Spracherkennern mit unterschiedlichen Aufgabengebieten, siehe Abbildung 4.
Das Problem der Spracherkennung bzw. Verarbeitung ist ein interdisziplinäres (Vgl. Abbildung 5). Im Folgenden werden kurz die einzelnen Bereiche und ihr Beitrag zum Erkennungsprozess erläutert.

Systemklasse	Verarbeitbare Äusserungen
Einzelworterkenner	Einzelne Wörter oder kurze Kommandos isoliert gesprochen, d.h. mit Pausen.
Keyword-Spotter	Einzelne Wörter oder kurze Kommandos in einer sonst beliebigen Äusserung.
Verbundworterkenner	Sequenz von fliessend gesprochenen Wörtern aus einem kleinen Vokabular (z.B. Telefonnummern).
Kontinuierlicher Spracherkenner	Ganze, fliessend gesprochene Sätze.

Abbildung 4: Einteilung von Spracherkennungssystemen, aus (Pfister und Kaufmann, 2008a, Seite 291)

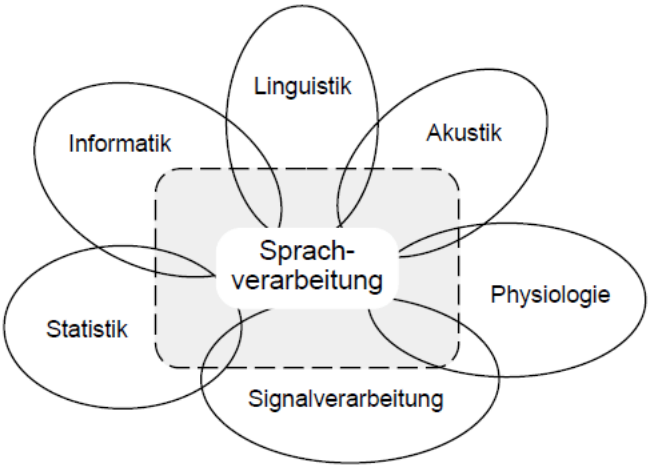

Abbildung 5: Disziplinen der Spracherkennung, aus (Pfister und Kaufmann, 2008a, Seite 22)

2.6.1. Signalverarbeitung

Sprache ist auf physikalischer Ebene gesehen nichts anderes als Schallwellen, die sich durch Luftdruckunterschiede manifestieren. Durch den Einsatz eines Mikrophons werden aus diesen Luftdruckunterschieden elektrische Signale generiert. Dies ermöglicht die weitere Verarbeitung zwecks Aufbereitung für die Informationsgewinnung. Da die Signalverarbeitung der erste Schritt auf dem Weg zum erkannten Text ist, hat sie Auswirkungen auf alle Folgeoperationen. Sie bildet somit das Fundament für den Erkennungsprozess.

Physiologie Das Fachgebiet der Physiologie, genauer der Humanphysiologie, beschäftigt sich mit der Funktionsweise der menschlichen Körpers (Vgl. (Guyton, 1991, Seite

3)). Der Aufbau des menschlichen Vokaltrakts beeinflusst die Ausbreitung der Schallwellen von den Stimmbändern. In Abbildung 6 sind die Bestandteile des Vokaltrakts zu sehen. Je nach Art ihres Zusammenspiels werden die unterschiedlichen Töne erzeugt (Vgl. Phonetics Flash Animation Project, University of Iowa,USA (2008)). Der Vokaltrakt steht somit in direktem Zusammenhang mit den Frequenzanteilen aus denen Sprachsignale aufgebaut sind. Je detaillierter das Verständnis seines Aufbaus ist, desto höher ist auch die Leistung beziehungsweise die Genauigkeit der Spracherkennung.

Abbildung 6: Aufbau des Vokaltrakts, aus Phonetics Flash Animation Project, University of Iowa,USA (2008)

Akustik Wie in den vorhergehenden Punkten erläutert bilden Schallwellen und deren Ausbreitung die Grundlagen der Sprache, die Akustik ist definiert als die Lehre dessen. Durch immer genauere Modelle für die Schallausbreitung und die Interferenz lassen sich die Einflüsse von Lärm auf das Sprachsignal besser approximieren und somit schon vor der Analyse beseitigen oder minimieren.

Linguistik Die Sprachwissenschaft beschäftigt sich mit der Erforschung und Beschreibung von Sprache. Neben der akustischen Grundlage bildet die Linguistik das semantische Modell der Spracherkennung. Sie beschreibt die Grundzüge der Sprache und wie sie sich definiert. Ohne diese Beschreibung ist eine Erkennung von Sprache unmöglich. Neben der Basisfunktion werden Ergebnisse der sprachwissenschaftlichen Forschung auch für die Verbesserung der Erkennung eingesetzt. Als Beispiel sei hier die Grammatik genannt. Anhand von grammatikalischen Regeln kann entschieden werden ob ein Wort Sinnvoll ist, oder ob besser eine ähnlich klingende Alternative erkannt werden sollte.

Statistik Statistische Erkenntnisse über die Häufigkeit von Lautkombinationen ermöglichen erst die stochastische Spracherkennung. Insbesondere der Einsatz von sogenannten Hidden-Markov-Modellen (HMM) hatte einen positiven Einfluss auf die Performanz von Spracherkennern. Wie genau HMMs eingesetzt werden lässt sich bei Rabiner (1989) nachlesen.

Informatik Die Informatik bildet die Schnittstelle zwischen Audiodaten und Datenverarbeitung. Die empfangenen Signale werden vom Computer ausgewertet und anhand der beschriebenen Merkmale verarbeitet und so in eine Textform überführt. Dabei werden zum Beispiel Algorithmen der Dynamischen Programmierung verwendet, um möglichst performant große HMMs zu untersuchen.

2.6.2. Spracherkennung in Smart-Home Umgebungen

Spracherkennung funktioniert dort gut, wo kontrollierte Bedingungen herrschen. Je unvorhersehbarer die Akustischen Umgebungsbedingungen sind, desto mehr verschlechtert sich die Qualität der erkannten Sprache. Kontrollierte Bedingungen bedeutet, dass beim Training des Erkenners die gleichen Parameterwerte aktiv sind wie später während der Ausführung. Einfluss haben zum Beispiel folgende Parameter:

- Mikrophontyp und der Abstand zum Sprecher
- Sprechervarianzen, wie unterschiedlicher Klang zum Beispiel durch eine Erkältung
- unterschiedliche Sprecher
- Lärmbedingungen, konstant wie intermittierend

Da es sich bei Smart-Home Umgebungen um normale Wohnräume handelt sind viele dieser Parameter starken Schwankungen unterworfen. Durch einfachen Positionswechsel ändert sich der Abstand zum Mikrophon. Durch Wechsel in einen anderen Raum ändert sich die Raumakustik (Hall). Vorbeifahrende Autos oder Nachbarn verursachen Geräusche. Insbsondere sprachähnlicher Lärm wie ein laufender Fernseher oder mehrere sich unterhaltende Personen sind der Erkennungsqualität abträglich.

Ein Smart-Home ist also eine denkbar ungünstige Umgebung für den Einsatz von Spracherkennung. Daher müssen Vorkehrungen getroffen werden um diesen Problematiken entgegen zu wirken (siehe Abschnitt 3.4). Viele vergleichbare Arbeiten setzen bei Ihren Evaluation daher entweder auf ein Headset-Mikrophon oder nutzen einen Wizard-of-Oz-Experiment Ansatz in dem ein Mensch die Rolle des Spracherkenners übernimmt (siehe Kapitel 3.2).

3. Analyse

Im folgenden Kapitel werden die für die Umsetzung der Zielsetzung notwendigen Anforderungen aus einem exemplarischen Szenario extrahiert, analysiert und problematisiert. Anhand der Ergebnisse existierender vergleichbarer Arbeiten wird auf Schwierigkeiten bei der Umsetzung dieser Arbeit geschlossen und etwaige Lösungsansätze werden diskutiert.

3.1. Szenario

Der folgende Abschnitt beschreibt das Hauptszenario das dieser Arbeit zu Grunde liegt. Es beinhaltet eine Vielzahl an möglichen Anwendungsfällen. Diese verdeutlichen wie kontextsensitive Interaktion in einer Smart-Home Umgebung stattfinden kann und welche Rolle dabei Multimodalität und insbesondere Spracherkennung spielen. Das Szenario stellt exemplarisch Abläufe dar die stellvertretend für viele weitere Interaktionsmöglichkeiten stehen.

3.1.1. Ein multimodaler Abend im der Wohnung der Zukunft

Nach einer anstrengenden Woche ist es endlich Freitag, später Nachmittag. X Y betritt fingerschnippend zum Beat der Musik seines Smartphones, seine modern eingerichtetes 1-Zimmer-Studio-Apartment. Wie auf Kommando erwacht das, wegen der heruntergelassenen Jalousien, noch in tiefes Schwarz gehüllte Ambiente zum Leben: Die Deckenbeleuchtung flammt auf und dimmt sich langsam auf angenehm warme 2700 Grad Kelvin, kleine ungefähr Tennisballgroße im Raum verteilte Kuben (Hamburg Cubical, siehe Gregor u. a. (2010)) melden über aufleuchtende LEDs Bereitschaft und die Musik die eben noch über die kleinen Ohrstöpsel piepste entfaltet auf der sauber eingemessenen Anlage ihre komplette Dynamik.

An einer Wand wird eine Projektion sichtbar. Sie zeigt verschiedene Symbole, einige blau umrandet, andere ausgegraut. Eine kurze Drehung am nächsten schwarzen Kubus und die Musik verschwindet. Aus den Lautsprechern ertönt: „Willkommen zu Hause X. Du hast neue Nachrichten.“. „Gib mir eine Nachrichtenübersicht“ spricht X in den Raum und setzt sich langsam auf ein großes gemütliches Sofa der erleuchteten Wand.

Die Projektion flackert kurz und zeigt nun eine Übersicht. Zwei Portrait-Fotos werden sichtbar auf denen eine kleine „1“ prangt. Am ersten Foto ist ein Briefumschlag zu erkennen, das zweite Foto ziert ein stilisierter Lautsprecher. Neben den beiden Fotos ist noch weiteres Symbol sichtbar, es zeigt eine Waschmaschine und ebenfalls eine Eins. „Zeig mir die Email von A“, das Portrait mit dem Briefumschlag leuchtet kurz auf und die Übersicht weicht einer Textdarstellung. Nach kurzem Überfliegen des Textes sagt X: „Ok, diese E-Mail löschen“. Die

Nachrichtenübersicht erscheint wieder. Ein kleiner Dreh erst um die Z-Achse auf eine andere Seite des Würfels und dann um Y-Achse und ein roter Rahmen taucht in der Projektion auf. Jede Drehung lässt den Rahmen ein Element weiter wechseln. Als er die kleine Waschmaschine erreicht hat sagt X „Nachricht öffnen“. „Der Trockengang ist abgeschlossen. Bitte Wäsche entnehmen“ erscheint in einer Sprechblase über der Waschmaschine. Seufzend steht X auf und schnappt sich einen im Regal verstauten Wäschekorb.

Mit der fertigen Wäsche und dem heißen Plätteisen bewaffnet beginnt X mit der Lieblingstätigkeit eines jeden Hemdträgers: „Bügeln“. Wieso gibt es dafür eigentlich noch keine Automatik?

Resignierend auf den weißen Wäscheberg starrend erinnert sich X an die noch wartende Sprachnachricht: „Spiele die Sprachnachricht ab.“. Einen Augenblick später ertönt eine weibliche Stimme über den Lautsprecher direkt über der errichteten Bügelstation. Sie berichtet ausführlich über tolles Wetter, epische Landschaften und nette Leute und schließt mit dem Satz: „Danke für die Super Empfehlung, kannst Du mir sagen, wo ich das Restaurant finde von dem Du so geschwärmt hast?“.

Das nächste weiße krause Hemd auf das Bügelbrett legend befiehlt X: „Starte Weltkarte“. Auf der Projektion erscheint ein langsam rotierender Globus. „Zeige mir ’Stadt’ in ’Land’“, der Globus dreht schnell in den richtigen Ausschnitt und vergrößert diesen bis die Stadt den Rahmen vollständig ausfüllt. „Suche Restaurants“, markiert mit kleinen Flaggen und den Namen der Gaststätten blitzen die verschiedenen Lokalitäten nach und nach auf der Karte auf. Neben einer der Flaggen, die mit einem R gekennzeichnet ist, befindet sich ein kleiner Fotoapparat, der signalisiert, dass eigene Fotos verfügbar sind. „Zeige mir „Restaurant R“. Der Kartenausschnitt zentriert sich über der Flagge und die vor Ort gemachten Fotos werden als kleine Vorschaubilder sichtbar. In Erinnerungen an das leckere Essen, das auf den Bildern zu erkennen, ist schwelgend, befördert X das nächste Hemd sauber zusammengelegt in den bereitgestellten Wäschekorb. „Route von hier“, auf der Karte erscheint ein kleiner Pin. „Zeige mir Hotel H“, der Kartenausschnitt bewegt sich und zeigt einen Gebäudekomplex direkt an der Küste. „(Route) Nach hier. Route speichern als „Hier lang““.

„Email an Y“. Die Projektion zeigt einen Texteditor. „Anlage hinzufügen, Route „Hier lang““. Unter dem Adressfeld erscheint eine Büroklammer die signalisiert, dass die Email nun eine Anlage besitzt. „Senden“. Die Projektion wechselt auf die Hauptebene vom Anfang zurück.

Noch schnell das Letzte nunmehr glatte und gefaltete Hemd im Wäschekorb verstaut und der anstrengende Teil des Abends ist beendet.

3.1.2. Analyse

Das Wohnen in einer Smart-Home Umgebung bietet den Bewohnern verschiedenste Vorteile und Erleichterungen. Auch wenn eine Bügelautomatik in den seltensten Fällen dazugehört. Die Adaption der Nutzervorlieben ist dabei ein wichtiger Bestandteil der die Akzeptanz eines Systems im besonderen Maße beeinflusst. Zum Beispiel der nahtlose Transfer der gehörten Musik vom Mobilgerät auf die statische HiFi-Anlage oder die automatische Anpassung der Lichtbedingungen.

Neben den automatischen Systemaktionen, die durch „Intelligente“ Auswertung verschiedener Sensoren und Regeln ausgelöst werden, fällt eine Schlüsselrolle der Interaktion zwischen Bewohner und Umgebung zu. Dem Wechsel vom Paradigma der indirekten Steuerung hin zur multimodalen Interaktion kann hier wohl die größte Wichtigkeit beigemessen werden. Die traditionelle Verteilung, ein Ausgabegerät und maximal zwei Eingabegeräte (Maus und Tastatur), wie man sie von normaler Computersteuerung gewöhnt ist, wird aufgebrochen und ersetzt. Adaptiv zur der Vorliebe des Anwenders kann indirekt (Würfel, Touchscreen aber auch Maus) oder direkt (Körper- und Handgesten, Sprache) interagiert werden. Auch Kombinationen verschiedener Modalitäten sind möglich. Die örtliche Gebundenheit verursacht durch Reichweite von Ein- und Ausgabegeräten wird kompensiert durch die Vielzahl verschiedener im Smart-Home verfügbarer Terminals. Bildschirme, Projektoren und Mobilgeräte (Tablet-PC, Smart-Phones) für die Informationsausgabe, sowie Mikrofone, Kameras und verschiedenste Human Interface Devices (HID) für die Eingabe.

Um diese Vielzahl an Interaktionsmöglichkeiten sinnvoll auswerten zu können müssen Informationen über den Anwender und seine aktuelle Tätigkeit gesammelt und verarbeitet werden. Wie auf eine Anwendereingabe reagiert wird, ist nun nicht mehr allein von der ausgeführten Aktion abhängig. Erst die Verknüpfung mit Kontextinformationen ermöglicht die Extrapolation der Semantik. Sprachbefehle wie „Suche Restaurants“ oder „Diese E-Mail löschen“ machen nur in ihrem jeweiligen Kontext Sinn. Erst die geographische Einschränkung durch die Wahl des Kartenausschnitts macht eine Suche nach Restaurants Sinnvoll. Eine globale Suche wäre zwar denk- und durchführbar aber durch die Masse an Ergebnissen nicht zielführend. Der Befehl „diese“ E-Mail zu löschen benötigt eine aktuell aktive Selektion ohne diese wäre der Befehl nicht ausführbar. Auch das Drehen des Würfels fällt in diese Kategorie. Zu Beginn des Szenarios ist nur die Audioausgabe und die Beleuchtung aktiv. Ein dreh am Würfel änderte in diesem Fall die Lautstärke der Ausgabe und nicht die Helligkeit. Welche Funktion aktiv ist lässt sich durch die Seite bestimmen auf der der Würfel sich gerade befindet. Später steuert er dann die Auswahl in der aktuellen Anwendung, ohne die Kontext-Information der aktiven Anwendung wäre dieses Verhalten nicht möglich.

3.2. Vergleichbare Arbeiten

Das Forschungsgebiet der multimodalen Interaktionen in Smart-Home Umgebungen ist ein sehr aktives (siehe Wahlster (2006), Abalos u. a. (2011), Aghajan u. a. (2009) oder Minker u. a. (2005b)). Verschiedene Systeme wurden auf diesem Gebiet bereits realisiert. Im Folgenden Abschnitt werden exemplarisch die Vorgehensweisen von vergleichbaren Arbeiten vorgestellt. So weit vorhanden werden die Ergebnisse dieser Projekte analysiert und hinsichtlich Richtlinien, Handlungsempfehlungen und nützlichen Vorgehensweisen ausgewertet.

3.2.1. INSPIRE

INSPIRE steht für **IN**fotainment management with **SP**eech, **I**nteraction via **RE**mote microphones and telephone interfaces ((Möller u. a., 2004, S.1)). Einige Ziele des von der EU im Rahmen des „Sixth Framework Programme“ unterstützten Projektes (Möller u. a., 2004, S.1) lauteten frei Übersetzt wie folgt:

- Anwenderfreundlicher Zugriff auf Informations- und Unterhaltungsgeräte in Smart-Home Umgebungen mit Hilfe der neuesten Sprachtechnologien.
- Anbieten eines „Portals für komplexe Geräte“ für behinderte oder technisch unbegabten Anwender Zuhause und am Arbeitsplatz.
- entwickeln eines benutzerfreundlichen, sprachgesteuerten Assistenten der per natürlichen Dialogen kommuniziert und zusätzliche Flexibilität hinsichtlich weiterer Szenarien, Technologischem Fortschritt und Mehrsprachigkeit bietet.

Das Ende August 2004 abgeschlossene Projekt hatte eine Laufzeit von 30 Monaten. Das Entwickelte System ermöglichte es, Einfluss zu nehmen auf die Funktion von Geräten im Wohnbereich einer Smart-Home Umgebung. Um dies zu erreichen wurde mit Hilfe von Mikrophon-Arrays, beziehungsweise kabellosen Headsets die Sprache des Anwenders aufgezeichnet und weiterverarbeitet. Diese Weiterverarbeitung umfasste die Sprecheridentifikation sowie die Spracherkennung durch zwei kommerziell erhältliche Spracherkenner (jeweils einer für die Projektsprachen Deutsch und Griechisch). Der Rückkanal bestand aus synthetisierten bzw. aufzeichneten Sprachmeldungen. Zusätzlich zur lokalen Interaktion war es möglich per Telefon Befehle und Anfragen zu äußern.

Ergebnis Da zum Zeitpunkt der Untersuchung das System noch nicht vollständig implementiert war, wurde eine Wizard-of-Oz-Simulation angewendet. Sowohl die Erkennung der Sprache also auch das Steuern der Geräte wurden durch einen Menschen durchgeführt. 24 Testpersonen bewerteten die Gesamtqualität des Dialogsystems mit einem Mittelwert von 3.3 (1-Schlecht, 5-Exzellent).

Fazit Die Akzeptanz der Gesamtprojektes liegt im Mittel bei Durchschnittlich. Bezüglich der Qualität der Spracherkennung verhielt sich das Headsetmikrophon am besten. Dies kann auf den statischen Aufbau zurückgeführt werden. Wie bereits beschrieben funktioniert Spracherkennung anhand von Mustererkennung. Die Mustererkennung benötigt für hohe Erkennungsraten den Trainingsdaten möglichst ähnliche Kandidaten. Insbesondere der Abstand zum Mikrophon und die Verrauschung des Sprachsignals wirken sich negativ auf die Erkennung aus. Bei einem Headsetmikrophon ist der Aufnahmeradius so gering gehalten das nur sehr wenig Rauschen das Sprachsignal verunreinigt. Zudem ist durch die Montage des Mikrophons der Abstand zur Sprachquelle konstant und ermöglicht ein homogenes Schalldruckspektrum.

3.2.2. Mayordomo

Mayordomo (Abalos u. a., 2011) ist ein multimodal steuerbares Dialogsystem. Die Hauptaufgabe des Systems besteht in der Zentralisierung der Steuerung verschiedener Haushaltstypischer Geräte.

Die Multimodalität setzt sich hierbei aus der Nutzung von Spracherkennung und traditioneller Tastatur/Maus zusammen.

Als Spracherkennungssoftware kommt die Microsoft Speech API zum Einsatz. Für jedes zu steuernde Gerät wurde eine Sprachgrammatik, im SRGS-Format[2] erstellt. Diese Grammatiken setzen sich aus Schlüsselworten zusammen, welche die gerätespezifischen Eigenheiten und Optionen repräsentieren. Zum Beispiel für ein Fernsehgerät enthält die Grammatik Schlüsselworte die beschreiben, wo sich das Gerät befindet, welche Attribute es besitzt (Lautstärke, Kanal) oder seine möglichen Aktionen (ein- und ausschalten, Kanal wechseln oder Lautstärke regulieren).

Anhand der erkannten Schlüsselwörter wird dann die gewünschte Aktion abgeleitet. Eine Aktion besteht aus vier Parametern:

Raum Die Lokation in der sich das zu steuernde Gerät befindet.

[2] SRGS steht für Speech Recognition Grammar Specification, ein vom W3 Konsortium erstelltes Format zu erstellung Spracherkennungs Grammatiken

Gerät Der Gerätekontext der Aktion.

Attribut Das zu veränderte Attribut des Geräts.

Wert Der Wert auf den das Attribut angepasst werden soll.

Sollte eine oder mehrere dieser Parameter fehlen werden diese abgefragt beziehungsweise aus anderen Parametern erschlossen. Wenn der Raum, Wert und Attribut bekannt sind aber das Gerät fehlt wird anhand dieser Werte nach einem passenden Gerät gesucht. Bei nicht eindeutiger Zuordnung wird nachgefragt, welches Gerät gemeint ist.
Neben der Steuerung von Geräten ist ein weiterer Interaktionstyp vorgesehen: die Informationsabfrage. Wenn der erkannte Sprachbefehl mit einem Frage Adverb beginnt oder aber mit einer Form des Verbs Sein wird dies als Informationsanfrage gewertet. Das sogenannte Provide Information-Modul beantwortet die Anfrage anhand der für das System vorliegenden Konfigurationsdateien.

Fazit Das Projekt enthält verschiedene für diese Arbeit interessante Ansätze. Die Abstraktion von einzelnen Funktionen in eine Aktions-Metapher ist besonders für die Steuerbarkeit sowie die Erweiterung des Systems interessant. Durch diese Abstraktion wird ein zentrales speichern von möglichen Aktionen möglich die durch verschiedene Oberflächen und über unterschiedliche Modalitäten abgefragt beziehungsweise angesteuert werden können. Zur Verbesserung des Konzeptes wäre es denkbar von Gerätespezifischen Grammatiken auf Domänenbezogene Steuerdateien zu wechseln. Also einen weiteren Steuerlevel über die Geräteebene zu legen um verschiedene geräteübergreifende Aktionen wie die getrennte Ausgabe von Video- und Audiodaten, zu ermöglichen.
Da MAYORDOMO ebenfalls für die Spracherkennung auf die MSSAPI zurückgreift kann angenommen werden, das ein Einsatz in Smart-Home Umgebungen durchführbar ist.

3.2.3. SmartKom

Das Ziel des vom Deutschen Institut für Künstliche Intelligenz (DFKI) initiierten SmartKom Projekts war die Erforschung von Multimodaler Interaktion unter realistischen Bedingungen ((Wahlster, 2006, S.5)). Die Projektgruppe wurde dabei von verschiedenen Vertretern aus der Wirschaft (unter anderem BMW) sowie der Forschung (verschiedenen Universitäten und einem Forschungsinstitut) gebildet. Geleitet wurde das Projekt von Professor Dr. Wolfgang Wahlster. Abgeschlossen wurde die Arbeit im September 2003.

Das System implementiert einen intelligenten Agenten („Smartakus“) der ähnlich der in der Darmstadt Challenge vorgestellten Metapher (siehe Augusto u. a. (2009)), als stiller, allgegenwärtiger Butler agiert. Dies natürlich nur in dem Maße, wie es sich bei den Tätigkeiten

eines Butlers um das bereitstellen von Informationen und verändern verfügbarer Parameter handelt. Anwender des Systems haben die Möglichkeit, mit Smartakus in verschiedenen Modalitäten zu interagieren. Dies wurde anhand von drei Anwendungsszenarien vorgestellt: SmartKom-Mobile, SmartKom-Public und SmartKom-Home. Dabei ist insbesondere die Home-Metapher für diese Arbeit von Interesse. Die Funktion des SmartKom-Home Szenarios ist beschränkt auf die Interaktion mit einem Medienabspiel System. Das bedeutet es werden Funktionen wie das Ein- und Ausschalten des Geräts oder das Programmieren von Aufnahmen unterstützt. Bedient wird das System dabei entweder über ein Touchpad oder per Sprachbefehl. Das System unterscheidet hierbei zwischen zwei Modi:

Lean-Forward Der Eingabefokus des Anwenders liegt auf dem Touchpad. Das System kann alle Ausgabemodalitäten nutzen.

Lean-Backward Der Anwender hat sich vom Touchpad abgewendet und sieht zum Beispiel fern. Das System verzichtet auf eine visuelle Ausgabe um das Fernsehbild nicht zu stören und stellt auf eine verbale Ausgabe um.

Für eine vollständige Liste der Funktionen siehe (Reithinger und Blocher, 2003, S. 7).

Das System ist implementiert als eine Multi-Blackboard-Architektur ((Reithinger und Blocher, 2003, Seite 8)). Jedes Modul (zum Beispiel Gesten- oder Sprachanalyse) liefert sein Ergebnis in einem XML-Dialekt, der sogenannten M3L (Multimodal Markup Language). Basierend auf einer Systemweiten Ontologie werden diese Ergebnisse weiterverarbeitet und führen schlussendlich zu einer oder mehreren Aktionen. Um die verschiedenen Modalitäten zu fusionieren wird auf ein sogenanntes Dialoggedächtnis zurückgegriffen, dieses enthält die für die Auflösung von Referenzierten Entitäten notwendigen Informationen.
Für nähere Informationen siehe Wahlster (2006),Reithinger und Blocher (2003) bzw. Engel (2006).

> Die Schnittstellen und die übergreifende Domänenrepräsentation sind in M3L (Multimodal Markup Language) definiert. M3L ist ein XML-Dialekt und definiert mittels XML-Schemata die Syntax, in der die Daten zum Austausch zwischen dem Modulen von SmartKom notiert werden. Integriert in diese Sprache ist die einheitliche, hierarchische Domänenmodellierung, die alle Anwendungen umfasst. Die unterliegende Wissensquelle ist eine Ontologie, die in OIL (OIL 2003) definiert wird und in ein XML-Schema umgesetzt wird. In SmartKom werden alle Verarbeitungsschritte und Zwischenergebnisse mittels M3L repräsentiert. Das inkludierte ontologische Wissen wird in den verschiedenen Modulen zusätzlich für Inferenzprozesse verwendet.

Fazit Der Funktionsvielfalt von SmartKom ist bedingt durch den Zeit- und Ressourceneinsatz mit dem dieser Arbeit nur schwer vergleichbar. Das Projekt bietet aber hinsichtlich der Aufteilung in Komponenten und dem Workflow gute Ansätze. Auch die Nutzung einer global verfügbaren Ontologie ist ein guter Ansatz um später dynamische Abfragen definieren zu können.

Die Spracherkennung kann in allen drei Szenarien gute Ergebnisse liefern, da jeweils kontrollierte Bedingungen vorliegen. Das SmartKom-Home Szenario ist beschränkt auf einen Raum und auf eine bzw. zwei Anwenderpositionen. Das bedeutet das nur geringe Varianz bei der Aufnahme des Sprachsignals zu erwarten ist. Abgesehen von Ausgabe des Fernsehgerätes. Diese kann aber dadurch das die Quelle des „Lärms“ bekannt ist über Spektrale Subtraktion aus dem Audiosignal entfernt werden. Bei einer konstant guten Qualität bietet sich die durchgeführte Aposteriori Sprachanalyse an, bei der die Semantik aus dem erkannten Freitext abgeleitet wird und nicht aus vorher definierten Grammatiken. Einzelne Entitäten sind in der Ontologie definiert und können so referenziert werden.

3.2.4. Analyse

Aus den vorgestellten Arbeiten lassen sich verschiedene Schlüsse für die Konzipierung dieses Arbeit ziehen.

Bezüglich der Spracherkennung lässt sich feststellen, dass in den Fällen, in denen die Bedingungen nicht optimal waren Vereinfachungen vorgenommen wurden (Einsatz eines Headsetmikrophons oder menschliche Transkription). Das lässt darauf schließen, das die für eine natürliche Interaktion notwendigen Erkennungsleistung in dynamischen Umgebungen nur schwer erreichbar ist. Dies muss bei der Realisierung berücksichtigt werden.

Der Einsatz von Freitexterkennung ohne feste Vorgabe von Grammatik und Semantik ist zwar wünschenswert allerdings in heterogenen Umgebungen nur schwierig durchzuführen, da eine hohe Genauigkeit der Erkennung benötigt wird. Dieses Anforderung würde es entweder notwendig machen das der Anwender einen bestimmten Bereich nicht verlässt oder mit einem Personengebundenen Mikrophon ausgestattet wird.

3.3. Anforderungen

Aus dem Szenario und den vergleichbaren Arbeiten lassen sich verschiedene Anforderungen ableiten. Im Folgenden Abschnitt werden diese analysiert, kategorisiert, problematisiert und erörtert.

3.3.1. Funktionale Anforderungen

Damit das zu entwickelnde System das Szenario umsetzen kann müssen verschiedene funktionale Anforderungen erfüllt werden. Diese werden zuerst überblickshaft aufgelistet und dann die wichtigsten in weiteren Abschnitten näher erläutert.

Ereignisse Auftretende Ereignisse müssen vom System erkannt werden. Ereignisse steht in diesem Fall für Sensorereignisse, zum Beispiel für das Betreten der Smart-Home Umgebung durch einen Anwender oder die Statusänderungen von Haushaltsgeräten (in diesem Fall der Waschmaschine).

Transparenz Das System muss die für verteilte Systeme typischen Formen von Transparenz unterstützen. Das bedeutet das möglichst alle Aktionen, unabhängig vom Ort ihrer Durchführung, über ein gemeinsames Interface steuerbar sein sollen (Zugriffs- und Ortstransparenz und „openness“ (Tanenbaum und Steen, 2006, S.8f)). Während der Anwender mit dem System interagiert soll er nach Möglichkeit auch seine Position ändern können und trotzdem weiterhin Zugriff auf die gesteuerte Komponenten behalten (Relocation-Transparenz).

Einfache Erweiterbarkeit Die möglichen Funktionen der im Systems zusammengeschlossenen Komponenten sollen durchsuchbar gestaltet werden um eine einfache Adaptierung des Systems an neue Anwendungen zu ermöglichen.

Erkennung und Verarbeitung von Spracheingaben Die vom Anwender geäußerte Sprache muss erkannt und in Systemkommandos transformiert werden. Dabei sollen auch referenzierte Entitäten erkannt und aufgelöst werden können.

Steuerbarkeit der Umgebung Die Einzelnen Bestandteile(Aktorik und Sensorik) der Smart-Home Umgebung müssen steuerbar sein. Das bedeutet das zum Beispiel die Beleuchtung über eine per Netzwerk ansprechbare Schnittstelle verfügt. Folgende Komponenten müssen steuerbar sein:

- Beleuchtung
- Audioausgabe/Hifi-Anlage
- Videoausgabe/Projektoren/Fernseher
- Weitere Peripheriegeräte (Cubicle, Waschmaschine)

Dashboard Die im Szenario angestrebte Verwaltungsumgebung soll mindestens folgende Anforderungen erfüllen:

Verwaltung von Informationen Eine Oberfläche für die Übersicht und Verarbeitung der eintreffenden und ausgehenden Nachrichten muss entwickelt werden. Dabei soll das erstellen von E-Mail Nachrichten möglich sein, sowie der damit verbundene Zugriff auf Kontaktinformationen eines Adressbuchs.

Statusanzeige Der aktuelle Status des Smart-Homes bezüglich einzelner Sensoren und Geräte soll übersichtlich dargestellt werden um einen Überblick um die aktuelle Situation zu bekommen und um die Fehlersuche zu erleichtern.

Bedienbarkeit Die Verwaltungsoberfläche soll sowohl mit den klassischen Eingabegeräten, Maus und Tastatur, wie auch mit alternativen Eingaben steuerbar sein. Das bedeutet ein Navigieren in den Menüs per Sprache und/oder Gesten soll möglich sein.

Routenplanung Die Geospatiale Routing Anwendung muss folgende Anforderungen erfüllen:

Anzeige von Karten und Satellitendaten Für die Visualisierung von Routen und Lokationsabhängigen Informationen muss die Anwendung Karten und Satellitenbilder der jeweiligen Region enthalten und wiedergeben können. Die Navigation innerhalb der Kartendaten muss ebenso möglich sein.

Umkreissuche Um lokal ansässige Branchen zu finden muss die Anwendung eine Datenbank enthalten die spatiale wie auch generelle Informationen über diese Branchen enthält. Dies ermöglicht eine Suche nach verschiedenen Begriffen in Abhängigkeit von einer bestimmten Position.

Darstellen von POIs Die Anwendung muss die Anzeige von Benutzer- bzw. Systemgenerierten Points of Interests unterstützen. Das bedeutet das Ort die von besonderer Bedeutung sind, oder die mit weiteren Benutzerspezifischem Inhalt versehen wurden dargestellt werden können müssen. Das ermöglicht es zum Beispiel Orte auf der Karte zu kennzeichnen, an denen der Benutzer Fotos aufgenommen hat oder auch das Darstellen von Kontakten deren Adressen bekannt sind (Geocoding).

3.3.2. Wissensverwaltung

Eine zentrale Rolle im System ist die Verwaltung von Wissen. Das bedeutet das Sammeln und Auswerten von Informationen die für die Abläufe in der Smart-Home Umgebung wichtig sind. Alle Systembestandteile sollen in der Wissensdatenbank repräsentiert werden können.

Ähnlich der im Abschnitt Vergleichbare Arbeiten vorgestellten SmartKom Umgebung soll auch in diesem Projekt eine Globale Begriffswelt beziehungsweise Ontologie entstehen. Diese ermöglicht konsistenten Zugriff auf alle Systembestandteile anhand von vordefinierten Bezeichnern. Diese Bezeichner sollten eine Typstruktur umfassen. Diese weist den Bestandteilen der Ontologie Typen zu, die zum Beispiel für deduktives Schließen oder beim Suchen von Begriffen Einsatz finden können.

Um die Ambiguität von Sprache zu kompensieren, sollte ein Konzept für Synonyme geschaffen werden. Das verschiedene Begriffe auf die gleiche Semantik abbildet.

Die Wissensbasis sollte über die Systemgrenzen hinaus mit Informationen versorgt werden können. Das bedeutet das diverse Wissensquellen über ein Adapterkonzept angeschlossen und verwendet werden können. Denkbar wären zum Beispiel E-Commerce Datenbanken um integriertes Online-Shopping anbieten zu können. Oder aber Musik-Portale die ausgewählte Musik-Abspiellisten enthalten. Damit bleibt das System erweiterbar und ermöglicht es den Anwender auch in Zukunft optimal unterstützen zu können.

3.3.3. Kontext

Um die Interaktion mit dem System möglichst intuitiv und natürlich zu gestalten muss der Kontext des Anwenders berücksichtigt werden. Dies macht eine Abbildung des abstrakten Konstrukts „Kontext“ auf ein konkretes Modell notwendig. Wie im Abschnitt Grundlagen beschrieben definiert sich ein Kontext über folgende Parameter:

Anwendung Eine in Software realisierte Entität die den Anwender bei der Lösung von Problemen unterstützt.

Anwender Der Bediener der Anwendung (Problemsteller).

Ereignisse Etwas, das eine Aktion auslöst bzw. den Zustand des Systems verändert.

Setting Die relativ statischen Eigenschaften der Umgebung in der sich der Anwender befindet.

In welcher Form und welcher Detaillierung diese Parameter zur Verfügung gestellt werden hängt sowohl von der Sensorausstattung wie auch dem Einsatzzweck ab. Die Abbildung des Kontextes muss Veränderungen zum Beispiel durch erweitern der Sensorik kompensieren können.
Zur Berücksichtigen ist weiterhin, welche Ereignisse für die jeweilige Anwendung von Interesse sind. Daher sollte die Anwendung selbst entscheiden können, ob und wenn ja über welche Ereignisse sie informiert wird. Es muss insbesondere das Beenden von Kontexten modelliert werden, da die aktiven Kontexte für die Steuerung des Gesamtsystems untersucht werden

und somit in direktem Zusammenhang mit dem Laden von Spracherkennungs-Grammatiken stehen.

Synchronisation Um eine zuverlässige Zuordnung von Ereignissen zu Anwendern und Anwendung zu ermöglichen, muss eine zeitliche Synchronität der teilnehmenden Sensoren und Geräte hergestellt werden. Dies ist insbesondere wichtig um die in Spracherkennung vorkommenden Deiktischen Referenzen aufzulösen:
„Setze Routenstart hierher“. Wenn diese Phrase erkannt worden ist, ist die Aktion die die Position des Routenstarts festlegt bereits geschehen. Um nun eine Korrelation herzustellen muss der Zeitpunkt der Sprachäußerung, „hierher“ mit den Ereigniszeitpunkten abgeglichen werden.

3.3.4. Kontexterkennung

Anwenderzahl Aus der Kontextdefinition ist ersichtlich, dass ein Kontextsensitives System wissen muss, welcher Benutzer gerade aktiv ist, und mit welcher Anwendung er interagiert. Im einfachsten Fall befindet sich nur ein Anwender in der Smart-Home Umgebung so dass dieser als Standardnutzer feststeht (Ausnahmen bilden z.B. reine Systemaktionen). Sobald jedoch mehrere Anwender präsent sind, muss unterschieden werden, wer am Anwendungskontext beteiligt ist. Dafür ist zunächst einmal eine Anwendererkennung notwendig.

Anwendererkennung Die Anwendererkennung stellt den Ausgangspunkt einer Aktion fest. Je nach genutzter Modalität ist dies mehr oder weniger Eindeutig.

Die folgende Tabelle listet beispielhaft verschiedene der möglichen Szenarien.

Anwendungserkennung Durch Kombination des Wissens über den Anwender, die Semantik der Aktion und die aktiven Kontexte lassen sich Rückschlüsse auf die Anwendung ziehen. Durch den Abgleich der aktiven Kontexte mit der Liste der beteiligten Anwender sowie der angeforderten Aktion sollte eine Zuordnung zu einer Anwendungsinstanz möglich sein. Im Falle von mehrdeutigen Abbildungen müssen Dialogaktionen gestartet werden um das Ziel der Aktion zu identifizieren.

Modalität	Beteiligte Komponenten	Erkennungsaktion	Problematik
Sprache	Mikrofon, Indoor-Positioning-System	Anhand des Zeitstempels des erkannten Sprachbefehls werden die Positionen der Anwender mit der Aktivierung des Mikrofons korreliert und anhand dessen der Nutzer festgestellt der am wahrscheinlichsten den Sprachbefehl abgesetzt hat. Um die Genauigkeit zu erhöhen ist der Einsatz einer Stimmenerkennung denkbar der die Sprachäußerung anhand der Stimme einem Nutzer zuordnet.	Die Zuordnung ist nicht vollständig verlässlich. Da die physikalische Anwesenheit kein eindeutiges Kriterium für die Zuordnung des Sprachbefehls darstellt. Diese Art der Erkennung bietet sich an, für die Wahl des Auswahlmediums (spatiale Nähe) sowie für nicht Nutzerabhängige Interaktion (z.B. Aktionen die unabhängig sind von Anwenderpräferenzen)
Touch	Eingabegerät	Die Zuordnung der Nutzers wird von der die Touchaktion empfangenen Applikation durchgeführt. Bei Multitouchsystemen an denen mehrere Nutzer beteiligt sind muss die Mehrdeutigkeit von der Anwendung aufgelöst werden.	Eine eindeutige Zuordnung von Nutzer zu Aktion ist auch hier nur über weitere Hilfsmittel wie visuelle Analyse machbar.

Tabelle 5: Benutzerkennung für unterschiedliche Modalitäten

3.3.5. Benutzerpräferenzen

Für eine angemessene und natürliche Unterstützung des Anwenders ist es notwendig, seine Präferenzen zu kennen (siehe auch (Augusto, 2004, S.3). Erst wenn die Vorlieben des Anwenders vom System berücksichtigt werden, ist eine natürliche Systeminteraktion gegeben. Das System soll einen Mechanismus bieten um Anwendervorlieben abzubilden und entsprechend verarbeiten zu können. Im Zusammenhang mit dem vorgestellten Szenario wäre dies zum Beispiel das automatische übernehmen der bevorzugten Raumbeleuchtung bei betreten des Smart-Homes.

3.3.6. Multimodale-Interaktion

Wie in Abschnitt 2.3 beschrieben findet die Interaktion zwischen einer Smart-Home Umgebung und dem Anwender auf verschiedenen Kanälen statt. Je nach Situation kann der

Nutzer dabei zwischen unterschiedlichen Ein- und Ausgabemodalitäten wählen. Voraussetzung dafür ist eine Unterstützung dieses Vorgangs von der Anwendung. Daher müssen die Anwendungen, die in der Smart-Home Umgebung eingesetzt werden sollen, eine Bidirektionale Steuerschnittstelle erhalten. Diese Schnittstelle ermöglicht die Kontrolle der Anwendung aus der „Ferne“ und somit auch eine Adaption nicht nativ unterstützter Modalitäten.

Beispiel Zur Verdeutlichung dient hier eine Schachspiel-Simulation. Diese ermöglicht eine Eingabe der Züge über Maus und Tastatur. Durch Unterstützung der Fernsteuerungsschnittstelle kann aber auch eine Sprachsteuerung mit wenig Aufwand realisiert werden. Die Befehle werden vom Spracherkennungsmodul erkannt und in Maus und Tastaturbefehle übersetzt. Zum Beispiel „Springer A3 auf C1“.

3.3.7. Spracherkennung

Die Eingabe via Sprache ist neben der Gestenerkennung die direkteste Modalität. Sie bietet sich insbesondere dann an, wenn andere Modalitäten nicht zur Verfügung stehen. Zum Beispiel wenn Hände und Arme andere Tätigkeiten vornehmen. Die unterschiedlichen Voraussetzungen der verschiedenen Modalitäten sind exemplarisch der Tabelle 6 zu entnehmen.

Modalität	Anforderungen Anwender	Anforderungen System
Sprache	Spracherzeugung (deutlich)	Sprachempfang (Mikrophone)
Berührung (Touch, Tastatur oder Maus)	freie Hand, spatiale Nähe zu Eingabegerät	verfügbare Eingabegeräte
Blicksteuerung	Sichtbare Augen	Eye-Tracker, Lichtbedingungen
Gesten	freie Extremitäten (Gestikulatoren)	Lichtbedingungen, optische Aufnahmegeräte

Tabelle 6: Eigenschaften von Modalitäten

Um Sprache als Eingabemedium nutzen zu können müssen verschiedene Voraussetzungen erfüllt sein:

Audioaufnahme Aufnahme der Sprachsignale wo sie zur Steuerung genutzt werden sollen.

Audioqualität Die Audiodaten müssen qualitativ hochwertig sein und ähnlich der Trainingsdaten des Spracherekenners.

Spracherkenner Eine Software die Schallwellen in ihre textuelle Repräsentation umwandelt (Transkription).

Semantik-Interpretation Aus der Transkription muss die jeweilige Bedeutung abgeleitet werden um sie an das System weitergeben zu können.

Dialog Von großer Bedeutung bei der sprachlichen Interaktion zwischen Mensch und System ist die Führung bzw. Unterstützung des Anwenders. Wie auch bei traditioneller Anwendungssoftware muss dem Nutzer vermittelt werden, welche Möglichkeiten er hat. Anschaulich lässt sich ein exklusiv Sprachgesteuerter Dialog mit einer Kommandozeilenumgebung vergleichen (Yankelovich, 1996). Welche Funktion die Anwendung unterstützt ist nicht ersichtlich und eröffnet sich erst durch Erfahrung mit dem System.
Die Ambiguität von Sprache und die Einschränkungen der heutigen Spracherkenner machen eine vollständig freie Sprachsteuerung äußerst komplex. Daher muss die Konversation zwischen Anwender und System in geordnete Bahnen gelenkt werden. Dies wird ermöglicht durch vorformulierte Dialoge.

Der zu erkennende Wortschatz vermindert sich durch diese Ablaufpläne radikal und ermöglicht Spracherkennern auch in suboptimalen Umgebungen (hoher Rauschanteil) gute Erkennungsraten. Zusätzlich lassen sich die Dialogbestandteile Apriori mit Semantik annotieren. So dass eine Auswertung, also eine Transformation der Sprachbefehle in Systemaktionen, vereinfacht wird. Abstriche müssen hier bei der Benutzerfreundlichkeit gemacht werden. Ein vorgefertigter Dialog erfordert vom Benutzer Kenntnisse über den Ablauf. Abmildern lässt sich dies durch eine Benutzerführung beziehungsweise Kontextbezogene Hilfetexte.

Es ist wichtig, dass sich Dialoge Systemweit konsistent verhalten: Dialogsteuerung, z.B. Schritt vor oder zurück, die Abfrage von Optionen und das Abbrechen des Vorgangs sollten in allen Dialogen gleichermaßen aufgerufen werden können.

3.4. Problematisierung

Bereits im Vorfeld lassen sich aus implementierten Systemen Rückschlüsse ziehen auf Problematiken die das angestrebte System mit sich bringt. Dieser Abschnitt beschreibt die wichtigsten Problematiken und etwaige Lösungsansätze.

3.4.1. User Acceptance

Das eine konzipierte Smart-Home Umgebung auch außerhalb von Forschungsstätten ihren Einsatz findet ist insbesondere von der Nutzerakzeptanz abhängig.

> The current challenge then is, ‘simply’, to satisfy the user. We already have all sort of smart environments exhibiting some degree of intelligence but AmI will not be adopted until the user can use the systems comfortably.(Augusto u. a., 2009, S.2)

Augusto et. al beschreiben dies als Zufriedenstellen des Anwenders. Erst wenn ein Anwender das System akzeptiert und es als angenehm empfindet mit ihm zu interagieren ist eine Anwendung außerhalb der Forschung denkbar.

Sprachsteuerung Insbesondere die Steuerung des Systems über Sprachbefehle kann für einen Anwender befremdlich wirken. Das Kommunizieren mit einer entkörperten nicht fassbaren Instanz erfordert besonders in Anwesenheit Anderer ein großes Maß an Überwindung. Um dem Anwender die Kommunikation zu erleichtern sollte verdeutlicht werden das eine Kommunikation mit dem System statt findet. Zum Beispiel durch Darstellung des Systems als animiertem Avatar (vgl. Smartakus im SmartKom Projekt).

Privacy Eine Smart-Home Umgebung soll primär den Benutzer bei seinen täglichen Arbeiten unterstützen. Dafür ist es notwendig, verschiedene Sensorik im Wohnbereich zu platzieren. Ohne explizite Sicherung stellt diese Sensorik einen groben Eingriff in die Privatsphäre einer Person da. Es muss sichergestellt sein, das alle Aufzeichnungen zeitnah gelöscht werden und somit ein Kompromittieren ausgeschlossen wird.

Untersuchungen die sich mit Vorbehalten gegenüber den Eingriffen in die Privatsphäre in Smart-Home Umgebungen beschäftigen beziehen sich meist auf den Bereich Ambient Assisted Living (siehe zum Beispiel Skubic u. a. (2009)). Das lässt sich mit der höheren Realitätsbewandtnis begründen. Durch die fortschreitende Überalterung der Gesellschaft gewinnen AAL Umgebungen immer mehr an Importanz. Das Ergebnis dieser Untersuchung ist, dass Nutzer die auf die Unterstützung durch das Smart-Home angewiesen sind, einen Eingriff in Privatssphäre eher in Kauf nehmen als Anwender die das System freiwillig nutzen.

Evaluation Um die Nutzerakzeptanz zu messen wurde die sogenannte „Darmstadt Challenge“Augusto u. a. (2009) entwickelt. Benannt nach dem Ort der Entstehung (Symposium on Wildlife and Horticultural Applications and their Ambient Intelligence 2007), stellt sie, ähnlich dem Turing-Test, eine Herausforderung dar. Sie besteht aus einem domänenspezifischen Fragenkatalog (siehe Anhang A.2), anhand dessen der Nutzen und insbesondere die Proaktivität eines Smart-Homes (bzw. Smart-Office, Öffentlicher Bücherei oder Klassenraum-Assistenten) beurteilt werden kann. Anhand dieses Fragenkatalogs soll eine Einschätzung der Nützlichkeit der Smart-Home Funktionen möglich werden.

3.4.2. Spracherkennung in Smart-Home Umgebungen

Die heute verfügbaren Spracherkenner ermöglichen unter kontrollierten Bedingungen hohe Erkennungs- und niedrige Fehlerraten (siehe Comerford u. a. (1997)). Dies setzt jedoch eine Ausführungsumgebung voraus, deren Charakteristik nicht oder nur minimal von der Trainingsumgebung abweicht. Je höher die Abweichung, umso schlechter die Qualität der Erkennung (Vgl. (Aghajan u. a., 2009, S. 6)). Die starke Varianz der Eingabebedingungen in Smart-Home Umgebungen stellt eine sehr schlechte Grundlage dar. Insbesondere müssen folgende Problematiken bewältigt werden:

Raum-Nachhall Durch die Raum-Charakteristik verursachte Reflektionen des Schalls die vom Mikrofon aufgenommen werden. Je nach Abstand des Sprechers zum aufnehmenden Mikrofon können diese Störgeräusche das Original Signal verunreinigen. Vgl. Nakamura u. a. (1996).

Rauschen Neben- und Hintergrundgeräusche die dem Sprachsignal hinzugefügt werden verschlechtern die Erkennungsleistung signifikant. In Smart-Home Umgebungen kann dieser Lärm z.B. von Küchengeräten, Medienabspielgeräten oder auch externen Lärmquellen stammen. Eine zusätzliche Störquelle ist weitere Sprache von Personen die sich in der Umgebung befinden aber nicht mit dem System kommunizieren. Dies bezeichnet man als das „cocktail party problem „Huang u. a. (2001)

Aufnahmegerät Eigenheiten Die Differenz in der Spektralcharakteristik zwischen dem Mikrophon, mit dem die Trainingsdaten aufgenommen wurden, und dem das die späteren Sprachebefehle erhält wirkt sich negativ auf die Erkennungsgenauigkeit aus.

Sprechervarianz Die Sprachäußerung eines Sprechers hängt nicht unerheblich von seiner aktuellen physischen sowie psychischen Verfassung ab. Die Spektrale Ausprägung eines Worts, geäußert in unterschiedlichen Verfassungen, variiert stark.

Deixis Von besonderer Bedeutung sind sogenannte Deiktische Ausdrücke (siehe 2.4.3). Sie ermöglichen die Auflösung von referentiellen Begriffen die in verschiedenen Befehlen vorkommen können. Voraussetzung dafür ist, dass das System die Referenz-Objekte kennt bzw. etwaige Referenzereignisse protokolliert. Um beispielsweise den lokaldeiktischen Begriff „hier“ aufzulösen ist es notwendig den Nutzerkontext zu kennen. Je nach Anwendung gibt es verschiedene Interpretationen:

Beleuchtungssteuerung Hier bezieht sich auf die lokale Position des Anwenders in der Smart-Home Umgebung zum Zeitpunkt der Sprachäußerung.

Geospatiale Routing- oder Mapping Anwendung Hier kann sich sowohl auf den gewählten Kartenausschnitt oder eine gerade markierte Landmarke beziehen.

Gesten Hier referenziert eine durch eine Zeigegeste beschriebene Position innerhalb der Smart-Home Umgebung.

Bereits anhand dieser wenigen Beispielfälle wird deutlich das für die Auflösung der deiktischen Begriffe eine Zeitsynchronisation notwendig ist. Zu dem Zeitpunkt an dem die Lautäußerung „hier“ erfasst wurde muss auch für die Auswertung herangezogen werden. Bereits minimale Abweichungen können die Semantik vollständig verändern.

Fazit Obwohl das Forschungsgebiet der Spracherkennung schon über eine halbe Dekade alt ist, ist es immer noch weit davon entfernt universell Einsetzbar zu sein. In kontrollierten Umgebungen, in denen nur ein Sprecher aktiv ist und der Rauschpegel konstant und bekannt ist, sind hohe Erkennungsraten möglich. Auch für Erkenner mit einem großen Erkennungsvokabular. Leider lassen sich die Umgebungsbedingungen einer Smart-Home Umgebung nicht in diese Kategorie einordnen. Die unterschiedlichen Raumcharakteristiken sowie die ständig wechselnden Lärmbedingungen, zum Beispiel durch vorbeifahrende Autos oder laute Musik aus der Nachbarwohnung, machen es notwendig stark eingeschränkte Grammatiken zu verwenden. Sind diese heterogen genug kann auch unter den widrigen Bedingungen eine Sprachsteuerung stattfinden. Eine freie uneingeschränkte Konversation mit dem System ist dadurch allerdings nicht mehr möglich.
Deutlich wurde dies auch in den vorgestellten vergleichbaren Arbeiten. Die Projekte die auf eine freie Konversation zwischen Nutzer und System gesetzt haben, haben entweder kontrollierte Bedingungen vorausgesetzt (3.2.3) oder bei der Evaluierung auf ein Wizard-Of-OZ Ansatz zurückgreifen müssen (3.2.1).
Die eingeschränkte Vielfalt in der Kommunikation wirkt sich negativ auf die Bedienbarkeit aus. Je genauer der Anwender über die möglichen Befehle Bescheid wissen muss desto mehr Führungsarbeit muss vom System geleistet werden (vgl. Yankelovich (1996). Hier kann die Multimodalität Abhilfe leisten. Visuelle Führung auf Projektionsflächen oder Displays hilft dem Nutzer bei der Auswahl und unterstützt somit die Bedienbarkeit und Akzeptanz des Systems.

3.4.3. Midas Touch Problem

Das sogenannte Midas Touch Problem leitet sich aus vom aus der griechischen Mythologie bekannten König Midas ab. Dieser bekam von Bacchus, dem Gott des Weins, die Gabe alles was er berührte in Gold zu transformieren. Schnell musste er feststellen, dass die Gabe mehr Fluch als Segen war. Da auch Essen, Pflanzen und Verwandtschaft nach Berührung sofort zu Gold erstarrten.

Robert Jacob hat diesen Mythos in seinem Aufsatz „WHAT YOU LOOK AT IS WHAT YOU GET“ (siehe Jacob (1990)) auf die Problematik des diffusen Eingabefokus bei Vision Tracking

bezogen. Die Möglichkeit allein durch Ansehen dem Computer Befehle zu erteilen entwickelt sich von einer „gottgegebenen“ Gabe schnell zum König Midas Problem. Alles was man ansieht wird als Befehl gewertet und löst eine Aktion aus.

Im Kontext der Sprachsteuerung ergibt sich ein ähnliches Problem wenn man die Ambiguität von Spracheingaben bzw. Sprache an sich betrachtet. Alle Sprachäußerungen, die von Mikrofonen aufgefangen werden, werden analysiert und, wenn möglich, auf Befehle abgebildet. Insbesondere während Gesprächen mit anderen Menschen kann dieses Verhalten schnell störend wirken. Um dieses Problem zu lösen beziehungsweise abzumildern gibt es verschiedene Heuristiken und Ansätze:

Selektion und Semantik Die grundsätzliche Vorgehensweise zu Vermeidung von Ambiguität ist die Erkennergrammatik möglichst klein zu halten. Nur die Phrasen und Sätze laden, die im aktuellen Kontext Bewandtnis haben. Dies ist insbesondere Problematisch, da der Anwender nicht weiß, welche Phrasen aktuell und geladen sind und welche nicht. Die Herangehensweise ist also, alle Äußerungen des Anwenders zuerst einmal als Kommando zu interpretieren und dann anhand der geladenen Grammatiken zu entscheiden ob ein Befehl vorliegt. Neben der Mengen-Beschränkung der Erkennergrammatik wäre auch ein Analyse der typischen Befehle hinsichtlich ihrer grammatischen Struktur denkbar.

Push-To-Talk Um eindeutig Befehle von konversationeller Sprache unterscheiden zu können lässt sich der Spracheingabe eine Vorbedingung voranstellen. Nur wenn diese Erfüllt ist, wird die folgende Sprache als Eingabe gewertet. Denkbar wären hier eindeutige Aktionen, wie Tastendrücke oder Gesten. Weiterhin möglich wäre eine identifizierende Erkennungsphrase (Vergleich siehe (Potamitis u. a., 2003, S.3)), also einen Namen, zu vergeben der jedem Kommando vorangestellt wird oder im Befehlssatz vorkommt.

Sprachanalyse Unter Annahme, dass Nutzer von Spracherkennung versuchen möglichst deutlich zu sprechen sowie einfache Sätze zu bilden wäre es Sinnvoll nur Befehle die mit hoher Wahrscheinlichkeit erkannt wurden als Eingabe zuzulassen. Weiterhin könnten von der Lautstärke der geäußerten Sprache Rückschlüsse auf deren Intention gezogen werden. Laute deutliche Sprache ist wahrscheinlich eher ein Befehl als leise. Diese Annahmen sollten in Evaluationen untersucht werden.

Sensorfusion Durch die hohe Sensordichte in einer Smart Home Umgebung ist es möglich, durch Kombinationen der Sensorwerte eine Aktivierung zu erkennen. Zum Beispiel durch Analyse der Blickrichtung des Anwenders mit Hilfe von Eyetrackern. Und vorhergehender Festlegung von Avataren oder Fixpunkten die als „Ansprechpartner“ fungieren (siehe zum Beispiel Vertegaal u. a. (2006)).

4. Design

Das Design Kapitel beschreibt die Umsetzung der Analyseergebnisse in eine Architektur. Es werden die Module und Abläufe des Systems vorgestellt und erläutert.

4.1. Systemarchitektur

Das angestrebte System ist hinsichtlich Architekturaspekten in die Kategorie (stark) verteilte Systeme einzuordnen. Die große Vielzahl verschiedener Funktionen und Anwendungen die in einer Smart-Home Umgebung zu finden sind, machen ein monolithisches rein zentrales System nicht anstrebenswert. Das System würde unübersichtlich und nahezu unwartbar werden. Daher bietet es sich an, die einzelne Funktionalität nach ihrer Aufgabe zu trennen und in einzelnen logische Modulen zu kapseln.

Damit diese Module ohne großen Aufwand zusammenarbeiten können muss ein Grundframework geschaffen werden das den Aufruf, die Abfrage und die Verarbeitung von Funktionen und ihren Ergebnissen unterstützt.

Als grundsätzliche Vorgehensweise bietet sich hier die Aufteilung der Funktion nach Services an. Also eine sogenannte Service Oriented Architecture (SOA).

4.1.1. Einteilung

Um die unterschiedlichen Klassen von Geräten die in einer Smart-Home Umgebung vorhanden sind zu verdeutlichen, werden diese unter spezifischen Begriffen zusammengefasst.

Sensor Ein Sensor ist eine technisches Konstrukt das Daten über die aktuelle Situation sammelt und diese dem System zur Verfügung stellt. Das Gerät muss, im Gegensatz zum Interaktionsgerät, über keine direkte Anwenderschnittstelle verfügen. Es reicht aus, wenn die gesammelten Daten z.B. über den Aufruf eines Dienstes an das System übermittelt werden können.

Aktor Als Aktor wird in diesem Zusammenhang ein Gerät bezeichnet das Einfluss nehmen kann auf die aktuelle Situation. Die Auswirkungen können hierbei physikalischer bzw. kinetischer Natur sein, zum Beispiel das Öffnen und Schließen von Türen, oder aber auch rein visuell wie das darstellen von Informationen über ein Display.

Interaktionsgerät Ein Interaktionsgerät ist im Grunde genommen eine Kombination aus Aktor und Sensor. Es ermöglicht dem Anwender mit dem System zu interagieren, das bedeutet sowohl Eingaben zu machen, wie auch Ausgaben zu empfangen. Grundsätzlich ist ein Interaktionsgerät vielfach leistungsstärker als ein Sensor bzw. Aktor.

4.1.2. Kommunikation

Damit das System als Ganzes zusammenarbeiten beziehungsweise Dienste und Unterstützung anbieten kann müssen die Einzelkomponenten zu einer Einheit verbunden werden. Dies geschieht durch den Austausch von Nachrichten. Dabei variiert die Form und der Umfang der Kommunikation stark je nach Systembestandteil. Es kann unterschieden werden in synchrone und asynchrone Kommunikation.

Synchrone Kommunikation setzt voraus, dass die sich austauschenden Komponenten das gleiche Protokoll implementieren und über eine gemeinsame Schnittstelle verfügen. Als asynchrone Kommunikation wird hier der ungerichtete Austausch von Informationen über ein Blackboard bezeichnet, dass bedeutet der Empfänger muss initial nicht feststehen.

Kategorie	Synchron	Asynchron
Geschwindigkeit	Abhängig von der Performanz der Übertragung	Hoch, der Sender kann nach erstellen des Ereignis auf dem Blackboard direkt weiterarbeiten.
Latenz	Niedrig, da Aufruf direkt an den Empfänger übermittelt und von ihm bestätigt wird.	Variiert, es ist nicht sicher, wann der Empfänger das im Blackboard erstellte Ereignisses abruft
Zuverlässigkeit	Hoch, es wird sichergestellt das die Nachricht ankommt (durch Bestätigung des Empfängers)	Ob die Nachricht (das Ereignis) angekommen ist kann nur durch ein Antwortereignis ermittelt werden.
Kopplung	Eng, da beide Parteien das gleiche Transferprotokoll und gleiche Datenstrukturen haben müssen	Lose, der Empfänger muss wissen für welche Ereignisse er sich interessiert. Allerdings müssen beide Parteien das Blackboard-Protokoll unterstützen.

Tabelle 7: Unterschiede asynchroner und synchroner Kommunikation

Wie aus Tabelle 7 zu entnehmen ist, bieten beide Kommunikationsformen Vor- und Nachteile. In Fällen in denen hohe Geschwindigkeit und Zuverlässigkeit (im Sinne von der direkten Übermittlung von Resultaten) erforderlich ist, sollte die Synchrone Kommunikation zum Einsatz kommen. Asynchrone Kommunikation wiederum bietet sich dort an, wo lose Kopplung erforderlich ist, also wo zum Beispiel der Empfänger einer Nachricht noch nicht feststeht.

Da der Zuverlässige Transfer von Nachrichten zwischen den Hauptkomponenten für die Funktion des Gesamtsystems von hoher Bedeutung ist, wird dafür auf synchrone Kommunikation gesetzt. Für weitere Funktionalität die das System nicht beeinträchtigt (zum Beispiel das Übermitteln von Sensorwerten) kann asynchrone Kommunikation eingesetzt werden.

Um eine möglichst große Anzahl von Geräten mit unterschiedlichen Leistungsspektren unterstützen zu können ist es Sinnvoll, auf ein Kommunikationsprotokoll zu setzen das weit verbreitet ist und nur geringen Anspruch stellt an die Leistungsfähigkeit. Für den SOA-Bereich hat sich die Ansteuerung über das SOAP über HTTP Protokoll bewährt.

4.1.3. Synchronität

Für die Erstellung von Kontexten, multimodaler Interaktion und die Auflösung deiktischer Referenzen ist eine chronologische Synchronisierung der Systembestandteile notwendig (siehe 4.4.1, 4.4.2).

Diese Anforderung lässt sich zum Beispiel durch Einsatz des Network Time Protocol (siehe RFC 958 Mills (1985)) erreichen. Inwiefern die Genauigkeit ausreicht muss evaluiert werden. Zwecks Vereinfachung können die Systemkomponenten aber auch auf einem Physikalischen Rechner ausgeführt werden. Damit entfällt der Synchronisierungszwang, da alle Ereignisse und Aktionen auf den gleichen Referenzzeitgeber zugreifen.

4.2. Designentscheidungen

Um die Anforderungen optimal zu erfüllen müssen verschiedene Entscheidungen getroffen werden. Diese werden im Folgenden kategorisiert nach der zugehörigen Anforderung aufgelistet und erläutert.

4.2.1. Erweiterbarkeit

Um das System möglichst einfach erweitern zu können, wird eine sogenannte Aktions-Metapher eingeführt. Diese bietet eine abstrakte normierte Schnittstelle auf die einnzelnen Komponenten. Weiterhin wird durch die Aufteilung nach Funktionalität und das Ansteuern über vordefinierte Dienstprotokolle (SOAP) ein hoher Grad von Verteilung möglich. Die wirkt sich vermindert die Komplexität von Hinzufügen und Entfernen neuer Module. In einer zentralen Wissensdatenbank werden neben Meta-Informationen (z.B. Positionen der verschiedenen Komponenten), auch operationelle Daten (bekannte Orte und Kontakte oder Informationen über die Mediensammlung des Anwenders) gespeichert. Neue Komponenten können diese Meta-Informationen durchsuchen und ihren Funktionsumfang erweitern.

4.2.2. Steuerbarkeit der Smart-Home Umgebung

Damit die sowohl Aktorik wie auch Sensorik der Smart-Home Umgebung steuerbar wird, müssen sie entweder eine Schnittstelle für die Aktions-Metapher anbieten oder aber über Ereignisse kommunizieren. Um die Steuerung über Ereignisse möglichst Einfach zu gestalten wird eine Blackboard-Architektur entwickelt. Diese muss über eine eine einfach zugängliche Schnittstelle verfügen (zum Beispiel HTTP). Um Geräte einzubinden die diese Schnittstelle nicht direkt unterstützen können, müssen Adapter implementiert werden. Die Adapter kommunizieren mit dem Geräteigenen nativen Protokoll und geben die Werte an das System über die oben genannte Schnittstelle weiter.

4.2.3. Transparenz/Openness

Die für das konzipierte verteilte System benötigte Transparenz wird über verschiedene Mechanismen erreicht:

Zugriffstransparenz Die Vereinheitlichung der Zugriffe auf die Funktionen des Systems werden über die Aktions- und die Ereignis-Metapher erreicht. Wie genau diese aussieht ist Abschnitt 4.3 zu entnehmen.

Ortstransparenz Die Interpretation des Begriffs Ortstransparenz ist eher dahingehend, dass die Wahl der Ein- und Ausgabegeräte anhand der Position des Anwenders festgelegt wird.

Die Position des Anwenders in der Smart-Home Umgebung wird durch verschiedene Sensoren überwacht. Sei es aktiv über ein Indoor Positioning System oder Passiv über Lichtschranken und Kontakte. Daher ist es möglich festzustellen, welche Aktorik sich in seiner Nähe befindet. Vorausgesetzt die Aktorik ist ebenfalls einer Positionbeschreibung versehen und im System eingetragen. Wenn es sich um mobile Aktorik handelt, wie zum Beispiel Tablet-PCs, sollte die Bewegung dieser Geräte ebenfalls erfasst und im System gespeichert werden. Bezüglich der Nutzung von Sprachsteuerung bleibt festzustellen, dass sie im Gesamten Wohnkomplex verfügbar sein soll (über das Aufstellen von Mikrophonen). Damit wird die Nutzung des Systems über die Sprachmodalität ortstranparent möglich.

Relokationstransparenz Laufende Anwendungen mit denen der Anwender interagiert können das Anzeigegerät wechseln. Sie sollen dem Anwender folgen wenn er Aufenthaltsort ändert. Dafür müssen sie ihren aktuellen Stand speichern und wiederherstellen können. Das System sorgt dafür, dass wenn der Anwender den festgelegten direkten Interaktions- bzw. Sichtbereich der jeweiligen Interaktoinsgerätes verlässt, die Anwendung auf das nächstmögliche Ausgabegerät migriert wird.

4.3. Begriffe

Dieser Abschnitt erläutert die im vorherigen Abschnitt einführten Begriffe und Metaphern.

4.3.1. Ereignis

Als Ereignis wird eine Information bezeichnet die auf dem Blackboard (also bei asynchroner Kommunikation) erstellt wird. Sie kann sowohl einfache Sensorwerte wie auch komplexe Eingaben repräsentieren.

4.3.2. Aktion

Das Gesamtsystem soll nach der Prinzip der Dienstorientierung aufgebaut werden. Das bedeutet, dass Funktionalität im System durch Dienste gekapselt wird. Ein Dienst bietet ein Portfoliot an unterschiedlichen Methoden und Prozeduren an, um einen bestimmten Zweck zu erfüllen. Zum Beispiel die Steuerung von Jalousien oder die Überwachung des Kühlschrankinhaltes. Um nun simplen und einheitlichen Zugriff auf die heterogenen Aktionen zu bekommen, muss eine einheitliche Schnittstelle geschaffen werden. Aktionen sind die Abstrahierung dieser Schnittstelle.

Ein Aktion ist im System eindeutig. Sie hat eine Bezeichnung und einen Namensraum der angibt in welche Kategorie sie gehört und anhand deren die passende Gegenstelle (Interpreter) gefunden werden kann. Außerdem enthält sie eine Textuelle Beschreibung ihrer Funktion. Anhand dieser Beschreibung kann das System nach benötigter Funktionalität durchsucht werden.

Aktionsparameter Die Funktionsattribute der zu Grunde liegenden Dienstmethoden werden durch sogenannte Parameter repräsentiert. Ein Parameter hat einen, in der Aktion, eindeutigen Namen, einen Werte-Typ und einen Wert. Zusätzlich weitere Meta-Informationen wie die Beschreibung der Funktion des Parameters und ob er Optional ist oder nicht. Die Werte-Typen sind zentral festgelegt so das eine einheitliche Domäne geschaffen wird.

Das Aktionskonzept erfordert eine zentrale Verwaltungsinstanz, die als Schnittstelle für die Ausführung der Aktionen dient. Diese Verwaltungsinstanz weiß welche Aktionen verfügbar sind und wie sie interpretiert werden müssen.

4.3.3. Anwendungs-Kontext

Als Anwendungs-Kontext werden diejenigen Informationen definiert, die für das korrekte Ausführen von Aktionen notwendig sind. Das bedeutet Informationen die es dem System ermöglichen zu entscheiden, wer die Aktion durchführen soll und von wem die Aktion initiiert wurde. Diese Kontextinformationen werden nicht explizit gespeichert sondern existieren nur zum Zeitpunkt der Aktionsausführung.

4.3.4. Benutzerdefinierter Kontext

Es bleibt den im System laufenden Anwendungen unbelassen, sich eigene Kontextdefinitionen anzulegen. Zum Beispiel definiert durch eine bestimmte Folge von Ereignissen. Damit andere Komponenten ebenfalls auf diese Kontexte zugreifen können, sollte der Lebenszyklus im System durch weitere Ereignisse repräsentiert werden. Durch Verknüpfung dieser Kontexte mit Benutzerpräferenzen können unter Anderem Entscheidungen hinsichtlich der Präsentation getroffen werden.

4.4. Interaktion

Die Interaktion von Benutzer und System ist ein Kernpunkt dieser Arbeit. Die Akzeptanz des Systems ist in hohem Maße korreliert mit der Natürlichkeit und Einfachheit mit der ein Anwender das System bedienen kann. Dieser Abschnitt präsentiert die Designentscheidungen die eine immersive Steuerung des Systems gewährleisten sollen.

4.4.1. Spracherkennung

Um die im Abschnitt Problematisierung benannten Unwegsamkeiten zu kompensieren, müssen verschiedene Designentscheidungen getroffen werden. Diese werden im Folgenden vorgestellt.

Erkennungsqualität und Robustheit

Da sowohl die Robustheit wie auch die Qualität der Spracherkenner direkt mit der Usability des Systems korrellieren sollten diese maximiert werden. Dafür werden verschiedene Richtlinien festgelegt:

minimale Grammatik Die Spracherkennung sollte nur die aktuell benötigten Grammatiken geladen haben. Nur wenn diese kein Ergebnis liefern, kann entschieden werden, ob der Erkennungsvorgang mit anderen Grammatiken wiederholt oder die Äußerung als Out-Off-Vocabulary, also als kein Befehl, klassifiziert und somit ignoriert wird.

Einsatznahes Training Das Audio-Modell des Erkenners sollte mit Aufnahmen trainiert werden die dem Einsatzort entsprechen. Dies minimiert die Diskrepanz zwischen Trainings- und Testdaten und ermöglicht eine hohe Qualität. Das bedeutet, dass für jedes Mikrophon in der Smart-Home Umgebung ein oder mehrere Audiomodelle trainiert werden müssen. Anhand der Position des Anwenders kann entschieden werden, welches Mikrophon für die Spracherkennung herangezogen wird und welches Audio-Modell dafür herangezogen wird. Unterschiedliche Audio-Modelle für ein Mikrophon könnten anhand des Sprecherabstandes und dem Geräuschanteil variiert werden.

Audiovorverarbeitung Um optimale Eingangsdaten für den Erkenner zu erhalten sollten die Rohaufnahmen vorverarbeitet werden. Das bedeutet dass versucht wird verschiedene Audio-Phänomene zu kompensieren (z.B. Nachhall). Weiterhin ist es denkbar dem System bekannte Audio-Ausgaben, die während der Aufzeichnung aktiv waren (Fernseher, Radio) per Spektraler Subtraktion vom Eingangssignal abzuziehen und somit die Qualität zu steigern (siehe Abad (2007)).

Stimmenerkennung Perspektivisch wird das System mit mehreren Sprechern gleichzeitig konfrontiert werden. Um trotzdem optimale Ergebnisse zu erhalten muss erkannt werden wer eine Äußerung getätigt hat und diese an den jeweilig trainierten Erkenner weitergeleitet werden. Vorausgesetzt die erkannte Person hat Steuerungsprivilegien.

Aktivierung

Das im Abschnitt Problematisierung beschriebene Midas Touch Problem führt häufig zu fehlerhaften Eingaben. Um dies nach Möglichkeit zu vermeiden wird auf die oben beschriebene Minimierung der Erkennungsgrammatik gesetzt. Das bedeutet das der Spracherkenner alle Eingaben ignoriert, die nicht in der Grammatik vorkommen. Sollte dieses Vorgehen nicht das gewünschte Ergebnis bringen, also viele Falscheingaben produzieren, kann auf die explizite Aktivierung gesetzt werden. Die explizite Aktivierung geschieht entweder durch Äußern einer Schlüsselphrase oder Namens, oder durch die Aktivierung über eine andere Modalität (zum Beispiel die Geste des Berührens des eigenen Ohrs).

Deixis

Ereignisse die deiktische Bedeutung erhalten können müssen explizit aufgezeichnet werden. Wichtig dabei ist die Erfassung des Zeitpunkts. Verursacht durch verteilten Charakter des Systems liegt die Verantwortlichkeit für die Erfassung von Ereignissen bei den Anwendungen.

Für die Auflösung von deiktischen Begriffen sollte ein Systemmodul erstellt werden. Dieses ermöglicht die Abfrage der Wissensdatenbank hinsichtlich typischer deiktischer Ereignisse und der Zeiträume in denen sie geäußert wurden.

Als Parameter für die erfolgreiche Auflösung einer deiktischen Referenz werden mindestens folgende Informationen benötigt:

Anwender Der Nutzer der die Anfrage stellt. Für die Interpretation von lokaldeiktischen Begriffen wie „hier“, „dort“ oder personaldeiktischen Ausdrücken wie „ich“,“du“ ist es notwendig die Quelle der Aktion zu kennen. Erst dadurch wird die korrekte Interpretation ermöglicht.

Anwendung Wenn eine Anwendung Ziel der Aktion ist. Benötigt um deiktische Befehle von globaler Ebene auch in der Anwendungsdomäne nutzen zu können. Im Multimodalitäts-Beispiel werden deiktische Begriffe genutzt, die sich auf die Anwendung beziehen, nicht auf den Status des Anwenders bezüglich der Smart-Home Umgebung.

Deixis-Typ Welche Art von Deixis muss aufgelöst werden. Anhand dieses Arguments wird festgelegt, nach welchen Typen von Ereignissen in der Wissensbasis gesucht wird.

Grammatik

Die Grammatik, also die Steuerdatei für den Spracherkenner, muss die Brücke schlagen zwischen Usability und Robustheit. Es muss ein Kompromiss gefunden werden der zwar minimal ist aber trotzdem alle Begrifflichkeiten zulässt die für eine intuitive Steuerung der jeweiligen Anwendung benötigt werden. Nur so kann eine robuste Spracherkennung in der stark Rauschanfälligen Umgebung einer Wohnung ermöglicht werden.

Da die für eine Anwendung zulässigen Steuerbefehle mit ihrer Funktion korrelieren, muss die Grammatik von der Anwendung vorgegeben werden. Um auch dynamisch auf Änderungen reagieren zu können (z.B. das aktivieren von Menüpunkten in einer Anwendung) muss die Sprachverarbeitung das temporäre hinzufügen von Sprachbefehlen ermöglichen.

Es kann vorkommen, dass Sprachbefehle mit der gleichen Terminologie unterschiedliche Semantische-Bedeutung haben. Diese Ambiguität muss vom System anhand der verfügbaren Kontextinformationen aufgelöst werden. Der aktuelle Anwendungskontext erhält dabei eine höhere Priorität als andere Anwendungen. Um feststellen zu können welche Semantik gemeint war, können auch durch visuelle Analyse erhaltene Informationen herangezogen werden. Also wo liegt der Fokus des Anwenders, wohin schaut er gerade. Diese Informationen sollten durch Ereignisse im System repräsentiert werden.

4.4.2. Multimodalität

Ob ein Modul beziehungsweise eine Anwendung multimodale Steuerung zulässt ist von der jeweiligen Implementation abhängig. Die Grundkomponenten, also die Wissensbasis und der Aktions-Manager (die zentrale Aktionsverwaltung) sowie das Konzept der Ereignisse ermöglichen es der Anwendung auf alle Anwenderaktionen Zugriff zu nehmen. Je nach Ausstattung der Smart-Home Umgebung kann dies mehr oder weniger Sensordaten umfassen und damit unterschiedliche Anzahl Modalitäten.

Alle Eingabeereignisse, die nicht von lokalen Benutzeroberflächen direkt verarbeitet werden, müssen über das Ereigniskonzept an das System übermittelt werden. Anhand dieser Ereignisse können Module und Anwendungen entscheiden ob eine Eingabe für Sie interessant ist oder nicht. Ein exemplarischer Ablauf für die multimodale Steuerung einer Anwendung wird im Folgenden beschrieben:

Exemplarische Anwendung

Die Funktionalität auf die sich diese Anwendung bezieht ist aus dem Szenario entnommen. Es handelt sich um das Navigieren und Interagieren mit einer Geospatialen Applikation (wie zum Beispiel Google Maps oder Microsoft Virtual Earth). Die Aufgabennstellung besteht im Zusammenstellen einer Route vom Ausgangsort, dem Campus der HAW Hamburg, zum Zielort einem Restaurant in Nähe des Hamburger Stadtparks.

Zur Steuerung für eine solche Aufgabe bieten sich exemplarisch folgende Modalitäten an:

Multitouch Der direkte Kontakt mit der Kartenoberfläche ermöglicht markieren von Orten, navigieren und zoomen durch Ziehen und Drehgesten.

Sprache Spracheingabe ermöglicht das schnelle Navigieren und Referenzieren von/zu bekannten Orten sowie die alterierung der Ansichtsparameter (Zoom, Ausrichtung, Detailgrad).

Blick und Zeigegesten Direktes (kontaktloses) markieren durch fokussieren der gewünschten Örtlichkeit mit dem Auge oder durch Zeigen durch ausgestreckten Zeigefinger.

Insbesondere die Kombination von Touch- und Spracheingaben kann in diesem Szenario zielführend sein. Die Berührungsinteraktion ermöglicht schnelles navigieren im lokalen Kartenausschnitt währned mit Spracheingaben zu Orten navigiert werden kann, die sich außerhalb des aktuellen Sichtbereichs befinden..

Ausgangssituation

Der Anwender befindet sich auf dem Sofa und hat einen Multitouch fähiges Ausgabegerät in Reichweite (Couchtisch). Zusätzlich befindet sich ein Mikrofon sowie ein Lautsprecher in unmittelbarer Nähe. Die Anwendung wird sowohl auf der gegen überliegenden Wand als Projektion also auch auf dem Multitouchgerät dargestellt.

Ablauf

Anwender Der Anwender zeigt mit seinem Zeigefinger auf einen Punkt der Karte (die HAW in Hamburg) und spricht: „Setze den Routenstart hierher“.

System (Verarbeitung) Das System wird über Ereignisse von der Zeigegeste des Anwenders in Kenntnis gesetzt, dieses Ereignis ist parametrisiert und enthält neben der Information das es sich um eine Zeigegeste handelt auch die Koordinaten des Ereignisses.

Parallel löst der Spracherkenner die Äußerung in einen Befehl bzw. eine Aktion auf (Route.SetzeStart). Da die Koordinaten nicht direkt aus der Sprachäußerung abgeleitet werden können muss die verarbeitende Anwendung auf die Ereignisse zurückgreifen. Durch Synchronisierung des Zeitpunktes, zu dem das Wort „hierher“ gesprochen wurde, mit den Ereignissen im Kontext der Anwendung bzw. des Benutzers kann festgestellt werden welche Koordinaten für den Startpunkt verwendet werden sollen.

System (Ausgabe) Das setzen des Startpunktes wird dem Anwender durch eine kleine Flagge auf der Karte bestätigt. Zusätzlich erfolgt über ein Sprachsynthese Modul die Ausgabe „Routenstart gesetzt“. Eine kleine rote Flagge erscheint im Bereich neben der Karte untertitelt „Routenende“.

Anwender „Zeige mir den Stadtpark in Hamburg“

System (Verarbeitung) Die Spracherkennung liefert die textuelle Repräsentation der Äußerung, der jeweilige Interpreter der für die Grammatik zuständig ist die das beste Ergebnis geliefert hat, verarbeitet die Eingabe. Resultat ist die Aktion „Zeigen“ mit dem Parameter „Stadtpark, Hamburg“. Ist dieser Begriff bekannt kann bereits der Interpreter diesen in Koordinaten umsetzen, ist dies nicht der Fall muss das von der Anwendungslogik übernommen werden.

System (Ausgabe) Die Karte bewegt sich und zentriert sich auf den Hamburger Stadtpark.

Anwender „Suche italienische Restaurants“

System (Verarbeitung) Die Eingabe wird umgesetzt in eine Aktion „Suche“ mit Parameter „italienische Restaurants“. Diese Aktion wird von der Anwendung umgesetzt, also um den Bereich des aktuellen Kartenausschnitts werden die gewünschten Restaurants gesucht.

System (Ausgabe) Neben der Karte erscheint eine Liste mit den gefundenen Ergebnissen. Eine Sprachausgabe teilt dem Anwender mit das drei Ergebnisse gefunden wurden.

Anwender Der Anwender drückt zweimal kurz hintereinander auf einen Eintrag der Ergebnisliste und die Karte bewegt sich zu der gewählten Lokation. Durch Drag-And-Drop zieht der Anwender die kleine rote Flagge auf die Karte und beantwortet die Nachfrage ob eine Route erstellt werden soll mit „Ja“.

System (Verarbeitung) Das System bekommt von der Anwendung die Aktion „Route“ mit den Argumenten „Start“ und „Ziel“ jeweils in Koordinatenform und berechnet daraus die schnellste Route.

System (Ausgabe) Die Route wird auf der Karte dargestellt.

Analyse

Die zentrale Ereignisverwaltung ermöglicht es dem System multimodale Interaktion zuzulassen. Wenn für die Durchführung einer Aktion Parameter unterspezifiziert sind oder fehlen können diese aus der Wissensbasis abgeleitet werden. Dafür muss die Kombination verschiedener Ereignisse zu einer eindeutigen Eingabe, und somit zu einer Systemaktion, von der Anwendung übernommen werden. Das System kann hierbei insofern unterstützend wirken, wie es Hilfsmittel anbietet um relevante Ereignisse zu gruppieren und abzufragen. Also zum Beispiel alle Ereignisse die im Kontext der Kartenanwendung mit einem bestimmten Benutzer aufgetreten sind und in ein bestimmtes Typenraster passen.

4.5. Systemkomponenten

Dieser Abschnitt beschreibt die aus den Designüberlegungen entstandenen Hauptkomponenten die das Rahmenwerk der Anwendung bilden. Abbildung 8 zeigt zwecks Verdeutlichung der Vorgänge einen groben Überblick über die Komponenten sowie den Informationsfluss.
Aus Diagramm 8 lassen sich konkretisierte Funktionen der Hauptbestandteile entnehmen.

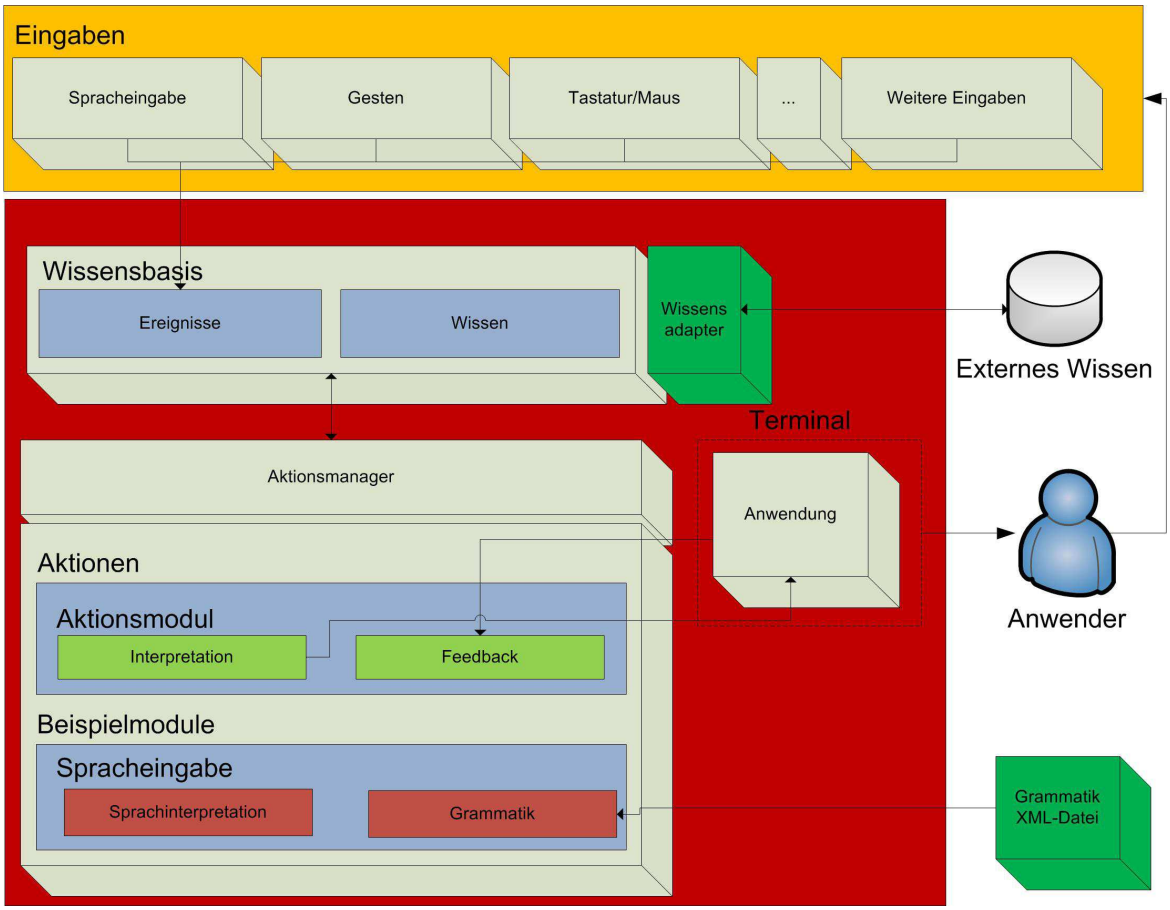

Abbildung 7: Überblick über die Systemkomponenten

4.5.1. Aktionsmanager

Der Aktionsmanager ist der zentrale Zugriffspunkt auf die Funktionen des Systems. Er ermöglicht das Durchsuchen, die Abfrage und den Aufruf von Aktionen. Eine Aktion besteht aus

einem Namen, einem Namensraum und einer Anzahl Parametern der sie eindeutig identifiziert. Das Voraussetzen eines Namensraums ermöglicht die Trennung der Verantwortlichkeit nach Domänen.

Die Aktions-Zwischenebene abstrahiert von konkreten Implementationen und legt eine verbindliche Daten-Typisierung fest.

Für eine Aktion müssen verschiedene Metadaten hinterlegt werden. Diese Metadaten beinhalten Beschreibungen der Parameter, der ausgeführten Aktion sowie deren Voraussetzungen und ermöglichen so ein Auffinden und Navigieren von bereitgestellter Funktionalität.

Diese Art der Modellierung begünstigt das Aufrufen von Aktionen mit unvollständigen Parametern. In Sprachgesteuerten Umgebungen kann es vorkommen, das ein Befehl nicht alle benötigten Informationen liefert. Zum Beispiel der Befehl: „Schalte den Backofen ein“. Dieser enthält zwar die Aktion, „schalten“ und die beiden Parameter „Gerät“ sowie „Modus“, für eine semantisch sinnvolle Interpretation fehlt jedoch der Temperaturparameter. Die Metadaten ermöglichen nun ein gezieltes Nachfragen des Systems nach dieser Information.

Der Aktionsmanager ist die Schnittstelle zu allen systemeigenen Komponenten die im Folgenden beschrieben werden. Eine Änderung der inneren Arbeitsweise diese Komponenten ist somit ohne Probleme möglich, vorausgesetzt die Parameter bleiben gleich.

Aktionsmodul Das Aktionsmodul kapselt die Funktionalität die ein Dienst anbietet. Ein Modul kann aus mehreren Komponenten bestehen mindestens jedoch aus einem Aktionsinterpreter.

Ein Aktionsinterpreter ist für die Ausführung der Aktionen zuständig. Jedes Modul das Aktionen anbietet benötigt also einen Interpreter der weiß, welche Aktionen das Modul anbietet und wie diese zu verarbeiten sind. Die Funktion eines Interpreters umfasst die zur Verfügungsstellung von Meta-Informationen über die jeweilig verfügbaren Aktionen und deren Parameter sowie das Aufrufen dieser Aktionen und Übermitteln der Resultate.

Trigger Um auf externe Ereignisse reagieren zu können, ohne dass eine zu enge Kopplung zwischen Modulen entsteht können Trigger eingesetzt werden. Ein Trigger besteht aus einer Beschreibung der auslösenden Aktion sowie einem Zeitpunkt. Ähnlich wie in einer relationalen Datenbank kann sich ein Modul so über Aktionen informieren lassen, die sein Verhalten direkt oder indirekt beeinflussen. Um zum Beispiel Audioaufnahmen an den Spracherkenner weiterzuleiten, kann das Modul das für die Steuerung der Spracherkennung einen Trigger implementieren der bei der Erstellung eines Audio-Ereignisess feuert.

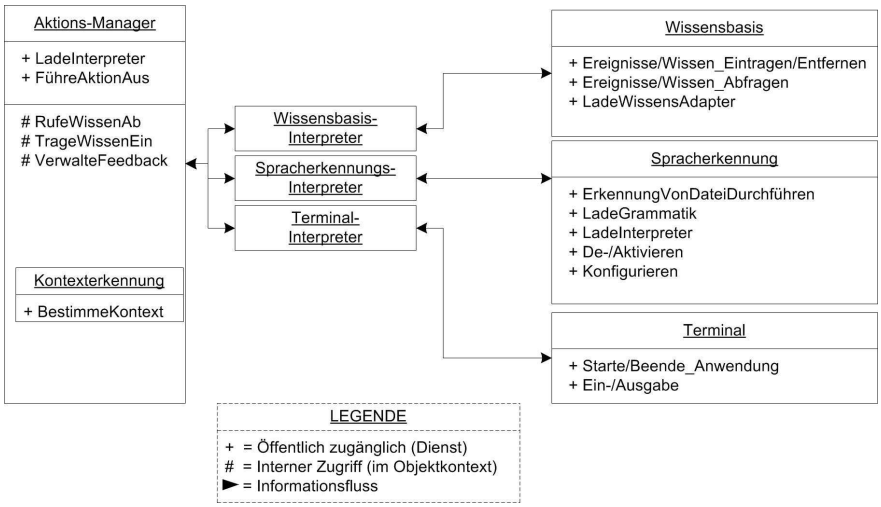

Abbildung 8: Grobübersicht über Funktionen der Systemkomponenten

4.5.2. Wissensbasis

Die Wissensbasis dient als zentrale Referenz für alle Komponenten, Ereignisse und Abläufe die sich im System befinden. Sie enthält Informationen über den aktuellen Status des Systems, die aktiven Benutzer, die angebotene Funktionalität einzelner Module sowie die verfügbaren Sensoren und Aktoren. Somit ist sie Speicher für Metadaten und Ausführungsdaten zugleich.

Die Hauptfunktion die die Dienstimplementation anbietet sind die Abfrage von Informationen, sowie das Andern und Erzeugen von Informationen. Der Zugriff auf die Daten erfolgt dabei nur über den Aktions-Manager beziehungsweise dem Wissensbasis-Aktionsinterpreter im Aktions-Manager geladen ist. Damit lassen sich alle Änderungen des Wissensbestandes an zentraler Stelle untersuchen und wenn gewünscht unterbinden oder augmentieren (durch den Einsatz eines Triggers).

Die Quellen der Informationen sind entweder die Systemmodule oder aber sogenannte Adapter. Diese dienen der Translation der Service-Aktionen (ändern, hinzufügen, löschen) auf Informationen von externen Quellen. Für die Umsetzung des Szenarios wurde zum Beispiel ein Adapter für Microsoft Outlook konzipiert. Dieser ermöglicht den Zugriff auf die Mails, Termine und Kontakte der jeweiligen Benutzer. Der Adapter übersetzt Abfragen an die Wissensbasis in Aktionen auf der jeweiligen externen Datenquelle. Dies erfüllt das bekannte Konzept der losen Kopplung.

Ob die externen Informationen und Ereignisse im System persistiert werden oder nur transienten Status erhalten ist von der jeweiligen Adapterimplementation abhängig. Damit eine

eindeutige Zuordnung der Informationen gewährleistet wird, ohne den Umweg über eine zentrale Vergabestelle zu gehen, werden als Identifikationsmerkmal GUIDs[3] eingesetzt. Diese können vom Adapter selbst vergeben werden und verhindern so einen Overhead bei der Aushandlung von Identifikationsnummern mit der Wissensbasis.

Die Informationen die in der Wissensbasis abgelegt werden sind relational organisiert. Einer Entität (identifiziert durch eine GUID), zum Beispiel einer Person, können beliebig viele Attribute und Kind-Entitäten zugeordnet werden. Dadurch ergibt sich ein hochgradig erweiterbares System. Anhand von vordefinierten Typbezeichnungen, wie zum Beispiel „/spatial/location/local“ bekommt eine Entität semantische Informationen verliehen. Anhand dieser kann die Entität gefunden werden und für bestimmte Aktionen, wie zum Beispiel das darstellen in einer Karte, nutzbar gemacht werden. Definiert sind diese Typenbezeichnungen in einer globalen Ontologie, ähnlich der Vorgehensweise im SmartKom-Projekt (siehe Engel (2006)).

[3] GUID - Globally Unique Identifier, eine 128-Bit pseudozufällige Ganzzahl die Globale Eindeutigkeit ermöglichen soll

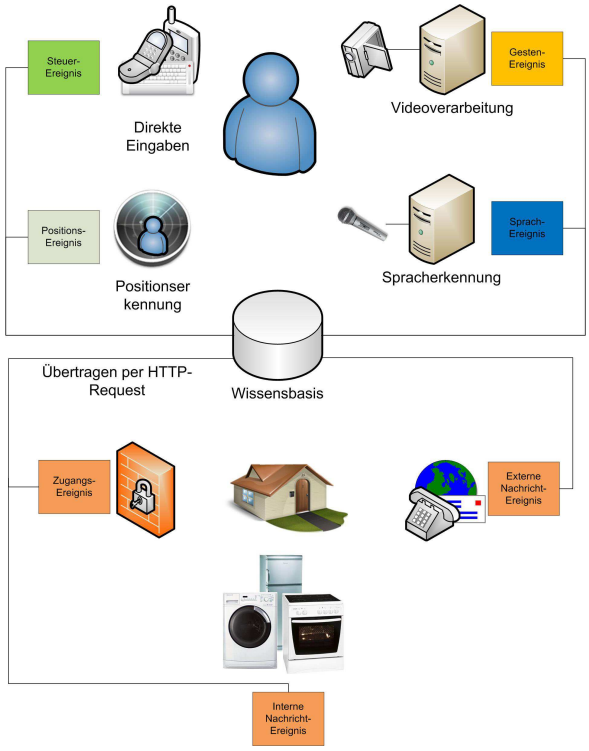

Abbildung 9: Übersicht über Ereignisse im Gesamtsystem

Ereignisse Zusätzlich zu der normalen Entitätsverwaltung unterstützt die Wissensbasis ein Ereignissystem. Ereignisse sind mit einem Verfallsdatum versehene zeitgebundene Systemvorkommnisse, siehe auch Abbildung 9. Sie sind der Grundbaustein für die Verarbeitung von multimodaler Interaktion. Ein Event besteht aus einem Typ, einem Zeitstempel, mehreren optionalen Parametern und einer Referenz auf eine Entität. Ein Ereignis bildet dynamisches Wissen ab, also Wissen das von seiner Definition her flüchtig nicht statisch ist. Die Bewegung des Anwenders in der Smart-Home Umgebung oder die Betätigung eines Schalters lösen Ereignisse aus. Durch Auswertung dieser Ereignisse oder Kombinationen derer können Aktionen ausgelöst werden (siehe a. und b. in Abbildung 10). Durch die Ereignismetapher wird eine Blackboard-Architektur (vgl. Abschnitt 2.2.2) implementiert.

4.5.3. Spracherkennung

Das Spracherkennungsmodul dient der Überführung von gesprochener Sprache in Aktionen. Dabei lädt es die zu erkennenden Phrasen aus Grammatiken die von anderen Modulen benötigt werden. Diese Grammatiken enthalten, neben den zu erkennenden Phrasen, auch die semantischen Informationen die für die Interpretation benötigt werden. Neben der statischen

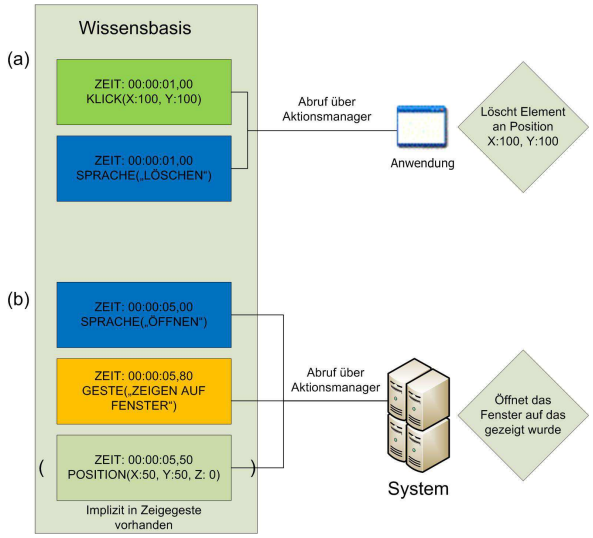

Abbildung 10: Multimodale Interaktion durch Auswertung von Ereignissen durch Anwendung (a) und System (b)

Definition der Grammatik ist es möglich auch dynamisch Einträge aus der Wissensdatenbank oder anderen Quellen abzufragen und einzugliedern (über die Referenzierung einer weiteren Grammatikdatei).

Grammatik Eine Grammatik ist das Hauptelement, sie enthält einen Namen der später für die Zuordnung zum jeweiligen Interpreter genutzt werden kann. Sie setzt sich zusammen aus verschiedenen Phrasen.

Phrase Eine Phrase stellt eine Abfolge von Wörtern und Begriffen dar die erkannt werden sollen. Eine Grammatik kann eine fast unbegrenzte[4] Anzahl an Phrasen enthalten. Eine Phrase setzt sich zusammen aus Fragmenten.

Fragment Fragmente enthalten Quantifikatoren, Semantische Indikatoren sowie den zu erkennenden Sprachausdruck (Term). Sie stellen die „Blätter" im Graphen der Grammatik dar, können aber auch weitere Unterfragmente enthalten (Composite-Pattern). Dies muss über ein Attribut gekennzeichnet werden (is_composite). Anhand der Quantifikatoren kann die Spracherkennungssoftware entscheiden ob ein Fragment optional ist (minimum = 0), wie oft es artikuliert werden muss für eine erfolgreiche Erkennung der Phrase.

[4] Durch die Leistungsgrenzen der Spracherkennungssoftware begrenzte.

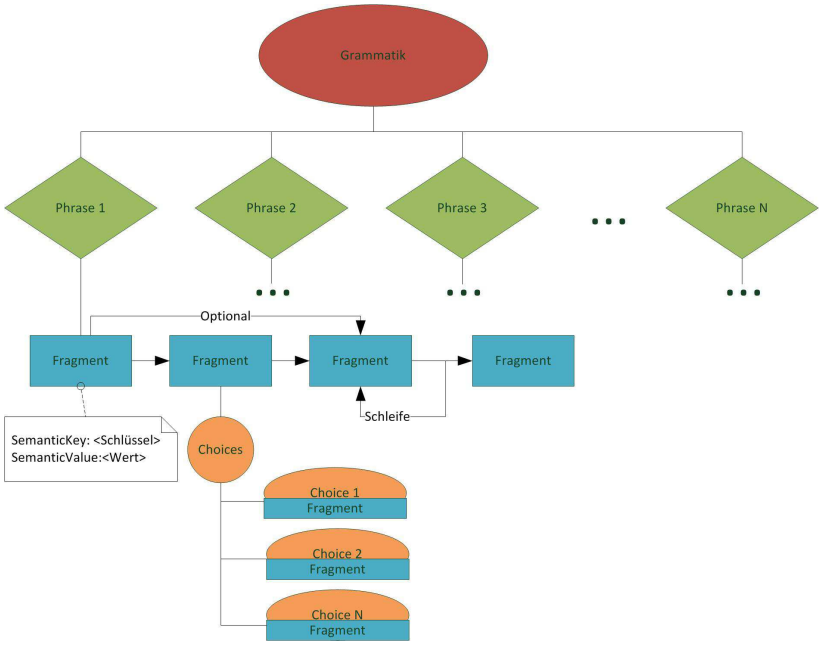

Abbildung 11: Aufbau der Sprachgrammatik

Die Semantische Annotierung ermöglicht es der Sprachinterpretation später, möglichst einfach auf Auswertungsinformationen zuzugreifen. Sie ermöglicht das Festlegen eines Schlüssel- und Wertepaares. Zum Beispiel um den Grundsätzliche Aktionsnamen festzulegen, der im Interpreter aufgerufen werden soll[5]. Wenn kein Semantischer Wert festgelegt wird, dann wird versucht diesen aus den Kindfragmenten zu extrahieren. Wenn keine Unterelemente vorhanden sind, nimmt der Term des Fragments den Platz des Semantischen Werts ein. Ist der Term-Wert leer, nimmt das System ein Diktat an, das bedeutet ein leerer Term ist eine Wildcard für eine beliebige Phrase[6].

Fragmente die sich auf einer Ebene befinden werden als Kette abgearbeitet. Um eine Auswahl verschiedener Fragmente zu ermöglichen wird ein Choices-Element deklariert.

Choices Ein Choices element bildet ein Entweder-Oder Konstrukt ab und ermöglicht eine selektive Auswahl von Fragmenten. Es besteht aus Choice-Elementen die wiederum Fragmente enthalten. Choices können aus externen Quellen nachgeladen werden. Zum Beispiel aus der Wissenbasis. Das ermöglicht es, eine Grammatik möglichst aktuell zu halten. Zum

[5] z.B. semanticKey = „command“, semanticValue=“Application.Navigation.GoogleMaps.show“ spezifziert das der Interpreter für den Namensraum „Application.Navigation.GoogleMaps“ aufgerufen werden soll um das Kommando „show“ zu verarbeiten

[6] Dies gilt nur für Fragmente die nicht als Elternfragmente gekennzeichnet sind

Beispiel können Systembekannte Orte referenziert werden, diese werden dann beim Laden der Grammatik nachgeladen und stehen als Auswahl zur Verfügung.

Ablauf Das Modul wird durch einen Aktions-Interpreter der standardmäßig geladen ist gesteuert. Die Aktivierung erfolgt über einen registrierten Trigger, der auf Audio-Sprachereignisse reagiert. Ein Sprachereignis enthält den Pfad zu einer zu interpretierenden Audiodatei und die jeweilige Quelle (also die Identifikation des Mikrophons).

Unabhängig von der Interpretation der Sprache kann vorher eine Audiovorverarbeitung statt finden die zum Beispiel Lärm reduziert oder eine Stimmenerkennung durchführt.

Eine Disambiguierung von gleichen Phrasen die in verschiedenen Grammatiken auftauchen und unterschiedliche Interpretationen zulassen findet nicht in diesem Modul statt. Es werden Aktionspakete geschnürt die alle erkannten Aktionen enthalten. Das weitere Vorgehen wird auf Aktionsinterpreter-Seite entschieden. Das bedeutet, je nach vorhandenem Kontext und agierendem Benutzer wird festgestellt welche Aktion am wahrscheinlichsten durchgeführt werden soll.

Sprachinterpreter Der Sprachinterpreter ist ein Bestandteil des Spracherkennungsmoduls. Er dient der Übersetzung der erkannten Wortfolge (bzw. der Semantik dieser Folge) in Aktionen. Er ist nur notwendig wenn eine simple Übersetzung anhand der annotierten Phrasen nicht möglich oder wünschenswert ist. Besonders für komplexere Befehle und Abfragen sollte also ein eigener Interpreter erstellt werden. Anwendungen die eigene Grammatiken spezifizieren müssen auch einen Sprachinterpreter schreiben.

Deixis Um deiktische Ausdrücke zu kennzeichnen, kann im jeweiligen Fragment der Semantische Wert auf eine bestimmte Konstante gesetzt werden. Wie diese Konstante aussieht muss vom jeweiligen Sprachinterpretationsmodul festgelegt werden.

Die aus der Sprachäußerung resultierende Aktion enthält, neben den festgelegten Parametern, auch Metadaten, die über die Zeitliche Abfolge der geäußerten Worte Auskunft gibt. Anhand dieser Metadaten ist es im Aktionsmanager bzw. im Aktionsinterpretationsmodul möglich eine Korrelation zwischen Systemereignissen und dem deiktischen Ausdruck herzustellen. Für eine exemplarische Beschreibung der Abläufe siehe Abschnitt 4.7.4.

Dynamische Anpassungen Der normale Ablauf sieht nur ein einmaliges Laden der Sprach-Grammatik bei Start der Anwendung[7] vor. Für Anwendungen deren Menü durch

[7] Da sich die Wissensbasis häufig ändert können Anwendungen ein erneutes Laden ihrer Grammatik anfordern. Um zum Beispiel Orte, die neu hinzugekommen sind, auch Auswählbar zu machen.

Spracheingaben gesteuert werden kann würde das bedeuten, dass immer alle Menüpunkte anwählbar wären. Besonders bei Menüpunkten die identische Aktivierungsphrasen haben aber unterschiedliche Ereignisse in der Anwendung auslösen ist dies nicht wünschenswert. Daher bietet die Sprachkomponente eine Möglichkeit des dynamischen Nachladens und Entfernens von temporären Grammatiken. Diese temporären Grammatiken sind meist kleine wenige Wörter und Sätze umfassende Vorgaben, die zusätzlich zu der Hauptgrammatik bei Bedarf geladen werden können. Somit ist es möglich für jede Menüebene einer Anwendung temporäre Grammatiken zu spezifizieren. Grundbefehle die global während der Ausführung der Anwendung zur Verfügung stehen bleiben in der Initialgrammatik während seltene Befehle ausgelagert werden und nur wenn es nötig ist geladen werden.

4.5.4. Anwendung

Die Anwendung stellt eine logische Kapselung verschiedener Funktionen dar. Sie bietet Aktionen an, verknüpft diese zum Beispiel mit Sprachbefehlen und reagiert auf Eingaben des Anwenders. Weiterhin kann sie eine standardisierte Schnittstelle für Feedback anbieten. Das Feedback kann sowohl uni- wie auch bidirektional gestaltet werden. Es ermöglicht die Abfrage von Wissensdaten zum Zwecke der Weiterverarbeitung (z.B. Abfrage von Fotos in einem bestimmten geographischen Raum oder aktuelle Systemnachrichten). In welcher Form das Feedback erfolgt ist Anwendungsabhängig. Möglich wären zum Beispiel für Web-Anwendungen der Austausch von Nachrichten per polling einer Datei oder aber für HTML5 fähige Browser WebSockets (siehe Hickson).

Eine Anwendung kann eine grafische Oberfläche anbieten sowie weitere Funktionen die nicht vom System über Aktionen genutzt werden können. Die Oberfläche wird bei Bedarf auf einem geeigneten Ausgabegerät dargestellt. Webanwendungen werden über den Standardbrowser geöffnet während interpretierte oder vorkompilierte Anwendung erst auf das Ausgabegerät übertragen und dann gestartet werden. Diese Aufgabe erledigt ein Clientprogramm das auf allen Ausgabegeräten eingesetzt wird, das Terminal.

4.5.5. Terminal

Die Terminalsoftware exponiert die Funktionalität von Hardware-Geräten für das System. Er stellt fest, welche Eigenschaften das Gerät besitzt und welche Ein- und Ausgabefähigkeiten vorhanden sind. Diese werden dem System gemeldet (in der Wissensbasis gespeichert) und stehen dann zur Verfügung.

Weiterhin ermöglicht ein Terminal das Starten/Beenden und Ausführen von Anwendungen. Diese werden in zwei Typen unterschieden:

Web-Anwendung Zentral auf einem Webserver laufende Web-Anwendungen werden über den Aufruf eines Browsers auf dem Terminalhost gestartet und können über einen REST-Service, WebSockets oder serverseitige Technologie Feedback an das System bzw. den Aktions-Interpreter liefern.

Lokale-Anwendung Anwendungen die nicht im Browser ausgeführt werden, sondern auf dem jeweiligen Zielsystem laufen müssen, können auch unterstützt werden. Das Terminal kopiert diese von einem zentralen Datenspeicher und führt sie mit den übergebenen Parametern aus.

Der Vorteil von lokalen Anwendungen ist ihr Leistungsumfang. Sie können die gesamte Leistung der Terminal-Host-Hardware nutzen. Web-Anwendungen hingegen sind beschränkt auf die Leistung der Laufzeitumgebung des jeweiligen Browsers. Der Vorteil den sie bieten ist die Kompatibilität, sie sind lauffähig auf allen Geräten die einen modernen Web-Browser beinhalten (z.B. Smart-Phones, Tablets und Fernseher).

4.5.6. Modul

Ein Modul ist ein Sammelbegriff für verschiedene Komponenten. Es kann bestehen aus den bereits beschriebenen Komponenten: Wissensbasis-Adapter, Sprach-Interpreter mindestens jedoch aus einem Aktionsinterpreter.

Beispielmodul „Routenplanung“

Zur Verdeutlichung des Modulkonzepts sollen die typischen Komponenten eines Moduls vorgestellt werden. Das Modul dient dem planen von Routen, finden von Wegpunkten, navigieren und lokaler Branchensuche. Sie wird gebildet aus den folgenden Bestandteilen:

Aktionsinterpreter Stellt die Schnittstelle zu den systemweit angebotenen Aktionen dar. Bietet diese mit den jeweiligen Parametern an. Zum Beispiel die Aktion „Zeigen“, diese stellt einen Ort auf der Karte dar. Diese kann entweder durch Koordinaten oder abstrakt durch Beschreibung spezifiziert werden.

Anwendung Eine Webbasierte Oberfläche die dem Anwender über ein visuelles Anzeigegerät spatiale/geographische Informationen präsentiert. Gesteuert wird die Anwendung entweder direkt über den Browser des anzeigenden Terminals oder über Befehle in einer Befehlsdatei.

Feedback Das Feedbackmodul bietet zum einen Zugriff auf die Wissensdatenbank um zum Beispiel zu den gezeigten Orten mit GPS versehene Fotos abzufragen oder Adressen von Personen aus der Kontaktdatenbank die sich in der Nähe befinden nachzuladen.

Zum zweiten ermöglicht das Modul das Eintragen von Ereignissen, wie die Auswahl einer Örtlichkeit durch Mausklick (das Ereignis enthält dann sowohl die Maus- wie auch die geospatialen Koordinaten). Diese Ereignisse werden unter anderem benötigt um lokal deiktische Phrasen aufzulösen.

Sprachinterpreter Erstellt die Grammatik die für die Steuerung der Anwendung genutzt werden kann. Mögliche Befehle wären „Zeige mir den Hauptbahnhof in Hamburg“, „Suche italienische Restaurants“ oder aber „Starte eine Route von hier... nach hier“. Die Phrasen werden hierbei zum Teil aus der Wissensdatenbank geladen, so dass bekannte Orte direkt mit Koordinaten versehen sind. Unbekannte Lokalitäten müssen von der Anwendung aufgelöst werden.

4.6. Systemmodule

Für die rudimentäre Funktion des Gesamtsystems werden bestimmte Module vorausgesetzt. Diese ermöglichen die Interaktion mit dem Anwender und den verschiedenen Komponenten. Diese Module werden im Folgenden vorgestellt.

4.6.1. Sensorschnittstelle

Damit Sensoren, die nicht über ein eigenes Modul bzw. eine Anwendung verfügen, ihre Informationen an das System übermitteln können, wird ein generisches Sensorschnittstelle implementiert. Dies ermöglicht ein standardisiertes Einfügen von Ereignisinformationen per SOAP bzw. REST Protokoll. Dadurch wird eine schnelle Integration von Sensoren in das System ermöglicht.

4.6.2. Kontexterkennung

Als Kontexterkennung wird das Ableiten eine Kontextentität aus verschiedenen Ereignissen und Systemzuständen bezeichnet. Diese Aufgabe wird nur zum Teil vom System übernommen. Das System versucht die für die Ausführung einer Aktion notwendigen Kontextinformationen, wie Ursprung und Ziel der Aktion, zu ermitteln um sie dem jeweiligen Aktionsinterpreter zur Verfügung zu stellen. Es wird also nur der Anwendungskontext bestimmt.

Zusätzlich zu den vom System bereitgestellten Kontextinformationen, können Anwendungen die im System ausgeführt werden, durch Auswerten verschiedener Sensorwerte für sich selbst Kontextdefinitionen schaffen. Dazu werden aus Low-Level-Ereignissen wie jenen Sensorwerten, über Regeln High-Level Informationen abgeleitet (vgl. Sokollek (2010). Aus den empfangen Ereignissen:

- Position des Anwenders entspricht Küche
- Backofen ist eingeschaltet
- Kühlschrank wird geöffnet/geschlossen und Sachen werden entnommen(siehe Hollatz (2008))

Kann zum Beispiel geschlossen werden, das der Anwender mit der Zubereitung von Speisen beschäftigt ist. Daraus resultierend könnten Ereignisse die den Start eines neuen Benutzerdefinierten Kontextes bekannt geben ausgelöst werden. Die Anwendungen im System bzw. die Module (zum Beispiel das Ausgabe-Modul) haben nun die Möglichkeit auf diese Information zu reagieren. Eine typische Anwendung wäre das Aktivieren kontaktloser Eingabemethoden, da mit hoher Wahrscheinlichkeit die Hände des Anwenders verschmutzt sind[8]. Da im Vorhinein nicht bekannt ist, welche Kontexte für die jeweiligen Anwendung von Bedeutung sind, ermöglicht diese Einteilung der Verantwortlichkeit in die Anwendungsdomäne ein Maximum an Flexibilität.

4.6.3. Benutzererkennung

Das Benutzerkennungsmodul verarbeitet Ereignisse und bietet Funktionalität zur Feststellung des aktuell aktiven Anwenders an. Wie in den Anforderungen beschrieben ist die Zuordnung von Aktionen zu einem Benutzer essentiell wichtig für die Kontexterkennung.

Dieses Modul kapselt verschiedene Methoden zur Erkennung, welcher Anwender tätig geworden ist und stellt somit die Basis für das Kontext-Erkennungsmodul.[9]

4.6.4. Spracherkennung

Für die Kommunikation mit dem Spracherkenner wird ein Aktionsmodul benötigt. Da die Spracherkennung im Mittelpunkt dieser Arbeit steht, sollte dieses Modul direkt verfügbar sein. Das Modul ermöglicht das Steuern der Spracherkennung, also das Laden, Aktivieren und Deaktivieren der Erkennung. Es lauscht über einen Trigger(siehe 4.5.1) auf Ereignisse die die Sprachsteuerung betreffen und verarbeitet diese wenn nötig, das heißt leitet Sie an die Spracherkennungssoftware weiter.

[8] Anhand der entnommenen Zutaten könnte festgestellt werden ob und wie wahrscheinlich eine Verschmutzung ist. Der Kontext könnte nach Erkennung der Tätigkeit „Hände waschen“ wieder deaktiviert werden (z.B. durch längeres Öffnen des Wasserhahns und von der Videoanalyse erkannten leeren Händen)

[9] In der späteren Realisierung durch die gemachten Einschränkungen nur ein aktiver Benutzer im System vorhanden sein, daher dient dieses Modul der Vereinfachung von späteren Erweiterungen.

4.6.5. Anwendungsverwaltung

Das Anwendungsverwaltungsmodul ist für das Starten und das Beenden von Anwendungen zuständig. Es verwaltet die laufenden Instanzen und kümmert sich, bei Bedarf um den Transfer einer Instanz an ein anderes Terminal.

Das Verwaltungsmodul überprüft beim Starten einer Anwendung den Ort des Anwenders der den Start angefordert hat und gleicht diese Position mit den Orten aktuell verfügbarer Terminals ab. Das jeweilig nächste Terminal wird dann instruiert die Anwendung zu starten. Die gestartet Instanz erhält eine eindeutige Identifikation und kann somit von anderen Modulen referenziert werden. Sollte kein Terminal verfügbar bzw. die Entfernung zum nächsten Terminal zu groß sein wird dies dem Anwender über das Ausgabemodul mitgeteilt.

Die Anwendungsverwaltung lädt bei Bedarf die Aktionsinterpreter der zu startenden Anwendungen in den Aktionsmanager, sowie etwaig vorhandene Grammatiken und Sprachinterpreter in die Spracherkennung.

4.6.6. Ausgabe

Das Ausgabemodul ist für die Ausgabe von Informationen an den Benutzer zuständig. Es wertet die Wissensbasis aus hinsichtlich verfügbarer Ausgabegeräte, Benutzerpräferenzen und dem aktuellen Kontext des Anwenders. Anhand dieser Auswertung entscheidet es dann welche Modalität für die Ausgabe an den Anwender genutzt werden soll und durch welches Terminal dies ausgeführt werden soll.

4.6.7. Dialog

Um einfach vom Anwender Informationen abfragen zu können wird ein Dialog-Modul benötigt. Diese Dialogmodul ermöglicht ein spezifizieren von Abfolgen von Anfragen und Antworten sowie das zuweisen dieser Antworten zu Parametern. Das Dialogmodul kümmert sich um das voranschreiten innerhalb der vorgegebenen Struktur und verwaltet die Resultate. Ein Dialog ist Benutzer-System-Modal, das bedeutet wenn ein Dialog aktiv ist, sind andere Wortäußerungen nicht zugelassen. Um wieder in den interaktiven Modus zu gelangen kann der Anwender einen Abbruch des Dialogs anfordern.

Reine Sprachdialoge sind mit Kommandozeilen Anwendungen zu vergleichen, das bedeutet dem Anwender wird nicht direkt vermittelt, welche Möglichkeiten er hat. Um diese Schwierigkeit zu kompensieren gibt es verschiedene Ansätze.

- Explizite Nennung der Optionen in Fragestellung

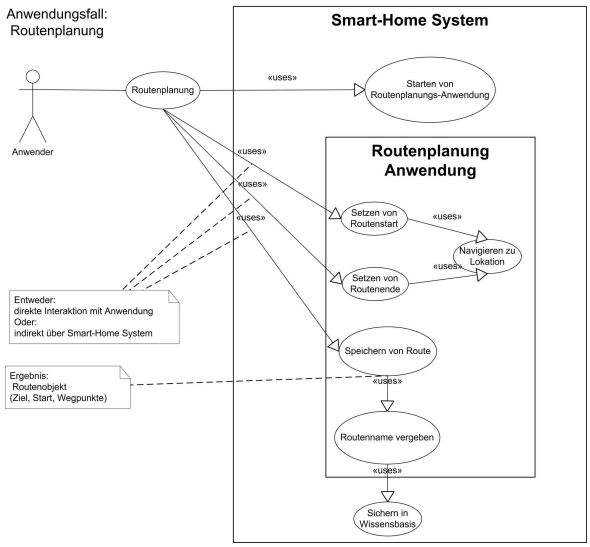

Abbildung 12: Anwendungsfall Routenplanung

- Hilfstexte die abgerufen werden können
- Nennung der Optionen bei unbekannter Eingabe

Die Verantwortung für die Vorgehensweise liegt also beim Anwendungsentwickler. Das Modul bietet nur die Verwaltung der jeweiligen Ein- und Ausgaben an.

4.7. Systemabläufe

Um zu verdeutlichen, wie die einzelnen Systembestandteile zusammenarbeiten, werden in diesem Abschnitt verschiedene graphische Veranschaulichungen aufgeführt und erläutert. Die Diagramme beschreiben jeweils einen Teilaspekt des Systems. Das Szenario dient hierbei als Basis für die beschriebene Funktionalität.

4.7.1. Anwendungsfall

Das Use-Case Diagramm beschreibt die Funktion Routenplanung. Die Funktion beschreibt das Finden einer Route von einem Startpunkt zu einem Zielpunkt. Die Funktion ist integriert in einer Anwendung, die Funktionalität für das darstellen und verknüpfen von Orten mit Daten aus der Wissensbasis bietet.

Der Anwender ruft die Funktion Routenplanung auf. Da er sich in einer Smart-Home Umgebung befindet, die eine Vielzahl von Ein- und Ausgabemodalitäten zur Bedienung aufweist,

kann er dies auf verschiedenste Art und Weise tun. Der Anwender entscheidet sich für das Starten per Sprachbefehl: „Starte Routenplanung“. Nachdem das System die Anwendung gestartet hat und die Ausgabe für den Anwender sichtbar ist (für nähere Informationen hierzu siehe Abschnitt 4.7.2) muss er den Startpunkt der Route angeben. Dazu navigiert er in der Anwendung zu der gewünschten Stelle und markiert Sie. Das kann je nach Position und Status des Anwenders erneut durch eine Spracheingabe („Zeige mir die Hamburger Straße in Hamburg“) oder zum Beispiel durch manuelles Navigieren per Berührung erfolgen. Nachdem der Startpunkt gesetzt ist kann der Anwender nun einen Zielpunkt setzen. Sobald die Routenparameter vollständig sind berechnet, die Anwendung die Wegpunkte der Route. Um später erneut auf die Route zugreifen zu können bietet das System die Möglichkeit, die Route unter einem Namen abzulegen.

4.7.2. Ablauf Routenplanung

Das Sequenz-Diagramm 13 beschreibt exemplarisch die Abläufe die beim Starten der Anwendung „Routenplanung“ aufgerufen werden. Die einzelnen Komponenten entsprechen den im vorherigen Abschnitt erläuterten Systembestandteilen (4.5).

Erläuterung

Nachdem das System und seine Bestandteile gestartet wurden beginnt die Initialisierung. Die Komponenten kennen sich über zentral abgelegte Konfigurationsdateien. Der Aktionsmanager lädt initial die Anwendungen die für die Grundfunktionalität in der Smart-Home Umgebung benötigt werden. Das ist zum Beispiel das Starten und Beenden von Anwendungen oder das Steuern von Geräten. Diese Anwendungen instruieren das Spracherkennungsmodul, bei Bedarf, Befehls-Grammatiken zu laden. Im vorliegenden Fall wären dies Befehle für das Starten der verfügbaren Anwendungen.

Wenn das System bereit ist kann der Anwender damit interagieren. Der Sprachbefehl „Starte Routenplanung“ wird vom Spracherkennungsmodul erkannt und anhand der geladenen Grammatik in eine Aktion aufgelöst. Die Aktion wird an den Aktionsmanager zur Ausführung übergeben. Der Aktionsmanager prüft ob das benötigte Modul für die Interpretation der Aktion geladen ist und lädt dieses bei Bedarf nach. Das starten einer Anwendung ist kein trivialer Vorgang, er beinhaltet eine Anzahl von Prüfungen und Unterprozessen.

Terminals

Eine Anwendung mit Benutzeroberfläche benötigt ein Ausgabeterminal. Wie bereits im Abschnitt 4.5 beschrieben sind Terminals Software die Funktionalität von komplexerer Hard-

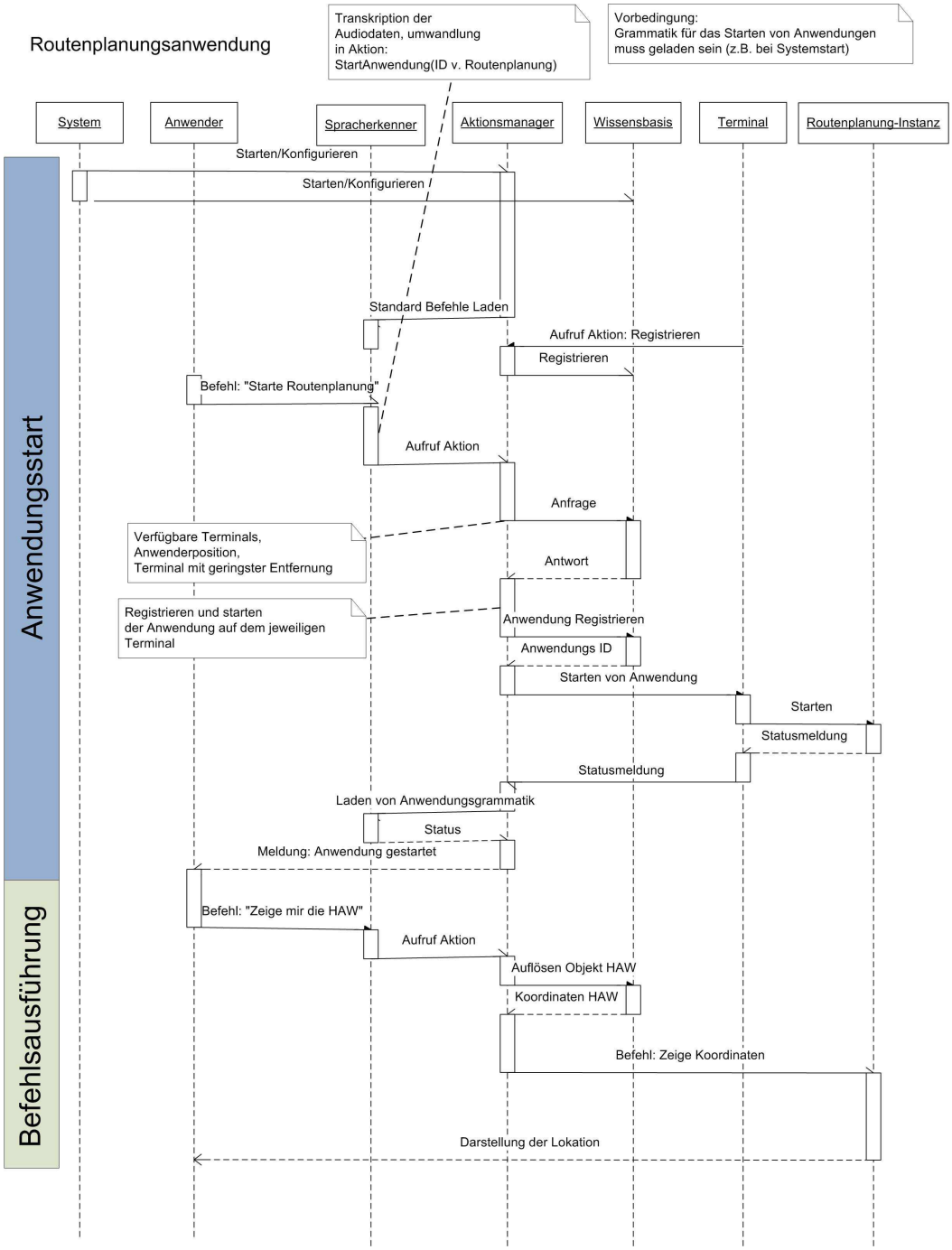

Abbildung 13: Ablauf der Anwendung Routenplanung

ware (z.B. PCs oder Embedded Systeme) exponieren und zugreifbar machen. Ein Terminal wird automatisch bei Start Hardware geladen und ausgeführt. Es meldet sich beim Aktionsmanager an und teilt ihm seine Adresse und Position sowie die verfügbare Funktionalität des Systems mit (zum Beispiel Video- und Audioausgabe oder angeschlossene Peripheriegeräte).

Anwendungsstart

Um eine Anwendung auf einem Terminal zu starten, muss zunächst einmal festgestellt werden, auf welchem Terminal die Anwendung laufen soll. Dies geschieht über den Abgleich der Position des Anwenders mit denen der Terminals. Das Terminal mit der geringsten Entfernung, wenn es alle Voraussetzungen erfüllt, wird instruiert die Anwendung zu laden und zu starten. Je nachdem ob es sich um eine zentral gehostete Anwendung handelt oder um Software die lokal auf dem Terminal läuft ändert sich die Verfahrensweise. Zentral gehostete Anwendungen werden in einem Browser geöffnet. Die Adresse (URL) der Anwendung wird annotiert mit einer Anwendungs-Identifikationsnummer. Das ermöglicht das mehrere Instanzen gleichzeitig störungsfrei laufen können. Für lokale Software wird die Anwendung von einem zentralen Datenspeicher kopiert und dann als Prozess auf dem Terminal ausgeführt. Wiederum wird als Start-Parameter die Identifikationsnummer übergeben.

Soll die Anwendung auch per Spracheingabe steuerbar sein, kann sie eine Befehlsgrammatik spezifizieren die vom Spracherkenner geladen werden soll. Wenn eine Grammatik existiert muss auch ein Interpretationsmodul existieren. Dieses Modul ist für Übersetzung der erkannten Semantik in Aktionen zuständig.

Durch das Erscheinen der Benutzeroberfläche auf dem Terminal weiß der Benutzer, dass die Anwendung geladen ist und auf Eingaben wartet. Zusätzlich kann über eine weitere Ausgabe explizit darauf hingewiesen werden, welche Anwendung wo gestartet wurde (zum Beispiel per Text-To-Speech).

Befehl „Zeige mir die HAW“

In der oben genannten Befehlsgrammatik ist auch die Phrase „Zeige [mir] [der|die|das](ORT|DIKTAT)“ vorhanden. Eine Beispielgrammatik für die Routenplanung ist im Anhang verfügbar, siehe 1. Zur Verdeutlichung der Struktur ist die Grammatik in Auszügen in Abbildung 14 dargestellt
Dem Befehl „Zeige“ ist intern eine Aktion zugeordnet (Application.Navigation.GoogleMaps.show). Anhand dieser Zuordnung kann der Spracherkenner entscheiden, welches Interpretationsmodul verwendet werden soll. In diesem Fall das Modul das für den Namensraum „Application.Navigation.GoogleMaps“ zuständig ist.

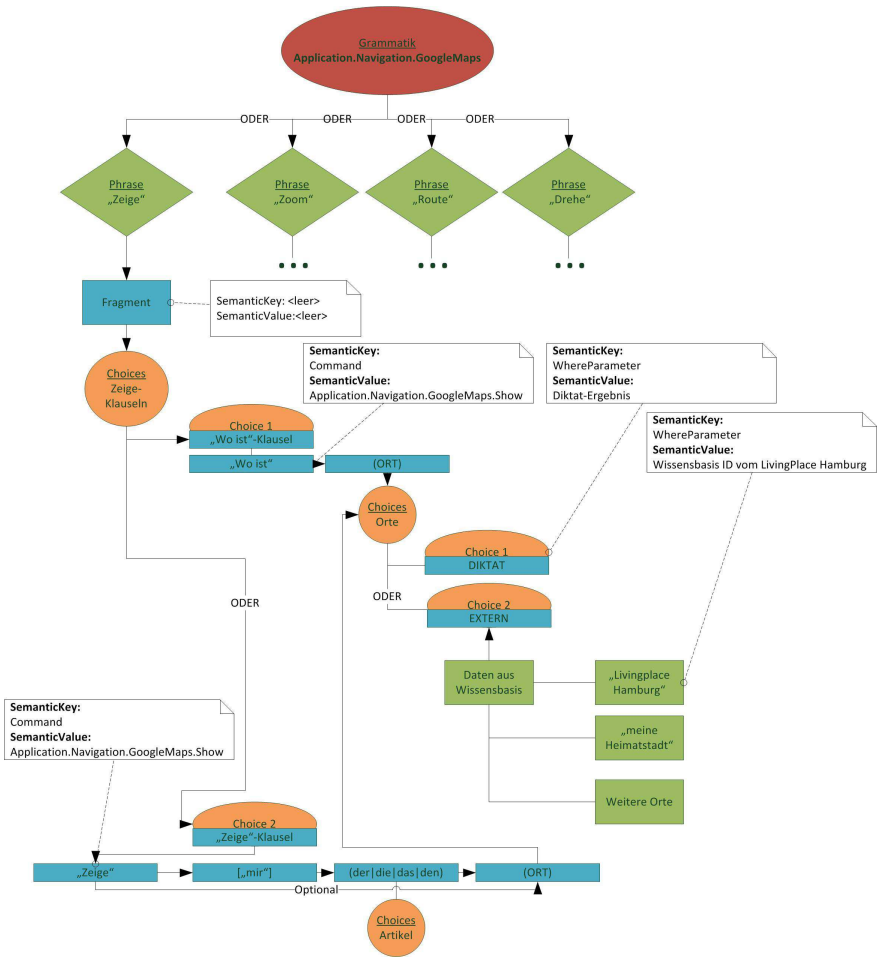

Abbildung 14: Beispielgrammatik

Die Klausel ORT steht hierbei als Platzhalter. Sie wird beim Laden der Grammatik durch die Wissensbasis aufgelöst in alle Entitäten, die als Lokation gekennzeichnet sind. Die ORT werden dann jeweils über ihren eindeutigen Identifikationsschlüssel referenziert. DieWissensbasis enthält zu diesen Orten dann Adressdaten oder auch globale Koordinaten. Die Klausel DIKTAT steht stellvertretend für das Akzeptieren von nicht vorher bekannten Wörtern. Bei Verwendung der freien Wortwahl muss das Gesagte von der Anwendung interpretiert werden. Das bedeutet in diesem Fall, in eine darstellbare Lokation aufgelöst werden.

Der Aktionsmanager verarbeitet die resultierende Aktion. Zu diesem Zeitpunkt steht noch nicht fest für welchen Anwendungskontext die Aktion bestimmt ist. Das Modul, das für die Interpretation der Aktion zuständig ist, muss also feststellen in welcher Anwendungs-Instanz die Aktion ausgeführt werden soll. Das geschieht durch einen Abgleich der aktiven Kontexte.

Kontexterkennung

Für die Kontexterkennung wird der handelnde Anwender benötigt. Anhand dessen kann der Raum der möglichen Anwendungs-Kontexte auf diejenigen mit seiner Beteiligung eingeschränkt werden. Sollten für den erkannten Anwender mehrere Anwendungs-Kontexte mit der jeweiligen Anwendungs-Identifikation existieren muss weiter unterschieden werden. Da Anwendungen Terminalmodal sind, also nur eine Anwendung zur Zeit auf einem Terminal aktiv sein kann, kann der Anwendungs-Kontext anhand der Position des Terminals festgelegt werden. Dafür muss die Position an der der Sprachbefehl getätigt wurde, abgeglichen werden mit den Positionen der in Frage kommenden Terminals. Probleme ergeben sich allerdings bei äquidistanten Terminals. In diesen Fällen muss der Anwender über einen Dialog festlegen, für welche Anwendung der Befehl gelten soll.

Aktionsverarbeitung

Die Aktion wird vom Aktionsmanager an das jeweilige Modul weitergeleitet inklusive der zuvor erkannten Kontextinformationen. Je nach Anwendungstyp kann das Modul dann auf unterschiedliche Art und Weise die Aktion veranlassen. Bei zentralen Web-Anwendungen die keine direkte Kommunikation unterstützen werden die Befehle in einer Kommandodatei abgelegt. Die Web-Anwendung pollt diese Datei und prüft ob Änderungen vorgenommen wurden. Lokale, also Terminal gebundene Anwendungen, sind selbst für die Kommunikation mit dem Aktionsinterpreter Modul zuständig. Das kann zum Beispiel durch einen von der Anwendungsinstanz gehosteten Webservice geschehen. Die Netzwerkadressen können aus den Kontextinformationen extrahiert werden.

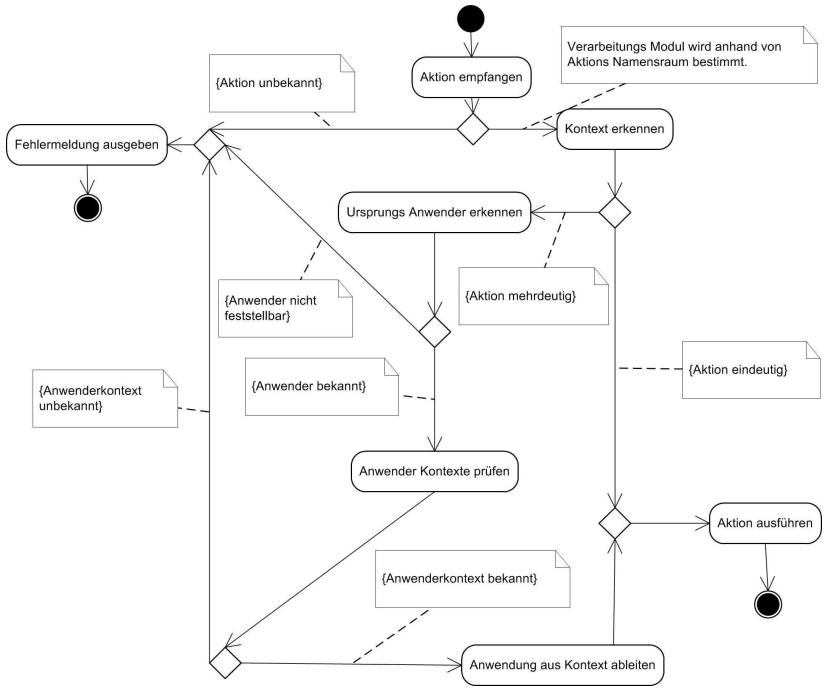

Abbildung 15: Aktivitätsdiagramm zur Kontexterkennung

4.7.3. Ablauf Multimodalität durch Ereignisverarbeitung

Im Sequenzdiagramm 16 ist beispielhaft dargestellt, wie durch die Auswertung verschiedener Ereignisse multimodale Eingaben entstehen. Die Anwendung präsentiert dem Anwender eine Auswahl an Menüpunkten. Neben der Möglichkeit dem Menüpunkt durch direkte Interaktion über Maus- und Tastatureingaben zu aktivieren bietet sich auch eine multimodale Variante: Durch zeigen auf einen Menüpunkt und gleichzeitigem äußern des Sprachbefehls „Öffnen“.

Die Zeigegeste vom Anwender wird über eine Videoanalyse registriert und in ein Ereignis umgewandelt. In diesem Ereignis sind die Art der Geste sowie ihre Parameter verzeichnet. In diesem Fall der Vektor der Zeigegeste, der Anwender von dem die Geste ausgeht sowie der Zeitpunkt zu dem sie erfolgte.

Die Spracherkennung erkennt den aufgenommenen Sprachbefehl „Öffnen“ und wandelt Ihn in eine ausführbare Aktion um. Die Aktion wird vom Aktionsmanager interpretiert, das bedeutet weitergeleitet an den jeweiligen Interpreter. Dieser versucht über vom Aktionsmanager angebotene Funktionen den in der Aktions-Spezifikation angegebenen Parameter aufzulösen, was geöffnet werden soll. Durch Abgleich der Zeitpunkte des Sprachbefehls mit den Zeitpunkten der Geste kann herausgefunden werden, welche Geste zum Zeitpunkt der Äußerung „aktiv“ war. Mit Hilfe der hinterlegten Koordinaten aus der Wissensbasis kann der

Vektor in Bildschirm/Anwendungs-Koordinaten umgewandelt werden. Die ergänzte Aktion wird an die jeweilige Anwendung weitergeleitet. Diese übersetzt die spezifizierten Koordinaten in einen Menüpunkt und aktiviert diesen.

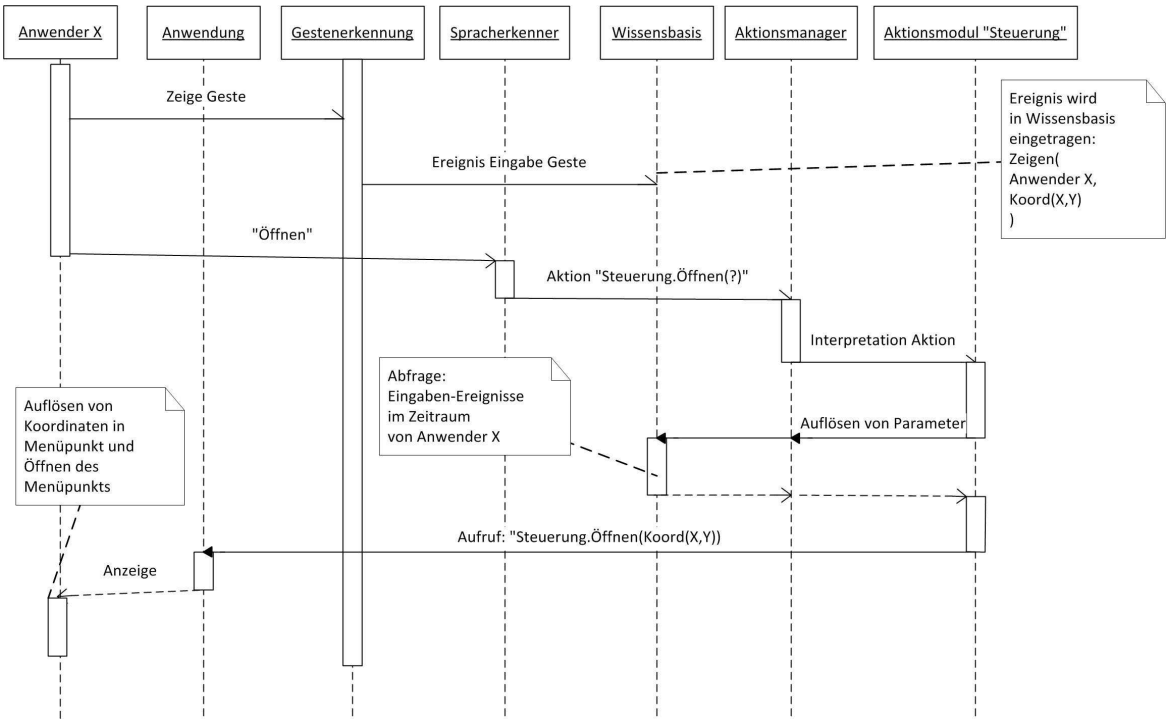

Abbildung 16: Sequenzdiagramm Multimodale Interaktion durch Ereignisverarbeitung

4.7.4. Auflösen von Deiktischen Begriffen

Damit sprachliche Interaktion so natürlich wie möglich gestaltet werden kann, muss das System das Auflösen von Deiktischen Begriffen unterstützen. Wie in Abschnitt 2.4.3 beschrieben, beschreibt Deixis das sprachliche Referenzieren von Orten, Personen oder Zeitpunkten. Im Folgenden wird der Ablauf erörtert, der beim auflösen eines Lokaldeiktischen Begriffs („hier“) ausgelöst wird.

Ausgangspunkt ist der Anwender. Er arbeitet mit einer Instanz der Routenplanungs-Anwendung, und bedient sie mit Maus und über Sprachbefehle.

Durch einen Klick auf die Karte (1a) wird ein „Input.Click“ Ereignis ausgelöst. Dieses Ereignis wird vom Aktionsinterpreter über den Feedback-Kanal der Anwendung empfangen[10]. Das

[10] In der Abbildung ist dieser Schritt vereinfacht. Das Ereignis wird hier direkt in die Wissensbasis eingetragen. Durch den Einsatz des Sensor-Terminal-Moduls (vgl. 4.6.1) ist dies auch durchführbar.

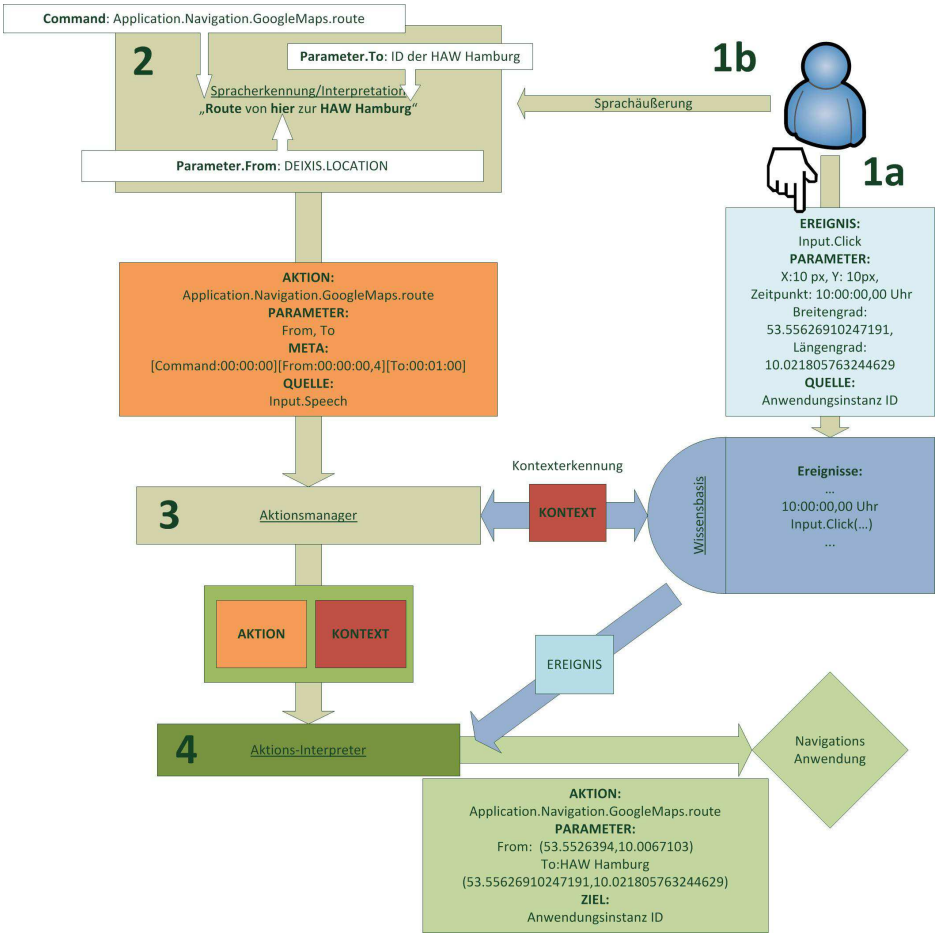

Abbildung 17: Exemplarischer Ablauf der Auflösung eines Deiktischen Begriffs

Ereignis enthält sowohl lokale Bildschirmkoordinaten (X,Y), wie auch die entsprechenden Breiten- und Längengradangaben des angeklickten Punkts. Zusätzlich verfügt das Ereignis über den Zeitpunkt der Aktion zur späteren Referenzierung.

Gleichzeitig zu der Klick-Eingabe äußert der Anwender den Sprachbefehl: „Route von hier zur HAW Hamburg“(1b). Die Grammatik der Routenplanungsanwendung hat diesen Satz definiert, so dass der Spracherkenner das Gesprochene als Befehl interpretieren kann.

Anhand der in der Grammatik festgelegten Semantik wird der Satz vom Routenplanungs-Sprachinterpretationsmodul in eine Aktion konvertiert (2). Das Fragment „Route“ definiert dabei, über den Semantik-Schlüssel-Wert „Command“ welche Aktion ausgeführt werden soll. Nämlich „Application.Navigation.GoogleMaps.route“. Der für die Aktion benötigte

Ziel-Parameter (Parameter.To) wird belegt mit einem System bekannten Ort[11]. Der Start-Parameter (Parameter.From) wird festgelegt auf einen von der Anwendung definierten Konstantwert (DEIXIS.LOCATION). Die Aktion die an den Aktionsmanager gesendet wird, enthält die benannten Parameter, den Aktionsnamen sowie weitere Metainformationen. Die Metainformationen sind optional.Sie werden im Falle des Aufrufs durch die Spracherkennung (Origin: Input.Speech) mit den chronologischen Parametern der Sprachäußerung gefüllt. Das bedeutet, dass zu den einzelnen erkannten Satzbestandteilen der Zeitpunkt ihrer Äußerung relativ zum Beginn gespeichert werden.

Der Aktionsmanager nimmt die Aktion entgegen, legt fest welchem Aktionsinterpreter-Modul sie zuzuweisen ist und augmentiert sie mit Kontextinformationen (3). Die Kontextinformationen beinhalten in diesem speziellen Falle für welche Anwendungsinstanz die Aktion wahrscheinlich bestimmt ist, sowie den Anwender von dem der Befehl geäußert wurde (siehe 4.7.2).

Im Aktionsinterpreter des Routenplaners wird die Aktion verarbeitet und in konkrete Befehle für die Anwendung übersetzt (4). Die Parameter werden auf ihren Inhaltstyp überprüft. Im Ziel-Parameter enthält die Identifikationsnummer des „HAW Hamburg“ Wissensbasiseintrags. Anhand dieser Identifikationsnummer kann die Anwendung über die Wissensbasis-Schnittstelle des Aktionsmanagers auf die Attribute der „HAW Hamburg“ zugreifen. Unter diesen Attributen befinden sich auch die Längen- und Breitengrade des Standorts.

Der Start-Parameter enthält zwar ein normales Literal, allerdings ist dies im Interpreter als Konstante für Lokaldeixis hinterlegt. Um einen lokaldeiktischen Begriff aufzulösen fordert der Interpreter die entsprechenden Ereignisse aus der Wissensbasis an. In diesem Fall bedeutet das, alle Ereignisse der Anwendungsinstanz, vom jeweiligen Anwender die Längen- und Breitengradinformationen enthalten. Zusätzlich eingeschränkt durch den Zeitpunkt der Äußerung der Konstante (Vier Zehntelsekunden nach Äußerungsbeginn) plus einen Puffer um etwaige Latenzen auszugleichen. Die Koordinaten des resultierenden Ereignis werden für den Konstantenwert substituiert.

Die vorbereitete Aktion wird nun an die Anwendung übermittelt und ausgeführt. Dem Anwender wird eine berechnete Routenempfehlung für die Strecke zur HAW Hamburg von den Koordinaten seines Klicks präsentiert.

[11] „HAW Hamburg“ ist in der Wissensbasis eingetragen inklusive globalen Koordinaten. Der Name des Orts wurde beim Laden der Grammatik aus der Wissensbasis in die Spracherkennung geladen.

5. Realisierung und Evaluation

Das Folgende Kapitel beschreibt die Realisierung und Evaluation des in der Zielsetzung beschriebenen Systems. Zuerst wird die Umgebung beschrieben, in der das System eingesetzt werden soll. Anschließend werden die für die Realisierung vorgenommenen Einschränkungen besprochen, sowie die Hauptkomponenten, ihre Funktionen und ihr Zusammenspiel erörtert.

5.1. Umgebung

Living Place Hamburg

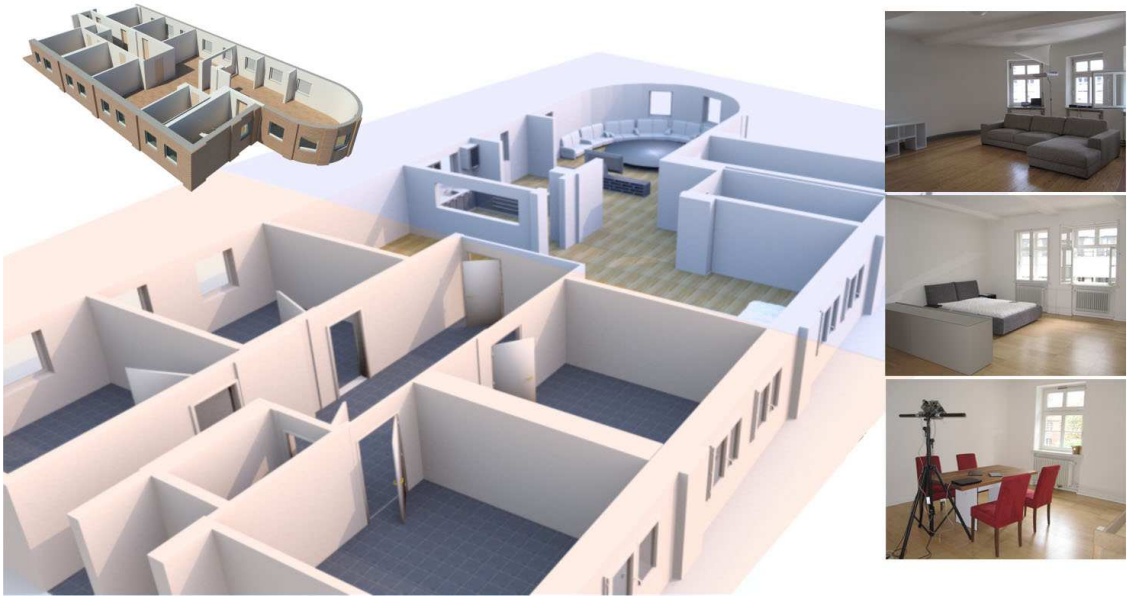

Abbildung 18: Übersicht über das Living Place Hamburg (blauer Bereich)

Auf dem Campus der Hochschule für angewandte Wissenschaften (HAW) in Hamburg entsteht seit Januar 2009 das Living Place Hamburg. Das Projekt wird zum Großteil von Finanzbehörde und der Behörde für Wissenschaft und Forschung finanziert. Es bietet Studenten verschiedener Fachrichtung die Möglichkeit ihre Forschungsarbeit auf dem Gebiet der Smart-Home-Technologien in der Praxis zu testen.

Das 140 Quadratmeter Fläche umfassende Studio-Apartment ist vollständig und funktional eingerichtet, inklusive funktionierendem Sanitärbereich. Der Grundsätzliche Aufbau lässt sich aus Abbildung 18 entnehmen.

Angrenzend an den Wohnraum befinden sich ein Entwicklungs- und Kontrollbereich. Aus dem Kontrollraum lassen sich die verfügbaren Sensoren überwachen und steuern. Diese umfassen verschiedene statische und schwenkbare Kameras sowie omnidirektionale und gerichtete Mikrophone. Somit sind sowohl autonome Testszenarien wie auch Wizard-of-OZ-Studien durchführbar. Die Büros dienen Studenten zur Arbeit vor Ort.
Das Ergebnis dieser Arbeit soll in dieser Umgebung eingesetzt und in Ihr evaluiert werden.

5.2. Einschränkungen

Der Umfang einer vollständig Integrierten Smart-Home Umgebung übersteigt den Rahmen dieser Arbeit. Daher werden für den Prototyp verschiedene Einschränkungen festgelegt, die eine Vereinfachung der Implementierung ermöglichen.

5.2.1. Anzahl der Benutzer

Die Zahl der gleichzeitigen Benutzer hat starken Einfluss auf die Komplexität des Gesamtsystems.
Um zum Beispiel bei der Spracherkennung eine möglichst hohe Erkennungsrate zu erzielen ist es notwendig den Erkenner auf eine Person zu trainieren. Mit jedem weiteren Benutzer steigt der Trainings- und Erkennungsaufwand und der Erkennungsprozess verlangsamt sich.

Auch bei gerätgebundenen Interaktionen wie zum Beispiel Multitouch Eingaben wird durch die Annahme von multiplen gleichzeitigen Verwendern die Komplexität der Auswertung stark erhöht. Die Zuordnung von Eingabeereignis zu Nutzer und somit auch der Kontext der Eingabeaktion muss über durch Anwendererkennung abgeleitet werden. Dieser Aufwand entfällt wenn sich nur ein Anwender im System befindet.

5.2.2. Freiheit der Spracheingaben

Die Eingabe durch freie Sprache ist bedingt durch die schwierigen Umgebungsbedingungen nicht ohne großen Mehraufwand durchführbar. Daher wird das System für jede Funktion die es erfüllen soll und die durch Sprache steuerbar ist bestimmte Grammatiken voraussetzen. Dies erhöht die Robustheit der Spracherkennung auch unter verrauschten Bedingungen. Die Erkennung ist pro aktiv nicht reaktiv. Es wird vorgegeben was erkannt werden soll und nicht aus freier Spracheingabe auf die Bedeutung geschlossen. Die Implikationen diese Entscheidung nach sich zieht wurden bereits in den vorhergehenden Abschnitten besprochen.

5.2.3. Kontext

Die Auswertung des Kontextes des Anwenders beschränkt sich auf das zuordnen von Anwenderaktionen zu Applikationen beziehungsweise zum System, so dass eine Auswertung von deiktischen Begriffen und des Anwendungskontexts ermöglicht wird. Verschiedene andere Parameter wie zum Beispiel der emotionale Kontext des Anwenders werden außer Acht gelassen, da das Erfassen und Auswerten dieser Kontexte den Rahmen dieser Arbeit übersteigt. Durch den Einsatz des Ereignissystems und der anwendungsgebundenen Interpretation ließen sich jedoch weitere Kontextinterpretationen ohne Schwierigkeiten nachgelagert implementieren.

5.2.4. Audiodaten Verarbeitung

Die Aufarbeitung von Audiodaten für die Verarbeitung mit Spracherkennung ist ein Forschungsgebiet für sich. Diese Arbeit befasst sich daher nicht explizit mit den verschiedenen Signalverarbeitungsprozessen. Da die Aufnahme und Verarbeitung der Lautsprache ein vorgelagerter Schritt ist, ist bleibt die Implementierung zum Beispiel von Filtern weiterhin möglich. Siehe dafür auch Maganti u. a. (2006),Chien und Lai (2004) oder Abad (2007). Ansätze dafür wären zum Beispiel die Fusion und Auswertung von Video und Positionsdaten für die Auswahl und Steuerung von Mikrophonen.

5.2.5. Laufzeitumgebung

Außer den Terminalmodulen laufen alle Systemkomponenten(siehe Abschnitt 4.5) während des Betriebs auf einer physikalischen Maschine. Das bietet verschiedene Vorteile. Zum einen entfällt die Notwendigkeit der Zeitsynchronisation da alle Bestandteile auf den gleichen Zeitgeber Zugriff haben. Zum zweiten vereinfacht sich die Fehlersuche und das aufwändige Debugging über Systemgrenzen hinweg entfällt. Weiterhin kann im System mit Maschinenrelativen Pfaden gearbeitet werden, so dass zum Beispiel Aufnahmen von einem Mikrofon lokal abgelegt und vom Sprachererkennungsmodul verarbeitet werden können. Ein komplexes Rechtesystem ist somit nicht notwendig.

5.2.6. Entwicklungsstand

Das realisierte System ist ein Prototyp der designten Architektur. Es wurden die Hauptkomponenten insoweit umgesetzt, wie sie für die Realisierung des Szenarios benötigt wurden. Wissensbasis, Aktionsmanager und Spracherkennung wurden ausimplementiert und bieten

die designte Funktionalität. Um weitere Anwendungsfälle umsetzen zu können soll das System Schritt für Schritt erweitert werden.

5.3. Audio- und Videoerfassung

5.3.1. Audiodatenerfassung

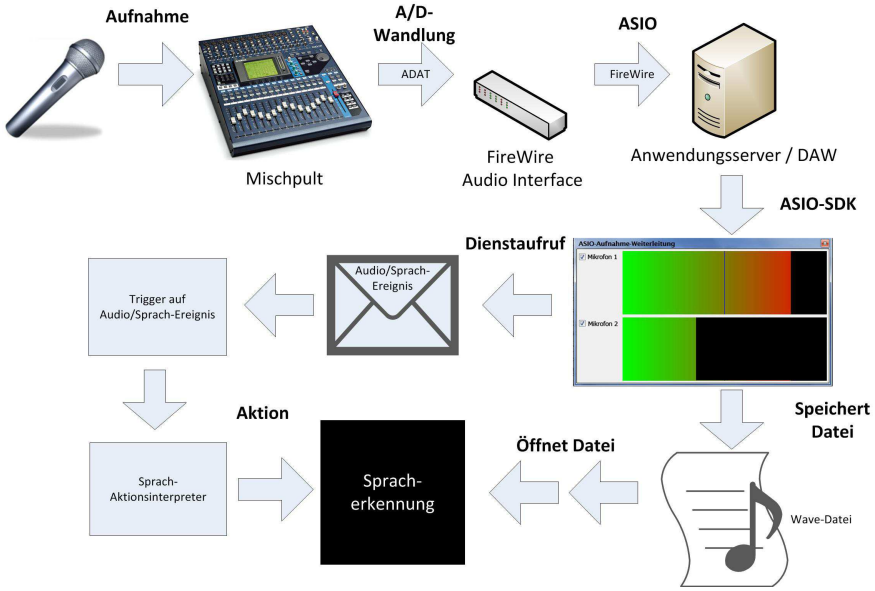

Abbildung 19: Audiodatenverarbeitung bis zur Spracherkennung

Die Sprachdaten werden von verschiedenen Mikrophontypen erfasst. Neben verschiedenen omnidirektionalen wurden auch mehrere gerichtete Mikrophone installiert. Die Richtmikrophone sind auf die Orte ausgerichtet, die durch ihre Ausstattung bedingt das größte Aktionspotential bieten. Neben den festinstallieren Mikrophonen lassen sich auch bestimmte „Terminals“ als Quelle für Audiodaten nutzen. Je nach Hardwarekonfiguration sind einige der die Terminalsoftware ausführenden Geräte mit integrierten Mikrofonen oder Schnittstellen ausgestattet, an die Mikrophone angeschlossen werden können (Zum Beispiel die für die Ansteuerung von großen Displays benötigten PCs). Die Verarbeitung dieser Audiosignale erfolgt dann, analog zum weiter unten beschriebenen Ablauf, über Audio-Ereignisse.

Die erfassten Audiodaten werden über Kabelleitung analog in den Kontrollraum übermittelt, wo sie in einem Mischpult erfasst werden. Das Mischpult bietet die Möglichkeit, über Softwaremodule, Raumakustische Anpassungen und Verbesserungen vorzunehmen. Die veränderten Daten fließen durch eine sogenannte ADAT Schnittstelle weiter an ein Audio-Interface das über eine IEEE 1394 Schnittstelle mit einer Digital Audio Workstation(DAW)

verbunden ist. Durch das parallele Übertragen der Audiodaten ist es möglich den Quellort zu bestimmen, zum Beispiel durch eine „Time-Difference of Arrival“-Analyse, oder auch die Ausrichtung des Sprechers (siehe Abad u. a. (2007)). Durch Kombination dieser Koordinaten und einem Indoor Positioning System ist eine Zuordnung der Sprache zu einer bekannten Person möglich.

Auf der DAW werden die Daten über einen ASIO-Treiber[12] weiterverarbeitet. Dieser ermöglicht das Umgehen der Audioarchitektur des Betriebssystems (Windows 7) und somit eine Latenzärmere Verarbeitung.

Die einzelnen Audiokanäle werden nach Filterung auf einen ausreichenden Aktivierungspegel geprüft. Wenn dieser Pegel erreicht wird, wird ein Stille-Timeout gestartet. Solange der Kanalpegel den Aktivierungspegel überschreitet wird der Timeout neu gestartet. Sollte die Lautstärke den Aktivierungspegel für einen bestimmten Zeitraum nicht erreichen und der Timeout somit ablaufen, wird das aufgezeichnete Audiomaterial als WAV-Datei zentral gespeichert.

Spracherkennung Über ein Audio-Ereignis wird dem System mitgeteilt das Audiodaten für die Spracherkennung verfügbar sind. Dieses Ereignis enthält auch die Quelle des Audiosignals, damit ein dementsprechendes Spracherkenner-Audio-Profil gewählt werden kann. Das Spracherkennungsmodul hat einen entsprechenden Trigger im Aktionsmanager registriert der auf diese Ereignisse lauscht. Wenn dieser Trigger feuert wird der Spracherkenner aktiv und verarbeitet die Audiodaten. Das weitere Vorgehen ist der Modulbeschreibung im Kapitel Design zu entnehmen, siehe 4.5.3.

5.3.2. Videodatenerfassung

Für die Erfassung von Gesten, Bewegungen und für die Aufzeichnung von Videodaten befinden sich verschiedene Kameras in der Evaluationsumgebung. Ihre Ausrichtung ist vom Kontrollraum steuerbar. Anhand der so gewonnenen Daten können über Videoanalysealgorithmen weitere Informationen abgeleitet werden. Dazu zählen zum Beispiel die einfache Bewegungserkennung und das Ableiten der Anwenderposition. Oder die Interaktion mit der Umgebung über Körpergesten. Die Videodaten werden in dieser Dieser Arbeit nicht explizit weiterverarbeitet stehen aber dafür zur Verfügung.

5.4. Technologische Bestandteile der Implementierung

Dieser Abschnitt listet die wichtigsten eingesetzten Technologien und deren Eigenheiten.

[12] Audio Stream Input/Output (ASIO) ein von Steinberg entwickeltes, mehrkanalfähiges Audiotransfer-Protokoll.

5.4.1. SAPI

Die Microsoft Speech API ist eine Programmierschnittstelle für die Spracherkennung und Sprachsynthese auf modernen Microsoft Betriebssystemen (ab Windows XP verfügbar). Sie bietet eine simple Schnittstelle auf die andernfalls komplexen Vorgänge Spracherkennung und -synthese. Über einen Assistenten kann sie auf einen Sprecher trainiert werden.

Sie bringt bereits trainierte Audio- und Sprachmodelle für die jeweilige Betriebssystemsprache mit. Im Vergleich mit häufig in der Forschung eingesetzten Spracherkennern (SPHINX oder Hidden Markov Toolkit) bietet die Microsoft Speech API Out-of-the-Box Funktionalität. Das bedeutet Sie muss nicht trainiert werden. Des Weiteren bietet der Spracherkenner die Möglichkeit Grammatiken mit Semantik zu annotieren. Wie bereits beschrieben ist dies eine Grundvoraussetzung für die Realisierung des Systems. Für weitere Informationen siehe http://www.microsoft.com/speech/

5.4.2. Windows Communication Framework

Das Windows Communication Framework(WCF) bietet die benötigte Funktionalität zur Verteilung der Komponenten. Es ist eine für das DOT.Net Framework entwickelte API die eine verteilte Implementierung mit minimaler Abhängigkeit von Transport-Protokollen ermöglicht. Eine Kompatibilität zu den Standardisierten Web-Dienst Technologien wie SOAP ist gegeben, so dass eine Kommunikation mit nicht in DOT.Net implementierten Anwendungen ermöglicht wird. Bei der Implementierung müssen verschiedene Attribute gesetzt werden die es der DOT.NET Laufzeitumgebung ermöglichen zu erkennen welche Dienste mit welchen Datentypen angeboten werden sollen. Folgt man diesen Vorgaben können verschiedenste Technologien für den Austausch von Informationen nur durch Anpassung einer Konfigurationsdatei Verwendung finden. Das WCF wird bei dieser Implementierung für die Kommunikation der Systemkomponenten untereinander verwendet. Weiterhin enthalten verschiedene Module Dienste zur Verständigung mit den Web-Oberflächen bzw. Anwendungen.

5.5. Umsetzung des Szenarios

Für die Umsetzung des Szenarios wurden zusätzlich zu den Systemkomponenten (Spracherkennung, Wissensbasis, Aktions-Manager) verschiedene weitere Module entwickelt. Diese bilden die benötigte Funktionalität ab und ermöglichen die gewünschte Interaktion mit dem Smart-Home System.

Abbildung 20: Bildschirmfoto des Routenplaner GUI

5.5.1. Routenplanung

Für das erstellen von Routen, das Darstellen von Geographischen Daten sowie das Suchen von Branchen wurde eine Schnittstelle zu Googles Kartendienst Google Maps(http://maps.google.com) geschaffen.

Das Modul besteht aus einer Anwendung (GUI, als Web-Anwendung implementiert), einem Sprachinterpreter, und einem Aktionsinterpreter (der auch Feedback von der Anwendung entgegen nimmt). Die Anwendung läuft zentral auf einem Webserver in der Smart-Home Umgebung und wird auf den jeweiligen Anzeigegeräten über einen Webbrowser aufgerufen. Sie ermöglicht die Interaktion mit Maus und Tastatur sowie über einfache Berührung (Touch).

Der Sprach-/Aktionsinterpreter ermöglicht unter anderem folgende Aktionen:

- Route erstellen und speichern
- Lokale Umkreissuche
- Zeigen von systembekannten und unbekannten Orten (Geocodierung wird durch Google Maps durchgeführt)

- Navigieren und anpassen der Zoomstufe

Der Sprachinterpreter lädt dafür aus einer XML-Grammatik-Datei (siehe A.1) die Phrasen die für die Steuerung der genannten Befehle verwendet werden können. Den Pfad dieser Datei erhält der Interpreter über Konfigurationdaten die beim Laden des Interpreters aus der Wissensbasis geladen werden.

Ein Feedback-Modul verbindet die Web-Anwendung mit dem Aktionsinterpreter. Auf diesem Wege können Aktionen die auf GUI Ebene ausgeführt werden an die Wissensbasis zurückgemeldet werden. Ein umgekehrter Informationsfluss ist ebenso möglich. Zum Beispiel um Bilder und Kontakte die zum aktuellen Standort passen abzurufen und darzustellen.

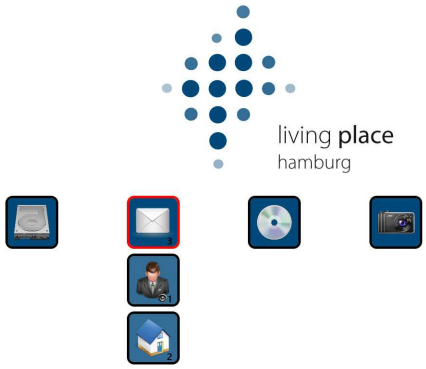

Abbildung 21: Bildschirmfoto des Dashboards

5.5.2. Dashboard

Für die Darstellung des Status der Smart-Home Umgebung und von eingehenden Nachrichten wurde eine „Dashboard“ Anwendung implementiert.

Das Modul enthält eine Web-Anwendung die für die Darstellung der Nachrichten zuständig ist. Ein Sprachinterpreter ermöglicht die Navigation in den Menüs der Anwendung über Sprachbefehle. Zusätzlich einen Aktionsinterpreter der verschiedene Trigger implementiert um den Systemstatus an die Web-Anwendung weitergeben zu können. Die Kommunikation übernimmt ein vom Aktionsinterpreter gestarteter WebSocket Server. Dieser ermöglicht eine direkte bidirektionale Kommunikation ohne Umwege über Polling von Dateien bzw. URLs.

5.5.3. Ausgabe

Dieses Modul dient der Ausgabe von Informationen an den jeweiligen Anwender. Nach Aktivierung überprüft es die Präferenzen des Nutzers hinsichtlich der Modalität und die verfügbaren Ausgabeterminals. Das Modul konvertiert, wenn nötig, die Information in die gewünschte Modalität und gibt diese an den Anwender aus. Textinformationen werden über einen Sprachsynthesedienst in Audiodaten umgewandelt und dann vom jeweiligen Terminal abgespielt. Zu diesem Zweck wird Googles Text-To-Speech Dienst (http://translate.google.com/translate_tts) verwendet. Dieses Modul ist eine minimale prototypische Implementierung des im Abschnitt 4.6.6 beschriebenen Ausgabemoduls.

5.5.4. Synchronisation

Das Synchronisationmodul übernimmt die Funktion der „Willkommen zu Hause“-Erkennung. Zu diesem Zweck registriert es sich für Positionsereignisse, sobald der Anwender in der Smart-Home Umgebung registriert wird, löst es verschiedene Aktionen aus um die Nutzerpräferenzen zu erfüllen (zum Beispiel Lichtsteuerung, Musik-Synchronisierung, start der Dashboard-Anwendung).

5.6. Hilfskomponenten

Zusätzlich zu den Modulen wurden verschiedene Komponenten entwickelt, die für die Kontrolle des Ablaufs und die Fehlersuche beziehungsweise -analyse benötigt werden. Diese Komponenten werden im folgenden Abschnitt kurz erläutert.

5.6.1. Hosting

Um die Systemdienste nach außen zur Verfügung zu stellen wurde ein Hostingprogramm erstellt. Dieses bietet eine Steuerungsoberfläche um einzelne Komponenten zu starten und zu beenden bzw. zu resetten. Es hält Referenzen auf die einzelnen Bestandteile und ermöglicht so ein Debugging während des Betriebs. Dies reduziert die Ansonsten in verteilten Systemen vorhandene Komplexität bei der Fehlersuche.

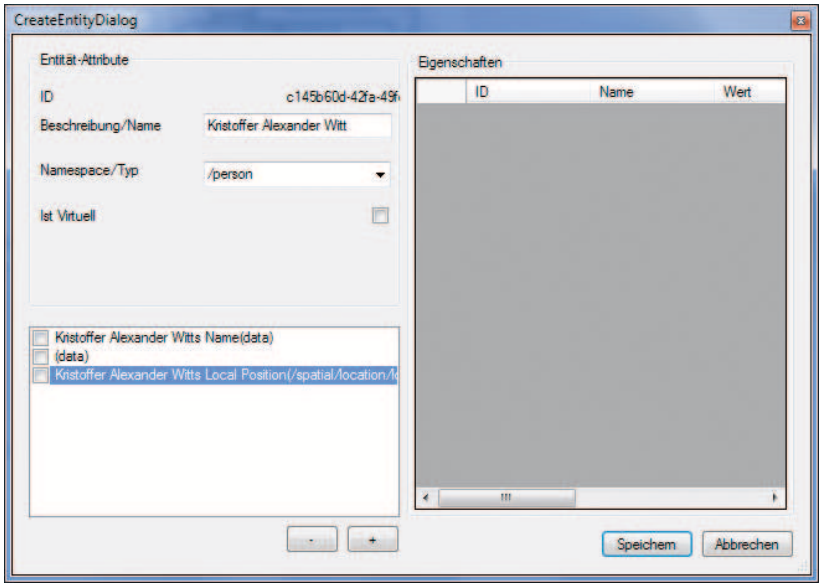

Abbildung 22: Bearbeiten von Einträgen der Wissensdatenbank

5.6.2. Verwaltung

Für die Verwaltung bzw. Steuerung und die vereinfachte Fehlersuche des Systems wurde eine Administrations-Anwendung entwickelt. Diese Anwendung ermöglicht neben der Überwachung des Systems auch das simulieren verschiedener Vorgänge, wie das Bewegen eines Anwenders im Raum und das explizite Starten von Anwendungen.

Auf einer graphischen Oberfläche, die eine Karte der Smart-Home Umgebung darstellt, können über Schaltflächen verschiedene Entitäten aus der Wissensbasis geladen und modifiziert werden (siehe Abbildung 23). Über eine direkte Kopplung erhält die Oberfläche aktuelle Ereignisse, wie zum Beispiel die Bewegung eines Anwenders im Raum und kann diese visualisieren. Neben der Steuerung ermöglichen verschiedene Übersichtsfenster eine Analyse der einzelnen Systembestandteile und des Gesamtstatus des Systems.

5.6.3. ASIO-Datenverarbeitung

Das ASIO-Datenverarbeitungsmodul dient dazu, über den ASIO-SDK empfangene Audiodaten als WAV-Datei zu speichern und an das System weiterzuleiten (über Audioereignisse) bzw. Aktivität auf den Aufnahmekanälen zu entdecken. Weiterhin ermöglicht die Komponente das spezifische De-/Aktivieren und verstärken einzelner Audioquellen zu Testzwecken. Wenn zum Beispiel nur ein Mikrophon Audiodaten aufzeichnen soll kann die Verarbeitung der anderen Mikrofondaten softwareseitig deaktiviert werden. Alternativ wäre das Herunter

regeln über das Mischpult, vorausgesetzt das Mikrofon wird vom Mischpult erfasst. Siehe 5.3.1.

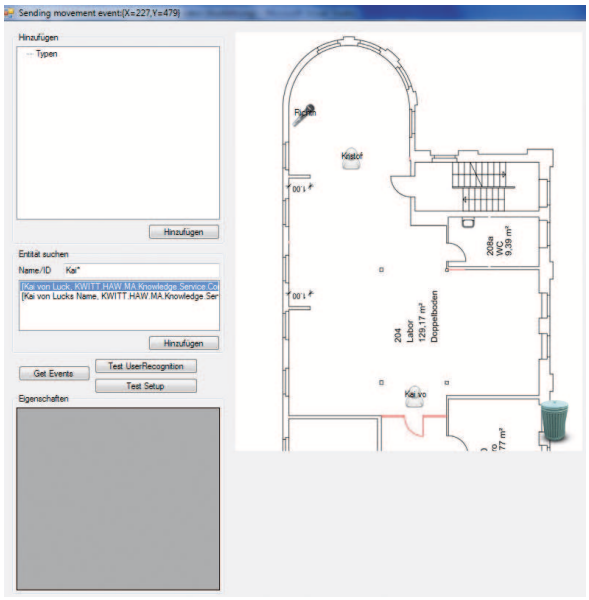

Abbildung 23: Hauptoberfläche der Simulationssoftware

5.7. Evaluation

Abschließend wird das Ergebnis der Systemevaluation vorgestellt. Die Evaluation des Systems dient dem Nachweis, dass das implementierte System die angestrebte Zielsetzung Ganz oder in Teilen erfüllt (Proof of Concept).

Die Evaulation des System gliedert sich in drei Teile. Im ersten Teil wird die grundsätzliche Vorgehensweise, der Ablauf und die Vorbedingungen der Evaluation erläutert. Der zweite Teil beschreibt das Ergebnis der Umsetzung in einer optimierten Testumgebung. Der zweite Teil die Ergebnisse in dem unter Abschnitt 5.1 aufgeführten Living Place Hamburg.

5.7.1. Was soll evaluiert werden?

Es soll gezeigt werden, dass die konzipierte Architektur für einen konkreten Anwendungsfall umsetzbar ist. Der Anwendungsfall orientiert sich an dem vorgestellten Szenario. Er enthält verschiedene Abläufe, die aufzeigen sollen, inwiefern die Zielsetzung erreicht worden ist, mit einer Smart-Home Umgebung kontextbasiert und multimodal zu interagieren. Dabei besteht die Multimodalität aus Kombination von Sprachsteuerung mit weiteren Modalitäten.

Zu zeigende Zielsetzungen:

I Kontextabhängige Interpretation von Eingabegeräten

II Multimodale Steuerung einer Anwendung mit Graphischer Oberfläche.

III Auflösung von deiktischen Referenzen in Sprachbefehlen

IV Dynamische Alterierung von Sprachgrammatiken zur Adaption des Anwendungskontexts.

V Automatische Anpassung an Ortskontext des Anwenders.

VI Steuerung der Smart-Home Umgebung mit Sprache

Wenn das Erreichen dieser Zielsetzungen gezeigt werden kann, können aufgrund dessen weitere mögliche Anwendungsfälle und Szenarien interpoliert werden. Allgemein kann somit gesagt werden das System ist multimodal und kontextabhängig steuerbar.

5.7.2. Implementierung

Für die Teilumsetzung des Szenarios wurden folgende Komponenten in Teilen oder Vollständig implementiert:

- Systemkomponenten (jeweils inklusive Aktionsinterpreter)
  - Aktionsmanager
  - Wissensbasis
  - Spracherkennung
  - Terminal-Software
- Systemmodule
  - Sensorschnittstelle
  - Benutzererkennung
  - Kontexterkennung
  - Anwendungsverwaltung
- Benutzermodule
  - Routenplanungsanwendung
  - Dashboardanwendung

5.7.3. Aufbau

Als Versuchsaufbau wurde in den beiden Testumgebungen jeweils folgendes Setup ausgeführt.

Terminal Es wurden zwei Terminalsoftware Instanzen gestartet, mit stark abweichenden Ortskoordinaten.

Anwender Es wurde ein Anwender über die Simulationsumgebung in der Nähe eines Terminals platziert. Das hat das Hinzufügen eines Positionsereignisses für diesen Anwender zur Wissensbasis zur Folge.

Module Initial wurden die Aktionsinterpreter für die Wissensbasis und die Spracherkennung sowie alle Systemmodule geladen.

Sonstige Eingabegeräte Dem Anwender stehen Tastatur und Maus sowie ein „Hamburg Cubical“Gregor u. a. (2010) zur Verfügung.

Audioverarbeitung Das Programm das den Schalldruckpegel überwacht und aus den Aufnahmen WAV-Dateien und Audio-Ereignisse generiert wurde gestartet.

5.7.4. Ablauf

Der Befehl „Starte Anwendung Routenplanung“ wurde gut wahrnehmbar in den Raum gesprochen. Die Lautstärke der Äußerung hat den Aktivierungspegel der Audioüberwachung (ASIO-Toolbar) überschritten und somit eine Aufzeichnung des Befehls ausgelöst. Nach einer Zeitspanne von eineinhalb Sekunden Stille nach der Äußerung hat die Audioüberwachung den Aufgezeichneten Befehl als WAV-Datei gespeichert und ein Audio-Ereignis wurde in der Wissenbasis erzeugt. Als Parameter die ID des Mikrophons das aktiviert wurde und der Pfad zu der Audiodatei. **(Zielsetzung VI)**

Der Befehl wurde von der Spracherkennung erkannt und der Grammatik des Anwendungsverwaltungsmoduls zugeordnet. In der Wissensbasis wurde ein dafür Sprach-Ereignis abgelegt, wieder mit der Quelle des Signals, der Mikrophon-ID, als Parameter. Der Sprachinterpreter des Anwendungsverwaltungsmoduls hat den Befehl umgesetzt in eine StartAnwendung-Aktion als Parameter wurde die ID der Routenplanungsanwendung eingesetzt. Die Aktion wurde an die Ausführungsschnittstelle des Aktionsmanagers weitergegeben.

Der Aktionsmanager ruft die Kontexterkennung auf, die wiederum die Benutzererkennung aufruft. Da sich nur ein Anwender im System befindet steht dieser als Ursprung fest (das System kann nicht Auslöser gewesen sein, da der Ursprung der Aktion die Spracherkennung ist). Der Aktionsmanager prüft die Aktionsinterpreter ob die Aktion verarbeitet werden kann und leitet sie inklusive der Kontextinformationen an den Anwendungsverwaltungs-Aktions-Interpreter weiter.

Die Anwendungsverwaltung vergleicht die Position des Anwenders (die Parameter des letzten Positionsereignisses des Anwenders) mit denen der gestarteten Terminals und gibt die Aktion an das jeweilige Terminal weiter (inklusive einer vergebenen Anwendungsinstanz-ID). Vorher wird noch, anhand der Informationen die in der Wissensbasis zur Anwendung Routenplanung hinterlegt sind, der Sprachinterpreter der Routenplanung in den Spracherkenner geladen. Sowie der Routenplanungs-Aktionsinterpreter in den Aktionsmanager, dieser aktiviert einen Trigger auf Eingabe-Ereignisse. **(Zielsetzung V)**

Das Terminal startet, da es sich um eine Web-Anwendung handelt, den Standardbrowser und übergibt öffnet die URL der Anwendung parametrisiert mit der Anwendungsinstanz-ID. Der Anwender sieht nun auf dem an das Terminal angeschlossenen Ausgabegerät, einem LCD-Bildschirm, die Oberfläche der Routenplanungsanwendung.

Der Cubical wird um 45 Grad mit dem Uhrzeigersinn gedreht. Circa 1 Minute später wird der Cubical auf eine andere Seite gelegt erneut um 45 Grad gedreht. Die Software die die Daten

des Cubicals empfängt ruft die Schnittstelle des Sensormoduls auf und instruiert sie, diese Eingabe-Ereignisse aufzunehmen.

Das Eintragen der Ereignisse löst den registrierten Trigger des Routenplanungsinterpreters aus. Dieser übersetzt die Ereignisse in zwei Aktionen (Unterschieden werden die Ereignisse anhand der mit übertragenen Seite auf der der Würfel liegt). Zum einen in ein Zoom-Ereignis und die zweite Drehung in ein Dreh-Ereignis. Der Aktionsmanager leitet diese Aktionen weiter an den Routenplaner-Aktionsinterpreter. Die Kontextinformationen enthalten wieder den Anwender sowie die ID der Anwendungsinstanz. Diese wurde anhand der Position des Terminals und der des Anwenders ermittelt. Da sich im System nur eine aktive Routenplaner-Instanz befindet kann angenommen werden, dass diese Instanz auch das Ziel der Aktionen ist.

Der Aktionsinterpreter beschreibt eine XML-Datei mit den beiden Aktionen. Diese XML-Datei wird von der Web-Anwendung in einem bestimmten Zeitintervall gepollt.

Die Web-Anwendung erkennt, dass neue Befehle vorhanden sind und setzt diese um. Das Bild dreht sich um 45 Grad und der Ausschnitt wird herangezoomt. **(Zielsetzung I)** *Erläuterung: Der Würfel kann je nach Trigger der auf seine Ereignisse wartet, unterschiedliche Aktionen auslösen.*

Ein Maus-Klick auf die Karte wird über das Sensormodul als Eingabe-Ereignis (Click) in die Wissensbasis übertragen, augmentiert mit den Koordinaten in Längen- und Breitengrad sowie der Anwendungsinstanz-ID. Gleichzeitig erfolgt der Sprachbefehl „Route von hier zum Stadtpark in Hamburg“.

Der Sprachbefehl wird vom Spracherkenner erkannt und an den Sprachinterpreter der Routenplanungsanwendung weitergeleitet[13]. Über die Semantik, die in der Grammatik der Anwendung festgelegt wurde, weiß der Interpreter das die Aktion Route ausgelöst werden muss mit den Parametern Startort und Zielort. Der Term „hier“ wurde in der Grammatik als Deiktische-Referenz gekennzeichnet. Der Term „Stadtpark in Hamburg“ wurde über eine Diktatregel erkannt. Würde stattdessen ein bekannter Ort bezeichnet, hätte dieser als Semantischen Wert die ID dieses Ortes. Der Sprachinterpreter setzt die Aktion „Route“ mit den genannten Parametern an den Aktions-Manager ab.

Die Aktion wird an den Aktionsinterpreter der Routenplanungsanwendung weitergeleitet, inklusive Kontextinformationen (Anwender, Anwendungsinstanz-ID). Der Interpreter erkennt das es sich beim Start Parameter um eine Deiktische Referenz handelt und verfährt wie unter Abschnitt 4.7.4 beschrieben. **(Zielsetzung II und III)**

Die Route wird von Web-Anwendung berechnet (Google Maps löst den Begriff Stadtpark Hamburg in eine Adresse auf und verwendet diese) und dargestellt. Durch ein Klicken auf

[13] Vorausgegangen ist wieder oben beschriebene Ablauf über die Audioüberwachung

eine einzelne Wegpunktbeschreibung wird der dazugehörige Wegpunkt auf der Karte dargestellt. Der Befehl „Suche Restaurants“ wird gesprochen und wie bereits beschrieben verarbeitet. Neben der Karte erscheinen die Suchergebnisse. Über die Feedbackfunktionalität der Anwendung, implementiert als REST-Aufruf eines in der Domäne des Anwendungsmanagers bzw. des Interpretationsmoduls laufenden Endpunktes, teilt die Anwendung die Daten der Suchergebnisse dem Interpreter mit.

Der Interpreter transformiert die Suchergebnisdaten in eine Grammatik und lädt diese über eine Aktion in den Spracherkenner. Die Grammatik ermöglicht nun ein ansteuern der Suchergebnisse per Sprachbefehl. **(Zielsetzung IV)**

5.7.5. Optimierte Testumgebung

Um zu überprüfen, inwieweit die konzipierte und in großen Teilen realisierte Architektur die Zielsetzung erfüllt, wurde der Anwendungsfall auf einem PC simuliert. Der PC diente somit als Host für die Systemkomponenten und als Schnittstelle für Benutzerinteraktion. Etwaig für den Ablauf benötigte Ereignisse[14] wurden, insofern wie sie noch nicht über Sensoren implementiert waren, über die Simulationsumgebung in das System eingegeben.

Spracherkennung

Die Spracherkennungskomponente war vor Testbeginn auf den Anwender trainiert worden. Die Testbedingungen, also Abstand zum Mikrophon, Hintergrundgeräusche und Sprecher waren identisch mit den Trainingsparametern. Daher konnten auch die normalerweise fehleranfälligen Diktatpassagen (z.B. Nennung eines Ortes der nicht in der Wissenbasis vorhanden ist) mit hoher Erkennungsrate durchgeführt werden.

Ergebnis

Unter den optimalen Bedingungen der Testumgebung funktionierte das Testsystem innerhalb der Erwartungen. Die angestrebte multimodale Interaktion wurde anhand der Kombination von Sprach- mit Maus- und Tastatureingaben nachvollzogen und lieferte das erwartete Ergebnis: Durch Verknüpfung der während der Bedienung des Systems auftretenden Ereignisse (Audio/Sprach-Ereignis, Eingabe-Ereignis) konnte die beschriebene Routenplanungsanwendung multimodal gesteuert werden.

[14] Zum Beispiel das Bewegen eines Anwenders durch das Living Place Hamburg, das zum Zeitpunkt des Tests noch nicht durch Sensoren erfasst wurde.

5.7.6. Living Place Hamburg

Der bisher implementierte Softwarestand, wurde von der Testumgebung für die Gegebenheiten der Smart-Home Umgebung angepasst, dafür wurden bestimmte Veränderungen vorgenommen:

Trennung Terminal und System Die Systemdienste wurden auf einem Serverrechner gestartet, die Terminal Software auf mehreren in der Räumlichkeit verteilten PCs.

Audiodatenerfassung Die Audio-Sprachdaten wurden von den Raummikrofonen erfasst digitalisiert und an die Spracherkennung übertragen.

Spracherkennung Für die Spracherkennung wurde ein Sprecherprofil trainiert. Die Parameter des Profils waren der fixe Abstand zu einem Raummikrophon bei optimalen Umgebungsbedingungen.

Ausgabe Zum Zeitpunkt des Tests war in der Smart-Home Umgebung noch keine zentral gesteuerte Audioausgabe möglich. Daher wurde die Audioausgabe über die Terminal Software realisiert, über an die Hostrechner angeschlossene bzw. integrierte Lautsprecher.

Spracherkennung

Die Erkennungsrate der Spracherkennung hat trotz Training des Audiomodells stark nachgelassen. Insbesondere kurze Sprachbefehle wurden vom System nicht oder nur sehr selten erkannt. Längere Sequenzen wie: „Starte Anwendung Routenplanung“ konnte das System in vielen Fällen richtig verarbeiten. Ursachen dafür sind mit großer Wahrscheinlichkeit die Ausgangsaudiomodelle des Erkenners. Diese Modelle sind für den Einsatz mit Headset bzw. Standmikrophonen optimiert. Das Training das die Spracherkennungsoftware anbietet, dient zum Großteil der Anpassung an die Sprechereigenheiten und nur zum Teil der Umgebungsbedingungen.

Diktateingaben waren zwar möglich wurden aber meistens Falsch erkannt. Somit war die Steuerung in der Kartenanwendung festgelegt auf Örtlichkeiten die dem System bekannt und somit in der Grammatik vorhanden waren.

Ergebnis

Ohne weitere Optimierungen ist eine Sprachsteuerung im Living Place Hamburg zwar möglich, aber durch die schlechte Erkennungsrate nur sehr sporadisch einzusetzen. Um bessere

Ergebnisse zu erhalten sollten weitere Versuche mit Headsetmikrofonen sowie weiteren angepassten Sprachmodellen durchgeführt werden. Weiterhin müssen Tests mit wechselnden LärmBedingugnen erfolgen um festzustellen, wie sich das System verhält. Neben der technischen Evaluation wäre auch Durchläufe mit weiteren Testpersonen Sinnvoll, um evaluieren zu können in wie weit das System benutzerfreundlich Bedienbar ist (im Vergleich mit per Maus und Tastatur bedienter Software).

6. Schluss

Der abschließende Teil gliedert sich in ein Fazit, das die grundlegenden Ergebnisse und die Vorgehensweise dieser Arbeit zusammenfasst und den Ausblick auf weitere Ansatzpunkte hinsichtlich Weiterentwicklung und Optimierung des entstandenen Systems.

6.1. Fazit

In dieser Arbeit wurde eine Architektur entwickelt die es ermöglicht kontextabhängig multimodal mit einer Smart-Home Umgebung zu interagieren. Der Schwerpunkt lag darin, die natürlichste Kommunikationsform, die Lautsprache, in allen interaktiven Bereichen des Systems verfügbar zu machen. Dabei wurde auch das für maschinelle Verarbeitung schwer erfassbare Konzept der Deiktischen Begriffe umgesetzt.

Es ist ein System entstanden, das sich durch seine allgemeine Schnittstellendefinitionen begünstigt dynamisch, einfach und schnell erweitern lässt. Die Architektur berücksichtigt die große Anzahl der in Smart-Homes verfügbaren Interaktionsmöglichkeiten und stellt diese dem jeweiligen Anwender zur Verfügung. Die Steuerung kann sowohl uni- wie auch multimodal erfolgen.

Anhand auftretender Ereignisse, deren Ursprung sowohl Sensorik wie auch Aktorik sein kann, wurden Kontexte gebildet. Diese ermöglichen ein adäquates, das bedeutet die Nutzerpräferenzen berücksichtigendes, reagieren auf verschiedenste Stimuli. Insbesondere die, in heterogenen Umgebungen nicht immer triviale, Aufgabe der Zuordnung von Aktion zu Empänger konnte mit Hilfe der umgesetzten Kontexterkennung gelöst werden.

Um das Verhalten der Architektur zu testen, wurde eine prototypischen Umsetzung sowohl unter Labor- wie auch unter Realbedingungen evaluiert. Unter optimalen Bedingungen konnte wie erwartet mit dem System interagiert werden. Der Test unter Realbedingungen ergab Schwierigkeiten bei der Genauigkeit der Sprachsteuerung aufgrund der Umgebungsbedingungen. Können diese Ungenauigkeiten beseitigt werden so ist ein Einsatz des Systems in weiteren Smart-Home Umgebungen denkbar.

6.2. Ausblick

Die in dieser Arbeit entstandene Architektur bietet verschiedene Ansatzpunkte für weitere Projekte. Exemplarisch sollen hier einige dieser Ideen für Optimierung und Erweiterung aufgelistet und beschrieben werden.

6.2.1. Unterstützung von mehreren Benutzern

Die Einschränkung auf einen aktiven Anwender macht das entwickelte System unattraktiv für einen Großteil der Bevölkerung. Daher sollte ein Ansatz gefunden werden, die Interaktion verschiedener Benutzer gleichzeitig zu verarbeiten. Dafür müssten Wege gefunden werden, Aktionen eindeutig Benutzer zuzuordnen. Im Falle von Sprachbefehlen wäre das zum Beispiel möglich durch eine vorgeschaltete Stimmerkennung. Für Gesten und andere Eingabeaktionen durch Auswertung von Videodaten.

6.2.2. Spracherkennung

Die aktuell implementierte Lösung funktioniert nicht optimal. Für die Verbesserung der Erkennungsperformanz und damit auch der Erhöhung der Benutzerakzeptanz des Gesamtsystems sind verschiedene Optimierungen denkbar. Zum einen in technischer Hinsicht der Umstieg auf ein Spracherkennungssystem welches mehr Anpassungsmöglichkeiten bietet. Also das Erstellen und Verbessern von Audio- und Sprachmodellen hinsichtlich des Anwendungsbereichs. Neben verbesserten Modellen ist es auch denkbar die Audiodaten vorzuverarbeiten. Zum Beispiel Filter einzusetzen, die die Eigenheiten der Raumakustik beachten und egalisieren.

Je besser die Leistung der Spracherkennung ist desto freier kann die Gestaltung der Erkennungsgrammatiken werden. Das Ziel sollte es sein, alle vom Anwender geäußerten Befehle in Text zu übersetzen und anschließend auf deren Semantik zu schließen. Das würde sich stark positiv auf die Benutzbarkeit des Systems auswirken. Als Vorstufe wäre es denkbar mit dem aktuellen System Usability-Studien durchzuführen um festzustellen welche Form die geäußerte Sprache annehmen kann und anhand dessen die aktuellen Grammatiken zu erweitern bzw. zu optimieren.

Im Zuge dessen wäre auch eine Adaptierung des semantischen Webs denkbar. Das bedeutet, dass anhand von semantischer Analyse natürlichsprachlich geäußerte Fragen über etwaige Web-Dienste wie Freebase (http://freebase.com) beantwortet werden könnten.

6.2.3. Weitere Modalitäten

Wie im Abschnitt 2.5 beschrieben dient Multimodalität insbesondere auch der Natürlichkeit der Interaktion mit einem Computersystem. Die Erkennung verschiedener weiterer Modalitäten, Körperhaltung, Gesten mit Arm und Bein wäre ein weiterer Schritt hinsichtlich der Akzeptanz des Gesamtsystems. Die entwickelte grundlegende Architektur sollte dazu in der Lage sein weitere Eingabe Ereignisse zu unterstützen und für die Verarbeitung zur Verfügung zu stellen.

Um die Zusammenarbeit noch harmonischer zu gestalten sollten auch die Befindlichkeiten des Anwenders mit in die Auswertung und Umsetzung von Aktionen einbezogen werden. Das bedeutet das neben der Interpretation der Bewegungsabläufe auch ein psychischer Aspekt evaluiert werden sollte. Die aktuelle emotionale Lage in der sich der Anwender befindet lässt sich aus verschiedenen (auch nicht invasiven, zum Beispiel Poh u. a. (2010)) Sensoren ableiten. Entsprechend kann das System so angepasst werden das es angemessen reagiert.

Literatur

[WebService 2004] Booth, David (Hrsg.) ; Haas, Hugo (Hrsg.) ; Mccabe, Francis (Hrsg.) ; Newcomer, Eric (Hrsg.) ; Champion, Michael (Hrsg.) ; Ferris, Chris (Hrsg.) ; Orchard, David (Hrsg.): Web Services Architecture / World Wide Web Consortium. February 2004. – Forschungsbericht

[Abad 2007] Abad, Alberto: *A Multi-microphone approach to speech processing in a smart-room environment*, Universitat Polit'ecnica de Catalunya, Dissertation, 2007

[Abad u. a. 2007] Abad, Alberto ; Segura, Carlos ; Nadeu, Climent ; Hernando, Javier: Audio-Based Approaches to Head Orientation Estimation in a Smart-Room . In: *INTERSPEECH-2007*, 2007, S. 590–593

[Abalos u. a. 2011] Abalos, Nieves ; Espejo, Gonzalo ; Lopez-Cozar, Ramon ; Callejas, Zoraida ; Griol, David: A Multimodal Dialogue System for an Ambient Intelligent Application in Home Environments. In: Sojka, Petr (Hrsg.) ; Horak, Ales (Hrsg.) ; Kopecek, Ivan (Hrsg.) ; Pala, Karel (Hrsg.): *Text, Speech and Dialogue* Bd. 6231. Springer Berlin / Heidelberg, 2011, S. 491–498

[Acampora und Loia 2008] Acampora, Giovanni ; Loia, Vincenzo: A proposal of ubiquitous fuzzy computing for Ambient Intelligence. In: *Information Sciences* 178 (2008), Nr. 3, S. 631 – 646. – URL http://www.sciencedirect.com/science/article/B6V0C-4PMJK4F-1/2/3f6c37 – Including Special Issue Ambient Intelligence. – ISSN 0020-0255

[Aghajan u. a. 2009] Aghajan, Hamid ; Augusto, Juan C. ; Delgado, Ramon Lopez-Cozar: *Human-Centric Interfaces for Ambient Intelligence*. Academic Press, 2009. – ISBN 0123747082

[Augusto u. a. 2009] Augusto, J. C. ; Bohlen, M. ; Cook, D. ; Flentge, F. ; Marreiros, Goreti ; Ramos, Carlos ; Qin, Weijun ; Suo, Yue: The Darmstadt Challenge - The Turing Test Revisited. In: (Filipe u. a., 2009), S. 291–296. – ISBN 978-989-8111-66-1

[Augusto 2004] AUGUSTO, Juan C.: Ambient Intelligence: the Confluence of Ubiquitous/-Pervasive Computing and Artificial Intelligence. (2004)

[Augusto 2009] AUGUSTO, Juan C.: Past, Present and Future of Ambient Intelligence and Smart Environments. In: (Filipe u. a., 2009), S. 11–18. – ISBN 978-989-8111-66-1

[Baker u. a. 2009] BAKER, J. ; DENG, Li ; GLASS, J. ; KHUDANPUR, S. ; LEE, Chin hui ; MORGAN, N. ; O'SHAUGHNESSY, D.: Developments and directions in speech recognition and understanding, part 1. In: *Signal Processing Magazine, IEEE* 26 (2009), May, Nr. 3, S. 75–80. – ISSN 1053-5888

[Bellik u. a. 2009] BELLIK, Yacine ; REBAÏ, Issam ; MACHROUH, Edyta ; BARZAJ, Yasmin ; JACQUET, Christophe ; PRUVOST, Gaëtan ; SANSONNET, Jean-Paul: Multimodal Interaction within Ambient Environments: An Exploratory Study. In: *INTERACT '09: Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction*. Berlin, Heidelberg : Springer-Verlag, 2009, S. 89–92. – ISBN 978-3-642-03657-6

[Burzagli u. a. 2007] BURZAGLI, Laura ; EMILIANI, Pier L. ; GABBANINI, Francesco: Ambient intelligence and multimodality. In: *UAHCI'07: Proceedings of the 4th international conference on Universal access in human-computer interaction*. Berlin, Heidelberg : Springer-Verlag, 2007, S. 33–42. – ISBN 978-3-540-73280-8

[Chien und Lai 2004] CHIEN, Jen-Tzung ; LAI, Jain-Ray: Use of Microphone Array and Model Adaptation for Hands-Free Speech Acquisition and Recognition. In: *J. VLSI Signal Process. Syst.* 36 (2004), February, S. 141–151. – URL http://portal.acm.org/citation.cfm?id=968399.968406. – ISSN 0922-5773

[Cohen u. a. 2004] COHEN, Michael H. ; GIANGOLA, James P. ; BALOGH, Jennifer: *Voice User Interface Design*. Redwood City, CA, USA : Addison Wesley Longman Publishing Co., Inc., 2004. – ISBN 0321185765

[Comerford u. a. 1997] COMERFORD, Richard ; MAKHOUL, John ; SCHWARTZ, Richard: The voice of the computer is heard in the land (and it listens too!). In: *IEEE Spectr.* 34 (1997), Nr. 12, S. 39–47. – ISSN 0018-9235

[Corkill 1991] CORKILL, Daniel: Blackboard Systems. In: *AI Expert* 6 (1991), January, Nr. 9. – URL http://mas.cs.umass.edu/paper/218

[De Mori u. a. 2008] DE MORI, R. ; BECHET, F. ; HAKKANI-TUR, D. ; MCTEAR, M. ; RICCARDI, G. ; TUR, G.: Spoken language understanding. In: *Signal Processing Magazine, IEEE* 25 (2008), May, Nr. 3, S. 50–58. – ISSN 1053-5888

[Dey 2001] DEY, Anind K.: Understanding and Using Context. In: *Personal Ubiquitous Comput.* 5 (2001), Nr. 1, S. 4–7. – ISSN 1617-4909

[Dimopulos u. a. 2007] DIMOPULOS, T. ; ALBAYRAK, Sahin ; ENGELBRECHT, Klaus-Peter ; LEHMANN, G. ; MÖLLER, Sebastian: Enhancing the Flexibility of a Multimodal Smart Home Environment. In: *Fortschritte der Akustik - DAGA 2007: Plenarvortr. u. Fachbeitr. d. 33. Dtsch. Jahrestag. f. Akust.* Berlin, Germany : Deutsche Gesellschaft für Akustik (DEGA), March 2007, S. 639ï¿–640

[(Editor) 2006] (EDITOR), J. C. Augusto (Editor) C. N.: *Smart Homes And Beyond: Icost 2006 (Assistive Technology Research Series)*. Ios Pr Inc, 2006. – ISBN 1586036238

[Engel 2006] ENGEL, Ralf: SPIN: A Semantic Parser for Spoken Dialog Systems. In: *Proceedings of the 5th Slovenian and First International Language Technology Conference (IS-LTC 2006)*, 2006, S. 1–100

[for Official Publications of the European Communities 2000] EUROPEAN COMMUNITIES, Office for Official Publications of the: *Scenarios for Ambient Intelligence in 2010 : final report*. Office for Official Publications of the European Communities, 2000. – ISBN 92-894-0735-2

[Filipe u. a. 2009] FILIPE, Joaquim (Hrsg.) ; FRED, Ana L. N. (Hrsg.) ; SHARP, Bernadette (Hrsg.): *ICAART 2009 - Proceedings of the International Conference on Agents and Artificial Intelligence, Porto, Portugal, January 19 - 21, 2009*. INSTICC Press, 2009. – ISBN 978-989-8111-66-1

[Galton und Augusto 2002] GALTON, Antony ; AUGUSTO, Juan C.: Two Approaches to Event Definition. In: *DEXA '02: Proceedings of the 13th International Conference on Database and Expert Systems Applications*. London, UK : Springer-Verlag, 2002, S. 547–556. – ISBN 3-540-44126-3

[Gilbert u. a. 2008] GILBERT, M. ; KNIGHT, K. ; YOUNG, S.: Spoken Language Technology [From the Guest Editors]. In: *Signal Processing Magazine, IEEE* 25 (2008), May, Nr. 3, S. 15–16. – ISSN 1053-5888

[Gregor u. a. 2010] GREGOR, S. ; RAHIMI, M. ; VOGT, M. ; SCHULZ, T. ; K.V.LUCK: Tangible Interaction - vom Konzept zur Realisierung. (2010)

[Grimaldi und Cummins 2008] GRIMALDI, Marco ; CUMMINS, Fred: Speaker Identification Using Instantaneous Frequencies. In: *IEEE Transactions on Audio, Speech and Language Processing* 16 (2008), Nr. 6, S. 1097–1111. – URL http://dblp.uni-trier.de/db/journals/taslp/taslp16.html#GrimaldiC08

[Guyton 1991] GUYTON, Arthur C.: *Textbook of Medical Physiology*. Philadelphia PA : W. B. Saunders Company, 1991. – ISBN 0-7216-3087-1

[Hannes Mügele 2006] HANNES MÜGELE, Florian S.: LREC06:SmartWeb UMTS Speech Data Collection, The SmartWeb Handheld Corpus. (2006), May

[Hayes-Roth 1985] HAYES-ROTH, Barbara: A blackboard architecture for control. In: *Artif. Intell.* 26 (1985), August, S. 251–321. – URL http://portal.acm.org/citation.cfm?id=4004.4005. – ISSN 0004-3702

[Hickson] HICKSON, I.: *The WebSocket API*. W3C Working Draft 22 December 2009

[Hollatz 2008] HOLLATZ, Dennis: *smart:shelf – Projektbericht*. http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08- 2008. – [Online; accessed 05-Feburary-2011]

[Huang u. a. 2001] HUANG, Xuedong ; ACERO, Alex ; HON, Hsiao-Wuen: *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, 2001. – ISBN 0130226165

[Jacob 1990] JACOB, Robert J. K.: What you look at is what you get: eye movement-based interaction techniques. In: *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 1990, S. 11–18. – ISBN 0-201-50932-6

[Johanson 2003] JOHANSON, Bradley E.: *Application coordination infrastructure for ubiquitous computing rooms*. Stanford, CA, USA, Dissertation, 2003. – AAI3085383

[Krumm 2009] KRUMM, John: *Ubiquitous Computing Fundamentals*. Chapman & Hall/-CRC, 2009. – ISBN 1420093606, 9781420093605

[Kuniavsky 2010] KUNIAVSKY, Mike: *Smart Things: Ubiquitous Computing User Experience Design*. Morgan Kaufmann, 2010. – ISBN 0123748992

[Maganti u. a. 2006] MAGANTI, Hari K. ; MOTLICEK, Petr ; GATICA-PEREZ, Daniel: Unsupervised Speech/Non-speech Detection for Automatic Speech Recognition in Meeting Rooms. Martigny, Switzerland : IDIAP, 2006. – Forschungsbericht

[Mehling 2010] MEHLING, Karin: Heute hier, morgen dort - Deixis und Anaphorik in der Deutschen Gebärdensprache (DGS) Analyse und Vergleich mit der deutschen Lautsprache. (2010)

[Melzer 2008] MELZER, Ingo (Hrsg.): *Service-orientierte Architekturen mit Web Services: Konzepte – Standards – Praxis*. 3. Heidelberg : Spektrum, 2008. – ISBN 978-3-8274-1993-4

[Mills 1989] MILLS, Dave L.: *Network Time Protocol (version 2) specification and implementation*. Network Working Group Request for Comments: 1119. September 1989

[Mills 1985] MILLS, D.L.: *Network Time Protocol (NTP)*. RFC 958. September 1985 (Request for Comments). – URL http://www.ietf.org/rfc/rfc958.txt. – Obsoleted by RFCs 1059, 1119, 1305

[Minker u. a. 2005a] MINKER, Wolfgang (Hrsg.) ; BÜHLER, Dirk (Hrsg.) ; DYBKJÆR, Laila (Hrsg.): *Text, Speech and Language Technology*. Bd. 28: *Spoken Multimodal Human-Computer Dialogue in Mobile Environments*. Dordrecht : Springer, 2005. – 37–57 S. – ISBN 978-1-4020-3073-4

[Minker u. a. 2005b] MINKER, Wolfgang (Hrsg.) ; BÜHLER, Dirk (Hrsg.) ; DYBKJÆR, Laila (Hrsg.): *Text, Speech and Language Technology*. Bd. 28: *Spoken Multimodal Human-Computer Dialogue in Mobile Environments*. Dordrecht : Springer, 2005. – ISBN 978-1-4020-3073-4

[Möller u. a. 2009] MÖLLER, Sebastian ; ENGELBRECHT, Klaus-Peter ; KÜHNEL, Christine ; WECHSUNG, Ina ; WEISS, Benjamin: *Evaluation of Multimodal Interfaces for Ambient Intelligence*. In: AGHAJAN, H. (Hrsg.) ; DELGADO, Ramon Lopez-Cozar (Hrsg.) ; AUGUSTO, J. C. (Hrsg.): *Human-Centric Interfaces for Ambient Intelligence*. Elsevier : untitled, 2009

[Möller u. a. 2004] MÖLLER, Sebastian ; KREBBER, Jan ; RAAKE, Alexander ; SMEELE, Paula ; RAJMAN, Martin ; MELICHAR, Mirek ; PALLOTTA, Vincenzo ; TSAKOU, Gianna ; KLADIS, Basilis ; VOVOS, Anestis ; HOONHOUT, Jettie ; SCHUCHARDT, Dietmar ; FAKOTAKIS, Nikos ; GANCHEV, Todor ; POTAMITIS, Ilyas: INSPIRE: Evaluation of a Smart-Home System for Infotainment Management and Device Control. In: *CoRR* cs.HC/0410063 (2004)

[Nakamura u. a. 1996] NAKAMURA, S. ; TAKIGUCHI, T. ; SHIKANO, K.: Noise and room acoustics distorted speech recognition by HMM composition. In: *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*. Washington, DC, USA : IEEE Computer Society, 1996, S. 69–72. – ISBN 0-7803-3192-3

[Nakashima u. a. 2009] NAKASHIMA, Hideyuki ; AGHAJAN, Hamid ; AUGUSTO, Juan C.: *Handbook of Ambient Intelligence and Smart Environments*. Springer Publishing Company, Incorporated, 2009. – ISBN 0387938079, 9780387938073

[Nakashima u. a. 2010] NAKASHIMA, Hideyuki (Hrsg.) ; AGHAJAN, Hamid (Hrsg.) ; AUGUSTO, Juan C. (Hrsg.): *Handbook of Ambient Intelligence and Smart Environments*. New York : Springer, 2010. – ISBN 978-0-387-93807-3

[Nijholt 2003] NIJHOLT, Prof.dr.ir. A.: *Multimodality and Ambient Intelligence*. 2003. – URL http://doc.utwente.nl/41394/

[O'Shaughnessy 2008] O'SHAUGHNESSY, D.: Invited paper: Automatic speech recognition: History, methods and challenges. In: *PR* 41 (2008), October, Nr. 10, S. 2965–2979

[Oulasvirta u. a. 2007] OULASVIRTA, Antti ; ENGELBRECHT, Klaus-Peter ; JAMESON, Anthony ; MÖLLER, Sebastian: Communication Failures in the Speech-Based Control of Smart Home Systems. In: *Proceedings of the Third International Conference on Intelligent Environments*, 2007, S. 135–143

[Payette 1994] PAYETTE, Julie: Advanced human-computer interface and voice processing applications in space. In: *HLT '94: Proceedings of the workshop on Human Language Technology*. Morristown, NJ, USA : Association for Computational Linguistics, 1994, S. 416–420. – ISBN 1-55860-357-3

[Pfister und Kaufmann 2008a] PFISTER, Beat ; KAUFMANN, Tobias: *Sprachverarbeitung - Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer, 2008. – ISBN 978-3-540-75909-6

[Pfister und Kaufmann 2008b] PFISTER, Beat ; KAUFMANN, Tobias: *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung (Springer-Lehrbuch) (German Edition)*. Springer, 2008. – ISBN 3540759093

[Phonetics Flash Animation Project, University of Iowa,USA 2008] PHONETICS FLASH ANIMATION PROJECT, UNIVERSITY OF IOWA,USA: *Phonetics: The sounds of German*. http://www.uiowa.edu/~acadtech/phonetics/german/frameset.html. 2008. – [Online; accessed 12-December-2008]

[Poh u. a. 2010] POH, Ming-Zher ; MCDUFF, Daniel J. ; PICARD, Rosalind W.: Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. In: *Opt. Express* 18 (2010), May, Nr. 10, S. 10762–10774. – URL http://www.opticsexpress.org/abstract.cfm?URI=oe-18-10-10762

[Potamitis u. a. 2003] POTAMITIS, Ilyas ; GEORGILA, K ; FAKOTAKIS, N. ; KOKKINAKIS, George: An integrated system for smart-home control of appliances based on remote speech interaction. In: *EUROSPEECH2003*, ISCA, 09/2003 2003, S. 0. – URL http://www.isca-speech.org/archive/eurospeech_2003/e03_2197.html

[Rabiner 1989] RABINER, L. R.: A tutorial on hidden Markov models and selected applications in speech recognition. In: *Proceedings of the IEEE* 77 (1989), Nr. 2, S. 257–286. – URL http://dx.doi.org/10.1109/5.18626

[Rabiner 2003] RABINER, Lawrence: COMPUTER SCIENCE: The Power of Speech. In: *Science* 301 (2003), Nr. 5639, S. 1494–1495. – URL http://www.sciencemag.org

[Raisamo 1999] RAISAMO, Roope: *Multimodal Human-Computer Interaction: a constructive and empirical study*. 1999

[Ramos u. a. 2008] RAMOS, Carlos ; AUGUSTO, Juan C. ; SHAPIRO, Daniel: Ambient Intelligencethe Next Step for Artificial Intelligence. In: *IEEE Intelligent Systems* 23 (2008), Nr. 2, S. 15–18

[Reithinger und Blocher 2003] REITHINGER, Norbert ; BLOCHER, Anselm: SmartKom - Multimodale Mensch-Technik-Interaktion (SmartKom - Multimodal Human Computer Interaction). In: *i-com* 2 (2003), Nr. 1, S. 4–10

[Ringlstetter u. a. 2007] RINGLSTETTER, Christoph ; SCHULZ, Klaus U. ; MIHOV, Stoyan: Adaptive text correction with Web-crawled domain-dependent dictionaries. In: *ACM Trans. Speech Lang. Process.* 4 (2007), Nr. 4, S. 9. – ISSN 1550-4875

[Skubic u. a. 2009] SKUBIC, Marjorie ; ALEXANDER, Gregory ; POPESCU, Mihail ; RANTZ, Marilyn ; KELLER, James: A smart home application to eldercare: Current status and lessons learned. In: *Technol. Health Care* 17 (2009), August, S. 183–201. – URL http://portal.acm.org/citation.cfm?id=1605363.1605370. – ISSN 0928-7329

[Sokollek 2010] SOKOLLEK, Wolfram: *smart:shelf – Projektbericht*. http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-p 2010. – [Online; accessed 05-Feburary-2011]

[Tanenbaum und Steen 2006] TANENBAUM, Andrew S. ; STEEN, Maarten V.: *Distributed Systems: Principles and Paradigms (2nd Edition)*. Prentice Hall, 2006. – ISBN 0132392275

[Treichler 2009] TREICHLER, J.: Signal processing: a view of the future, part 2. In: *Signal Processing Magazine, IEEE* 26 (2009), May, Nr. 3, S. 83–86. – ISSN 1053-5888

[Turing 1995] TURING, A. M.: Computing machinery and intelligence. (1995), S. 11–35. ISBN 0-262-56092-5

[Turunen u. a. 2009a] TURUNEN, Markku ; KALLINEN, Aleksi ; SÀNCHEZ, Ivàn ; RIEKKI, Jukka ; HELLA, Juho ; OLSSON, Thomas ; MELTO, Aleksi ; RAJANIEMI, Juha-Pekka ; HAKULINEN, Jaakko ; MÄKINEN, Erno ; VALKAMA, Pellervo ; MIETTINEN, Toni ; PYYKKÖNEN, Mikko ; SALORANTA, Timo ; GILMAN, Ekaterina ; RAISAMO, Roope: Multimodal interaction with speech and physical touch interface in a media center application. In: *ACE '09: Proceedings of the International Conference on Advances in Computer Enterntainment Technology*. New York, NY, USA : ACM, 2009, S. 19–26. – ISBN 978-1-60558-864-3

[Turunen u. a. 2009b] TURUNEN, Markku ; MELTO, Aleksi ; HELLA, Juho ; HEIMONEN, Tomi ; HAKULINEN, Jaakko ; MÄKINEN, Erno ; LAIVO, Tuuli ; SORONEN, Hannu: User expectations and user experience with different modalities in a mobile phone controlled home

entertainment system. In: *MobileHCI '09: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA : ACM, 2009, S. 1–4. – ISBN 978-1-60558-281-8

[Vertegaal u. a. 2006] VERTEGAAL, Roel ; SHELL, Jeffrey S. ; CHEN, Daniel ; MAMUJI, Aadil: Designing for augmented attention: Towards a framework for attentive user interfaces. In: *Computers in Human Behavior* 22 (2006), Nr. 4, S. 771 – 789. – URL http://www.sciencedirect.com/science/article/B6VDC-4J84SP8-2/2/9773a0 – Attention aware systems - Special issue: Attention aware systems. – ISSN 0747-5632

[Wahlster 2006] WAHLSTER, Wolfgang (Hrsg.): *SmartKom: Foundations of Multimodal Dialogue Systems*. Berlin : Springer, 2006. – ISBN 3-540-23732-1

[Want 2011] WANT, Roy: *http://www.roywant.com/cv/vita.htm*. WWW. 2011

[Weiser 1999] WEISER, Mark: The computer for the 21st century. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 3 (1999), Nr. 3, S. 3–11. – ISSN 1559-1662

[Winograd 2001] WINOGRAD, Terry: Architectures for context. In: *Hum.-Comput. Interact.* 16 (2001), Nr. 2, S. 401–419. – ISSN 0737-0024

[Wolters u. a. 2009] WOLTERS, Maria ; GEORGILA, Kallirroi ; LOGIE, Robert ; MACPHERSON, Sarah ; MOORE, Johanna ; WATSON, Matt: Reducing Working Memory Load in Spoken Dialogue Systems. In: *Interacting with Computers* 21 (2009), Nr. 4, S. 276–287

[Yankelovich 1996] YANKELOVICH, Nicole: How do users know what to say? In: *interactions* 3 (1996), Nr. 6, S. 32–43. – ISSN 1072-5520

Tabellenverzeichnis

Abbildungsverzeichnis

A. Anhang

A.1. Beispielgrammatik

Listing 1: Beispiel Grammatik für den Spracherkenner und die Anwendung Routenplanung

```
<Grammar description="Application.Navigation.GoogleMaps">
  <Phrase>
    <Fragment  index="0" minimum="1" maximum="1" term=""
       semanticKey="" semanticValue="">
      <Choices description="ShowKlauseln">
        <Choice>
          <Fragment index="0" minimum="1" maximum="1" iscomposite="
             True">
            <Fragment index="0" minimum="1" maximum="1" term="wo"
               semanticKey="command" semanticValue="Application.
               Navigation.GoogleMaps.show" />
            <Fragment index="0" minimum="1" maximum="1" term=""
               semanticKey="" semanticValue="">
              <Choices description="FormenVonSein">
                <Choice>
                  <Fragment  index="0" minimum="1" maximum="1" term
                     ="bin" semanticKey="verb" semanticValue="" />
                </Choice>
                <Choice>
                  <Fragment  index="0" minimum="1" maximum="1" term
                     ="ist" semanticKey="verb" semanticValue="" />
                </Choice>
                <Choice>
                  <Fragment  index="0" minimum="1" maximum="1" term
                     ="sind" semanticKey="verb" semanticValue="" />
                </Choice>
                <Choice>
                  <Fragment  index="0" minimum="1" maximum="1" term
                     ="seid" semanticKey="verb" semanticValue="" />
                </Choice>
              </Choices>
            </Fragment>
            <Fragment  index="0" minimum="1" maximum="1" term=""
               semanticKey="" semanticValue="">
```

```
25              <Choices description="FormenVonIch">
26                <Choice>
27                  <Fragment entity="edc053f1 160d 4ceb be9f 7
                        ae2564492d2" index="0" minimum="1" maximum="1
                        " term="ich" semanticKey="parameter"
                        semanticValue="edc053f1 160d 4ceb be9f 7
                        ae2564492d2" />
28                </Choice>
29                <Choice>
30                  <Fragment entity="edc053f1 160d 4ceb be9f 7
                        ae2564492d2" index="0" minimum="1" maximum="1
                        " term="meine␣wenigkeit" semanticKey="
                        parameter" semanticValue="edc053f1 160d 4ceb
                        be9f 7ae2564492d2" />
31                </Choice>
32              </Choices>
33            </Fragment>
34          </Fragment>
35        </Choice>
36        <Choice>
37          <Fragment index="0" minimum="1" maximum="1" iscomposite=
              "True">
38            <Fragment index="0" minimum="1" maximum="1" term="
                Zeige" semanticKey="command" semanticValue="
                Application.Navigation.GoogleMaps.show" />
39            <Fragment index="0" minimum="0" maximum="1" term="mir"
                semanticKey="possessiv" semanticValue="" />
40            <Fragment index="0" minimum="1" maximum="1"
                iscomposite="True">
41              <Fragment index="0" minimum="0" maximum="1" term=""
                  semanticKey="" semanticValue="">
42                <Choices description="Artikel">
43                  <Choice>
44                    <Fragment index="0" minimum="1" maximum="1"
                        term="der" semanticKey="Artikel"
                        semanticValue="" />
45                  </Choice>
46                  <Choice>
```

```
47                  <Fragment index="0" minimum="1" maximum="1"
                        term="die" semanticKey="Artikel"
                        semanticValue="" />
48                </Choice>
49                <Choice>
50                  <Fragment index="0" minimum="1" maximum="1"
                        term="das" semanticKey="Artikel"
                        semanticValue="" />
51                </Choice>
52                <Choice>
53                  <Fragment index="0" minimum="1" maximum="1"
                        term="den" semanticKey="Artikel"
                        semanticValue="" />
54                </Choice>
55                <Choice>
56                  <Fragment index="0" minimum="1" maximum="1"
                        term="ein" semanticKey="Artikel"
                        semanticValue="" />
57                </Choice>
58                <Choice>
59                  <Fragment index="0" minimum="1" maximum="1"
                        term="einen" semanticKey="Artikel"
                        semanticValue="" />
60                </Choice>
61                <Choice>
62                  <Fragment index="0" minimum="1" maximum="1"
                        term="eine" semanticKey="Artikel"
                        semanticValue="" />
63                </Choice>
64              </Choices>
65            </Fragment>
66            <Fragment index="0" minimum="1" maximum="1" term=""
                semanticKey="showparameter" semanticValue="">
67              <Choices referenceurl="knowledgebase:http://
                    localhost:55119/KnowledgeBaseQueryRESTService.
                    svc/QueryHTTPasXML?entityId=cb0ab981
                     6189 4087 919e 2838f5876782&amp;
                    subEntityTypeFilter=/spatial/location/global&amp
                    ;subEntityPropertiesCSV=longitude,latitude"
                    description="" />
```

```
68                </Fragment>
69              </Fragment>
70            </Fragment>
71          </Choice>
72        </Choices>
73      </Fragment>
74    </Phrase>
75    <Phrase>
76      <Fragment  index="0" minimum="1" maximum="1" term=""
          semanticKey="" semanticValue="">
77        <Choices  description="ZoomKommandos">
78          <Choice>
79            <Fragment  index="0" minimum="1" maximum="1" term="Zoome"
                semanticKey="command" semanticValue="Application.
                Navigation.GoogleMaps.zoom" />
80          </Choice>
81          <Choice>
82            <Fragment  index="0" minimum="1" maximum="1" term="
                vergrï¿¬ï¿¬ern" semanticKey="command" semanticValue="
                Application.Navigation.GoogleMaps.zoom in" />
83          </Choice>
84          <Choice>
85            <Fragment  index="0" minimum="1" maximum="1" term="
                verkleinern" semanticKey="command" semanticValue="
                Application.Navigation.GoogleMaps.zoom out" />
86          </Choice>
87        </Choices>
88      </Fragment>
89      <Fragment  index="0" minimum="0" maximum="1" term=""
          semanticKey="zoomvalue" semanticValue="">
90        <Choices  description="ZoomParameter">
91          <Choice>
92            <Fragment  index="0" minimum="1" maximum="1" term="rein"
                semanticKey="parameter" semanticValue="in" />
93          </Choice>
94          <Choice>
95            <Fragment  index="0" minimum="1" maximum="1" term="raus"
                semanticKey="parameter" semanticValue="out" />
96          </Choice>
97          <Choice>
```

```
98          <Fragment index="0" minimum="1" maximum="1" term="hinein
                " semanticKey="parameter" semanticValue="in" />
99        </Choice>
100       <Choice>
101         <Fragment index="0" minimum="1" maximum="1" term="heraus
                " semanticKey="parameter" semanticValue="out" />
102       </Choice>
103       <Choice>
104         <Fragment index="0" minimum="1" maximum="1" term="weg"
                semanticKey="parameter" semanticValue="out" />
105       </Choice>
106       <Choice>
107         <Fragment index="0" minimum="1" maximum="1" term="hin"
                semanticKey="parameter" semanticValue="in" />
108       </Choice>
109     </Choices>
110   </Fragment>
111 </Phrase>
112 <Phrase>
113   <Fragment index="0" minimum="1" maximum="1" term="weiter"
        semanticKey="command" semanticValue="Application.Navigation.
        GoogleMaps.repeat" />
114 </Phrase>
115 <Phrase>
116   <Fragment index="0" minimum="1" maximum="1" term="Route"
        semanticKey="command" semanticValue="Application.Navigation.
        GoogleMaps.route" />
117   <Fragment index="0" minimum="1" maximum="1" term="von"
        semanticKey="test" semanticValue="" />
118   <Fragment index="0" minimum="1" maximum="1" term=""
        semanticKey="route_start" semanticValue="">
119     <Choices referenceurl="knowledgebase:http://localhost:55119/
          KnowledgeBaseQueryRESTService.svc/QueryHTTPasXML?entityId=
          cb0ab981 6189 4087 919e 2838f5876782&amp;
          subEntityTypeFilter=/spatial/location/global&amp;
          subEntityPropertiesCSV=longitude,latitude" description=""
          />
120   </Fragment>
121   <Fragment index="0" minimum="1" maximum="1" term="nach"
        semanticKey="test1" semanticValue="" />
```

```
122     <Fragment  index="0" minimum="1" maximum="1" term=""
           semanticKey="route_finish" semanticValue="">
123       <Choices referenceurl="knowledgebase:http://localhost:55119/
             KnowledgeBaseQueryRESTService.svc/QueryHTTPasXML?entityId=
             cb0ab981 6189 4087 919e 2838f5876782&amp;
             subEntityTypeFilter=/spatial/location/global&amp;
             subEntityPropertiesCSV=longitude,latitude" description=""
             />
124     </Fragment>
125   </Phrase>
126    <Phrase>
127     <Fragment index="0" minimum="1" maximum="1" term="Route"
           semanticKey="command" semanticValue="Application.Navigation.
           GoogleMaps.route_set_route_start" />
128     <Fragment index="0" minimum="1" maximum="1" term="von"
           semanticKey="test" semanticValue="" />
129     <Fragment index="0" minimum="1" maximum="1" term="hier"
           semanticKey="location" semanticValue="Application.Navigation
           .GoogleMaps.Constant.Here" />
130   </Phrase>
131    <Phrase>
132     <Fragment index="0" minimum="1" maximum="1" term="Ã-ber"
           semanticKey="command" semanticValue="Application.Navigation.
           GoogleMaps.set_route_over" />
133     <Fragment index="0" minimum="1" maximum="1" term="hier"
           semanticKey="location" semanticValue="Application.Navigation
           .GoogleMaps.Constant.Here" />
134   </Phrase>
135    <Phrase>
136     <Fragment index="0" minimum="1" maximum="1" term="nach"
           semanticKey="command" semanticValue="Application.Navigation.
           GoogleMaps.route_set_route_end" />
137     <Fragment index="0" minimum="1" maximum="1" term="hier"
           semanticKey="location" semanticValue="Application.Navigation
           .GoogleMaps.Constant.Here" />
138   </Phrase>
139   <Phrase>
140     <Fragment  index="0" minimum="1" maximum="1" term="Suche"
           semanticKey="command" semanticValue="Application.Navigation.
           GoogleMaps.search" />
```

```
141     <Fragment  index="0" minimum="1" maximum="1" term=""
            semanticKey="parameter" semanticValue="" />
142   </Phrase>
143 </Grammar>
```

A.2. Darmstadt Challenge

Der Fragenkatalog für die Evaluierung eines Smart-Homes.

> Sample of evaluation for a Smart Home:
>
> - Does the introduction of AmI technology change the look or feel of the house?
> - What changes in daily life are needed to make use of AmI technology?
> - For how much of the house is smart home assistance available?
> - How much effort is required to request assistance from the home?
> - Does the quality of the assistance increase with use and time?
> - Does the assistance customize itself to the residents of the home?
> - Does the assistance improve your productivity at home?
> - Does the assistance improve your health and/or safety at home?
> - Which aspects of the Smart Home were useful?
> - Which aspects were disappointing?
> - Would you recommend use of the Smart Home to a friend or family member?

(Augusto u. a., 2009, S.6)

Versicherung über Selbstständigkeit

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

Hamburg, 7. Februar 2011
Ort, Datum

Unterschrift