Magisterarbeit

Integration eines Spracherkenners in ein Rauminformationssystem

erstellt von

Stefan Schaffer

[redacted]

Betreuer:

Prof. Dr. Stefan Weinzierl

FG Audiokommunikation, TU Berlin

Dr. Florian Metze

Quality and Usability Lab, TU Berlin

Technische Universität Berlin, Fachgebiet Audiokommunikation

Institut für Sprache und Kommunikation

Berlin, September 2008

Danksagung

Zunächst möchte ich Professor Dr. Stefan Weinzierl vom Fachgebiet Audiokommunikation der TU Berlin für die Übernahme der Betreuung dieser Arbeit danken, die die Anfertigung überhaupt erst möglich machte.

Mein Betreuer, Dr. Florian Metze, gab mir die Gelegenheit, im Rahmen dieser Magisterarbeit bei den Deutsche Telekom Laboratories tätig zu sein, und hatte stets ein offenes Ohr für meine Fragen und Probleme. Dafür möchte ich meinen außerordentlichen Dank aussprechen. Die Arbeit im inspirierenden Umfeld der T-Labs vereinte spannende Themen, neue Herausforderungen und Spaß, so dass mein Interesse an der Forschung noch wuchs und ich täglich gerne dazulernte.

Besonderer Dank gilt Ina, Christine und Julia von den T-Labs, die mich in wertvollen Gesprächen mit Anmerkungen und Anregungen versorgten, und mir mit ihrem Wissen und Erfahrungen beratend zur Seite standen. Danke auch an die Schar von Kollegen, die ihre Zeit für die Teilnahme an meinen Experimenten geopfert haben. Der Kicker-Crew sei herzlich für die stets gute Stimmung und die gemeinsamen revitalisierenden Momente gedankt.

Bettina, Bernd und Oli danke ich für den bereitwilligen Einsatz ihrer korrigierenden Kenntnisse der Sprachvarietäten des Deutschen.

Ebenso möchte ich meiner Familie für Unterstützung und Geduld während meines Studiums danken: Ohne euren Zuspruch und eure Förderung wäre ich so weit vielleicht gar nicht gekommen.

Schließlich danke ich meinen Freunden, die mir während der Magisterarbeitszeit Zuwendung, Kraft, Ablenkung und Ermutigung spendeten, und vor allem Sonja, die mich zum Glück regelmäßig daran erinnerte, dass das Leben noch mehr zu bieten hat, als die Integration von Spracherkennern.

Live long and prosper!

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Ziel der Arbeit 2

1.2 Gliederung 3

2 Multimodale Dialogsysteme **5**

2.1 Aufbau eines multimodalen Dialogsystems 6

2.1.1 Eingabemodule 6

2.1.2 Multimodale Verarbeitung 9

2.1.3 Ausgabemodule 10

2.2 Dialogsysteme als Zustandsautomaten 12

2.3 Testkriterien und Evaluation 13

2.3.1 Technische Kriterien 15

2.3.2 Usability Evaluation 16

2.3.3 Benutzerevaluation 18

3 Automatische Spracherkennung **21**

3.1 Die Fundamentalformel der Spracherkennung 21

3.2 Aufbau eines Spracherkenners 22

3.2.1 Vorverarbeitung 23

3.2.2 Akustische Modelle 24

3.2.3 Sprachmodelle 25

3.2.4 Dekodierung 25

3.3 Grammatik und Vokabular 27

4 Architektur des Rauminformationssystems **31**

4.1 Die Arbeitsplatz-Map (AP-Map) 31

4.2 Der Spracherkenner 33

4.3 Datenbankanbindung 36

5 Integration der Sprachsteuerung **39**

5.1 Integrationskonzept 40
5.2 Protokolldefinition 42
5.3 Voice-AP-Map Interface 46
5.3.1 Implementierung der Schnittstelle 46
5.3.2 Integration in Dialogmanager 48
5.4 Erweiterungen der AP-Map 49

6 Tests und Evaluierung 51
6.1 Fragestellung 51
6.2 Vortests und Testmethode 53
6.2.1 Vortests 53
6.2.2 Testmethode 55
6.2.2.1 Aufgaben 55
6.2.2.2 Probanden 55
6.2.2.3 Versuchsmaterial 56
6.2.2.4 Versuchsaufbau 58
6.2.2.5 Versuchsablauf 59
6.3 Technische Tests und Performanz Daten 61
6.4 Subjektive Daten 64
6.5 Vergleich von subjektiven Daten und Performanz Daten 67
6.6 Diskussion und Implikationen 68
6.6.1 Diskussion der Versuchsergebnisse 68
6.6.2 Weitere Implikationen 73

7 Zusammenfassung 75

A Anhang 79
A.1 Grammatik 79
A.2 Instruktionen 81
A.3 Personenbezogener Fragebogen 85
A.4 Aufgaben 87
A.5 SASSI 89

Literaturverzeichnis 95

Abbildungsverzeichnis

2.1 Allgemeine Architektur eines multimodalen Dialogsystems, nach Delago und Araki [DA05] 7
2.2 Links: Zustandsgraph eines Endlichen Automaten Rechts: Beispiel eines einfachen Dialogsystems als Zustandsautomat 12

3.1 Allgemeine Architektur eines statistischen Spracherkennungssystems, nach [Met05] 22
3.2 Beispiel eines Hidden-Markov-Modell für die Spracherkennung, nach [YEG+06] 23
3.3 Hypothesengraph 26
3.4 Tiefensuche mit einer vollständig expandierten Hypothese (links). Breitensuche mit parallel expandierten Hypothesen (rechts). 26
3.5 Ein einfaches Dialogsystem mit unterschiedlichen aktiven Regeln pro Zustand 29

4.1 Ein Anwender bei der Verwendung der AP-Map. 31
4.2 Eventscreen 32
4.3 Stockwerkanzeige mit einem markierten Raum und einem Detailfenster. 32
4.4 Mitarbeitersuche 33
4.5 Bestehende Infastruktur der sprachverarbeitenden Einheit vor der Integration 34
4.6 Datenbankanbindung 37

5.1 Infastruktur des Gesamtsystems. Die hier beschriebenen Module sind blau hinterlegt. 39
5.2 Überblick über den im Dialogmanager implementierten Zustandsautomat. 41
5.3 Nachrichtenfluss bei einem Touch-Event. 45
5.4 Nachrichtenfluss bei einem ASR-Event bei Synchronität. 46
5.5 Nachrichtenfluss bei einem ASR-Event. 47
5.6 Klassendiagramm des Voice-AP-Map Interface. 48

6.1 Skizzierter Versuchsaufbau 59
6.2 Versuchsablauf eines Probanden 60
6.3 Bewertungen der einzelnen SASSI Skalen und der Gesamtqualität. Die Fehlerbalken stellen eine Standardabweichung dar. 65

Tabellenverzeichnis

2.1 Sensorische Wahrnehmung, Sinnesorgane und Modalitäten [Met07] 5

3.1 Wortschatzumfang potentieller Spracherkennungsanwendungen, nach [ST95] 29

4.1 Größe des Vokabulars mit und ohne Aussprachevarianten. 36

5.1 Vom Dialogmanager unterschiedene Zustände und korrespondierende Bildschirmanzeigen der AP-Map. 40

5.2 Funktionsnamen und Parameter, die vom Dialogmanager an die Voice-AP-Map übertragen werden. 43

6.1 Minimal nötige Anzahl an Interaktionsschritten pro Aufgabe 55

6.2 Anpassung der SASSI Items 7, 16 und 22. 57

6.3 Durchschnittliche Bearbeitungszeiten der Modalitäten in [Minuten:Sekunden] 61

6.4 Ergebnisse des T-Tests zur Expertise der Probanden 62

6.5 Bevorzugte Modalität bei multimodaler Bedienung in [%]; $p<0.05$. 62

6.6 Benutzung von Sprache bei multimodaler Bedienung in [%]. Gruppe A hatte Sprache als ersten Block. Gruppe B hatte Sprache als zweiten Block. 63

6.7 Benutzung von Sprache durch Experten/Novizen bei multimodaler Bedienung in [%]. 63

6.8 CER für Bedienung mit Sprache in [%] und Anzahl an Spracherkennungsfehlern. 64

6.9 OOG-rate für Bedienung mit Sprache in [%] 64

6.10 Einfluss der Modalitäten auf die Benutzerwahrnehmung; $p<0.05$ . . 65

Abkürzungen

A/D	Analog/Digital
AP	Arbeitsplatz
ASR	Automatic Speech Recognition
BN	Broadcast News
CER	Concept Error Rate
CFG	Context-Free Grammar
CW	Cognitive Walkthrough
D	Deletion
EE	Experimentelle Evaluation
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HCI	Human Computer Interaction
HE	Heuristische Evaluation
HMM	Hidden-Markov-Modell
I	Insertion
JRTk	Janus Recognition Toolkit
JSGF	Java Speech Grammar Format
LDA	Lineare Diskriminanz Analyse
MFCC	Mel Frequency Cepstral Coeffcients
N	Number (dt. Anzahl)
NLG	Natural Language Generation
NLP	Natural Language Processing
OOG	Out Of Grammar
PARADISE	Paradigm for Dialogue System Evaluation
PDA	Personal Digital Assistant
PROMISE	Procedure for Multimodal Interactive System Evaluation
S	Substitution
SASSI	Subjective Assessment of Speech System Interfaces
SUMI	Software Usability Measurement Inventory
TTS	Text-to-speech
VAD	Voice Activity Detection
WER	Word Error Rate

Bezeichnungen

H	Hidden-Markov-Modell
M	Mittelwert
O	Beobachtungssequenz
$P(W)$	A-priori-Wahrscheinlichkeit für Ereignis W
$P(W\|O)$	A-posteriori-Wahrscheinlichkeit
SD	Standard Deviation (dt. Standardabweichung)
W	Wortsequenz

1 Einleitung

In kaum ein Lebensgebiet haben technische Systeme noch nicht Einzug erhalten. Für viele Menschen ist es bereits Normalität, dass Dienstleistungen wie Telefonauskunft oder Fahrkartenverkauf heutzutage nicht mehr von einem menschlichen Gegenüber ausgeführt werden. Um eine angemessene Möglichkeit zur Bedienung solcher Systeme zu schaffen, sind ausgefeilte Schnittstellen für den menschlichen Benutzer nicht wegzudenken. Aus diesem Grund herrscht auf dem Gebiet der Mensch-Maschine-Interaktion bereits seit vielen Jahren rege Forschungsaktivität [Cos05].

Der rasante technologische Fortschritt bringt immer leistungsfähigere Geräte hervor. Moderne Computersysteme erlauben es, rechenintensive Aufgaben wie Spracherkennung oder Gesichtserkennung in Echtzeit durchzuführen [CCC+96, GY93]. Zum Teil werden diese neuen Interaktionsformen bereits in kommerziellen Systemen genutzt. So z. B. die automatische Spracherkennung bei Navigationsgeräten im Automobil. Andere Anwendungen, wie Blickrichtungserkennung, rücken in greifbare Nähe, sind aber in den meisten Fällen über prototypische Anwendungen noch nicht hinaus.

Natürliche Sprache nimmt einen besonderen Stellenwert in der zwischenmenschlichen Kommunikation ein. Ihr wird zugeschrieben, die für den Menschen natürlichste Form der Kommunikation zu sein. Daher ist es wünschenswert, Sprache auch zur Mensch-Maschine-Interaktion effizient einsetzen zu können [WWL00]. Vor allem im Bereich der telefon- und internetbasierten Informations- und Service-Angebote sind Sprachdialogsysteme heutzutage bereits weit verbreitet [KWL99].

Einen zukunftsweisenden Trend im Bereich der Mensch-Maschine-Interaktion stellen multimodale Systeme dar. Multimodal bedeutet, dass mehrere Medien oder Kommunikationsmittel zur Kommunikation verwendet werden. So kann z. B. der akustische Kanal zur auditiven Wahrnehmung und der optische Kanal zur visuellen Wahrnehmung parallel genutzt werden [Möl07]. Seit dem Aufkommen von Richard Bolt's [Bol80] „Put That There“ System, das Sprache parallel mit händischem Deuten

verarbeitet, ist eine Vielzahl von multimodalen Systemen entstanden. Die meisten der bisher entwickelten Systeme sind Prototypen, die HCI-Forschungsprojekten (HCI: Human-Computer-Interaction[1]) entstammen. Multimodale Dialogsysteme werden im nächsten Kapitel noch ausführlich betrachtet.

Das Rauminformationssystem *Voice-AP-Map*[2] ist ein solches multimodales Dialogsystem, das in den *Deutsche Telekom Laboratories (T-Labs)*[3] entwickelt wird. Es versorgt den Nutzer mit Informationen über Räume und Raumbuchungen in den *T-Labs* und bietet die Möglichkeit, die aktuellen Buchungen eines Mitarbeiters zu suchen und anzuzeigen. Neben der Bereitstellung dieser Funktionalität ist die verwendete Benutzerschnittstelle ein zentraler Aspekt bei der Entwicklung eines solchen Systems. Die *Voice-AP-Map* ermöglicht eine taktile Interaktion, über einen großen an einer Wand montierten Touchscreen, und eine Sprachsteuerung. Dabei sind alle wesentlichen Funktionen per Sprache oder per Touchscreen nutzbar. Eine möglichst intuitive Bedienung des Systems ist wünschenswert, da die Akzeptanz seitens des Benutzers unter anderem von dem für die Verwendung notwendigen Lernaufwand abhängt.

Die Erweiterung des Rauminformationssystems, um eine weitere Eingabemodalität, soll der Benutzerfreundlichkeit des Systems zuträglich sein. Die Benutzerfreundlichkeit ist maßgeblich durch den Begriff der Qualität und der Gebrauchstauglichkeit[4] bestimmt. Diese Begriffe haben eine herausragende Bedeutung in der Informations- und Kommunikationstechnik sowie der Informatik erlangt und stellen in diesen Bereichen eine eigene Forschungsdisziplin dar [Nie93].

1.1 Ziel der Arbeit

Im Rahmen dieser Arbeit wurde die *AP-Map* um eine Sprachsteuerung erweitert. Das resultierende System ist die *Voice-AP-Map*. Durch die Integration der Sprachsteuerung wurde ein multimodales Dialogsystem geschaffen, das es dem Benutzer ermöglicht, Aktionen wahlweise per Touchscreen oder per Sprache auszuführen. Ziel dieser Arbeit ist es, die Integration der Sprachsteuerung zu dokumentieren und das entstandene Gesamtsystem auf Qualität und Gebrauchstauglichkeit zu untersuchen.

[1] Deutsch: Mensch-Maschine-Interaktion
[2] AP steht für „**A**rbeits**P**latz“
[3] `http://www.laboratories.telekom.com`
[4] Engl.: Usability

Auf Grundlage der Evaluation sollen Vorschläge zur Verbesserung des Systems gemacht werden.

Die Evaluation multimodaler Dialogsysteme ist Gegenstand aktueller Forschung. Um relevante Parameter zur Bestimmung der Benutzerwahrnehmung solcher Systeme zu ermitteln, ist es notwendig, Evaluationsdaten, wie die dieser Arbeit, mit den bekannten Parametern zu generieren [Möl08a]. Auf Grundlage dieser Daten können vergleichende Untersuchungen mit anderen Evaluationen von multimodalen Dialogsystemen oder Erweiterungen der *Voice-AP-Map* durchgeführt werden.

1.2 Gliederung

Im Anschluss an diese Einleitung werden in Kapitel 2 multimodale Dialogsysteme im Allgemeinen betrachtet. Neben dem Aufbau werden die Modellierung basierend auf Zustandsautomaten und Evaluationsmethoden für multimodale Dialogsysteme erörtert. Kapitel 3 ist speziell dem Thema automatische Spracherkennung gewidmet und liefert einen Überblick über die wichtigsten Konzepte moderner Spracherkennungssysteme.

Im praktischen Teil dieser Arbeit soll die Integration des Spracherkenners beschrieben werden. Daher wird in Kapitel 4 zunächst die Architektur der einzelnen Module vor der Integration beschrieben. Die Implementierungen für die Kommunikationsschnittstelle zwischen dem Rauminformationssystem und dem Spracherkenner sind Gegenstand von Kapitel 5. Dabei steht im Vordergrund eine anschauliche Dokumentation der Konzeptionierung des Interface und der Modifikationen am Dialogmanager zu liefern.

Die durchgeführten Tests und die Testergebnisse werden in Kapitel 6 dargestellt. Neben einleitenden Ausführungen zu Fragestellung und Hypothesen finden sich die Diskussion der Ergebnisse und Ausblicke für zukünftige Arbeiten am Ende des Kapitels. Abschließend bietet Kapitel 7 eine Zusammenfassung der Arbeit.

2 Multimodale Dialogsysteme

Dialogsysteme sollen den Benutzer auf möglichst einfache Weise mit Informationen versorgen. Dabei geht ein Mensch einen Dialog[1] mit einem Computersystem ein. Bei multimodalen Dialogsystemen können mehrere Modalitäten zur Interaktion verwendet werden. [Möl07] beschreibt in diesem Zusammenhang den Begriff des *Mediums*, als

> *„[...] ein Kommunikationsmittel (Material oder Gerät), welches einen bestimmten physikalischen (z. B. akustischen, optischen) Kanal benutzt, [...]“*

und den Begriff der *Modalität*, als

> *„[...] die Verwendung dieses Mediums zur Kommunikation, z. B. in Form von Intonation, Blick, Geste, Mimik, etc.“*

Tabelle 2.1 zeigt, welche Modalitäten für die menschliche Wahrnehmung identifiziert werden können. Für die Mensch-Maschine-Interaktion sind hierbei vor allem die visuelle, die auditive und die taktile Modalität von Interesse [Cha94].

Sensorische Wahrnehmung	Sinnesorgan	Modalität
Sehsinn	Augen	visuell
Hörsinn	Ohren	auditiv
Tastsinn	Haut	taktil
Geruchssinn	Nase	olfaktorisch
Geschmackssinn	Zunge	gustatorisch
Gleichgewichtssinn	Vestibularorgan	vestibular

Tabelle 2.1: Sensorische Wahrnehmung, Sinnesorgane und Modalitäten [Met07]

Bislang ist keine umfassende Theorie der Modalitäten verfügbar [Möl07]. [Ber02] beschreibt Eigenschaften von Ein- und Ausgabemodalitäten. Abhängig von den dort beschriebenen Eigenschaften sind bestimmte Modalitäten offenbar mehr oder weniger geeignet, bestimmte Informationen zu transportieren. Zusätzlich ist es auch von der

[1] von altgriech. dialégesthai: sich unterhalten, sich unterreden; dialogein: einander zurechnen

Aufgabe abhängig, welche Modalitäten sich zur Integration in das jeweilige System anbieten.

Multimodale Interaktion bringt Vorteile auf der Eingabeseite, bei der Verarbeitung und auf der Ausgabeseite des Systems mit sich. Nutzer können z. B. die Eingabemodalität aufgrund persönlicher Präferenzen oder Umgebungsbedingugen wählen. Bei der Verarbeitung können Schwächen einzelner Modalitäten kompensiert werden. Auf der Ausgabeseite kann die Information über die geeignetste Modalität oder auch mehrere Modalitäten gleichzeitig dargeboten werden. Eine ausführliche Beschreibung des Nutzens multimodaler Interaktion liefert [DA05]. Der erste Abschnitt des vorliegenden Kapitels dient der Beschreibung der technischen Komponenten. In Abschnitt zwei wird die zustandsbasierte Modellierung von Dialogsystemen vorgestellt. Die Evaluation multimodaler Dialogsysteme ist Gegenstand des dritten Abschnitts.

Eine Liste von Beispielen für multimodale Dialogsysteme ist bei [Lar08] zu finden. Bislang haben es nur sehr wenige Systeme bis zur Anwendungsreife gebracht.

2.1 Aufbau eines multimodalen Dialogsystems

Die allgemeine Architektur eines multimodalen Dialogsystems ist in Abbildung 2.1 dargestellt. Die einzelnen Elemente der Eingabe-Schnittstelle (*Input Interface*), der Einheit zur Verarbeitung der multimodalen Informationen (*Multimodal Processing*) und der Ausgabe Schnittstelle (*Output Interface*) werden in den folgenden Abschnitten erläutert.

Die Ausführungen in diesem Abschnitt sind zum Teil [DA05] und [Möl07] entnommen.

2.1.1 Eingabemodule

Auf der Eingabeseite (Input interface) werden unterschiedliche Module zur Verarbeitung der einzelnen Eingabemodalitäten bereitgestellt, von denen jedes ein eigenes großes Forschungsgebiet repräsentiert. Daher können nicht alle Module im Rahmen dieser Arbeit ausführlich betrachtet werden. Da die automatische Spracherkennung

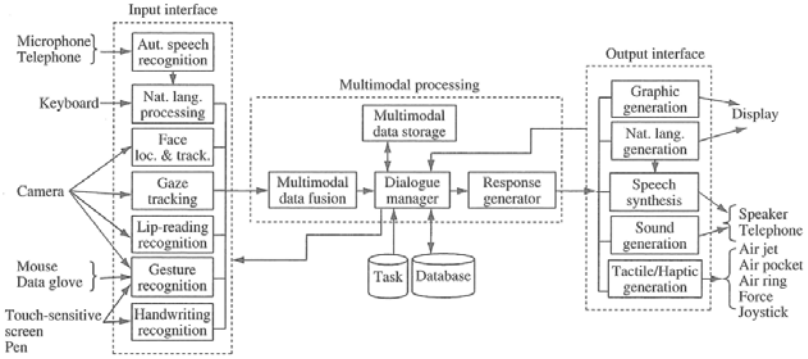


Abbildung 2.1: Allgemeine Architektur eines multimodalen Dialogsystems, nach Delago und Araki [DA05]

fundamentaler Bestandteil dieser Arbeit ist, wird diese in Kapitel 3 näher erläutert.

Automatische Spracherkennung (Automatic Speech Recognition, ASR)

Aufgabe der ASR ist es Sprachsignale in eine Folge von Wortsequenzen zu dekodieren und die Äußerungen eines Sprechers festzustellen. Diese, für einen Menschen einfach zu lösende Aufgabe, stellt ein schwer zu lösendes Problem für aktuelle ASR Technik dar, da die menschliche Fähigkeit, Sprache zu erkennen auf der Nutzung von vielfältigen Wissenquellen beruht. Der Umsetzung von der Signal- auf die Symbolebene schließt sich eine Weiterverarbeitung der extrahierten Information durch eine sprachverstehende Einheit an.

Sprachverstehen (Natural Language Processing, NLP)

Als Eingabe der sprachverstehenden Einheit dient orthografischer Text, der z. B. über eine Tastatur eingegeben oder von der Spracherkennung ermittelt wird. Die für den Dialog wichtigen Informationen werden hierbei aus dem Text extrahiert. NLP wird bei [MS00] beschrieben.

Gesichtserkennung und -verfolgung (Face Localisation and Tracking)

Die Gesichtserkennung wertet die Bilder einer Kamera aus. Ist ein Gesicht erkannt worden, können spezifische Informationen über Gesichtsmerkmale eines Nutzers gewonnen werden. Dazu gehören z. B. die Position der Lippen oder der Augen. Weiterverarbeitende Verfahren können die Leistung rein akustischer ASR verbessern, indem visuelle Information mitverarbeitet wird.

Blickbewegungserkennung (Gaze Tracking)

In der Mensch-Mensch-Kommunikation dient die Blickrichtung unter anderem dazu, den Fluss der Interaktion zu kontrollieren und spielt eine wichtige Rolle in der Koordination von wechselnder Rede. Bei multimodalen Dialogsystemen

wird sie vor allem genutzt, um Information über das aktuelle Interesse des Nutzers zu erhalten. Dieses Wissen kann zur Unterstützung der ASR verwendet werden oder helfen, den Wechsel der Dialoginitiative zu regulieren. Direkt als Eingabemodalität kann die Blickrichtungserkennung z. B. zur Steuerung des Mauszeigers auf einem Bildschirm verwendet werden.

Lippen-Erkennung (Lip-reading Recognition)

Visuelle Signale beeinflussen die Wahrnehmung von Sprache vor allem, wenn sich die Sprecher von Angesicht zu Angesicht gegenüber stehen. Ein anschauliches Beispiel hierfür liefert der so genannte McGurk-Effekt[2]. Die Auswirkung der dabei beobachteten Viseme[3] macht sich die audio-visuelle Spracherkennung zunutze. Dabei wird visuelle und gesprochene Information auf der Visem/Phonem-Ebene zusammengeführt, um die Spracherkennung robuster zu gestalten.

Gesten-Erkennung (Gesture Recognition)

Gesten sind wichtiger Bestandteil der menschlichen Kommunikation. Um Gesten zur HCI einsetzen zu können, werden intrusive (z. B. Datenhandschuhe) oder nicht intrusive (z. B. Kameras) Geräte benutzt. Mit Datenhandschuhen können 3D-Gesten verarbeitet werden, während 2D-Gesten mittels berührungssensitiver Bildschirme erkannt werden können. Bei mobilen Geräten wie PDAs (PDA: Personal Digital Assistant) werden auch digitale Stifte oder Stylus zur Bedienung eingesetzt. Eine Übersicht zur Bedienung von Touchscreens findet sich bei [Dah06].

Handschrift-Erkennung (Handwriting Recognition)

Computer werden immer kleiner, Tastaturen sind jedoch nicht beliebig verkleinerbar. Daher ist die Handschrift-Erkennung interessant für die HCI. Auf einem kleinen berührungssensitiven Bildschirm kann mittels eines digitalen Stiftes oder Stylus durch Schreiben anstatt durch Tippen Text in einen Computer eingegeben werden.

[2] Beim McGurk-Effekt (benannt nach dem Entwicklungspsychologen Harry McGurk) wurde festgestellt, wenn ein Video z. B. so beeinflusst wird, dass ein 'ga' Sound zu hören ist während eine Person im Video einen 'ba' Sound artikuliert, nehmen die meisten Menschen, die das Video sehen einen 'da' Sound wahr [MM76].

[3] Angelehnt an das Phonem auf der akustisch/auditiven Ebene, beschreibt das *Visem* die Bewegung des Mundes und des Gesichtes während der Äußerung eines Phonems. Dabei können sich mehrere Phoneme dasselbe Visem teilen.

2.1.2 Multimodale Verarbeitung

Das zentrale Modul bei der multimodalen Verarbeitung (Multimodal Processing) ist der Dialogmanager, in dem die „Intelligenz“ des Systems implementiert ist. Er leitet aus den Informationen, die er von den übrigen Modulen erhält, die nächste Aktion des Systems ab.

Fusion (Multimodal Data Fusion)

Wie in Abbildung 2.1 zu sehen ist, werden die Informationen der Eingabemodalitäten integriert. Dieser Vorgang wird als *Fusion* bezeichnet. Dabei wird eine Repräsentation der vom Nutzer kommunizierten Ziele und Aktionen erzeugt. Ein Kriterium, das häufig zur Integration der Informationen herangezogen wird, ist die zeitliche Synchronität von Eingaben. Die eigentliche Fusion wird auf Signal- oder semantischer Ebene durchgeführt. Audiovisuelle Spracherkennung funktioniert z. B. durch Kombination akustischer Merkmale und Merkmale der Lippenbewegung auf der Signalebene. Bei der Fusion auf semantischer Ebene wird zunächst die Bedeutung der einzelnen Modalitäten bestimmt. Diese Interpretationen werden im Anschluss zusammengeführt. Welche Variante der Fusion geeignet ist, hängt sowohl von den Modalitäten, als auch von der jeweiligen Aufgabe ab. Ein jüngerer Forschungsbeitrag von Oviatt befasst sich mit unterschiedlichen *Fusion*-Techniken [OLC05].

Multimodaler Datenspeicher (Multimodal Data Storage)

Die Informationen, die im multimodalen Datenspeicher abgelegt werden, um sie dem Dialogmanager zur Verfügung zu stellen, setzen sich zusammen aus Nutzer-Informationen, wie z. B. Hypothesen der Spracherkennung, und Informationen, die von den Eingabemodulen produziert werden. Letztere beinhalten u.a. Konfidenzwerte und Zeitstempel. Durch Konfidenzen zusammengehöriger Eingabemodalitäten kann eine Gesamterkennungsrate ermittelt werden.

Dialogmanager (Dialogue Manager)

Der Dialogmanager entscheidet über die nächste vom System auszuführende Aktion. Die Entscheidung wird bestimmt durch die Aufgabe (beschrieben im nächsten Punkt), ein domainabhängiges Dialogmodel für die Interaktion zwischen Mensch und System und die Interaktionsstrategien, durch die die Initiative bei der Dialogführung festgelegt wird. Die Initiative kann z. B. auf Systemseite, auf Nutzerseite oder gemischt sein. Typische Aktionen des Systems sind Datenbankabfragen, Generierung von Prompts und Hilfestellung für den Nutzer. Prompts ermöglichen die Interaktion mit dem Nutzer. Durch sie kann

nach zusätzlich benötigter Information gefragt werden und bereits erhaltene Information bestätigt werden. Die Bestätigung kann implizit, explizit oder durch Wiederholung einer Frage erfolgen. Einen Überblick zum Thema Dialogmanagement liefert [CAS97].

Aufgabe (Task)
Die Aufgabe kann in Form einer Schilderung des Services, der vom Dialogsystem bereitgestellt wird, separat gespeichert werden. Durch die Benutzung eines Aufgaben-Moduls wird eine Trennung zwischen dem Wissen über den Dialog und dem Wissen über die Aufgabe herbeigeführt. Durch aufgabenunabhängige Plattformen soll es möglich gemacht werden, Systeme zu schaffen, die sich einfach an unterschiedliche Aufgaben anpassen können.

Datenbank (Database)
Die Datenbank enthält die Informationen, an denen der Benutzer möglicherweise interessiert ist. Es können mehrere Datenbanken genutzt werden, die lokal oder serverseitig verfügbar sind. Manche Systeme speichern auch Informationen über Nutzer zur Identifikation oder zur Personalisierung der Interaktion.

Antwortgenerierung (Response Generator)
Der Dialogmanager veranlasst das Modul, zur Antwortgenerierung die ausgewählte Systemantwort auszugeben. In Bezug zur bereits besprochenen Fusion, nennt man diesen Vorgang *Fission*. Dabei werden die Ausgaben der einzelnen Ausgabemodalitäten ausgewählt, koordiniert und synchronisiert. Mögliche Ausgabemodalitäten werden im nächsten Abschnitt erläutert.

2.1.3 Ausgabemodule

Die Kombination mehrerer Ausgabemodalitäten kann es dem Nutzer erleichtern, den Leistungsumfang eines Systems und den aktuellen Dialogzustand zu erfassen. Dadurch lässt sich die Möglichkeit zur Fehlerkorrektur verbessern und die Interaktion wird aufgewertet. Es folgt eine kurze Übersicht über die Module der Ausgabemodalitäten (Output interface).

Grafikgenerierung (Graphic Generation)
Bei der Grafikgenerierung wird ein Bildschirm zur Anzeige von visuellen Objekten genutzt. Animierte Agenten dienen vor allem der Erhöhung der Verständlichkeit der Sprachausgabe, indem sie z. B. auf andere Bezugsobjekte auf dem Bildschirm deuten oder schauen, oder ihre Mitteilung durch einen bestimmten

Gesichtsausdruck unterstreichen. Weitere grafische Information kann dem Nutzer je nach Anwendungsgebiet in Form von Karten, Bildern, Icons und anderen Anzeigen vermittelt werden.

Erzeugung natürlicher Sprache (Natural Language Gereration, NLG)
Die Erzeugung natürlicher Sprache ist ein nicht triviales Problem aus dem Bereich der Computerlinguistik, bei dem es um die Erzeugung von textuellen Antworten geht. Das System muß passende Worte und Ausdrücke in einem bestimmten Zusammenhang auswählen. Sprachliche Phänomene wie Ellipsen und Anaphern müssen geplant und die Information phrasiert werden, um Klarheit zu erreichen und Redundanzen zu vermeiden. Ausführlich wird NLG bei [DMS00] beschrieben.

Sprachsynthese (Speech Synthesis)
Das Sprachsynthesemodul stellt Information in sprachlicher Form bereit. Vorleseautomaten[4] sind für Dialogsysteme besonders geeignet, da sie das Sprachsignal aus Rechtschrift-Text erzeugen können. Dies ermöglicht, die Antwort der NLG direkt als Eingabe zu verwenden. TTS Systeme ermöglichen jeden Satz durch parametrische Verfahren, ohne vorher aufgezeichnete Worte, zu erzeugen. Die Qualität der so synthetisierten Sprache ist jedoch immer noch recht begrenzt. Daher kommen in kommerziell genutzten Systemen meist datenbasierte Verfahren zum Einsatz, bei denen der Sprechschall aus Daten vorher aufgezeichneter Sprache zusammengesetzt wird.

Tonerzeugung (Sound Generation)
Auch andere Töne als Sprechschall können die Interaktion bereichern. Mit einer taktilen Eingabe sind z. B. häufig *Auditory Icons* verbunden, durch die sehr einfach und schnell eine Rückbestätigung gegeben werden kann. Weiterhin ist es möglich, einen Bezug zwischen Lautstärken- oder Tonhöhenwechseln und Datenveränderungen, wie z. B. Anstieg und Abfall von Aktienkursen, herzustellen.

Taktile/Haptische Erzeugung (Tactile/Haptic Generation)
Taktiles Feedback wird bislang in multimodalen Dialogsystemen wenig genutzt. Es kann pneumatisch (mittels Luftdruck), vibrotaktil (z. B. durch Ausnutzung des piezoelektrischen Effekts), elektotaktil (Anregung durch Elektroden) oder funktional-neuromuskulär (direkte Anregung des neuromuskulären Systems) erfolgen.

[4] Engl.: text-to-speech (TTS) synthesiser

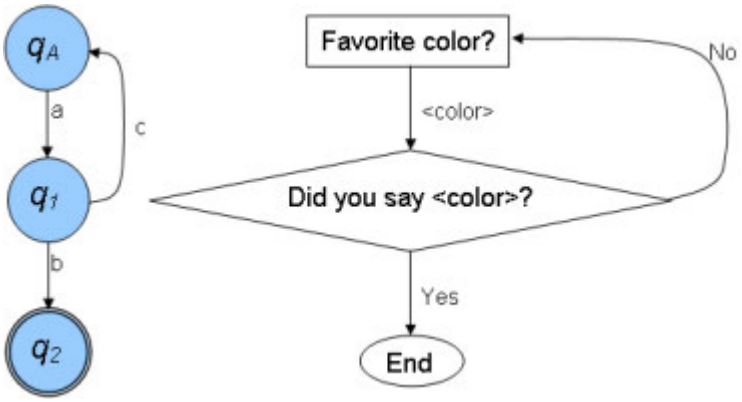


Abbildung 2.2: Links: Zustandsgraph eines Endlichen Automaten
Rechts: Beispiel eines einfachen Dialogsystems als Zustandsautomat

2.2 Dialogsysteme als Zustandsautomaten

Der Ansatz, Dialogsysteme als Zustandsautomaten zu modellieren, hat bereits eine relativ lange Geschichte [PS84]. Häufig wird statt dem Begriff Zustandsautomat auch auch der Ausdruck *Dialog-Grammatik* verwendet. Man darf in diesem Zusammenhang jedoch davon ausgehen, dass man es mit endlichen Automaten und regulären Grammatiken im Sinne der Chomsky Hierarchie[5] zu tun hat. Das heißt, man kann zu jedem endlichen Automaten eine reguläre Grammatik konstruieren und umgekehrt. Die beiden Konstrukte sind also äquivalent [HU69].

Zustandsgraphen sind eine anschauliche Darstellungsform von Zustandsautomaten, durch die implizit alle Bestandteile eines Automaten gegeben sind. Sie bestehen aus einer endlichen Anzahl an Knoten, die durch gerichtete Kanten verbunden sind. Abbildung 2.2 links zeigt einen Zustandsgraphen mit dem Anfangszustand q_A und zwei weiteren Zuständen q_1 und q_2. Zustandsübergänge sind unter den Bedingungen a, b und c über die entsprechenden verbindenden Kanten möglich. In Zustand q_2 terminiert der Zustandsautomat. Diesen Zustandsgraphen kann man verwenden, um ein sehr einfaches System, wie das Sprachdialogsystem in Abbildung 2.2 rechts, zu konstruieren. Die Initiative liegt in diesem Beispiel auf Systemseite, wobei Zustandsübergänge durch Äußerungen des Nutzers herbeigeführt werden. Der Zustand in Rautenform kennzeichnet die Stelle im Dialog, an der das System explizit eine Bestätigung der Eingabe fordert. Zum Thema Initiative bei der Dialogführung und Bestätigung von

[5] Die Chomsky-Hierarchie (benannt nach dem Linguisten Noam Chomsky) bezeichnet vier Klassen formaler Grammatiken. Ausgehend von einer nicht eingeschränkten Grundgrammatik, entstehen die weiteren Grammatiktypen durch zunehmende Einschränkung der Produktionsregeln [BS03].

Eingaben sei auf den Punkt *Dialogmanager* im Abschnitt 2.1.2 und die dort angegebene Literatur verwiesen.

Mit Hilfe von Zustandsautomaten lässt sich die Interaktion zwischen Nutzer und System auf einfache Weise modellieren. Sie eignen sich besonders für Systeme, bei denen das Dialogmodell und das Aufgabenmodell dieselbe Struktur aufweisen. Da die Initiative zur Dialogführung auf Systemseite liegt, können die Äußerungen des Nutzers antizipiert werden, was einfach zur Verbesserung der Spracherkennungsleistung verwendet werden kann [CAS97]. Beispielsweise können in jedem Zustand nur bestimmte Grammatikregeln des Spracherkenners aktiviert werden[6].

Kritik wird an Zustandsübergangsmodellen vor allem wegen ihrer Unflexibilität und der mangelhaften Verwendbarkeit für komplexe Aufgaben geübt. Alternativ können z. B. plan-basierte Ansätze verwendet werden, mit denen auch eine komplexere Modellierung des Dialogs möglich ist [Coh97]. Allerdings kommt das Prinzip der Zustandsautomaten immer noch häufig zum Einsatz, was die rege Verwendung von Standards wie *VoiceXML*[7] zeigt.

2.3 Testkriterien und Evaluation

Die Evaluation multimodaler Dialogsysteme ist Gegenstand aktueller Forschung [DHM07]. In diesem Abschnitt werden aufbauend auf einführenden Erläuterungen die Erkenntnisse einiger aktueller Forschungsarbeiten zusammengestellt. Zur Evaluation von multimodalen Systemen muss neben der Überprüfung der Leistung einzelner Systemkomponenten auch die Qualität der Interaktion zwischen Benutzer und Gesamtsystem gemessen werden [Möl08a]. Eine zeitgemäße Definition von *Qualität* liefert [Jek00]:

> *Qualität: „Ergebnis eines Wahrnehmungs- und Beurteilungsprozesses, bei dem der/die Beurteilende die wahrgenommenen Merkmale einer Einheit vergleicht mit individuellen Erwartungen, sachgerechten Erfordernissen oder gesellschaftlichen Forderungen.“*

Die wahrgenommenen Merkmale einer Einheit (auch Qualitätsmerkmale genannt) werden durch die so genannten Qualitätselemente beeinflusst [Jek04]. Für den Fall eines multimodalen Dialogsystems sind die Qualitätselemente durch die einzelnen

[6] Mehr zum Thema Grammatiken in Abschnitt 3.3

[7] http://www.w3.org/TR/voicexml20/

Komponenten des Systems (siehe Abschnitt 2.1) gegeben. Über die Qualitätsmerkmale multimodaler Systeme ist bislang nur sehr wenig bekannt [Möl08b]. Einfluss auf die Qualität des Wahrgenommenen haben aber nicht nur die verschiedenen Interaktionsmodalitäten, sondern auch Benutzereigenschaften, die Beurteilungssituation und die Evaluationsmethode. Zur Feststellung der vom Benutzer wahrgenommenen Qualität werden Interaktionstests mit menschlichen Benutzern durchgeführt, bei denen Informationen über den Verlauf der Interaktion gespeichert werden, und die Benutzerwahrnehmung z. B. mittels Fragebögen oder Interviews festgehalten wird.

Ein weiterer wichtiger Teilaspekt der Qualität interaktiver Systeme ist *Usability*[8]. Laut ISO 9241 Teil 11 [1199] ist sie wie folgt definiert:

> *Usability: „The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.“*[9]

Dabei lassen sich Effektivität (Grad der Zielerreichung) und Effizienz (Aufwand und Ressourcen im Verhältnis zum Grad der Zielerreichung) neben anderen Dimensionen, wie Intuitivität oder Erlernbarkeit, wiederum mittels Fragebögen messen.

Einen weiteren Ansatz zur Evaluation multimodaler Systeme bietet das PROMISE[10] Framework [BKL+02], das eine Weiterentwicklung des PARADISE[11] Frameworks [WLKA97] darstellt. Letzteres wurde zur Evaluation von Sprachdialogsystemen entwickelt. Bei diesen Qualitätsvorhersagemodellen wird ein Wert für die Systemperformance anhand von quantifizierter *Nutzerzufriedenheit* und so genannten *Dialogkosten* bestimmt. Details hierzu sind in der angegebenen Literatur zu finden.

Bei Dybkjær et al. [DLW04] wird die Evaluation in die drei Teilbereiche technische Evaluation, Usability Evaluation und Benutzerevaluation gegliedert. Dieser Einteilung folgend, werden in den nächsten Unterabschnitten Methoden zur Qualitätsmessung multimodaler Systeme vorgestellt.

[8] Deutsch: Gebrauchstauglichkeit

[9] Deutsch: Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.

[10] Procedure for Multimodal Interactive System Evaluation

[11] Paradigm for Dialogue System Evaluation

2.3.1 Technische Kriterien

Die technische Evaluation kann auf *Systemebene* und auf *Sub-Systemebene* durchgeführt werden. Die Qualität des Gesamtsystems hängt unter anderem von der Leistung der einzelnen Subsysteme ab. [Möl08b] beschreibt einige Verfahren zur Leistungsmessung multimodaler Systemkomponenten, darunter unter anderem:

Input Performance Die Leistung der einzelnen Eingabemodule kann als Genauigkeit oder Fehlerrate des entsprechenden Erkenners angegeben werden. Als Beispiel sei hier die Wortfehlerrate (*Word Error Rate*, WER) eines Spracherkenners genannt, die sich wie folgt berechnet:

$$WER = \frac{S + D + I}{N} \tag{2.1}$$

dabei ist S die Anzahl an Ersetzungen (Substitutions), D die Anzahl an Löschungen (Deletions), I die Anzahl an Einfügungen (Insertions) und N die Anzahl der betrachteten Worte. Ein verwandtes Maß ist die *Concept Error Rate* (*CER*, die den Prozentsatz korrekt verstandenener semantischer Einheiten pro Dialog misst [12]. Ähnliche Rechnungen kann man z. B. für Gestenerkenner oder Emotionserkenner anstellen. Der Satz erkennbarer Gesten und Emotionen ist, wie das Vokabular eines Spracherkenners, begrenzt. Für Spracherkenner lässt sich in diesem Zusammenhang die *Out of Vocabulary Rate* ableiten. Ein Maß für die Grammatikabdeckung der Benutzeräußerungen ist die *In Grammar Rate*. Ein Überblick über spezielle Maße für Spracherkenner ist bei [PHS07] zu finden.

Dialogue Management Performance Die Leistung des Dialogmanagers kann durch vielfältige Techniken, die von den jeweils implementierten Funktionen des Interfaces abhängen, ermittelt werden. Häufig wird der Parameter *Task Success* zu Hilfe gezogen, der die Fähigkeit, den Dialog zum gewünschten Ziel zu führen, misst. Eine ausführliche Beschreibung dieses und weiterer Parameter liefert [Möl08a].

Der Einfluss der Leistungen der Subsysteme auf die Qualität des Gesamtsystems ist bislang noch nicht vollständig erfasst. Für die Evaluation auf Systemebene gibt es derzeit einige Bemühungen, Verfahren zur Evaluation von Sprachdialogsystemen für multimodale Dialogsysteme zu erweitern [BKL+02]. Nachstehende Parameter zur

[12] Siehe auch Abschnitt 6.2.2.3

Evaluation von sprachbasierten Systemen stellen einen Auszug aus einer von Möller [Möl08a] zusammengestellten Liste dar:

- *Dialogue Duration* (Dialogdauer) misst die Dauer des gesamten Dialogs
- *User Turn Duration* (Benutzeräußerungsdauer) misst die durchschnittliche Dauer einer Nutzeräußerung
- *Number of User Turns* (Anzahl der Schritte des Nutzers) misst die Anzahl der Schritte, die in einem Dialog geäußert wurden
- *Task Success* (Aufgabenerfolg) misst Anzahl der am Ende des Dialogs erreichten Ziele

Üblicherweise werden zur Evaluation, abhängig von der Fragestellung, wie hier einige passende Maße ausgewählt. Die Quantifizierung der Parameter geschieht in der Regel durch Auswertung von Logfiles. Einige Parameter können rein instrumentell gemessen werden. Die *User Turn Duration* ergibt sich z. B. über die durchschnittliche Dauer einer Nutzeräußerung, gemessen vom Beginn der Äußerung bis zur Systemreaktion. Weitere technische Kriterien sind z. B. Verzögerungszeiten, Echtzeit-Fähigkeit und Speicher- und Prozessorauslastung. Delago und Araki (2005) [DA05] beschreiben außerdem noch Ansätze zur Evaluation fusionierter Modalitäten und animierter Agenten.

2.3.2 Usability Evaluation

Zur Usability Evaluation werden üblicherweise *analytische* und *empirische* Methoden eingesetzt [PRS02]. Zwei Beispiele für analytische Methoden sind:

Cognitive Walkthrough Durch den Cognitive Walkthrough[13] (CW) sollen potenzielle Usability Probleme, wie sie z. B. bei der Erlernbarkeit eines Interfaces auftreten können, aufgedeckt werden. Meist wird der CW durch Experten durchgeführt, die Schritt für Schritt die Interaktionen durchgehen, die zur Lösung mit dem Interface erfüllbarer Aufgaben nötig sind. Die Experten prüfen dabei für jede Aufgabe sorgfältig, welche Auswirkung die Interaktion auf den Benutzer hat, welches Vorwissen er benötigt und was er erlernen muss, um die Aufgabe zu erfüllen [WRLP94].

[13] Im übertragenen Sinn: „Durchdenken eines Problems“

Heuristische Evaluation Bei der Heuristischen Evaluation (HE) wird das System von einzelnen Experten untersucht, um es auf bestimmte Usability-Kriterien (Heuristiken) zu prüfen. Die Experten sollen festhalten, was gut und was schlecht am Interface ist. Eine Übersicht über viel genutzte Heuristiken ist bei [Nie94] zu finden. Die dortigen Angaben sind jedoch auf unimodale Systeme bezogen.

Durch die Anwendung empirischer Methoden werden Daten gewonnen, die einen Indikator für die Leistung des Systems darstellen. Um derartige Daten zu erhalten, muss die Wahrnehmung von Benutzern, die mit dem System gearbeitet haben, in irgendeiner Form festgehalten werden. Bei der Erfassung dieser vom Nutzer gefühlten Qualität, spielen die zu Beginn dieses Abschnitts erwähnten Teilspekte der Qualität, wie Effizienz, Effektivität, Zufriedenheit und Erlernbarkeit, eine wichtige Rolle. An dieser Stelle sollen zwei wichtige empirische Methoden zur Abschätzung der Usability eines Systems vorgestellt werden.

Experimentelle Evaluation Bei der experimentellen Evaluation (EE) werden spezifische Aspekte des interaktiven Verhaltens eines Systems untersucht, wofür bestimmte Hypothesen formuliert werden [Stu05]. Die Bedingungen unter denen ein Experiment stattfindet, dürfen sich nur in einigen kontrollierten Variablen unterscheiden. EE mit Nutzern findet entweder in einem Usability-Labor oder im Feld statt. Laboruntersuchungen garantieren qualitativ bessere Daten, allerdings ist die Laborsituation oft weit von der Realität entfernt.

Befragungstechniken Die Befragung der Benutzer kann z. B. durch Fragebögen erfolgen. Aufgrund der ständigen Weiterentwicklung interaktiver Systeme, müssen auch die Fragebögen zur Evaluation der Systeme immer wieder überarbeitet werden. Ein Fragebogen, der speziell für grafische Nutzerschnittstellen entwickelt wurde, ist das Software Usability Measurement Inventory (SUMI) [KC93]. Aus diesem wurde für sprachbasierte Systeme das Subjective Assessment of Speech System Interfaces (SASSI) [HPGL00] abgeleitet. SASSI besteht aus 34 Items, die auf einer Fünf-Punkte Likert-Skala[14] bewertet werden. Bei Fragebögen werden in der Regel mehrere Items zu einem Faktor zusammengefasst. SASSI beinhaltet folgende sechs Faktoren:

- Die *System Response Accuracy* („Genauigkeit der Systemantwort“) gibt

[14] Eine Likert-Skala (benannt nach dem Psychologen Rensis Likert) ist eine psychometrische Skala, die in Fragebögen verwendet wird. Die Probanden tragen den Grad ihrer Zustimmung zu einer bestimmten Aussage auf der Skala ein [Lik32].

Auskunft darüber, ob die Nutzereingabe vom System richtig erkannt wurde und ob die Intentionen und Erwartungen des Nutzers erfüllt wurden.

- Der Faktor *Likeability* („Gefallen“) beinhaltet Angaben zur Meinung des Nutzers und dessen Empfindung zum System.
- *Cognitive Demand* („Kognitive Beanspruchung“) umfasst die vom Nutzer wahrgenommene kognitive Anstrengung zur Benutzung des Systems.
- Der Faktor *Annoyance* („Ärgernis“) beinhaltet Items wie „Die Interaktion mit dem System ist monoton/langweilig/irritierend“.
- *Habitability* („Bewohnbarkeit“) bezieht sich darauf, inwiefern sich die Systemvorstellungen des Nutzers mit den Systemvorstellungen der Entwickler decken.
- *Speed* (Geschwindigkeit) bezieht sich auf die wahrgenommene Geschwindigkeit des Systems.

Fragebögen wie *AttrakDiff*[15] [HBK03] nutzen ein semantisches Differential, um die vom Nutzer wahrgenommene Qualität des Systems zu bemessen. Dabei werden gegensätzliche Wortpaare (z. B.: gut/schlecht) als Extrema einer Skala gegenübergestellt. Der Benutzer gibt an, zu welchem Adjektiv das System seiner Ansicht nach eher tendiert.

Für diese Arbeit wurden drei SASSI Fragebögen verwendet, die entsprechend der dargebotenen Modalität modifiziert wurden. Als Vortest wurde eine Expertenevaluation mit Hilfe der Cognitive Walkthrough-Methode durchgeführt. Die detaillierte Beschreibung der Experimente findet in Kapitel 6 statt.

2.3.3 Benutzerevaluation

Durch die Auswertung von Fragebögen erhält man Ergebnisse, die sehr von der subjektiven Meinung des Nutzers geprägt sind. Zur Bewertung von Systemen sind aber auch objektive Daten erwünscht. Dabei ist für multimodale Dialogsysteme vor allem auch interessant, welche Modalität vom Nutzer tatsächlich verwendet wird. Diese muss nicht unbedingt dieselbe sein, die er in Fragebögen als die bevorzugte Modalität angibt [Lar03a].

[15] http://www.attrakdiff.de/

Der bereits erwähnte Parameter *Task Success* kann in manchen Fällen hoch sein, obwohl Nutzer die Effizienz des Systems niedrig bewertet haben. Auch die wahrgenommene Länge der Interaktion (*Dialogue Duration*) muss nicht tatsächliche Messergebnisse wiederspiegeln [Möl08a].

Die hier genannten Parameter lassen sich hauptsächlich durch die Auswertung von Logdaten gewinnen, wobei oft deren Korrelation mit den Ergebnissen einer Benutzerbefragung geprüft wird.

3 Automatische Spracherkennung

Das vorliegende Kapitel liefert einen Überblick über die wichtigsten Konzepte moderner Spracherkennungssysteme. Der Fokus ist dabei auf die Komponenten gerichtet, die im Rahmen dieser Arbeit beeinflusst wurden. Eine ausführlichere Darstellung des Themas ist z. B. bei [Eul06, ST95] zu finden.

3.1 Die Fundamentalformel der Spracherkennung

Statistisch gesehen kann der Erkennungsprozess als Dekodierung einer Sequenz gesprochener Wörter W in eine Sequenz von Merkmalsvektoren O (auch Beobachtungen genannt), die das Sprachsignal möglichst präzise repräsentieren, betrachtet werden. Gesucht ist die Sequenz W^*, die der ursprünglich geäußerten Wortfolge W, ausgehend von O, am besten entspricht. Die Lösung ist demnach der Wortfolge W^* mit der maximalen A-posteriori-Wahrscheinlichkeit $P(W|O)$ und lässt sich mit Hilfe des Satzes von Bayes wie folgt formulieren:

$$W^* = \underset{W}{\operatorname{argmax}}\, P(W|O) = \underset{W}{\operatorname{argmax}}\, \frac{P(O|W) \cdot P(W)}{P(O)} = \underset{W}{\operatorname{argmax}}\, P(O|W) \cdot P(W) \tag{3.1}$$

Die Beobachtungswahrscheinlichkeit $P(O)$ ist konstant bezüglich W und kann daher hier ignoriert werden [ST95]. Die in 3.1 auftretenden Verteilungsfunktionen sind nicht bekannt. Daher ist es zum Entwurf eines Spracherkennungssystems nötig, möglichst exakte Näherungen für $P(O|W)$ (Akustisches Modell) und $P(W)$ (Sprachmodell) zu finden.

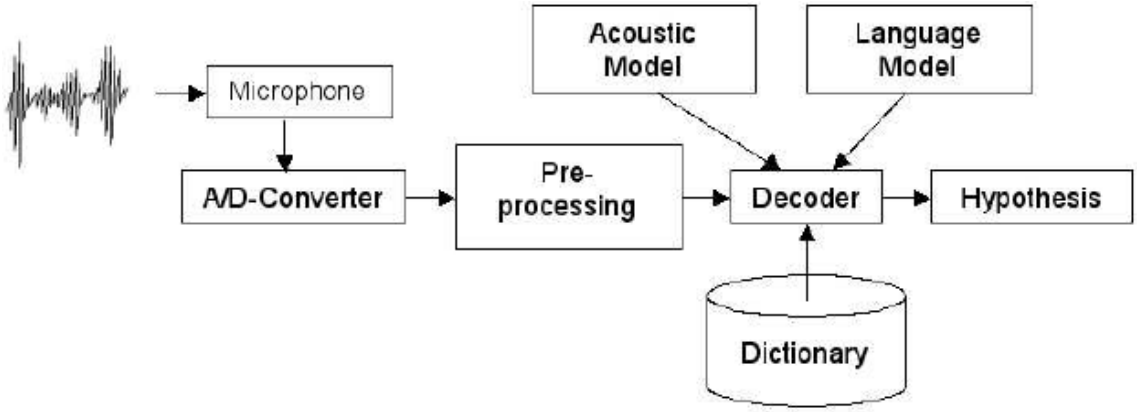


Abbildung 3.1: Allgemeine Architektur eines statistischen Spracherkennungssystems, nach [Met05]

3.2 Aufbau eines Spracherkenners

Die grundlegenden Bestandteile eines Spracherkenners finden sich in Abbildung 3.1 wieder. Aus dem mit Hilfe eines A/D-Wandlers (A/D-Converter) digitalisierten Sprachsignal werden zunächst bei der Vorverarbeitung (Preprocessing) geeignete Merkmale extrahiert. Im Dekodierer (Decoder) werden anhand der zur Verfügung stehenden Wissensquellen, namentlich des akustischen Modells (Acoustic Model), des Sprachmodells (Language Model) und des Lexikons (Dictionary), Hypothesen generiert, die typischerweise in ihrer orthografischen Form ausgegeben werden.

Ein gesprochenes Wort liegt nach der Berechnung der Merkmale als eine Folge von Vektoren vor, die sowohl in ihrer zeitlichen Struktur als auch in ihrer spektralen Zusammensetzung erhebliche realisierungsbedingte Verzerrungen aufweisen [ST95]. Die Verzerrungen ergeben sich dadurch, dass ein und derselbe Sprecher dasselbe Wort immer wieder anders und in unterschiedlicher Umgebung sagt. Es wird ein Mechanismus benötigt, der diesen speziellen Anforderungen gerecht wird. Durch *Hidden-Markov-Modelle* (HMMs) sind die zeitlichen Varianzen der Sprachproduktion durch eine sequenzielle Zustandsabfolge und die spektralen Beziehungen durch Beobachtungswahrscheinlichkeiten modellierbar. Abbildung 3.2 zeigt ein HMM mit fünf Zuständen, wie es zur Modellierung von Sprache denkbar ist. Drei der Zustände sind Beobachtungswahrscheinlichkeiten b_i zugeordnet, die z. B. durch Gaußsche Mischverteilungsdichten (GMM: Gaussian Mixture Model) [ED81] gebildet werden können. Die Übergangswahrscheinlichkeiten a_{ij} geben an, unter welcher Wahrscheinlichkeit von Zustand i nach Zustand j gewechselt wird.

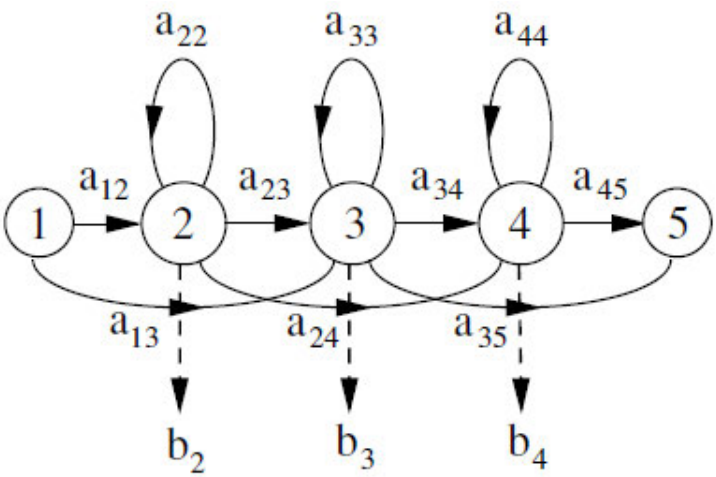


Abbildung 3.2: Beispiel eines Hidden-Markov-Modell für die Spracherkennung, nach [YEG+06]

Bei der Verwendung von HMMs müssen drei grundlegende Probleme bewältigt werden, für die verschiedene Algorithmen zur Verfügung stehen:

Evaluierungsproblem: Berechne die Wahrscheinlichkeit einer Beobaochtungssequenz bei gegebenem Modell. → *Forward-Algorithmus*

Dekodierungsproblem: Finde die wahrscheinlichste Zustandssequenz innerhalb eines HMMs für die gemachten Beobachtungen. → *Viterbi-Algorithmus*

Lernproblem: Lerne die Modellparameter, die eine gewünschte Ausgabesequenz am wahrscheinlichsten erzeugen. → *Forward-Backward-Algorithmus*

Diese Probleme greifen in gewisser Weise ineinander: Bevor das HMM überhaupt effektiv verwendet werden kann, müssen die Modellparameter in einer Trainingsphase gelernt werden. Nachdem das Evaluierungsproblem gelöst ist, kann über den Viterbi-Algorithmus die Zustandssequenz für die wahrscheinlichste Beobachtungssequenz ermittelt werden. Eine genauere Beschreibung der Funktionsweise der Algorithmen liefert z. B. [Kol08]. Weitere Informationen zum Einsatz von HMMs sind in den Abschnitten 3.2.2 und 3.2.4 zu finden.

3.2.1 Vorverarbeitung

Ziel der Vorverarbeitung ist es, relevante Merkmale aus dem Sprachsignal zu extrahieren und in eine geeignete Repräsentation zu überführen. Bewährte Verfahren extrahieren lautspezifische Informationen im Vokaltrakt und berücksichtigen Aspekte der menschlichen Wahrnehmung. Häufig verwendet wird dabei zum Beispiel das

Mel-skalierte Cepstrum, welches eine Trennung von sprachlichem Anregungssignal und Lautformung durchführt. Die resultierenden Merkmale werden *Mel Frequency Cepstral Coefficients* (MFCC) genannt. Zu deren Gewinnung wird das Zeitsignal zunächst blockweise in den Spektralbereich transformiert und das Betragsspektrum gebildet. Durch ein spezielles Gewichtungsfilter und eine logarithmische Frequenzskalierung werden Prozesse der menschlichen Schallverarbeitung nachgebildet. Nach einer Rücktransformation (üblicherweise mittels Diskreter Kosinustransformation) erhält man typischerweise 13 cepstrale Koeffizienten. Die resultierenden Merkmalsvektoren beinhalten üblicherweise die Koeffizienten sowie deren erste und zweite Ableitung, durch die Informationen über die zeitliche Veränderung der Koeffizienten hinzugefügt werden. Dadurch steigt die Dimensionalität des Vektors auf 39.

3.2.2 Akustische Modelle

Das akustische Modell behandelt die Wahrscheinlichkeiten $P(O|W)$, wobei O eine Beobachtungssequenz und W eine Wortsequenz darstellt. W ist mit Hilfe des Lexikons auf eine Sequenz von Zuständen des HMMs H abbildbar, so dass man auch $P(O|H)$ schreiben kann. Zur Bestimmung der Parameter des akustischen Modells ist somit eine möglichst genaue Schätzung der Übergangswahrscheinlichkeiten a_{ij} und der Emissionswahrscheinlichkeiten b_i des HMMs wichtig (Lernproblem). Für die Optimierung der Parameter ist umfangreiches Trainingsmaterial erforderlich, das aus textannotierten Sprachdaten gewonnen wird und in Form von Merkmalsvektoren vorliegt. Auf Basis des Trainingsmaterials können die Modellparameter durch den Forward-Backward-Algorithmus, ausgehend von einer initialen Schätzung, adaptiert werden. Ziel der Adaption ist es, das alte Modell so zu verbessern, dass die Produktionwahrscheinlichkeiten der Trainingsdaten durch das neue Modell gesteigert werden. Für das verbesserte Modell H' soll demnach $P(O|H') > P(O|H)$ gelten.

Moderne Spracherkenner nutzen häufig kontextabhängige Modelle (*Polyphone*) zur akustischen Modellierung, um koartikulatorischen[1] Effekten Rechnung zu tragen. Bei der Verwendung von Triphonen werden, abhängig vom linken und rechten Kontext, unterschiedliche Modelle zur Modellierung eines Phonems verwendet. Ein vorhergehendes Phonem, das eigentliche Kernphonem, und ein nachfolgendes Phonem werden als Phonemtripel im Kontext betrachtet und durch ein HMM mit bis zu fünf Zuständen modelliert [ST95].

[1] Koartikulation: Beeinflussung eines Lautes durch den lautlichen Kontext.

3.2.3 Sprachmodelle

Das Sprachmodell beschreibt die Wahrscheinlichkeiten $P(W)$ aus Gleichung 3.1. Für einfache Modelle können kontextfreie Grammatiken (context free grammars, CFG) im Sinne der Chomsky-Hirarchie verwendet werden. Der Lösungsraum wird dabei durch syntaktische Regeln eingeschränkt, wodurch sich die Erkenugsleistung erhöhen lässt. Algorithmen zur datenbasierten Ermittlung der grammatikalischen Struktur funktionieren bisher nur unzureichend, daher werden CFGs üblicherweise von Hand erstellt [Met05]. Weitere Details zu CFGs finden sich in Abschnitt 3.3.

Für Aufgaben mit komplexerer sprachlicher Struktur werden oft statistische n-Gramm-Modelle verwendet, bei denen n aufeinander folgende Worte betrachtet werden. Die Wahrscheinlichkeit des nächsten Wortes hängt somit von den $n-1$ vorangegangenen Worten ab. Aus praktischen Gründen, wie dem Mangel an Trainingsdaten und Speicher, sind derzeit häufig 3-Gramm- und 4-Gramm-Modelle im Einsatz [Met05]. Die freien Parameter, also die Wahrscheinlichkeiten bestimmter aufeinander folgender Wörter, werden auf Grundlage umfangreicher Textproben geschätzt.

3.2.4 Dekodierung

Die Dekodierung einer Folge von Merkmalsvektoren entspricht der Lösung des in Gleichung 3.1 beschriebenen Problems. Das Finden der optimalen Lösung W^* ist in erster Linie ein Ressourcenproblem, daher werden schnelle Algorithmen für die Suche benötigt. Abbildung 3.3 zeigt einen baumförmigen Hypothesengraphen. Derartige Datenstrukturen ergeben sich bei Durchsuchung des Lösungsraumes. Die beiden „klassischen“ Dekodierungsverfahren automatischer Spracherkennung sind der *strahlgesteuerte Viterbi-Algorithmus* und die auf dem A^*-Algorithmus basierende Kellersuche [ST95].

Die Kellersuche ist ein zeitasynchrones Tiefensuchverfahren (Abbildung 3.4 links). Zum Durchlaufen des Hypothesengraphen wird Schritt für Schritt der Lösungsraum partieller Hypothesen expandiert, bis eine vollständige Hypothese der kompletten akustischen Eingabe O gefunden ist. Die vollständige Hypothese mit maximaler Wahrscheinlichkeit entspricht W^*.

Der strahlgesteuerte Viterbi-Algorithmus ermöglicht eine zeitsynchrone Suche. Das

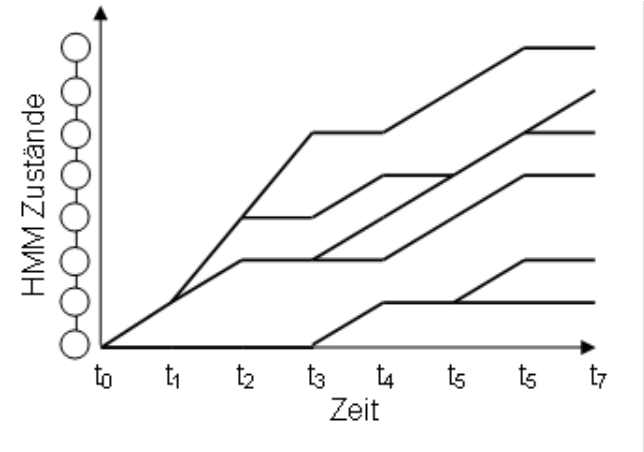


Abbildung 3.3: Hypothesengraph

Risiko fehlerhafter Entscheidungen ist bei diesem Verfahren höher als bei der Kellersuche. Dennoch wird es aus Effizienzgründen bei den meisten modernen Spracherkennungssystemen eingesetzt. Die Basis des verwendeten Graphsuchverfahrens bildet eine Breitensuche (Abbildung 3.4 rechts), bei der alle Hypothesen parallel expandiert werden. Zu jedem Zeitpunkt gibt es eine bestimmte Anzahl „aktiver“ Zustände. Die aktiven Zustände des Zeitpunktes t_{i+1} setzen sich aus allen Folgezuständen des Zeitpunktes t_i zusammen. Aufgrund des Anstiegs an Verzweigungen wächst die Anzahl der möglichen Hypothesen mit dem Zeitverlauf. Um eine Explosion des Lösungsraumes zu vermeiden, kann der Aufwand beschränkt werden. Durch die strahlgesteuerte Suche gelangen nur hochwahrscheinliche Zustände zur Weiterverarbeitung durch den Viterbi-Algorithmus.

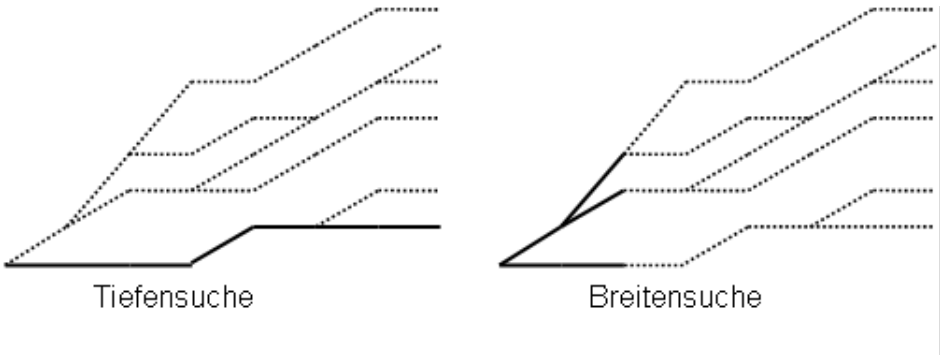


Abbildung 3.4: Tiefensuche mit einer vollständig expandierten Hypothese (links). Breitensuche mit parallel expandierten Hypothesen (rechts).

Das Ergebnis der Dekodierung ist die wahrscheinlichste Wortfolge (sofern keine Suchfehler durch verfrühtes verwerfen von Hypothesen vorliegen). In vielen Systemen wird ein Spracherkenner als Eingabemodul verwendet. Wird das Erkennungsergebnis einer

nachfolgenden semantischen Analyse unterzogen, kann die wahrscheinlichste Wortfolge als nicht sinnvoll verworfen werden. Ist dies der Fall, können die besten Alternativen des Ergebnisses als Lösung herangezogen werden. Mögliche Realisierungsformen sind nach Wahrscheinlichkeiten geordnete Listen mit den N besten Wortfolgen (N-best list) oder gerichtete, zyklenfreie Wortgraphen (auch „lattice“ genannt) [ST95, Met05].

3.3 Grammatik und Vokabular

Grammatik und Vokabular stellen zwei wichtige Entwurfsparameter spracherkennender Systeme dar, für die sich je nach Anwendung unterschiedliche Freiheitsgrade ergeben. Ein Kommando-System für die Menüsteuerung oder einfache Kontroll- und Abfragefunktionen akzeptiert beispielsweise nur wenige, spezielle Befehle. Bei einem Diktiersystem möchte man dem Anwender möglichst keine Beschränkungen auferlegen [Eul06, ST95]. Durch CFGs kann eine endliche Menge an syntaktischen Regeln angegeben werden. Sie sind vor allem für Einsatzgebiete mit einfacher Grammatik geeignet. Durch das starre Konzept einer CFG erfolgt das Parsing und die semantische Interpretation einer Äußerung so zu sagen automatisch. Bei stark variierenden Benutzeräußerungen sind CFGs in der Regel weniger geeignet, da die Anzahl an möglichen Formulierungen nur schwer durch eine CFG abgebildet werden kann.

Das *Java Speech Grammar Format* (JSGF) [Hun00] ist ein Beispiel für einen Formalismus, der Grammatiken für die Spracherkennung beschreibt. Es bedient sich des Stils und der Konventionen der Programmiersprache Java[2] und spezieller Notationen für Grammatiken. Die folgende Beispielgrammatik ist an das in Abschnitt 2.2 vorgestellte Dialogsystem angelehnt:

```
1 #JSGF V1.0 UTF-8;
2 grammar colorGrammar;
3
4 public <favoriteColor> = [(MY FAVORITE COLOR | IT) IS] <color> ;
5 public <confirmation> = YES | NO ;
6
7 <color> = RED | BLUE | GREEN | YELLOW ;
```

[2] http://java.sun.com/

Zeile 1 enthält Angaben zur verwendeten JSGF-Version und zur Zeichenkodierung. In Zeile 2 wird der Name der Grammatik (*colorGrammar*) festgelegt. Ab Zeile 4 folgen syntaktische Regeln. Auf der linken Seite einer Regel wird ihr Bezeichner in spitzen Klammern angegeben. Jede Regel darf nur einmal vorkommen. Durch die Angabe *public* wird eine Regel zur *aktiven* Regel. Das heißt, die Regel kann von einem Spracherkenner direkt genutzt werden, um eine Hypothese zu bestimmen [Hun00]. Die rechte Seite kann (terminale) Worte oder (nicht-terminale) Regelbezeichner enthalten. Ausdrücke können durch Klammern zu Gruppen zusammengefasst werden. Eckige Klammern bezeichnen optionale Elemente. Alternative Ausdrücke können durch einen senkrechten Strich "|" getrennt angegeben werden.

Einige Beispielsätze, die mit der angegebenen Grammatik realisierbar sind:

```
MY FAVORITE COLOR IS BLUE
MY FAVORITE COLOR IS GREEN
IT IS RED
IT IS YELLOW
YES
NO
```

Durch die *public* Deklaration ist es auch möglich Regeln zu aktivieren und zu deaktivieren. Dadurch können den Zuständen eines Dialogsystems gezielt bestimmte Regeln zugeordnet werden. In Abbildung 3.5 sind die Regeln der hier definierten *colorGrammar* den Zuständen des in Abbildung 2.2 dargestellten Dialogsystems zugeordnet. Zustandsübergänge werden durch sprachliche Äußerungen herbeigeführt. Wird in einen anderen Zustand gewechselt, treten zustandsspezifische Regeln in Kraft.

Art und Größe des Vokabulars sind abhängig vom jeweiligen Anwendungskontext (auch Domäne oder Domain genannt). Domänen und deren durchschnittlicher Wortschatzumfang sind in Tabelle 3.1 aufgeführt. Sie reichen vom einelementigen Wortschatz eines Alarmstoppschalters (z. B. ein „Stopp!" zur Beendigung eines Weckalarms) bis zum Gesamtumfang des deutschen Hochsprachvokabulars. Der Umfang variiert vor allem je nach Anforderung an die Stilfreiheit der benutzerseitigen Formulierung. Schon bei stark eingeschränktem Anwendungsbereich (z. B. Hotelbuchung, Fahrplanauskunft etc.) kann das Vokabular auf einige Hundert oder Tausend Wortformen anwachsen [ST95].

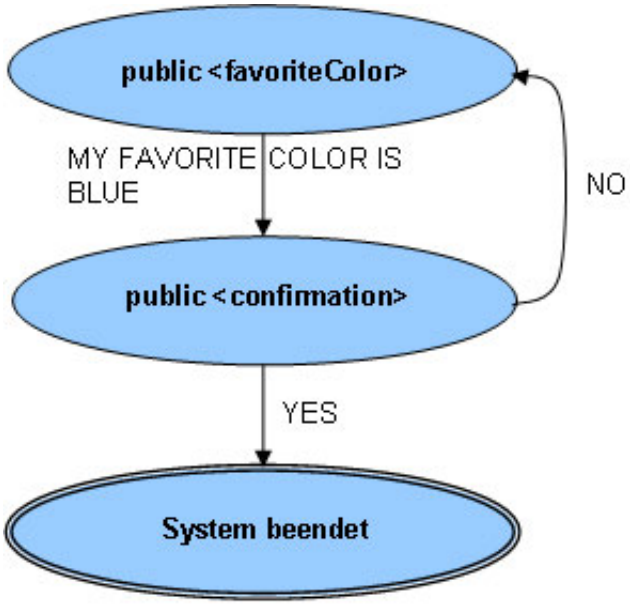


Abbildung 3.5: Ein einfaches Dialogsystem mit unterschiedlichen aktiven Regeln pro Zustand

Domain	Umfang
Alarmstoppschalter	1
Menü-Steuerung (ja/Nein)	2
Zahlen/Ziffern	10+x
Gerätebedienung	20-200
Auskunftsdialog	500-2000
Alltagssprache	8 000-20 000
Diktiermaschine	20 000-50 000
Deutsch ohne Fremdwörter	ca. 300 000

Tabelle 3.1: Wortschatzumfang potentieller Spracherkennungsanwendungen, nach [ST95]

4 Architektur des Rauminformationssystems

Das vorliegende Kapitel beschreibt die Ausgangssituation des Spracherkenners und des Rauminformationssystems vor der Integration der Sprachsteuerung. Das als Flash-Anwendung realisierte grafische Benutzerinterface *„Arbeitsplatz-Map“* (AP-Map) wird in Abschnitt 4.1 vorgestellt. Abschnitt 4.2 gibt einen Überblick über den verwendeten Spracherkenner. Abschließend erfolgt in Abschnitt 4.3 eine Darstellung der Datenbankanbindung der beiden Module.

4.1 AP-Map

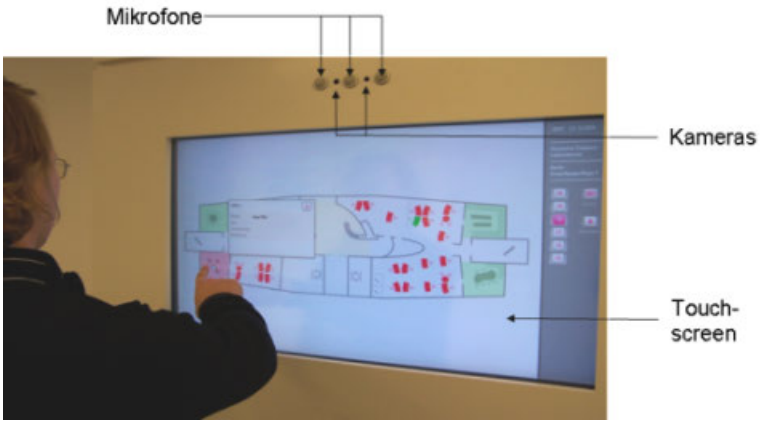


Abbildung 4.1: Ein Anwender bei der Verwendung der AP-Map.

Das unimodale Rauminformationssystem AP-Map soll die optimale Nutzung vorhandener Ressourcen wie Räume und Arbeitsplätze fördern und dem Benutzer bei der Suche nach Mitarbeitern behilflich sein. Abbildung 4.1 zeigt einen Anwender bei der Interaktion über den Touchscreen. Derartige Panels befinden sich auf jeder Etage der T-Labs. Die Mikrofone und Kameras werden von der Ausgangsversion des Rauminformationssystems noch nicht verwendet. Sie wurden für die geplanten Erweiterungen der Benutzerschnittstelle integriert.

Die AP-Map wird im Wechsel mit dem so genannten *Eventscreen* (Abbildung 4.2) angezeigt. Dieser bietet eine Übersicht über die aktuellen Veranstaltungen. Zeit, Name und Ort jeder Veranstaltung sind zeilenweise aufgelistet. Bei Berührung des Eventscreen an beliebiger Stelle wird zum *Mainscreen* der AP-Map gewechselt. Welche Etage dabei angezeigt wird, hängt davon ab, auf welcher Etage man sich gerade befindet. Beispielsweise entspricht auf dem Panel der 17. Etage die Stockwerkanzeige der 17. Etage der Mainscreen. Berührt man gezielt einen Raumnamen auf dem Eventscreen, wird direkt zu dem Stockwerk gewechselt, auf dem sich der Raum befindet. Der gewählte Raum ist rot blinkend gekennzeichnet.

Abbildung 4.2: Eventscreen

Tippt man auf einen Raum oder einen Arbeitsplatz auf der Stockwerkanzeige, wird dieser ebenfalls rot blinkend dargestellt. Zusätzlich öffnet sich ein Detailfenster mit Informationen zum Buchungsstatus (Abbildung 4.3).

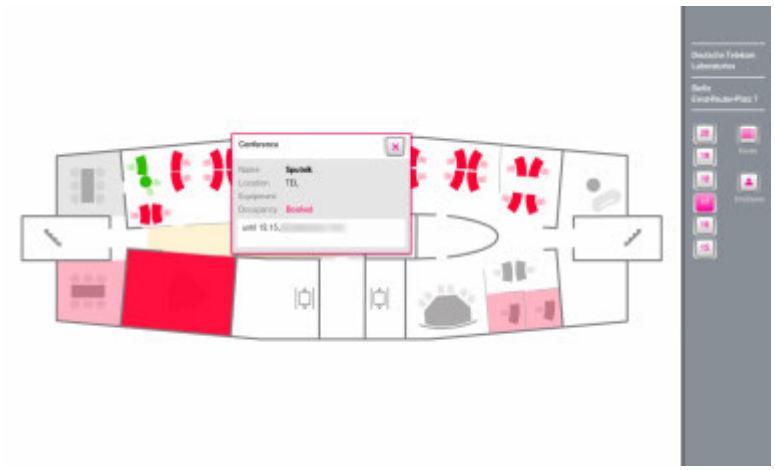


Abbildung 4.3: Stockwerkanzeige mit einem markierten Raum und einem Detailfenster.

Auf der rechten Seite der Stockwerkanzeige befinden sich mehrere Schaltflächen. Über die nummerierten Buttons kann man zwischen den Stockwerkanzeigen hin und her springen. Der Button mit der Unterschrift „Events“ dient der Anzeige des Eventscreens. Der Button mit der Unterschrift „Employees“ öffnet das Fenster zur Mitarbeitersuche. Nach Eingabe des gesuchten Namens über die Bildschirmtastatur werden die gefundenen Mitarbeiter aufgelistet. Durch Antippen eines Namens erhält man Informationen zu den aktuellen Buchungen eines Mitarbeiters (Abbildung 4.4). Sind mehrere Buchungen vorhanden, kann zwischen diesen mittels Pfeiltasten geblättert werden. Der Button „Show“ wechselt zurück zur Stockwerkanzeige, wobei der Ort der Buchung rot blinkend dargestellt wird.

Abbildung 4.4: Mitarbeitersuche

4.2 Der Spracherkenner

Einen Überblick über die bestehende Infrastruktur bietet Abbildung 4.5. Der ASR-Server erhält Konfigurations- und Initialisierungsangaben über die Socket Schnittstelle[1] vom Dialogmanager. Die Audiodaten kommen von der Soundkarte des Rechners und werden vom ASR-Server verarbeitet. Die Hypothesen werden an den Dialogmanager weitergeleitet.

Der Spracherkennener nutzt das Janus Recognition Toolkit (JRTk) [FGH+97] und den Ibis-Dekoder [SMFW01]. *JRTk* ist eine Erweiterung der Skriptprogrammiersprache Tcl/Tk [Tcl08] für die Entwicklung von Spracherkennungssystemen und stellt den

[1] Ein Socket ist eine bidirektionale Software-Schnittstelle zur Kommunikation zwischen Prozessen. Die Kommunikation kann lokal oder über ein Netzwerk erfolgen.

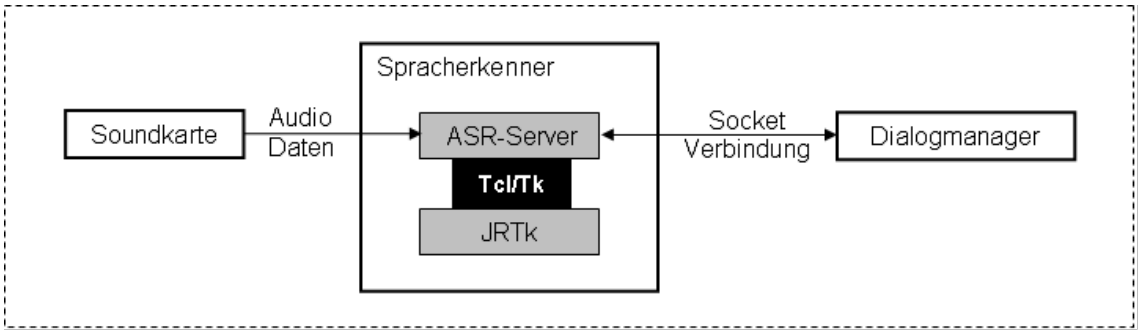


Abbildung 4.5: Bestehende Infastruktur der sprachverarbeitenden Einheit vor der Integration

Kommandozeileninterpreter *janus* zur Verfügung. Durch die Möglichkeit der Verarbeitung von Tcl-Skripten ist der Erkenner leicht konfigurierbar und praktisch beliebig erweiterbar. Zusätzlich werden spezielle Objektklassen zur Verfügung gestellt, durch die wichtige Bestandteile des Erkenners, wie z. B. Wörterbücher, Kodebücher und der Dekoder, instanziiert und durch objektspezifische Methoden manipuliert werden können.

Der *Ibis*-Dekoder kann statistische *n*-Gramm-Sprachmodelle und CFGs verarbeiten. Weiterhin werden Module zur Beschreibung des akustischen Modells bereitgestellt, die unter anderem der Erstellung einer HMM-Topologie dienen. Für diese Arbeit wird eine CFG im JSG-Format verwendet. Die Grammatik wurde im Lauf der Entwicklung mehrmals angepasst und erweitert. In Anhang A.1 ist die Fassung der Grammatik zu finden, wie sie für die durchgeführten Tests (siehe Kapitel 6) verwendet wurde.

Der Spracherkenner wird durch das Tcl-Skript *ASR-Server.tcl* implementiert. Nach dem Start von ASR-Server.tcl werden folgende Aspekte des Spracherkenners durch den *Dialogmanager* konfiguriert:

Audioquelle Auswahl des Mikrofons als Audioquelle (zu Testzwecken kann auch eine gespeicherte Audioaufnahme erneut verarbeitet werden).

Dekoder Auswahl eines echtzeitfähigen, Ibis-basierten, run-on-recognition-fähigen[2] singlepass[3]-Dekoders.

Akustisches Modell Auswahl eines akustischen Modells, das für distant speech[4] in Englisch trainiert ist. Das Modell beinhaltet 42 englische Phoneme und vier

[2] Die Verarbeitung einer Äußerung kann beginnen, obwohl das Mikrofon noch offen ist.
[3] Es erfolgt nur ein Dekodierungsdurchgang über die Audiodaten ohne Adaption des Dekoders.
[4] Deutsch: „Raumakustik“

Geräusch-/Pausen-Modelle (Garbage, Breath, Filler, Silence). Weiterhin ist das akustische Modell wie folgt charakterisiert:

- 2000 akustische Modelle (Kodebücher, entsprechen HMM Zuständen) mit 64 Gauß-Kurven mit diagonalen Kovarianzmatrizen pro Kodebuch. Jedes Phonem wird im Kontext von +/- zwei angrenzenden Phonemen (sog. Quintphone) durch eine tristate (begin, middle, end) HMM-Topologie modelliert.
- Die Vorverarbeitung erfolgt MFCC-basiert (siehe Abschnitt 3.2.1) mit einer 69-dimensionalen globalen semi-tied Kovarianzmatrix [Gal99]. Nach einer Linearen Diskriminanz-Analyse (LDA) ergibt sich ein 42 dimensionaler Merkmalsraum.
- Cepstrale Mittelwertsubtraktion und Varianznormalisierung (zur Normalisierung der Mittelwerte im Cepstralbereich).
- Training mit ca 90 Stunden Broadcast News (BN) und ca 100 Stunden „Meeting"-Daten mit Raumakustik aus der NIST [5] RT-04S (rich transcription) Evaluation [MFP+04], welche von ICSI[6], CMU[7], LDC[8] und NIST zur Verfügung gestellt wurden.

Sprachmodell Auswahl der CFG mit folgenden Parametern:

- sechs aktive (public) Regeln
- 16 Subregeln
- Das Vokabular ergibt sich aus den Worten in der Grammatik und vier speziellen Klassen (*@firstName*, *@lastName*, *@completeName* und *@roomName*), die innerhalb der Grammatik als Variablen für Mitarbeiter- und Raumnamen verwendet werden. Die Namensliste wird dem Spracherkenner bei der Konfiguration durch den Dialogmanager übergeben. Zum Zeitpunkt der Funktionstests umfasste das Wörterbuch 524 Eintrage inklusive Aussprachevarianten (siehe Tabelle 4.1).

Ist die Konfiguration abgeschlossen, kann die Spracherkennung initialisiert werden. Nach dem Start des Erkennungsprozesses sendet der Dialogmanager Verarbeitungsanfragen an den ASR-Server. Sobald ein grammatikalisch korrekter Satz erkannt

[5] National Institute of Standards and Technology
[6] International Computer Science Institute
[7] Carnegie Mellon University
[8] Linguistic Data Consortium

	ohne Varianten	mit Varianten
Grammatik	87	109
@firstName	60	143
@lastName	62	169
@completeName	66	69
@roomName	14	34
gesamt	223	524

Tabelle 4.1: Größe des Vokabulars mit und ohne Aussprachevarianten.

wurde und der Sprecher eine ausreichende Pause macht, wird der Dialogmanager benachrichtigt, der daraufhin die Hypothese des Spracherkenners anfordert. Die Implementierung des Dialogmanagers ist auf ein zustandsbasiertes System, wie es in Abschnitt 2.2 beschrieben wurde, ausgerichtet. Während des Betriebs werden daher Grammatikregeln gemäß dem aktuellen Zustand durch den Dialogmanager aktiviert und deaktiviert. Eine Übersicht über die Zustände der AP-Map wird in Abschnitt 5.1 des nächsten Kapitels gegeben.

Weitere Funktionen des Dialogmanagers sind in Abschnitt 4.3 erläutert. Abschnitt 5.3 gibt Aufschluss über die Erweiterungen des Dialogmanagers, die zur Integration des Spracherkenners in das Rauminformationssystem nötig waren.

4.3 Datenbankanbindung

AP-Map benötigt Informationen über Mitarbeiter und Buchungen von Räumen sowie Arbeitsplätzen. Die Angaben müssen immer auf dem aktuellen Stand gehalten werden, um eine effektive Mitarbeitersuche und korrekte Ressourcenanzeige zu gewährleisten. Die benötigten Informationen über Mitarbeiter und Ressourcen sind in einer Datenbank gespeichert und über Webservices[9] verfügbar.

Der Informationsaustausch zwischen der Datenbank und den einzelnen Modulen ist in Abbildung 4.6 dargestellt. Auch der Dialogmanager bezieht die notwendigen Auskünfte über Mitarbeiter- und Raumnamen indirekt von der Datenbank. Das in JAVA implementierte Modul *GetUsers* nutzt die Webservices, um die Namen abzurufen. Mit Hilfe der Sprachsynthese-Software Festival[10] wird automatisch für jeden Namen eine lautsprachliche Verschriftung erzeugt. In einer Textdatei (*allNames.txt*) werden

[9] Einen Webservice ist ein Programm das einen Datenaustausch zwischen Applikationen über internetbasierte Protokolle ermöglicht.

[10] http://www.cstr.ed.ac.uk/projects/festival/

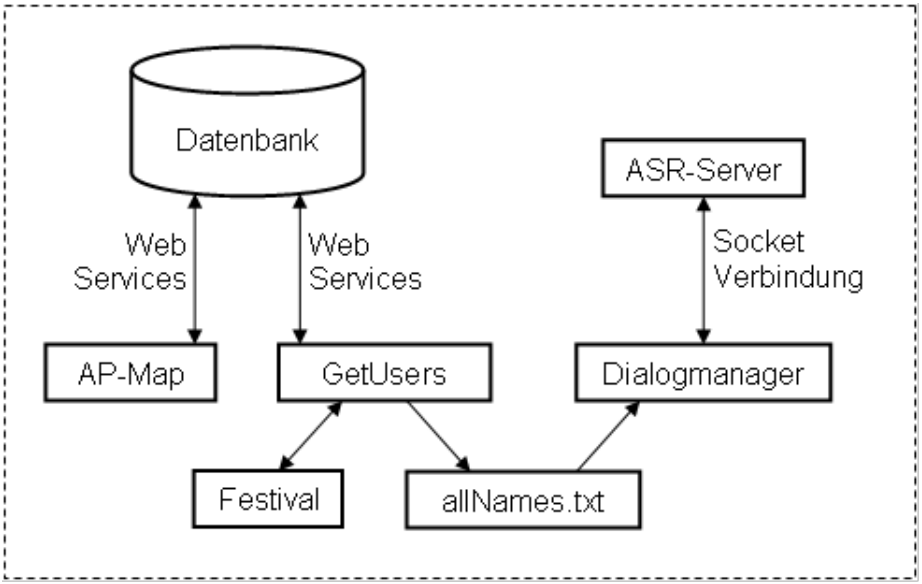


Abbildung 4.6: Datenbankanbindung

die Informationen zu jedem Namen gespeichert. Beim Start des Dialogmanagers wird die Textdatei eingelesen, um die Namen dem Spracherkenner hinzufügen zu können.

5 Integration der Sprachsteuerung

Die um die Spracheingabe erweiterte AP-Map wird als *Voice-AP-Map* bezeichnet. Die Voice-AP-Map ist genauso wie die AP-Map über den Touchscreen bedienbar. Die Steuerung über den Touchscreen wurde in Abschnitt 4.1 erläutert. Zusätzlich soll die Voice-AP-Map per Sprache steuerbar sein. Dabei sollen alle Buttons des Touchscreens durch einen Sprachbefehl abgebildet werden. Hautaufgabe bei der Integration der Sprachsteuerung war die Implementierung der Schnittstelle zwischen Dialogmanager und Voice-AP-Map, und der Entwurf eines Protokolls, um die Kommunikation zwischen den beiden Modulen zu ermöglichen. Die Beschreibung der erforderlichen Erweiterungen des Dialogmanagers sind Gegenstand des vorliegenden Kapitels. Abbildung 5.1 zeigt die Infrastruktur des Gesamtsystems. Die in diesem Kapitel behandelten Module sind blau hinterlegt. Die Datenbankanbindung wird in dieser Ansicht nicht berücksichtigt, da sie für die Beschreibungen nicht wichtig ist.

Die Zusatzimplementierungen seitens der Voice-AP-Map wurden von *Ivistar*[1], der Firma die die AP-Map erstellt hat, durchgeführt und werden nicht im Detail erläutert.

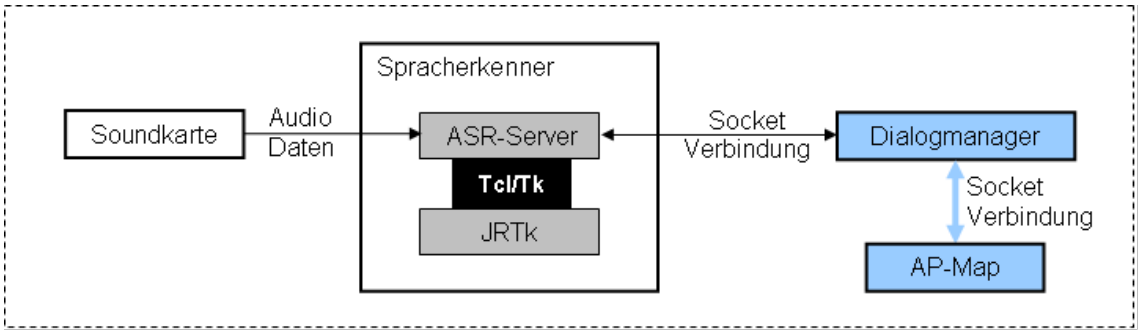


Abbildung 5.1: Infastruktur des Gesamtsystems. Die hier beschriebenen Module sind blau hinterlegt.

[1] http://www.ivistar.de/

Zustand	Bildschirmanzeige der AP-Map
L1	Stockwerkanzeige / Mainscreen
E1	Eventscreen
T1	Ressource blinkt mit Detailfenster
T2	Ressource blinkt ohne Detailfenster
MS1	Mitarbeitersuche
MS2	Mitarbeitersuche mit Fenster für Buchungsinformationen

Tabelle 5.1: Vom Dialogmanager unterschiedene Zustände und korrespondierende Bildschirmanzeigen der AP-Map.

5.1 Integrationskonzept

Zustandsbasierte Sichtweise

Versteht man die einzelnen Bildschirmanzeigen als Zustände, lässt sich die AP-Map einfach als zustandsbasiertes System verstehen. In Tabelle 5.1 sind die Zustände aufgelistet, die vom Dialogmanager unterschiedenen werden.

Für jeden Dialogzustand wurde eine Grammatik definiert, die beim Wechsel in diesen Zustand aktiviert wird. In Abbildung 5.2 ist der Zustandsautomat abgebildet, der im Dialogmanager implementiert ist. Jedem Zustand ist eine aktivierbare Grammatik zugeordnet. In Zustand *L1* (Stockwerkanzeige) ist beispielsweise die Grammatik *public <L1>* aktiviert; alle anderen Grammatiken sind deaktiviert. In der Grammatik des Dialogsystems (Anhang A.1) sind jeder aktivierbaren Grammatik weitere Untergrammatiken zugeordnet. Über die Untergrammatiken werden Zustandsübergänge ermöglicht. So kann von *L1* z.B. über *<showEvents>* nach *E1* gewechselt werden.

Synchronisierung

Zustandsübergänge können sowohl durch den Touchscreen (Touch-Events), als auch durch die Sprachsteuerung (ASR-Events) ausgelöst werden. Zur effizienten Nutzung des Systems ist es notwendig, das sich die Voice-AP-Map und der Dialogmanager möglichst immer im gleichen Dialogzustand befinden. Beide Module müssen daher Informationen über Zustandsübergänge zur gegenseitigen Synchronisierung austauschen.

Erfolgt ein Touch-Event während eine Spracheingabe verarbeitet wird, kann es vorkommen, dass die beiden Module kurzzeitig nicht synchron arbeiten. Dies ist der

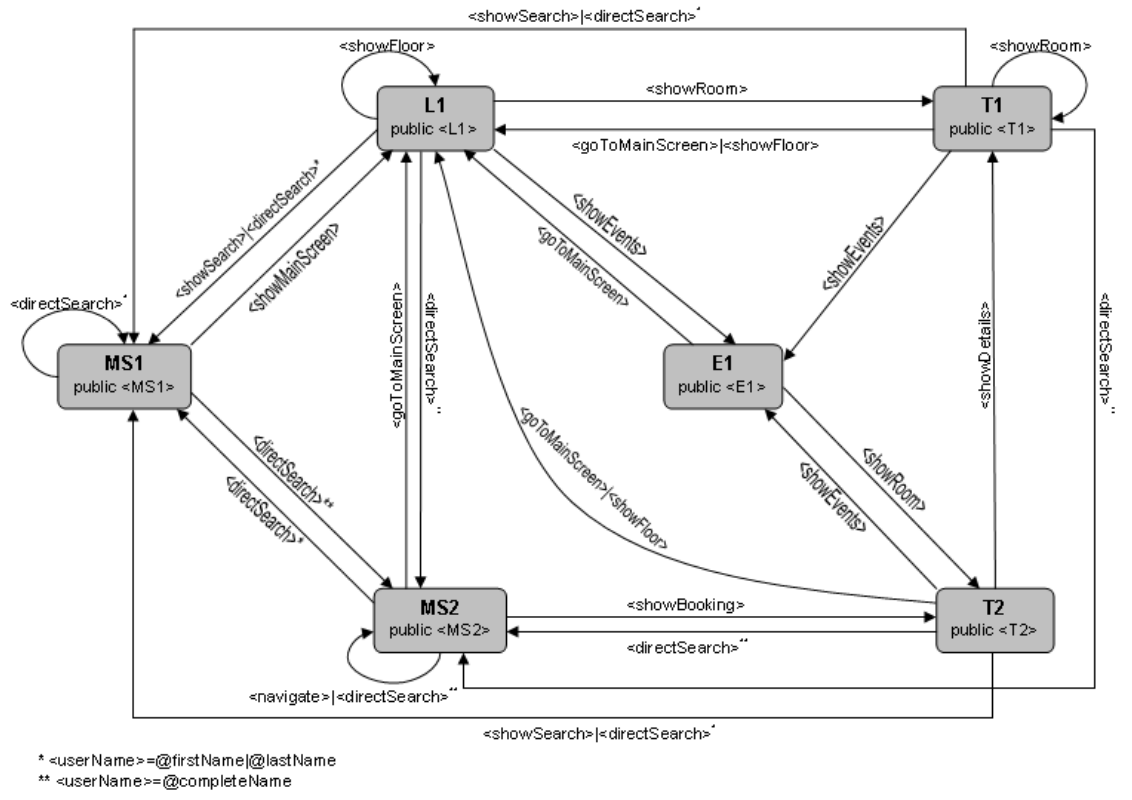


Abbildung 5.2: Überblick über den im Dialogmanager implementierten Zustandsautomat.

Fall, wenn durch die Berührung eines Buttons die Voice-AP-Map in einen anderen Zustand wechselt und gleichzeitig die Verarbeitung einer Spracheingabe noch nicht abgeschlossen ist. Die Hypothese der Spracherkennung bezieht sich dann auf einen vorherigen Zustand der Voice-AP-Map. Die aktive Grammatik liefert somit nicht die adäquaten Sprachbefehle für die Buttons der aktuellen Anzeige. Bekommt die Voice-AP-Map Nachrichten vom Dialogmanager, die nicht zum eigenen aktuellen Zustand passen, wird die Nicht-Synchronität dem Dialogmanager gemeldet. Zur Resynchronisierung schaltet der Dialogmanager in den Zustand, in dem sich die Voice-AP-Map befindet.

Initialisierung

Die Voice-AP-Map kann wie die AP-Map verwendet werden, wenn der Dialogmanager nicht gestartet ist. Eine Bedienung per Sprache ist dann nicht möglich. Im Unterschied zur AP-Map versucht die Voice-AP-Map in regelmäßigen Abständen sich mit dem Dialogmanager zu verbinden.

Für den Dialogmanager ist nicht festgelegt, welcher Dialogzustand initial eingenommen wird. Nach dessen Start wird daher auf eine eingehende Verbindungsanfrage gewartet. Wurde die Verbindung hergestellt, kann der Dialogmanager den aktuellen Zustand einnehmen, sobald die erste Zustandsaktualisierung von der Voice-AP-Map eingeht. Das für den Informationsaustausch definierte Protokoll wird im anschließenden Abschnitt 5.2 beschrieben.

5.2 Protokolldefinition

Wie die Informationen zwischen Dialogmanager und Voice-AP-Map ausgetauscht werden, wird durch ein Protokoll festgelegt. Zur strukturierten Darstellung der Daten wird das XML-Format[2] verwendet. Die auszutauschenden Informationen betreffen Touch-Events, ASR-Events und Bestätigungen über Zustandswechsel.

Touch-Events

Jede Berührung eines Buttons auf dem Touchscreen wird dem Dialogmanager von der Voice-AP-Map durch *state*-Nachrichten gemeldet. Diese sind wie folgt aufgebaut:

```
<state>
  <param comment="LevelButton.onRelease()" type="string"
  value="L1" name="current_state" />
</state>
```

Die *state*-Nachrichten beinhalten einen *param*-tag[3], der mit den vier Attributen *comment*, *type*, *value* und *name* versehen ist. In erster Linie ist das Attribut *value* von Bedeutung. Es beinhaltet den aktuellen Zustand der Voice-AP-Map. Falls sich der Dialogmanager nicht im selben Zustand befindet, muss er in den übermittelten Zustand wechseln. Das Attribute *comment* enthält die zuletzt seitens der Voice-AP-Map ausgeführte Aktion. *Type* gibt den Datentyp von *value* an. *Name* ist ein Bezeichner für die Art des Zustands.

[2] http://www.w3.org/XML/
[3] Deutsch: Kennung

function	param		
id	id	value	type
show_L1	levelid	$\in \mathbb{N}$	number
show_E1	-	-	-
show_T1	resourceid	$\in \mathbb{N}$	number
show_T2	resourceid	$\in \mathbb{N}$	number
show_MS1	name	$\in \{A, \ldots, Z\}^*$	string
show_MS2	userid	$\in \mathbb{N}$	number
showBooking	-	-	-
nextBooking	-	-	-
prevBooking	-	-	-

Tabelle 5.2: Funktionsnamen und Parameter, die vom Dialogmanager an die Voice-AP-Map übertragen werden.

ASR-Events

Die Hypothesen des Spracherkenners werden vom Dialogmanager einer semantischen Analyse unterzogen, bei der die Daten, die an die Voice-AP-Map übertragen werden müssen, ermittelt werden. Die zur Übertragung verwendete *command*-Nachricht hat folgenden Aufbau:

```
<command>
  <current_state id="L1" />
  <function id="show_MS2">
    <param id="userid" value="4480" type="number" />
  </function>
</command>
```

Der Wert des Attributs *id* des *current_state*-tags entspricht dem aktuellen Dialogzustand. Der Wert des Attributs *id* des *function*-tags benennt die Funktion, die von der Voice-AP-Map ausgeführt werden soll, um den gewünschten Zustandsübergang herbeizuführen. Die Voice-AP-Map stellt für jeden Zustandswechsel eine eigene Funktion zur Verfügung. Die Funktionen werden mit unterschiedlichen Parametern aufgerufen. Die Parameter werden im enthaltenen *param*-tag übertragen. Tabelle 5.2 gibt einen Überblick über die vorkommenden Fuktionen und deren Parameter.

Das hier angegebene Beispiel für eine *command*-Nachricht beauftragt die Voice-AP-Map einen Zustandswechsel von $L1$ nach $MS2$ herbeizuführen und die Buchungsinformationen des Mitarbeiters mit der *userid* 4480 anzuzeigen.

Returns

Als Antwort auf die *command*-Nachrichten sendet die Voice-AP-Map *return*-Nachrichten an den Dialogmanager. Diese dienen der Behandlung von Nicht-Synchronitäten und geben Auskuft darüber, ob ein durch Sprache initiierter Zustandsübergang erfolgreich war. Bei Synchronität hat die *return*-Nachricht folgenden Aufbau:

```
<return>
  <param type="boolean" value="true" name="success" />
</return>
```

Der enthaltene *param*-tag hat die drei Attribute *type*, *value* und *name*. Der Wert des Attributs *name* teilt mit, dass es sich um eine Nachricht über den Erfolg[4] des Zustandswechsels handelt. *Type* trifft eine Aussage über den Datentyp des Attributs *value*. *Value* hat den Wert *„true“*, was bedeutet, dass die Voice-AP-Map den Zusandswechsel vollzogen hat. Nach Erhalt einer positiven *return*-Nachricht (value=“true“) schaltet der Dialogmanager in den Zielzustand.

Einr negative *return*-Nachricht (value=“false“) ist folgendermaßen aufgebaut:

```
<return>
  <param type="boolean" value="false" name="success" />
  <param type="string" value="E1" name="current_state" />
</return>
```

Bei Nicht-Synchronität hat eine *return*-Nachricht zwei *param*-tags. Der erste ist im Prinzip identisch mit dem *param*-tag einer Synchonitäts-Nachricht, mit dem Unterschied, dass *value* den Wert *„false“* hat. Das Attribut *name* des zweiten *param*-tag gibt Aufschluss darüber, dass es sich bei der enthaltenen Information um den aktuellen Zustand[5] der Voice-AP-Map handelt. *Type* bestimmt wieder den Datentyp und *value* den Zustand in den der Dialogmanager wechseln muss, um wieder synchron mit der Voice-AP-Map zu laufen.

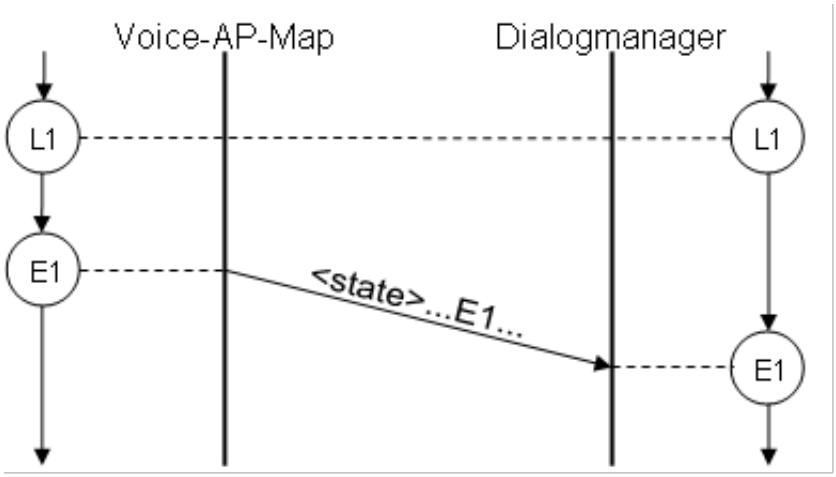


Abbildung 5.3: Nachrichtenfluss bei einem Touch-Event.

Nachrichtenfluss

Der Nachrichtenfluss bei einem Touch-Event besteht lediglich aus einer *state*-Nachricht (siehe Abbildung 5.3). Beide Module befinden sich im Zustand *L1*. Nach der Berührung des Touchsrceens wechselt die Voice-AP-Map sofort in den Zielzustand *E1* und verschickt die Nachricht an den Dialogmanager. Dieser nimmt den übermittelten Zustand ein.

Der Nachrichtenfluss bei einem ASR-Event bei Synchronität ist in Abbildung 5.4 dargestellt. Solange kein Touch-Event während der Verarbeitung einer Spracheingabe eingeht, laufen beide Module synchron. Beide Module befinden sich in Zustand *L1*. Nach Verarbeitung einer Spracheingabe sendet der Dialogmanager eine *command*-Nachricht an die Voice-AP-Map. Abhängig vom jeweiligen Zustandswechsel werden ein oder mehrere *state*-Nachrichten an den Dialogmanager gesendet. Die letzte *state*-Nachricht wird nach dem Wechsel in den gewünschten Zielzustand *MS2* geschickt. Nach der positiven *return*-Nachricht wird der Zustandswechsel auch vom Dialogmanager ausgeführt.

Abbildung 5.5 stellt den Nachrichtenfluss bei Nicht-Synchronität dar. Durch ein Touch-Event wird seitens der Voice-AP-Map ein Zustandswechsel von *L1* nach *E1* durchgeführt, was dem Dialogmanager durch eine *state*-Nachricht mitgeteilt wird. Die Nachricht trifft ein, während Sprachdaten verarbeitet werden. Der Dialogmanager schickt die *command*-Nachricht sofort nachdem die Hypothese des Spracherkenners feststeht. Der aktuelle Zustand der Dialogmanagers (*L1*) und somit der

[4] Engl.: success
[5] Engl.: current state

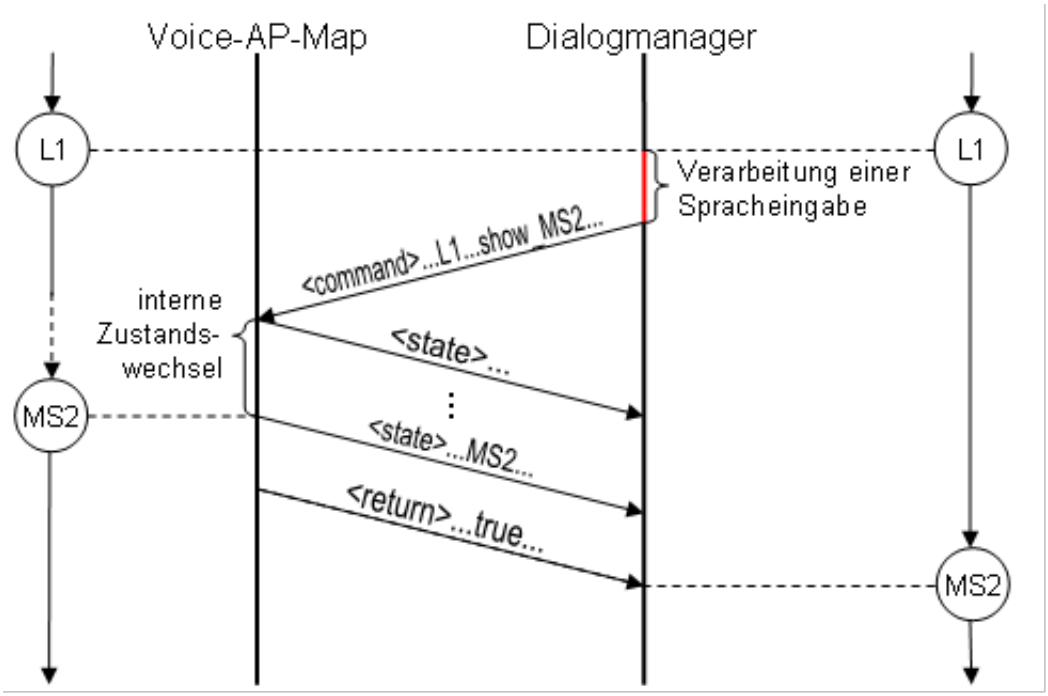


Abbildung 5.4: Nachrichtenfluss bei einem ASR-Event bei Synchronität.

current_state in der *command*-Nachricht, stimmen nicht mit dem *current_state* der Voice-AP-Map (*E1*) überein. Daher antwortet die Voice-AP-Map mit einer negativen *return*-Nachricht, woraufhin der Dialogmanager den Zustand einnimmt, der in der *return*-Nachricht übermittelt wird.

5.3 Voice-AP-Map Interface

Das Voice-AP-Map Interface ist ein Zusatzmodul des Dialogmanagers. Es bietet der Voice-AP-Map die Möglichkeit, eine Verbindung mit dem Dialogmanager herzustellen, um den Datenaustausch zwischen den Applikationen zu ermöglichen. Die Kommunikation mit der Voice-AP-Map muss parallel zu den bisherigen Aktivitäten des Dialogmanagers erfolgen.

5.3.1 Implementierung der Schnittstelle

Die Klassen des Voice-AP-Map Interface sind in einem Paket[6] zusammengefasst. Sie ergänzen den bisher bestehenden Java[7] Programmcode des Dialogmanagers. Die

[6] Ein Paket ist eine Sammlung zusammengehöriger Java-Klassen.
[7] http://java.sun.com/

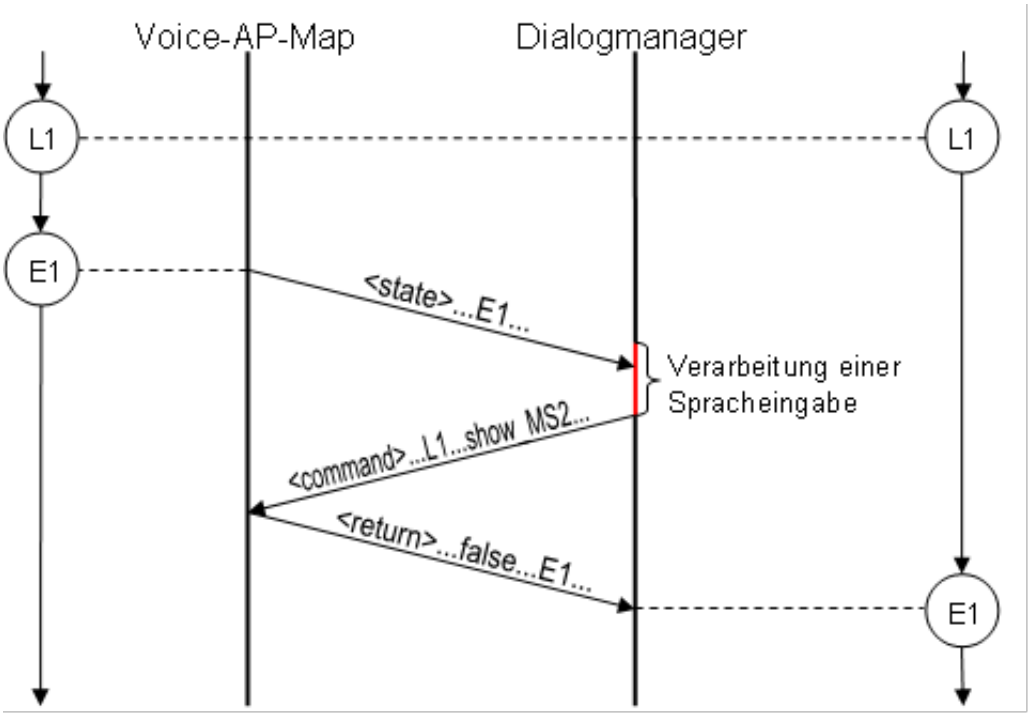


Abbildung 5.5: Nachrichtenfluss bei einem ASR-Event.

Struktur der Implementierung ist in dem Klassendiagramm in Abbildung 5.6 dargestellt.

Die Klasse *APMNetworkInterface* bietet neben dem Konstruktor[8] die Methoden *sendMessage(message:String)* und *handleReturn()* an. Erstere wird aufgerufen, um eine Nachricht an die Voice-AP-Map zu schicken. Dabei wird ein „String"-Datenstrom über den Socket abgesetzt. Die Methode *handleReturn()* verarbeitet *return*-Nachrichten, die dem Dialogmanager als Antwort auf eine *command*-Nachricht gesendet werden. Die Attribute *currentState* und *nextState* werden dabei für den Synchronisierungsmechanismus verwendet.

Bei der Instantiierung des *APMNetworkInterface* wird ein *SocketListener*-Objekt erzeugt. Durch den *SocketListener* kann der Dialogmanager seine eigentlichen Aktivitäten und den Empfang von Nachrichten quasi parallel ausführen. Ermöglicht wird dies, indem der *SocketListener* als Unterklasse der Klasse *Thread*[9] implementiert ist. Die Methode *listen(String address, int port)* dient der Initialisierung der Socket-Verbindung. Hat sich die Voice-AP-Map verbunden, wird der *SocketListener-Thread*

[8] Ein Konstruktor ist eine spezielle Prozedur, die bei der Erzeugung eines Objektes aufgerufen wird, um das Objekt in einen definierten Anfangszustand zu bringen. Zu Konstruktoren in Java siehe [Ull06].

[9] Ein Thread ist ein Ausführungsstrang eines Prozesses. Innerhalb eines Prozesses kann es mehrere Threads geben, die untereinander auf ihre Daten zugreifen können. Zu Threads in Java siehe [Ull06].

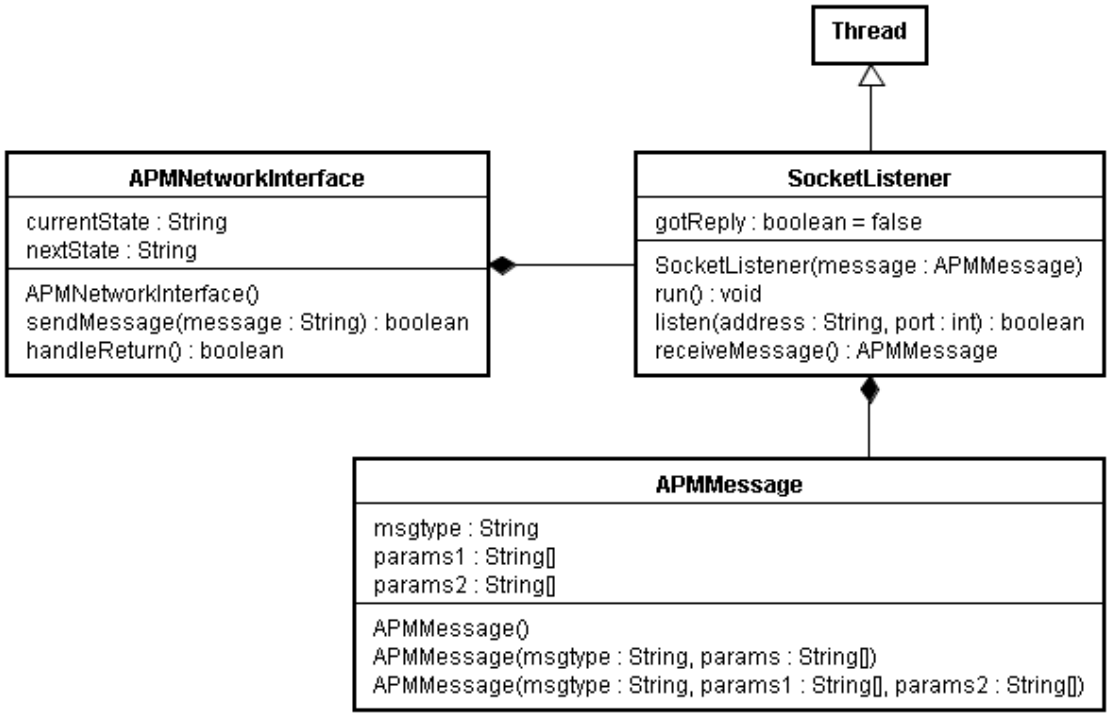


Abbildung 5.6: Klassendiagramm des Voice-AP-Map Interface.

gestartet. Dazu wurde die *run()*-Methode implementiert. In dieser wird *receiveMessage()* in einer Endlosschleife ausgeführt. Die Methode dient dem Empfang der Nachrichten von der Voice-AP-Map. Zur strukturierten Speicherung von Nachrichten wird vom *SocketListener* ein *APMMessage*-Objekt erzeugt. Dafür stehen mehrere Konstruktoren zur Auswahl, um die Anzahl der Parameter (*params1, params2*) an die übertragenen Daten anpassen zu können.

5.3.2 Integration in Dialogmanager

Die durch das Voice-AP-Map Interface gegebene Funktionalität musste in verschiedene Methoden des Dialogmanagers integriert werden. Die Funktionsweise dieser Methoden wird hier nicht im Detail erläutert. Es wird jedoch kurz auf ihren Nutzen und auf die jeweilige Verwendung des Voice-AP-Map Interface Bezug genommen.

JRTkDemo Die Hauptklasse des Dialogmanagers heißt *JRTkDemo*. Sie enthält einen Konstruktor bei dessen Aufruf das Voice-AP-Map Interface erzeugt wird.

main In der *main*-Methode von *JRTkDemo* wird der *SocketListener*-Thread gestartet und auf eine erste Nachricht von der Voice-AP-Map gewartet, um festzustellen, welcher Zustand initial eingenommen werden muss.

processAudio Die Methode, die den Spracherkennungsprozess abwickelt ist *processAudio*. Sie wird nach der vollständigen Initialisierung des Dialogmanagers in einer Schleife ausgeführt. Am Anfang der Methode wurde Code eingefügt, über den geprüft wird, ob eine neue Nachricht über den *SocketListener* eingegangen ist. Ist dies nicht der Fall, wird der Spracherkennungsprozess angestoßen. Ist eine Nachricht eingegangen, muss im Falle eines Zustandswechsels zuerst die Grammatik aktualisiert werden.

listenerAPMResult Zustandswechsel bei Touch-Events werden von der Methode *listenerAPMResult* in die Wege geleitet. Dem Voice-AP-Map Interface muss dabei der neue aktuelle Zustand mitgeteilt werden.

listenerASRResult Zustandswechsel bei ASR-Events werden von der Methode *listenerASRResult* veranlasst. Über den integrierten Code werden Nachrichten an die Voice-AP-Map gesendet und die Synchronität der Module sicher gestellt.

5.4 Erweiterungen der AP-Map

Die Voice-AP-Map unterscheidet sich optisch nicht von der AP-Map. Nach dem Start ist sie genau wie die AP-Map über den Touchscreen benutzbar. Um die Steuerung per Sprache zu ermöglichen, müssen jedoch einige Zusatzfunktionen implementiert werden. Diese werden im Folgenden nur kurz im Prinzip dargelegt. In ihrer Struktur unterscheiden sich die Erweiterungen meist nur wenig von denen, die für den Dialognamager erläutert wurden.

Verbindungsaufbau Um die Sprachsteuerung nutzbar zu machen, muss die Voice-AP-Map regelmäßig im Abstand von wenigen Sekunden versuchen, eine Verbindung mit dem Dialogmanager aufzubauen.

Senden und Empfangen von Nachrichten Bei Berührung des Touchscreens wird zwischen den unterschiedlichen Anzeigen der AP-Map hin und her gewechelt. Damit diese Zustandswechsel dem Dialogmanager mitgeteilt werden können, müssen dem definierten Protokoll entsprechende Nachrichten versendet werden. Umgekehrt muss eine Möglichkeit geschaffen werden, Nachrichten vom Dialogmanager zu empfangen.

Parsen und reagieren auf Nachrichten Eingehende Nachrichten müssen geparst werden, um relevante Informationen zu extrahieren. Es muss gewährleistet werden, dass alle Nachrichten, die dem gemäß des Protokolls definiert sind, interpretierbar sind. Alle Zustandsübergänge in Abbildung 5.2 müssen dabei ermöglicht werden. Der bestehende Programmcode muss so erweitert werden, dass ASR-Events die gleichen Reaktionen am Touchsrceen auslösen wie korrespondierende Touch-Events.

Synchronisierungsinformation Die Voice-AP-Map führt selbst keine Zustandswechsel zur Synchronisierung durch. Es muss aber festgestellt werden, ob der *current_state* der eingehenden Nachrichten dem aktuellen Zustand der Voice-AP-Map entspricht. Die *return*-Nachrichten müssen gemäß der Lage der Synchronisierung gewählt werden.

6 Tests und Evaluierung

Nach aktuellen Forschungserkenntnissen ist noch ungeklärt, welchen Einfluss die Erweiterung unimodaler Systeme zu Systemen mit multimodalen Interaktionsmöglichkeiten auf die Usability hat [Ovi99]. Dies ist auch darin begründet, dass benötigte Evaluationsmethoden noch in der Entwicklungsphase sind (siehe Kapitel 2.3). Zur Evaluation des Rauminformationssystems wurde daher eine Methode erarbeitet, die hier neben der eigentlichen Fragestellung beschrieben werden soll. Im Anschluss finden sich die die Versuchsergebnisse und deren zusammenfassende Diskussion.

6.1 Fragestellung

Die Erweiterung des Rauminformationssystems, um die Sprachsteuerung, soll der Benutzerfreundlichkeit des Systems zuträglich sein. Die Mitarbeitersuche ist durch Sprachbefehle mit weniger Interaktionsschritten, als mit Touchscreen möglich. Die Raumsuche per Sprache stellt eine Erweiterung des Systems dar, da dem Touchscreen eine entsprechende Funktion fehlt.

In dieser Arbeit werden drei Implementierungen des Systems miteinander verglichen: (1) ein System mit Touchscreen, (2) ein System mit Spracheingabe und (3) ein System mit multimodaler Eingabe (Touch und Sprache). Es soll festgestellt werden, welchen Einfluss die Veränderung der Eingabemodalitäten auf die Usability des Systems hat. Die dabei eingesetzten Methoden zur technischen und zur Usability-Evaluation werden in Abschnitt 6.2 erläutert.

Neben dieser grundsätzlichen Qualitätsbewertung des Systems wurden Hypothesen entwickelt, die im Rahmen der Untersuchung verifiziert werden sollen.

Vorwissen ermöglicht Benutzern die Bedienung eines Systems mit geringerem Lernaufwand [TSK03, Stu00]. Experten sollten die Bedienung des Systems aufgrund ihres

Vorwissens schneller erlernen, was sich in geringeren Bearbeitungszeiten bemerkbar machen kann. Daher lautet die erste Hypothese:

Hypothese 1 Experten lösen die Aufgaben schneller als Novizen.

Durch die Integration der Spracherkennung ist es möglich ganze Wörter (wie z. B. Namen von Mitarbeitern) durch eine Benutzeräußerung zu erkennen [ST95]. Sprachbedienung bietet somit eine Abkürzung an, wenn per Touchscreen mehrere einzelne Buchstaben eingegeben werden müssen. Daraus folgt die Annahme:

Hypothese 2 Sprache ist die bevorzugte Modalität für Aufgaben, die durch Sprache mit weniger Interaktionsschritten ausgeführt werden können.

Jedes Resultat des Spracherkennungsvorganges ist zu einem gewissen Anteil mit Fehlern behaftet, was die Zuverlässigkeit der Sprachsteuerung beeinträchtigt [ST95]. Die Bedienung mit Touchscreen hingegen ist weniger fehleranfällig. Dies dürfte sich auf das Interaktionsverhalten der Benutzer auswirken. Demnach wurden folgende Hypothesen aufgestellt:

Hypothese 3 Der Touchscreen ist die bevorzugte Modalität für Aufgaben, die mit einem einzigen Interaktionsschritt ausgeführt werden können.

Hypothese 4 Der Touchscreen wird insgesamt besser beurteilt als die Spracheingabe.

Die Bereitstellung einer weiteren Eingabemodalität sollte die Bedienung des Systems für den Nutzer einfacher gestalten [Möl07]. Die entsprechende Annahme lautet:

Hypothese 5 Die vom Benutzer empfundene Qualität ist am höchsten bei multimodaler Bedienung.

Der Vergleich von gemessenen Bearbeitungszeiten und Nutzerurteilen zeigt in unterschiedlichen Untersuchungen widersprüchliche Befunde [NW08]. Für das Rauminformationssystem wird angenommen, dass längere Bearbeitungszeiten zu einer schlechteren Beurteilung führen, woraus sich folgende Hypothese ableitet:

Hypothese 6 Die vom Benutzer wahrgenommene Systemgeschwindigkeit wird bei längerer Bearbeitungszeit schlechter beurteilt.

6.2 Vortests und Testmethode

6.2.1 Vortests

Es wurde ein *Cognitive Walkthrough* (siehe auch Abschnitt 2.3.2) mit sechs Usability Experten der T-Labs durchgeführt, um möglichst viele Usability-Probleme vor den eigentlichen Tests zu beseitigen. Die Experten lösten die Aufgaben (siehe nächster Abschnitt 6.2.2.1 (Aufgaben)) einmal per Touchscreen und einmal per Sprache, und protokollierten die gefundenen Usability-Probleme. Der erste Durchgang (Touch) wurde an einem Touch-Panel durchgeführt. Für den zweiten Durchgang (Sprache) wurde ein Besprechungsraum genutzt, um Störgeräusche zu reduzieren. Dort war ein Versuchsrechner mit 19" Touchmonitor und Mikrofon platziert.

Die am häufigsten genannten Probleme betrafen die Module Spracherkennung, grafische Schnittstelle, Dialogmanagement und das Zusammenspiel im Gesamtsystem.

Probleme der Spracherkennung

- Teilweise gab es bei der Interaktion Inkonsistenzen zwischen Beschriftungen der grafischen Schnittstelle und durch diese implizierte Sprachbefehle, die weitestgehend behoben werden konnten.
- An einigen Stellen waren erwünschte Befehle nicht verfügbar, da sie nicht Bestandteil der Grammatik waren. Unter anderem wurden konsistentere Befehle zwischen Mitarbeitersuche und Raumsuche gewünscht. Dies ist zwar theoretisch möglich, steigert jedoch die Anzahl an Varianten für die Spracherkennung. Dadurch sinkt in der Regel die Erkennungsleitung. Die Grammatik wurde daher überarbeitet, indem jeweils zwei bis vier häufig genutzte und möglichst eindeutige Befehle pro Zustandsübergang eingeführt wurden.
- Stark kritisiert wurde die hohe Fehlerrate der Spracherkennung. Es wurde versucht, die Erkennungsleitung über eine Erhöhung der minimal zulässigen Konfidenz[1] zu verbessern. Auch das erläuterte Vorgehen bei der Konstruktion der Grammatik sollte der Erkennungsleistung zuträglich sein. Der erzielte Effekt war allerdings gering.

[1] Die Konfidenz ist ein Maß für die Präzision der Hypothese. Sie wird üblicherweise durch einen Wert zwischen 0 und 1 angegeben, wobei ein höherer Wert eine höhere Präzision darstellt.

Probleme der grafischen Schnittstelle

- Größter Kritikpunkt war, dass die GUI (Graphical User Interface, deutsch: grafische Benutzerschnittstelle) keine Raumsuche zur Verfügung stellt. Eine Aufgabe für die Versuchspersonen bestand jedoch darin, einen bestimmten Raum zu finden.
- Für den Wechsel von E1 nach L1 (Bezeichnungen siehe Tabelle 5.1) wurde ein eigener Button erwünscht, da ein einfaches Berühren des Screens für viele Nutzer nicht intuitiv ist.
- In E1 kann zum Ort einer Veranstaltung nur durch direktes Antippen des Raumnamens gesprungen werden. Da dieser nicht als Button gekennzeichnet ist, sollte stattdessen ermöglicht werden, die ganze Zeile einer Veranstaltung anzutippen und so zum entsprechenden Raum weitergeleitet zu werden.

Probleme des Dialogmanagements Wenn bei der Mitarbeitersuche bereits ein Vor- oder Nachname erkannt wurde, genügt es bei der aktuellen Implementierung nicht, nur den zugehörigen Nach- oder Vornamen zu nennen. Es ist in diesem Fall nötig Vor- und Nachnamen gemeinsam und in dieser Reihenfolge auszusprechen, um sich die Buchungsliste des gesuchten Mitarbeiters anzeigen zu lassen.

Probleme des Gesamtsystems Das System wurde als zu langsam empfunden. Dies kann Nutzer z.B. dazu veranlassen, einen Sprachbefehl zu wiederholen, obwohl das System noch verarbeitet. Grund hierfür könnten die Leistungsmerkmale des bei den Vortests genutzten Rechners gewesen sein. Für die Usability Tests wurde ein leistungsstärkerer Rechner verwendet.

Die Probleme der grafischen Schnittstelle konnte im Rahmen dieser Evaluation nicht gelöst werden. Der Quelltext des Moduls stand nicht zur Verfügung. Sofern es der Aufwand zuließ, wurden die übrigen Probleme durch das angegebene Vorgehen behandelt. Die verwendete Grammatik ist in Anhang A.1 zu finden.

Aufgaben	Beschreibung	Interaktionsschritte	
		Sprache	Touch
A1	Anzeige Mainscreen	1	1
A2	Anzeige 18. Etage	1	1
A3	Mitarbeitersuche	3	6
A4	Raumsuche	2	-
A5	Anzeige Eventscreen	1	1
A6	Anzeige Raum zu Event	1	1

Tabelle 6.1: Minimal nötige Anzahl an Interaktionsschritten pro Aufgabe

6.2.2 Testmethode

Dieser Abschnitt beschreibt die Methode, die zur Durchführung der Tests und zur Usability-Evaluation erarbeitet wurde. Es wird dargelegt, in welchem Rahmen die Experimente durchgeführt wurden und welche Daten gesammelt wurden.

6.2.2.1 Aufgaben

Die sechs von den Versuchsteilnehmern zu bearbeitenden Aufgaben bestanden im wesentlichen darin, zwischen Screens zu wechseln und nach Buchungen von Mitarbeitern und Räumen zu suchen. Die detaillierte Formulierung der Aufgaben kann in Anhang A.4 nachgeschlagen werden. Zur Lösung der Aufgaben A1, A2, A5 und A6 genügte ein Interaktionsschritt. Zur Lösung der Aufgaben A3 und A4 (Raumsuche) waren mehrere Interaktionsschritte nötig (siehe Tabelle 6.1). Eine grafische Eingabemaske, wie für die Mitarbeitersuche, ist für die Raumsuche nicht vorhanden. Daher ist eine gezielte Suche nach Räumen über den Touchscreen nicht möglich. Per Sprache werden Räume durch die Äußerung eines einfachen Befehls, wie „Show Sputnik“, auf dem Display angezeigt. Bei der Mitarbeitersuche per Sprache steht eine Abkürzung zur Verfügung. Während die Namen auf dem Touchscreen eingetippt werden müssen, genügt es per Sprache einen Befehl wie „Search Sebastian Möller“ zu nennen.

6.2.2.2 Probanden

An den Evaluationstests nahmen 17 Männer und 19 Frauen im Alter von 21 bis 39 Jahren teil ($M_{alter} = 27.19, SD_{alter} = 3.45$). Die Hälfte der Teilnehmer waren T-Labs Mitarbeiter, die in der Verwendung des Rauminformationssystems, aufgrund

ihres Vorwissens über das unimodale Touch-System, als *Experten* bezeichnet wurden. Die andere Hälfte bestand aus Studenten, die für die Teilnahme an dem Experiment bezahlt wurden. Sie wurden hinsichtlich der Systembenutzung als *Novizen* bezeichnet. Alle Versuchspersonen waren deutsche Muttersprachler. Alter und Geschlecht war in beiden Gruppen gleich ausbalanciert.

6.2.2.3 Versuchsmaterial

Personenbezogener Fragebogen (Anhang A.3) Dieser Fragebogen diente zur Erfassung demografischer Daten, wie Alter, Geschlecht und Beruf bzw. Studiengang der Teilnehmer. Es wurde außerdem abgefragt, ob die Teilnehmer bereits Erfahrung mit Sprachsteuerungen und Bewegungssteuerungen haben. Weiterhin sollten die Teilnehmer ihre Englischkenntnisse auf einer 5-stufigen Skala, von „keine“ bis „Muttersprache, perfekt“, einschätzen, da das System englischsprachig ist.

Subjektive Beurteilung (Anhang A.5) Das Experiment wurde in drei Blöcken durchgeführt. Die Versuchspersonen sollten Aufgaben entweder mit Hilfe des Touchscreens, mit der Sprachsteuerung, oder multimodal bearbeiten. Am Ende jedes Blocks sollte ein für die entsprechende Eingabemodalität (Sprache, Touch oder multimodal) modifizierter SASSI Fragebogen zur subjektiven Beurteilung des Systems ausgefüllt werden. SASSI wurde für unimodale sprachbasierte Systeme entworfen. Daher wurden drei Items für Touchscreen und multimodale Bedienung angepasst (siehe Tabelle 6.2). SASSI hat negative und positive Items, wie z. B.:

```
NEGATIV: Das System reagierte zu langsam.
POSITIV: Die Interaktion mit dem System ist schnell.
```

Den negativen und positiven Aussagen entsprechend, haben die zugehörigen Skalen eine negative oder positive Polung. Negative Items wurden umgepolt, dass ein höherer Zahlenwert einer besseren Bewertung entsprach. Weitere Details zu SASSI wurden bereits in Abschnitt 2.3.2 erläutert.

Logfiles Mit den Logfiles wurden die Hypothesen der Spracherkennung zur Bestimmung des Parameters *Concept Error Eate* (CER) für Sprach- und multimodale Bedienung erfasst. Die *Concept Error Rate* (siehe auch Abschnitt 2.3.1) ist ein spracheingabebezogener Parameter. Im Gegensatz zur *Sentence Error Rate* (SER), bei der die

Item 7	
Sprache	Es ist klar, wie mit dem System zu sprechen ist.
Touch	Es ist klar, wie mit dem System zu agieren ist.
Multimodal	Es ist klar, wie mit dem System zu agieren ist.
Item 16	
Sprache	Ich fragte mich manchmal, ob ich das richtige Wort benutze.
Touch	Ich fragte mich manchmal, ob ich den richtigen Button benutze.
Multimodal	Ich fragte mich manchmal, ob ich die richtige Aktion ausführe.
Item 22	
Sprache	Ich wusste zu jeder Zeit, was ich sagen konnte.
Touch	Ich wusste zu jeder Zeit, welchen Button ich benutzen konnte.
Multimodal	Ich wusste zu jeder Zeit welche Aktion ich ausführen konnte.

Tabelle 6.2: Anpassung der SASSI Items 7, 16 und 22.

Ausgabe des Systems nur dann als korrekt gezählt wird, wenn alle Worte richtig erkannt wurden, bezieht sich die CER auf Fähigkeiten bezüglich des Sprachverstehens. Sie ermöglicht eine Bewertung der Parserleistung auf einer semantischen Ebene und lässt dabei eine unvollständige Analyse zu [BEG+96]. Fehler der Spracherkennung, die keine falsche Systemreaktion verursachen[2], werden dadurch vernachlässigt. Die CER berechnet sich wie folgt:

$$CER = \frac{S + D + I}{N_{ig}} \tag{6.1}$$

Dabei ist S die Anzahl der ersetzten Konzepte (Substitutions), D die Anzahl der gelöschten Konzepte (Deletions), I die Anzahl der eingefügten Konzepte (Insertions) und N_{ig} (ig = in grammar) die Anzahl derjenigen von einem Probanden geäußerten Konzepte, die mit der Grammatik produzierbar waren. Äußerungen der Probanden, die nicht mit der verwendeten Grammatik produzierbar waren, wurden in diese Berechnung nicht aufgenommen.

Weiterhin wurde anhand der Logfiles die Bearbeitungszeit bei der Bedienung mit Touchscreen ermittelt. Die Auswertung der Logfiles geschah teils automatisch, teils expertenbasiert.

[2] D.h. das Konzept wurde richtig erkannt

Audioaufnahmen Während Sprach- und multimodaler Bedienung wurden die Äußerungen der Versuchspersonen aufgezeichnet. Für die Aufnahmen wurden Referenztranskriptionen auf Wortebene zur Ermittlung der CER und der *Out-Of-Grammar Rate* (OOG) angefertigt. Die OOG-Rate berechnet sich wie folgt:

$$OOG = \frac{N_{oog}}{N_{total}} = \frac{N_{oog}}{N_{ig} + N_{oog}} \tag{6.2}$$

dabei ist N_{oog} (oog = out of grammar) die Anzahl der Äußerungen, die nicht mit der Grammatik produzierbar waren, N_{ig} (ig = in grammar) die Anzahl der Äußerungen, die mit der Grammatik produzierbar waren und N_{total}, die Anzahl der Äußerungen, die kein *offtalk*[3] [SSB01] waren.

Außerdem wurden die Bearbeitungszeiten für Sprach- und multimodale Bedienung aus den Aufnahmen gewonnen.

Beobachtungsprotokolle In den Beobachtungsprotokollen wurde festgehalten, welche Modalität bei multimodaler Bedienung zur Lösung einer Aufgabe verwendet wurde. Dabei wurde die Modalität, mit der zuerst versucht wurde eine Aufgabe zu lösen, als *bevorzugte Modalität* bezeichnet.

6.2.2.4 Versuchsaufbau

Der Versuch fand im Usability-Labor der T-Labs statt und wurde mit einem Probanden pro Sitzung durchgeführt. Abbildung 6.1 zeigt eine Skizze des Versuchsaufbaus. Ein 32" Breitformat Display (NEC Multisync LCD 3210) mit Touch-overlay (Interactive Technologies Eclipse LCD 432) fungierte als externer Monitor des Versuchsrechners (Fujitsu Simens Computers S Series Lifebook), der mit einem 2,0 GHz Dualcore Prozessor und 3 GB Arbeitsspeicher ausgestattet war. Zum Anschluss des verwendeten Grenzflächenmikrofons (Beyerdynamic MPC 66 V) wurde ein externes Audiointerface (EDIROL UA-25) verwendet. Als Browser wurde der Internet Explorer 6 und als Betriebssystem Windows XP Professional genutzt.

Das Display wurde in Augenhöhe aufgestellt, dass es für einen Erwachsenen im Stehen einfach per Touchscreen zu bedienen und einzusehen war. Das Mikrofon lag mittig kurz vor dem Touchscreen.

[3] Mit *offtalk* werden Äußerungen bezeichnet, die nicht direkt an das System gerichtet waren.

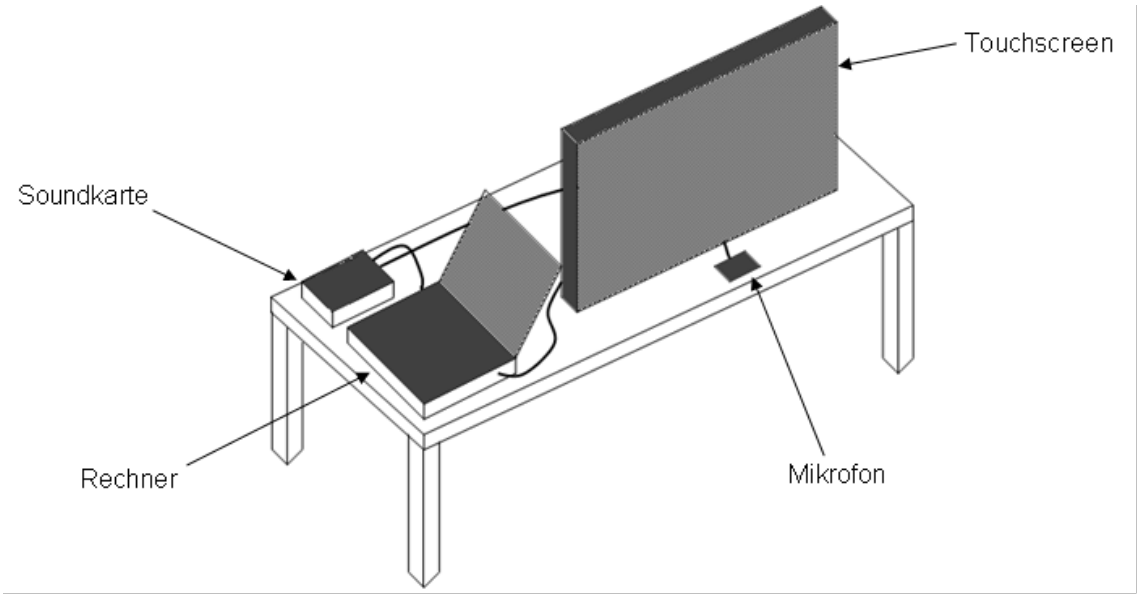


Abbildung 6.1: Skizzierter Versuchsaufbau

6.2.2.5 Versuchsablauf

Der Versuch dauerte zwischen 45 und 60 Minuten. Zu Beginn des Versuchs bekamen die Teilnehmer schriftliche Erläuterungen zur Applikation und zum Versuchsablauf (siehe Anhang A.2). Sofern die Erläuterungen nicht ausreichten erklärte der Versuchsleiter zusätzlich den Ablauf und das Funktionsprinzip des Rauminformationssystems. Im Anschluss erhielten die Versuchspersonen den personenbezogenen Fragebogen. Danach begann der eigentliche Versuch. Wie schon erwähnt, wurde das Experiment in drei Blöcken (1. Sprache, 2. Touch, 3. Multimodal) durchgeführt, um Vergleiche zwischen den drei Systemen mit unterschiedlichen Eingabemöglichkeiten ziehen zu können. Der Versuchsablauf ist in Abblidung 6.2 schematisch dargestellt. Die Reihenfolge der ersten beiden Blöcke wurde nach jedem Probanden vertauscht, um evtl. beim Probanden auftretende Lerneffekte zu berücksichtigen. Jeder Block begann mit der Bearbeitung der Aufgaben durch den Probanden und wurde durch einen modalitätsspezifischen SASSI Fragebogen abgeschlossen. Die Teilnehmer wurden dazu angehalten die Aufgaben in der gegebenen Reihenfolge zu bearbeiten. Wurde die Interaktion bei der Bedienung per Sprache aufgrund von Erkennungsfehlern gestört, war es den Teilnehmern erlaubt, sich per Touchscreen bis zu dem Interaktionsschritt zu bewegen, zu dem sie per Sprache bereits gelangt waren, um von dort aus mit der Lösung fortzufahren.

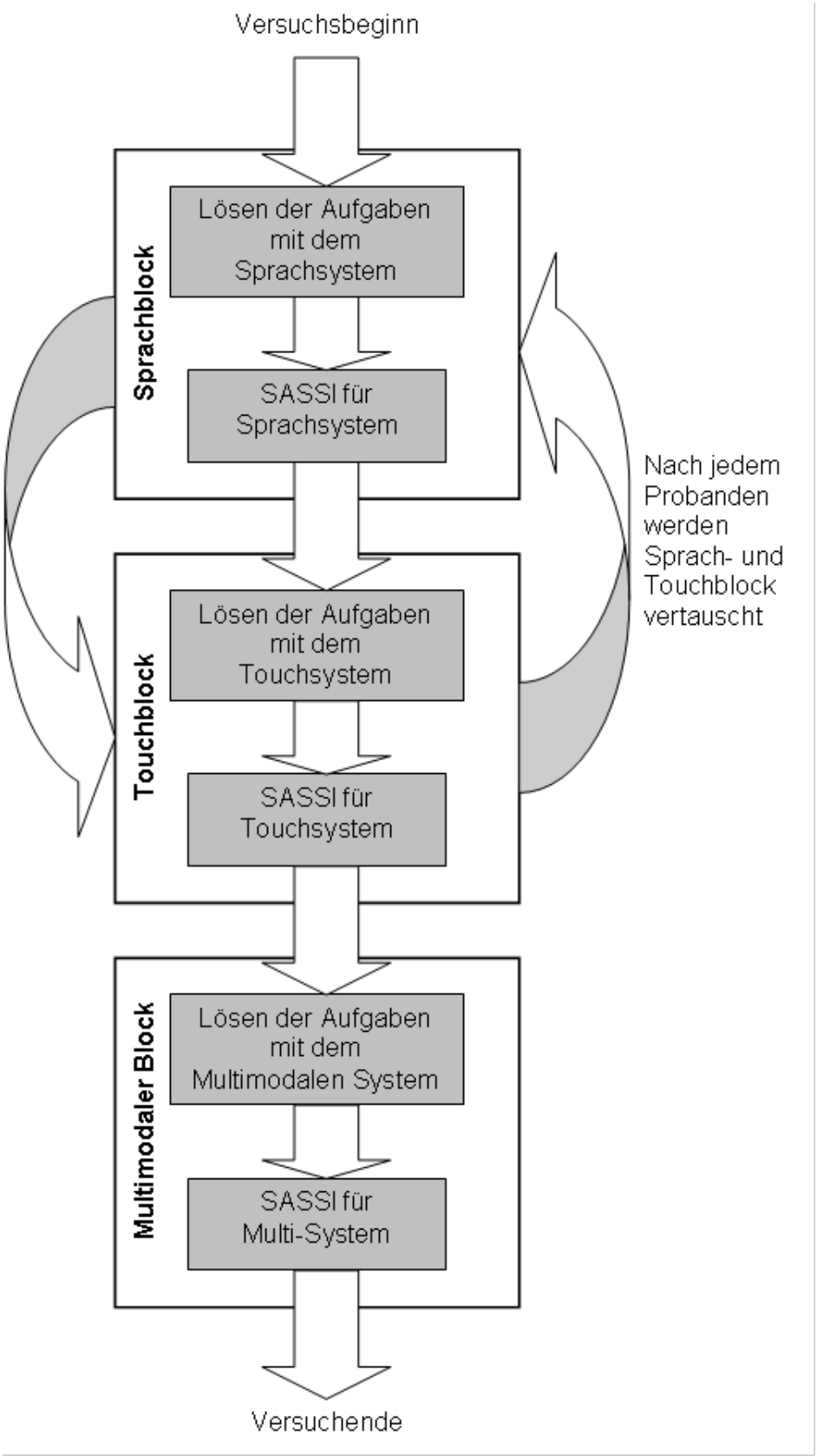


Abbildung 6.2: Versuchsablauf eines Probanden

		Sprache N=35	Touch N=35	Multi N=35
Experten	Mittelwert	8:44	2:46	1:46
	Standardabweichung	2:46	0:54	0:45
Novizen	Mittelwert	8:50	3:17	2:00
	Standardabweichung	2:41	1:12	0:57
Total	Mittelwert	8:47	3:02	1:53
	Standardabweichung	2:41	1:05	0:51

Tabelle 6.3: Durchschnittliche Bearbeitungszeiten der Modalitäten in [Minuten:Sekunden]

6.3 Technische Tests und Performanz Daten

Für sämtliche statistischen Tests wurde ein α-Niveau von .05 verwendet. Wenn der angegebene p-Wert kleiner oder gleich dem α-Niveau ist, liegt Signifikanz vor [BD06].

Bearbeitungszeiten In Tabelle 6.3 sind die Bearbeitungszeiten der einzelnen Durchläufe aufgelistet. Mit über acht Minuten benötigen die Versuchspersonen zur Aufgabenbearbeitung mit der Sprachsteuerung wesentlich länger als mit dem Touchscreen. Am schnellsten war die multimodale Bearbeitung. Allerdings fand die multimodale Bearbeitung immer im dritten Block statt, während die ersten beiden Blöcke (Sprache und Touchscreen) nach jedem Probanden vertauscht wurden. Somit ist im dritten Block ein Lerneffekt bezüglich der Interaktion mit dem System zu erwarten. Eine Varianzanalyse zeigte signifikante Unterschiede zwischen den Bearbeitungszeiten der einzelnen Modalitäten ($F_{2,68} = 185,02; p = .000; part.eta^2 = .845$).

Ein t-Test bezüglich der Expertise ergab keine signifikanten Unterschiede zwischen *Experten* und *Novizen* (siehe Tabelle 6.4) [BD06]. Hypothese 1 kann somit nicht bestätigt werden.

Modalitätennutzung Die Untersuchung der Modalitätennutzung bei multimodaler Bedienung ergab, dass Sprache für die Aufgaben $A3$ und $A4$ häufiger genutzt wurde (siehe Tabelle 6.5). Ein χ^2-Test zeigte bei diesen Aufgaben signifikante Unterschiede

Input	t(df)	p
Touch	t(33)=.142	.166
Sprache	t(34)=.298	.767
Multi	t(34)=.807	.425
gesamt	t(33)=.736	.466

Tabelle 6.4: Ergebnisse des T-Tests zur Expertise der Probanden

	bevorzugt				
Aufgabe	Sprache	Touch	χ_1^2	p	N
A1	50	50	.000	1.000	36
A2	22,2	77,8	11.11	**.001**	36
A3	75	25	9	**.004**	36
A4	83,3	16,7	16	**.000**	36
A5	57,1	42,9	.714	.500	35
A6	26,5	73,5	5.77	**.024**	34
∅	52,35	47,65			

Tabelle 6.5: Bevorzugte Modalität bei multimodaler Bedienung in [%]; $p<0.05$

zwischen der Modalitätennutzung [BD06]. Bei Aufgaben mit mehreren Interaktionsschritten wurde die Spracheingabe somit häufiger genutzt. Hypothese 2 lässt sich somit bestätigen.

Bei den Aufgaben bei denen ein Interaktionsschritt zur erfolgreichen Lösung genügte, ergab sich für die Modalitätennutzung: Aufgabe A2 und A6 wurden überwiegend per Touchscreen gelöst, Aufgabe A1 wurde genau von der Hälfte der Probanden per Touchscreen gelöst, Aufgabe A5 wurde etwas häufiger mit Sprache als mit Touchscreen gelöst. Ein χ^2-Test zeigte signifikante Unterschiede zwischen der Modalitätennutzung bei den Aufgaben A2 und A6. Für die Aufgaben A1 und A5 zeigte sich kein signifikanter Unterschied. Hypothese 3 wird dadurch nur teilweise bestätigt.

Es wurden Reihenfolgeneffekte der ersten beiden Blöcke ermittelt. Gruppe A bezeichnet die Probanden, die die Aufgaben zuerst per Sprache und dann per Touchscreen lösten. Gruppe B bezeichnet die Probanden, die die Aufgaben zuerst per Touchscreen und dann per Sprache lösten. Für beide Gruppen wurde die Modalitätennutzung für den dritten (multimodalen) Block ermittelt. Tabelle 6.6 stellt den Anteil der Sprachbenutzung der beiden Gruppen gegenüber. Ein χ^2-Test zeigte keine statistisch signifikanten Unterschiede zwischen Gruppe A und Gruppe B.

Aufgabe	Gruppe A	Gruppe B	χ^2_1	p	N
A1	44,4	55,6	.444	.370	36
A2	22,2	22,2	.000	.655	36
A3	77,8	72,2	.148	.500	36
A4	88,9	77,8	.800	.329	36
A5	41,2	72,2	3.441	.065	35
A6	35,3	23,5	.567	.354	34
∅	53,9	51,6			

Tabelle 6.6: Benutzung von Sprache bei multimodaler Bedienung in [%]. Gruppe A hatte Sprache als ersten Block. Gruppe B hatte Sprache als zweiten Block.

Zur Bestimmung des Effekts des Vorwissen der Probanden wurde ein weiterer χ^2-Test durchgeführt. Für die Modalitätennutzung bei Block drei konnten keine statistisch signifikanten Unterschiede zwischen Experten und Novizen festgestellt werden. Tabelle 6.7 stellt den Anteil der Sprachbenutzung der beiden Gruppen gegenüber.

Aufgabe	Experten	Novizen	χ^2_1	p	N
A1	44,4	55,6	.444	.370	36
A2	27,8	16,7	.643	.345	36
A3	44,4	50	.111	.500	36
A4	88,9	72,2	1.596	.201	36
A5	55,6	58,8	.038	.558	35
A6	33,3	18,8	.926	.285	34
∅	53,9	51,6			

Tabelle 6.7: Benutzung von Sprache durch Experten/Novizen bei multimodaler Bedienung in [%].

Concept Error Rate Tabelle 6.8 fasst die Ergebnisse der CER zusammen. Die Werte der Experten und der Novizen unterscheiden sich nur geringfügig. Die Berechnung eines Mann-Whitney-U-Tests ergab, dass die Unterschiede nicht signifikant sind ($Mann-Whitney-U = 27.000; p_{zweiseitig} = .626$) [BD06]. Insgesamt ist die CER mit 66,8% sehr hoch.

Out-of-grammar rate Tabelle 6.9 fasst die Ergebnisse der OOG-rate zusammen. Die Werte der Experten und der Novizen unterscheiden sich wiederum nur geringfügig. Die Berechnung eines Mann-Whitney-U-Tests ergab, dass die Unterschiede nicht

	Experten N=8	Novizen N=8	Total N=16
CER	59,2	65,9	63
Insertion	48	56	104
Substituion	38	34	72
Deletion	24	28	52

Tabelle 6.8: CER für Bedienung mit Sprache in [%] und Anzahl an Spracherkennungsfehlern.

signifikant sind ($Mann - Whitney - U = 22.000; p_{zweiseitig} = .310$). Insgesamt ist die OOG-rate mit 58% sehr hoch.

	Experten N=8	Novizen N=8	Total N=16
OOG-Rate	61,7	54,3	58

Tabelle 6.9: OOG-rate für Bedienung mit Sprache in [%]

6.4 Subjektive Daten

Für sämtliche statistischen Tests wurde ein α-Niveau von .05 verwendet. Wenn der angegebene p-Wert kleiner oder gleich dem α-Niveau ist, liegt Signifikanz vor [BD06].

Im Folgenden werden die Unterschiede zwischen den Benutzerbewertungen der drei Systeme anhand der SASSI Skalen erläutert. Für die Gesamtbewertung wurde der Durchschnitt der Skalen gebildet. Die Ergebnisse der Varianzanalyse sind in Tabelle 6.10 aufgeführt. Sofern kein Post-Hoc-Test durchgeführt wurde, gab es Unterschiede zwischen allen drei Systemen. Durch den Post-Hoc-Test werden die Systeme bei denen keine Unterschiede zu beobachten waren, nochmals auf Signifikanz untersucht [BD06].

In Abbildung 6.3 sind die Bewertungen der einzelnen SASSI Skalen und der Gesamtqualität für alle Modalitäten als Balkendiagramm gegenübergestellt.

Accuracy (Genauigkeit)
Es ergab sich ein signifikanter Effekt zwischen allen Modalitäten. Am besten bewerteten die Versuchsteilnehmer die Genauigkeit des Systems mit Touchscreen ($M_{Touch} =$

SASSI Skala	F		p	$part.Eta^2$	N
Accuracy	(2,68)	128,86	**.000**	.791	35
Likeability	(2,68)	139,22	**.000**	.804	35
Cognitive Demand	(2,70)	116,68	**.000**	.769	36
Annoyance	(2,70)	79,16	**.000**	.693	36
Hability	(2,68)	62,70	**.000**	.648	35
Speed	(2,70)	5,34	**.007**	.132	36
Gesamt	(2,70)	180,25	**.000**	.837	36

Tabelle 6.10: Einfluss der Modalitäten auf die Benutzerwahrnehmung; $p<0.05$

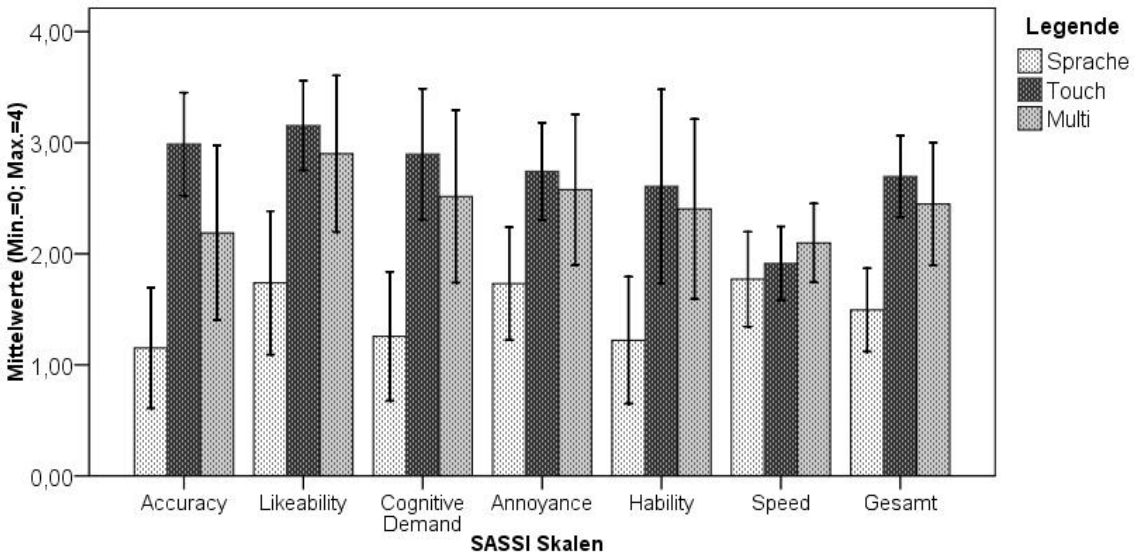


Abbildung 6.3: Bewertungen der einzelnen SASSI Skalen und der Gesamtqualität. Die Fehlerbalken stellen eine Standardabweichung dar.

$2.99, SD_{Touch} = 0.46$). Danach folgte die multimodale Bedienung ($M_{Multi} = 2.20, SD_{Multi} = 0.79$). Am schlechtesten wurde die Genauigkeit des Systems mit Spracheingabe beurteilt ($M_{Sprache} = 1.18, SD_{Sprache} = 0.53$).

Likeability (Gefallen)
Es ergab sich ein signifikanter Effekt zwischen allen Modalitäten. Am besten gefiel den Versuchsteilnehmern das Systems mit Touchscreen ($M_{Touch} = 3.16, SD_{Touch} = 0.40$). Danach folgte das multimodale System ($M_{Multi} = 2.95, SD_{Multi} = 0.65$). Am schlechtesten gefiel das System mit Spracheingabe ($M_{Sprache} = 1.78, SD_{Sprache} = 0.58$).

Cognitive Demand (Kognitive Beanspruchung)
Es ergab sich ein signifikanter Effekt zwischen allen Modalitäten. Am besten wurde das System mit Touchscreen beurteilt ($M_{Touch} = 2.87, SD_{Touch} = 0.60$). Danach folgte das multimodale System ($M_{Multi} = 2.52, SD_{Multi} = 0.78$). Am schlechtesten fiel die Bewertung des Systems mit Spracheingabe aus ($M_{Sprache} = 1.25, SD_{Sprache} = 0.57$).

Annoyance (Ärgernis)
Es ergab sich ein signifikanter Effekt zwischen Sprachsteuerung und Touchscreen, und Sprachsteuerung und multimodaler Eingabe. Dem Urteil der Versuchsteilnehmer zu folge, bereitet die Sprachsteuerung ($M_{Sprache} = 1.73, SD_{Sprache} = 0.50$) mehr Unannehmlichkeiten als der Touchscreen ($M_{Touch} = 2.72, SD_{Touch} = 0.46$) und auch mehr als die multimodale Eingabe ($M_{Multi} = 2.58, SD_{Multi} = 0.68$). Ein Post-Hoc-Test zeigte keine signifikanten Unterschiede zwischen Touchscreen und multimodaler Bedienung [BD06].

Habitability (Weiß der Nutzer was er sagen kann und was das System tut?)
Es ergab sich ein signifikanter Effekt zwischen Sprachsteuerung und Touchscreen, und Sprachsteuerung und multimodaler Eingabe. Die Art der Umsetzung der Sprachsteuerung ($M_{Sprache} = 1.22, SD_{Sprache} = 0.57$) entsprach weniger den Systemvorstellungen der Versuchsteilnehmer, als die Art der Umsetzung des Touchscreens ($M_{Touch} = 2.64, SD_{Touch} = 0.85$) und die Art der Umsetzung der multimodalen Eingabe ($M_{Multi} = 2.44, SD_{Multi} = 0.78$). Ein Post-Hoc-Test zeigte keine signifikanten Unterschiede zwischen Touchscreen und multimodaler Bedienung.

Speed (Systemgeschwindigkeit)
Es ergab sich ein signifikanter Effekt zwischen Touchscreen und multimodaler Eingabe, und Sprachsteuerung und multimodaler Eingabe. Die Versuchsteilnehmer fanden das System mit multimodaler Eingabe ($M_{Mulit} = 2.10, SD_{Multi} = 0.35$) schneller als den Touchscreen ($M_{Touch} = 1.90, SD_{Touch} = 0.33$) und auch schneller als die Sprachsteuerung ($M_{Sprache} = 1.81, SD_{Sprache} = 0.47$). Ein Post-Hoc-Test zeigte keine signifikanten Unterschiede zwischen Touchscreen und multimodaler Bedienung.

Gesamtbewertung
Es ergab sich ein signifikanter Effekt zwischen allen Modalitäten. Am besten viel die Gesamtbewertung für das System mit Touchscreen ($M_{Touch} = 2.99, SD_{Touch} = 0.46$) aus. Danach folgte die multimodale Bedienung ($M_{Multi} = 2.20, SD_{Multi} = 0.79$). Am schlechtesten fiel die Gesamtbewertung für das System mit Spracheingabe ($M_{Sprache} = 1.18, SD_{Sprache} = 0.53$) aus. Hypothese 4 kann somit bestätigt werden. Hypothese 5 kann nicht bestätigt werden.

6.5 Vergleich von subjektiven Daten und Performanz Daten

Für sämtliche statistischen Tests wurde ein α-Niveau von .05 verwendet. Wenn der angegebene p-Wert kleiner oder gleich dem α-Niveau ist, liegt Signifikanz vor [BD06].

Zur Untersuchung der Wirkungszusammenhänge zwischen subjektiven Daten und Performanz Daten wurden die SASSI Skalen und die technischen Messwerte auf Korrelation getestet.

Eine bivariate Korrelation über alle Modalitäten hinweg zeigte, wie erwartet, einen negativen Zusammenhang zwischen der Speed-Skala des SASSI und den erhobenen Bearbeitungszeiten (Pearson's r= -.217, p =.012). Das bedeutet bei längeren Bearbeitungszeiten wurde die Systemgeschwindigkeit schlechter bewertet. Dadurch wird Hypothese 6 bestätigt.

6.6 Diskussion und Implikationen

6.6.1 Diskussion der Versuchsergebnisse

Ergebnisse der Hypothesen

Ausgehend vom aktuellen Forschungsstand [TSK03, Stu00] wurde angenommen, dass *Experten* die Aufgaben aufgrund ihres Vorwissens über das System schneller lösen, als *Novizen* (Hypothese 1). Es zeigte sich zwar bei allen Blöcken ein numerischer Unterschied zu Gunsten der *Experten*, dieser war jedoch nicht signifikant. Das Vorwissen der Probanden stellte keine Einflussgröße der Modalitätennutzung dar. Dies kann darin begründet sein, dass der Touchscreen auf möglichst intuitive Bedienbarkeit ausgelegt ist und *Experten* wie *Novizen* ausführliche Instruktionen zur Funktionsweise des Systems bekamen. Dies kann den Mangel an Erfahrung bei den *Novizen* weitgehend kompensiert haben. Die Sprachbedienung war für beide Gruppen neu und ihr Vorwissen gereichte den Experten hier offenbar nicht zum Vorteil.

Bezüglich der Modalitätennutzung ließ sich gemäß den Erwartungen feststellen, dass die Sprachsteuerung bei multimodaler Bedienung für Aufgaben, die durch Sprache mit weniger Interaktionsschritten gelöst werden konnten, häufiger genutzt wurde (Hypothese 2). Bei der Mitarbeitersuche bevorzugten es die Probanden den Namen auszusprechen, anstatt ihn per Touchscreen einzugeben. Dies deutet darauf hin, dass die Sprachsteuerung hier als Vorteil empfunden wurde. Da per Touchscreen keine direkte Raumsuche zur Verfügung stand, war es zu erwarten, dass hierbei die Sprachsteuerung bevorzugt wurde.

Obwohl die Zuverlässigkeit der Sprachsteuerung durch Erkennungsfehler beeinträchtigt ist [ST95] wurden nur zwei von vier Aufgaben, die mit nur einem Interaktionsschritt lösbar waren, signifikant häufiger mit Touchscreen gelöst (Hypothese 3). Die beiden Aufgaben bei denen keine Signifikanz festgestellt werden konnte, sind A1 und A5. Interessanterweise handelt es sich dabei um den Wechsel vom Eventscreen zum Mainscreen (A1) bzw. vom Mainscreen zum Eventscreen (A5). A1 könnte deswegen bevorzugt mit Sprache gelöst worden sein, da es keine eindeutige Schaltfläche auf dem Touchscreen für den Wechsel gibt[4]. Vielleicht wurden die Aufgaben A1 und A5 beim Sprachblock häufiger erfolgreich gelöst, als die anderen Aufgaben mit

[4] Stattdessen kann der Touchscreen an beliebiger Stelle berührt werden.

nur einem Interaktionsschritt, und daher im multimodalen Block hier häufiger Sprache genutzt. Um diese These zu untersuchen, bedarf es allerdings weiterer Versuche.

Entsprechend Hypothese 4 wurde die Bedienung mit Touchscreen insgesamt besser beurteilt, als die Bedienung mit Sprache. Womöglich ist die Fehleranfälligkeit reiner Sprachbedienung der Grund für das Ergebnis [ST95].

Multimodale Bedienung sollte gegenüber unimodaler Bedienung die Benutzbarkeit eines Systems vereinfachen [Möl07]. In der Praxis zeigt sich jedoch auch, dass die Bereitstellung einer zusätzlichen Eingabemodalität nicht automatisch zu einer besseren Qualitätsbewertung führt [Ovi99]. Dies war auch bei dieser Untersuchung der Fall. Hypothese 5 kann daher nicht bestätigt werden. Die Sprachsteuerung bietet dem Nutzer zwar Vorteile (z. B. Abkürzungen oder die Möglichkeit der Interaktion in *hands-busy* Situationen), stört aber die Interaktion auch häufig durch Erkennungsfehler. Die schlechten Werte der CER belegen die hohe Fehlerrate des verwendeten Erkenners und können eine Ursache für das schlechtere Ergebnis der multimodalen Steuerung sein.

Die SASSI Speed-Skala und die Bearbeitungszeiten zeigen negative Korrelation. Das heißt die vom Nutzer wahrgenommene Geschwindigkeit des Systems wurde bei längeren Bearbeitungszeiten schlechter beurteilt (Hypothese 6). Demnach korrelieren kürzere Bearbeitungszeiten mit einer besseren Bewertung der Systemgeschwindigkeit. Die Bearbeitungszeiten sind möglicherweise von den Interaktionsproblemen der Probanden abhängig. Daher könnte auch die Bewertung schlechter ausfallen, wenn viele Interaktionsprobleme zu beobachten sind. Die genaueren Zusammenhänge könnten Gegenstand einer anschließenden Untersuchung sein.

Ergebnisse der subjektiven Beurteilung

Am besten wurde die Accuracy des Touchscreens bewertet. Bei multimodaler Bedienung litt die Präzision des Systems vor allem an Einfügungen der Spracherkennung. Die Gegenüberstellung von Hypothesen und Referenztranskriptionen zeigte, dass die meisten Fehler aus Einfügungen von kurzen Worten des Vokabulars bestanden. Derartige Einfügungen kommen durch Störschall zustande, der aus Tippgeräuschen, nicht-sprachlichen Äußerungen und anderen Nebengeräuschen resultieren kann. Diesem Problem könnte Sorge getragen werden, indem die *Garbagemodelle*[5] [BG00]

[5] Garbagemodelle sind spezielle akustische Modelle zur Klassifikation von Worten außerhalb des Vokabulars und nicht-sprachlichen Äußerungen.

des Spracherkenners um häufig auftretende Störgeräusche und Worte außerhalb des Vokabulars erweitert werden. Besonders bei der Suche des Mitarbeiters „Sebastian Möller“ kam es häufig zu Vertauschungen. Dabei wurde meist der Nachname nicht erkannt, weswegen das Konzept *find <completeName=SebastianMöller>* mit dem Konzept *find <firstName=Sebastian>* verwechselt wurde. Die Erkennung des *<completeName>* war elementar wichtig für die erfolgreiche Absolvierung einer Aufgabe. Es ist daher davon auszugehen, dass sich dieses Verhalten bei der Bewertung der Accuracy bemerkbar gemacht hat. Die schlechte Erkennungsleitung ist wahrscheinlich dadurch zu erklären, dass dem Spracherkenner ein englischsprachiges phonetisches Alphabet zugrunde lag und das [œ][6] in „Möller“ keinen englischsprachigen Vokal darstellt [TE07]. Um das Problem zu beheben, könnte das akustische Modell des Spracherkenners um die Modelle der fehlenden nicht-englischen Phoneme erweitert werden. Die Korrektheit der Wörterbucheinträge müsste jedoch für jeden nicht-englischen Namen separat geprüft werden.

Die Bewertung der Likeability des Systems fiel ähnlich wie die Accuracy Berwertung aus. Der Abstand zwischen Touchscreen und multimodalem System ist geringer als bei der Accuracy, jedoch immer noch signifikant. Interessant wäre, ob eine Steigerung der Accuracy der Sprachstuerung auch zur Steigerung der Likeability des multimodalen Systems führen würde. Zur Beantwortung dieser Frage wäre eine vergleichende Evaluierung einer Weiterentwicklung des Systems wünschenswert.

Die kognitive Beanspruchung war den Nutzerurteilen zufolge bei der Sprachsteuerung am höchsten. Dies kann daran gelegen haben, das feste Befehle durch die Grammatik vorgegeben waren. Erschwerend kam hinzu, dass die Befehle den Nutzern vorher nicht bekannt waren. Stattdessen mussten sie selbst überlegen welcher Sprachbefehl für welchen Interaktionsschritt passend sein könnte und diesen in Englisch formulieren. Insofern entspricht es den Erwartungen, dass den Probanden die Nutzung des Touchscreens am einfachsten fiel und die multimodale Steuerung die mittelere Position einnahm. Flexiblere Sprachbefehle würden die Bedienung des Systems intuitiver gestalten. Dies könnte die kognitive Beanspruchung senken. Der Einsatz eines statistischen Sprachmodells statt einer Grammatik wäre ein guter Ansatz zur Modellierung der stellenweise komplexeren Dialogzustände und würde die Flexibilität der möglichen Sprachbefehle erhöhen.

Der Faktor Annoyance war bei Sprachbedienung am höchsten, was in Anbetracht der

[6] Symbol gemäß Internationalem Phonetischen Alphabet [IPA05].

negativen Bewertungen der bereits diskutierten Skalen zu erwarten ist. Bemerkenswert ist jedoch, dass es keinen signifikanten Unterschied zwischen dem Touchscreen und dem multimodalen System gibt, welches durch den Einfluss der Sprachsteuerung fehleranfälliger ist. Man kann annehmen, dass die Nutzer hier Vorteile bei der Benutzung von Sprache erkannt haben. Diese haben sich jedoch bei der Bewertung mit den Nachteilen gegeneinander aufgehoben, sodass die Annoyance von Touchscreen und multimodalem System insgesamt ungefähr gleich bewertet wurden. Interessant wäre eine weiterführende Untersuchung, bei der überprüft wird, inwiefern sich eine Verbesserung von Accuracy, Likeability und Cognitive Demand auf die Bewertung der Annoyance auswirken.

Auch bei dem Faktor Habitability gibt es keinen signifikanten Unterschied zwischen Touchscreen und multimodaler Bedienung. Allerdings schneidet die Sprachsteuerung deutlich schlechter ab, was auch hier vermuten lässt, dass sich die bereits erläuterten Vor- und Nachteile multimodaler Bedienung bemerkbar machen. Auch hier wäre es interessant, welchen Effekt eine Veränderung der übrigen Skalen auf die Habitability hat.

Multimodale Bedienung wurde von den Nutzern als am schnellsten empfunden. Dies war aus mehreren Gründen zu erwarten: Es konnten Abkürzungen per Sprache genutzt werden. Auf Grund der höheren Erfolgsquote ist die Bedienung mit Touchscreen bei Aufgaben mit nur einem Interaktionsschritt oft schneller. Im dritten Block ist ein Lerneffekt bezüglich der Interaktion mit dem System zu erwarten. Erstaunlich ist jedoch, dass die Bedienung mit Touchscreen und Sprache, was die Systemgeschwindigkeit angeht, als ungefähr gleich bewertet wurden, obwohl sich im Durchschnitt eine deutlich schnellere Bedienung mit Touchscreen ergab. Nutzer nehmen die Bearbeitungszeiten unterschiedlicher Modalitäten scheinbar unterschiedlich wahr.

Über alle Skalen wurde entgegen dem wünschenswerten Ergebnis, dass das multimodale System am besten bewertet wird [Möl07], das System mit Touchscreen am besten bewertet. Die Gesamtbewertung setzt sich aus den Einzelbewertungen der SASSI Skalen zusammen. Daher sollte eine Verbesserung der Usability des multimodalen Systems durch die bei den Einzelskalen bereits vorgeschlagenen Eingriffe erreicht werden können.

Weitere Versuchsergebnisse

Die insgesamt sehr hohe CER ist vor allem durch die hohe Anzahl an Einfügungen und Vertauschungen begründet, die wie bei den Ergebnissen der Qualitätsbewertung

schon erwähnt, evtl. durch unzureichende *Garbagemodelle* und fehlende akustische Modelle für Phoneme des Deutschen begründet werden kann. In Anbetracht der speziellen Domain von Personen- und Raumnamen ist es, um eine Reduktion der CER herbeizuführen, wahrscheinlich unumgänglich, das akustische Modell des Spracherkenners entsprechend anzupassen. Das Phoneminventar des Englischen reicht nicht aus um realistische Transkriptionen für alle Namen zu erzeugen. In hohem Maße dürften sich auch die erschwerenden Bedingungen durch die Raumakusik bemerkbar gemacht haben. Die akustischen Modelle sind zwar speziell für Raumakustik trainiert. Dies ändert jedoch nicht die Tatsache, dass im Gegensatz zu Modellen, die mit „clean speech“ und für ein Mikrofon zur Nahbesprechung trainiert sind, der Informationsverlust bei den Modellen für Raumakustik hoch ist. Eine Überprüfung der Audiodaten ergab weiterhin, dass viele Einfügungen aufgrund sehr kurzer Schallereignisse stattfanden. Solcher Schall sollte eigentlich durch die *voice activity detection* (VAD)[7] abgefangen werden. Es sollte daher geprüft werden, ob man die verwendete VAD noch verbessern kann.

Das hohe Maß an OOG Äußerungen deutet darauf hin, dass die benutzte Grammatik dem Anspruch der intuitiven Bedienbarkeit des Systems nicht gewachsen ist. In ihrer aktuellen Form wurde darauf geachtet, möglichst eindeutige Befehle für die möglichen Zustandsübergänge zu verwenden. In der Praxis hat sich jedoch gezeigt, dass viele Nutzer denselben Befehl für mehrere Zustandsübergänge benutzen, sobald dieser Befehl einmal funktioniert hat. Beispielsweise kam es vor, dass Nutzer bei denen *„show <roomName>“* funktionierte, auch *„show <floorName>“* ausprobierten, was jedoch mit der verwendeten Grammatik nicht produzierbar war. Eine entsprechende Erweiterung der Grammatik könnte Abhilfe verschaffen. Dies führt jedoch auch dazu, dass der Raum der mit der Grammatik produzierbaren Lösungen wächst, was wiederum die Erkennungsleistung senken kann. Da die Namens-Domain ohnehin schon als problematisch bezüglich des Suchraums bezeichnet werden kann, und da die intuitive Bedienbarkeit und eine praktikable feste Grammatik schwer in Einklang zu bringen sind, sollte der Versuch unternommen werden, ein statistisches Sprachmodel einzusetzen. Letzteres könnte den Anforderungen des Systems besser gewachsen sein, da es eine flexiblere Modellierung von Benutzeräußerungen ermöglicht als eine CFG. Allerdings müsste ein statistisches Modell für die Systeminteraktion noch trainiert werden. Dies könnte durch einen Wizard-of-Oz Test [Fra97] mit anschließender statistischer Auswertung der Benutzeräußerungen erfolgen .

[7] Die VAD wird bei Spracherkennern verwendet, um Sprachschall von Geräuschen die Sprache sehr unähnlich sind zu unterscheiden.

Reihenfolgeneffekte der Versuchsblöcke konnten nicht beobachtet werden. Diese sind ohnehin nicht gewollt, da sie unerwünschte Auswirkungen auf die Bewertung des dritten Blocks haben können. Um evtl. auftretende Effekte zu kompensieren, wurde die Reihenfolge der ersten beiden Aufgaben randomisiert.

Auch das Vorwissen über das unimodale Touch-System hatte keinen signifikanten Einfluss auf die Modalitätennutzung. *Experten* und *Novizen* benutzten Sprache und Touchscreen in ähnlichem Ausmaß. Dies deutet darauf hin, dass *Experten* keine offensichtlichen Vorteile hatten und *Novizen* mit ähnlichen Schwierigkeiten bei der Interaktion umzugehen wussten wie *Experten*.

6.6.2 Weitere Implikationen

Bei den Vortests zeigten sich einige Probleme der GUI. Deren Beseitigung könnte die Usability des Touchscreens weiter steigern. Vor allem die fehlende Raumsuche der GUI wurde von den Probanden während der Versuche stark kritisiert.

Das bei den Vortests bereits aufgedeckte Problem des Dialogmanagements, dass die Mitarbeitersuche per Sprache nur nach einem festen Schema funktioniert, konnte im Vorfeld dieser Evaluation aus zeitlichen Gründen nicht behoben werden. Für eine Weiterentwicklung wäre es jedoch sinnvoll, eine flexiblere Interaktion bei der Mitarbeitersuche zuzulassen. Diese könnte auch berücksichtigen, dass Nutzer oft nur den Nachnamen sagen, wenn der Vorname schon angezeigt wird, und umgekehrt. Um hierfür den Implementierungsaufwand im Dialogmanager zu reduzieren, wäre es am einfachsten die Kommunikation mit der Voice-AP-Map zu erweitern. Die Voice-AP-Map erzeugt eine für die Erweiterung nötige Namensliste, die an den Dialogmanager geschickt werden sollte.

Eine weitere Verbesserung würde die Ein- und Ausschaltbarkeit der Spracherkennung darstellen. Alle auftretenden Schallereignisse werden vom Rauminformationssystem als potentielle Spracheingaben verarbeitet. Unterhält man sich in Reichweite des Mikrofons mit einer anderen Personen, wird der eingehende Sprachschall unweigerlich interpretiert. Dies verursacht scheinbar wahllose Zustandswechsel. Eine interessante Zusatzfunktion wäre, die Spracherkennung nur dann zu aktivieren, wenn ein integrierter Gesichtserkenner ein dem Display zugewandtes Gesicht erkannt hat. Besonders für Demosituationen wäre diese Erweiterung sinnvoll.

An die Idee des Gesichtserkenners anknüpfend, könnte man versuchen, die Leistung

des Systems durch audiovisuelle Spracherkennung zu steigern. Die Informationen des visuellen Kanals könnten wertvolle Zusatzinformation liefern, da das System ohnehin schon die schwierige Aufgabe erfüllen muss, Sprache eines entfernten Sprechers in einer verhallten Umgebung zu erkennen.

7 Zusammenfassung

Gegenstand dieser Arbeit ist die Integration eines Spracherkenners in ein Rauminformationssystem. Dadurch wurde ein multimodales Dialogsystem geschaffen, das dem Benutzer die Bedienung per Touchscreen oder per Sprache ermöglicht. Die Erweiterung um die Spracheingabe soll der Benutzerfreundlichkeit des Systems zuträglich sein. Um festzustellen welchen Einfluss die Erweiterung auf die vom Benutzer empfundene Qualität hat, wurden Tests zur Usability-Evaluation des Systems durchgeführt. Die vorliegende schriftliche Arbeit dokumentiert und diskutiert die dabei gewonnenen Ergebnisse.

Eingangs wurden allgemeine Prinzipien multimodaler Dialogsysteme erläutert. Neben dem Aufbau multimodaler Systeme wurde das Konzept der zustandsbasierten Modellierung vorgestellt. Letzteres ist elementarer Bestandteil des Entwurfskonzepts des weiterentwickelten Rauminformationssystems. Weiterführend erfolgte die Darstellung des aktuellen Forschungsstands zu Evaluationsmethoden multimodaler Systeme. Im Anschluss wurde ein Überblick über die wichtigsten Konzepte moderner Spracherkennungssysteme gegeben. Besonderes Augenmerk wurde dabei auf die Modellierung von Sprachmodellen durch kontextfreie Grammatiken gelenkt, da im Rahmen dieser Arbeit eine solche Grammatik erstellt wurde.

Auf Basis dieser grundlegenden Betrachtungen wurden anschließend die einzelnen Module vor der Integration des Spracherkenners vorgestellt. Dabei wurde das Rauminformationssystem *AP-Map* sowie der *janus*-basierte Spracherkenner eingehend beschrieben. Besondere Aufmerksamkeit galt der Konfigurierbarkeit des Spracherkenners durch den *Dialogmanager*. Weiterhin wurde dargestellt, wie sich die Module Informationen über Mitarbeiternamen und Ressourcen aus einer Datenbank holen.

Nach der Vorstellung der einzelnen Module wurde mit der Beschreibung der Integration des Spracherkenners begonnen. Zunächst wurde das grundlegende Integrationskonzept erläutert. Dazu gehört die Beschreibung des Zustandsautomaten im Dialogmanager, des Synchronisierungsmechanimus zwischen der erweiterten AP-Map und

dem Dialogmanager und der Initialisierung der Schnittstelle. Es wurde ein Protokoll zur Kommunikation zwischen den Modulen entworfen und der Aufbau der unterschiedlichen Nachrichten sowie der Nachrichtenfluss zwischen den Modulen erläutert. Die Implementierung der Schnittstelle und die Integration in den Dialogmanager wurden dokumentiert. Weiterhin wurde auf nötige Erweiterungen seitens der AP-Map hingewiesen.

Im Anschluss wurde die Vorgehensweise bei der durchgeführten Usability-Evaluation beschrieben. Um möglichst viele Usability-Probleme bereits vor den eigentlichen Benutzertests zu erkennen und, falls möglich, noch beseitigen zu können, wurde eine Expertenevaluation durchgeführt. Bei den Benutzertests wurden drei unterschiedliche Implementierungen des Systems miteinander verglichen: (1) ein System mit Sprachsteuerung, (2) ein System mit Touchscreen und (3) ein System mit multimodaler Eingabemöglichkeit. Die Benutzertests wurden dem entsprechend in drei Blöcken abgewickelt, bei denen jeweils die gleichen Aufgaben gelöst werden sollten. Jeder Block wurde durch einen adaptierten modalitätsspezifischen SASSI Fragebogen abgeschlossen. Die Ergebnisse der Expertenevaluation und der Benutzertests wurden ausgewertet und dokumentiert.

Die Experimente, die im Rahmen dieser Arbeit durchgeführt wurden, zeigten unterschiedliche Ergebnisse:

- Vorwissen über das unimodale Touch-System machte sich bei der Qualitätsbewertung nicht als Einflussgröße bemerkbar. Dies kann darin begründet sein, dass der Touchscreen sehr intuitiv bedienbar ist und die Sprachbedienung für Experten und Novizen neu war.
- Abkürzungen per Sprache wurden bei multimodaler Bedienung genutzt, was darauf hindeutet, dass die Sprachbedienung hier als Vorteil empfunden wurde.
- Bei der Qualitätsbewertung schnitt das System mit Touchscreen besser ab als das System mit Sprachsteuerung. Dieses Ergebnis kann vermutlich der Fehleranfälligkeit der Spracherkennung zugeschrieben werden.
- Bei der Qualitätsbewertung schnitt das System mit Touchscreen besser ab als das multimodale System. Auch bei multimodaler Bedienung machen sich Spracherkennungsfehler bemerkbar.
- Die Systemgeschwindigkeit wurde bei längeren Bearbeitungszeiten schlechter

beurteilt. Möglicherweise sind die Bearbeitungszeiten von den Interaktionsproblemen der Benutzer abhängig. Bei einer hohen Anzahl von Interaktionsproblemen könnte auch die Bewertung der Systemgeschwindigkeit schlechter ausfallen.

- Die Systemgeschwindigkeiten des Systems mit Touchscreen und des Systems mit Sprachsteuerung wurden gleich bewertet, obwohl der Sprachblock im Durchschnitt deutlich länger dauerte. Nutzer nehmen die Bearbeitungszeiten unterschiedlicher Modalitäten scheinbar unterschiedlich wahr.

Neben diesen Ergebnissen deckte die Untersuchung einige Möglichkeiten zur Optimierung und Weiterentwicklung des Systems auf:

- Aufgrund vieler Einfügungen und Vertauschungen ist die CER sehr hoch. Ein Grund könnten fehlende akustische Modelle für die korrekte Aussprache von manchen Namen sein. Daher könnte eine entsprechende Erweiterung der akustischen Modelle die Erkennungsleistung steigern. Weiterhin könnten spezielle Garbagemodelle für den in der Umgebung des Rauminformationssystems auftretenden Störschall die CER senken.

- Die hohe OOG-Rate deutet darauf hin, dass die Grammatik dem Anspruch der intuitiven Bedienbarkeit nicht gewachsen ist. Ein statistisches Sprachmodell eignet sich evtl. besser für die stellenweise komplexen Dialogzustände und könnte die Flexibilität der möglichen Sprachbefehle erhöhen. Vor allem bei der Mitarbeitersuche sollte eine flexiblere Sprachsteuerung geschaffen werden.

- Die Möglichkeit die Sprachsteuerung ein- und auszuschalten könnte von manchen Benutzern als Vorteil gewertet werden. Manche Interaktionsprobleme, die durch Spracherkennungsfehler ausgelöst wurden, könnten damit vermieden werden.

- Die Integration audiovisueller Spracherkennung würde sich anbieten, da Kameras in den Panels des Rauminformationssystems schon integriert sind.

- Die GUI sollte um eine Raumsuche erweitert werden, um die Bedienungsmöglichkeiten per Sprache und Touchscreen konsistent zu halten.

A Versuchsmaterial

A.1 Grammatik

```
#JSGF V1.0 UTF-8;
grammar mainGrammar;

public <T2>  = <showFloor> | <showRoom> | <showEvents> | <showSearch> | <directSearch> | <goToMainScreen> | <showDetails>;
public <T1>  = <showFloor> | <showRoom> | <showEvents> | <showSearch> | <directSearch> | <goToMainScreen>;
public <L1>  = <showFloor> | <showRoom> | <showEvents> | <showSearch> | <directSearch>;
public <MS2>  = <navigate> | <showBooking> | <directSearch> | <goToMainScreen>;
public <MS1>= <directSearch> | <goToMainScreen>;
public <E1>  = <goToMainScreen> | <showRoom>;

<showRoom>= (WHERE IS | SHOW | HIGHLIGHT | GO TO) <roomName> [PLEASE];
<directSearch>= (SEARCH | FIND | SHOW [ME] | LOOK FOR) <userName> [PLEASE];
<roomName>= [[THE] ROOM] @roomName;
<showFloor>= (GO | CHANGE | SWITCH) TO <floorName> [PLEASE];
<floorName>= [FLOOR | LEVEL] <floorNumber> | <floorIndex> [FLOOR | LEVEL];

<showEvents>= (DISPLAY | GO TO) ((EVENT [SCREEN] | EVENTS) | SCHEDULE | TABLE | TIMETABLE);
<showSearch>= [SHOW | DISPLAY | SWITCH TO] (SEARCH | EMPLOYEE | EMPLOYEES) [LIST] [PLEASE];
<floorIndex>= TWENTIETH | NINETEENTH | EIGHTEENTH | SEVENTEENTH | SIXTEENTH | FIFTEENTH;
<floorNumber>= TWENTY | NINETEEN | EIGHTEEN | SEVENTEEN | SIXTEEN | FIFTEEN;
<userName>= @firstName =SILENCE= | @lastName | @completeName;
<showDetails>= [SHOW] (DETAIL | DETAILS) | OPEN | ENTER | [MORE] INFORMATION;
<goToMainScreen>= GO TO MAIN SCREEN | CLOSE [SCREEN | [THE] WINDOW] | ESCAPE | EXIT | CANCEL | MAP;
<showBooking>= SHOW [ME] [[THE | THAT | THIS] BOOKING] | OKAY | YES;
<navigate>= <next> | <previous>;
<next>= ((([GO TO | SHOW [ME]] [THE] NEXT ) [BOOKING]) | SKIP [FORWARD] | FORWARD | [GO] RIGHT) [PLEASE];
<previous>= ((([GO TO | SHOW [ME]] [THE] PREVIOUS ) [BOOKING])| ([SKIP] (BACK | BACKWARDS)) |[GO] LEFT) [PLEASE];
```

A.2 Instruktionen

Lieber Teilnehmer,

Vielen Dank, dass Sie sich die Zeit für diesen Test nehmen. Im Folgenden werden Sie gebeten ein multimodales Dialogsystem zu bewerten. Mit Hilfe des Systems ist es möglich Informationen über Räume und Mitarbeiter der T-Labs zu erlangen.
Für den Test bitten wir Sie, verschiedene Aufgaben mit Hilfe dieses Rauminformationssystems zu erledigen. Die Aufgaben sollen insgesamt dreimal bearbeitet werden:

1. nur mit Hilfe des Touchscreens
2. nur mit Hilfe der Sprachsteuerung
3. mit Hilfe des multimodalen Systems (Touchscreen und Sprachsteuerung).

Im letzten Fall können Sie bei jedem Interaktionsschritt frei wählen, ob Sie den Touchscreen oder die Sprachsteuerung zur Bedienung benutzen möchten.

Uns sind Ihr spontanes Verhalten und Ihre Meinung über das System wichtig. Wir wollen nicht Sie testen, sondern unser System. Wenn es Schwierigkeiten mit der Bearbeitung der Aufgaben gibt, ist das kein Fehler Ihrerseits, sondern ein Problem unseres Systems. Sie helfen uns mit diesem Test bei der Weiterentwicklung, daher sind wir dankbar für jeden Fehler, den Sie finden. Alle während des Versuchs aufgenommenen Daten werden vertraulich behandelt und anonym ausgewertet.
Bei Fragen oder Unklarheiten können sie sich jederzeit an die Versuchsleiter wenden.

Bearbeiten Sie bitte alle Aufgaben der Reihe nach. Sie haben mehrere Versuche, um eine Aufgabe zu erfüllen, daher sollten Sie es auf möglichst verschiedene Weise probieren, wenn Sie nicht beim ersten Versuch zum Ziel gelangen.

Nach jedem der drei Durchläufe der Aufgabenbearbeitung bitten wir Sie, einen Fragebogen auszufüllen. Dabei sollen Sie das System bzw. die Interaktion mit dem System, im Hinblick auf die jeweils genutzte Interaktionsmöglichkeit bewerten.
Falls Sie Hinweise und Kommentare haben, die durch die Fragen nicht abgedeckt werden, bitten wir Sie, diese am Ende des Fragebogens festzuhalten.

Für die Sprachsteuerung stehen Ihnen Kommandos in englischer Sprache zur Verfügung. Sie sollten dem System klare Befehle geben, die nur aus wenigen oder einzelnen Wörtern bestehen. Eigennamen können Sie sowohl deutsch, als auch englisch aussprechen.

Damit Sie vor dem Versuch einen Überblick über die grundlegenden Funktionen des Rauminformationssystems erhalten, haben wir auf den nächsten Seiten einige Abbildungen für Sie vorbereitet. Schauen Sie sich diese vor Versuchsbeginn in aller Ruhe an. Auch wenn Sie hierzu Fragen haben, können Sie sich gern an die Versuchsleiter wenden.

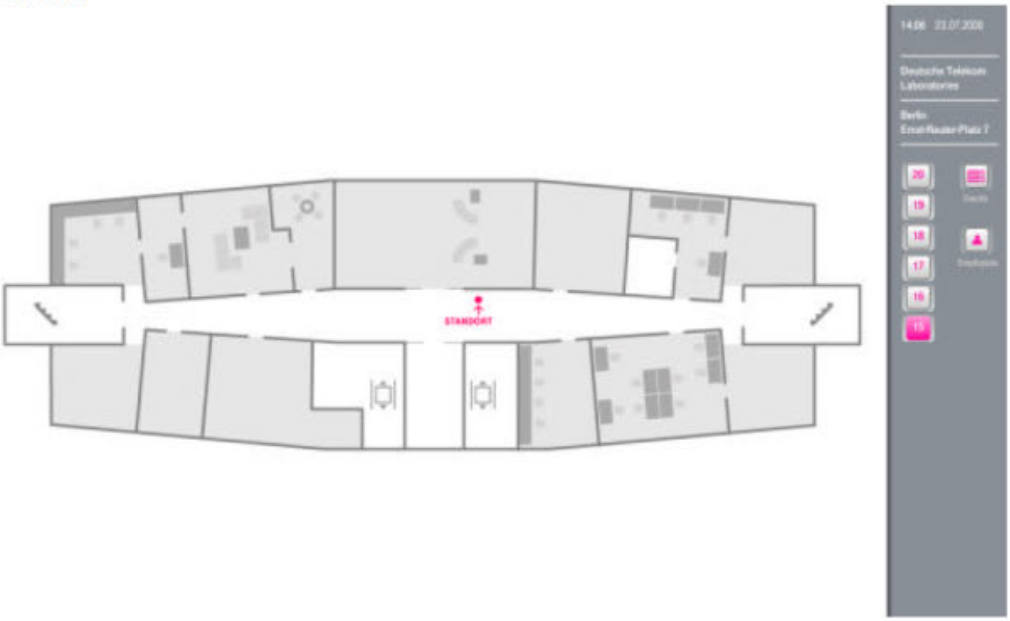
Eventscreen

Deutsche Telekom Laboratories — 11:22, 23.07.2008

Time	Event	Location
00:00-23:59	[illegible]	TEL 17 Flores
00:00-23:59	[illegible]	TEL 18 Kon-Tiki
08:00-18:00	[illegible]	TEL 16 ThinkTank5
08:00-22:00	[illegible]	TEL 16 ThinkTank7
08:00-18:00	[illegible]	TEL 20 Auditorium 2
09:30-11:30	[illegible]	TEL 19 Iceland
10:00-12:00	[illegible]	TEL 18 Spirit
10:00-12:00	[illegible]	TEL 13 Consilium (TEL 1315)
10:00-12:00	[illegible]	TEL 17 TelePresence
10:30-12:00	[illegible]	TEL 17 Sputnik
10:30-12:00	[illegible]	TEL 18 Green Box

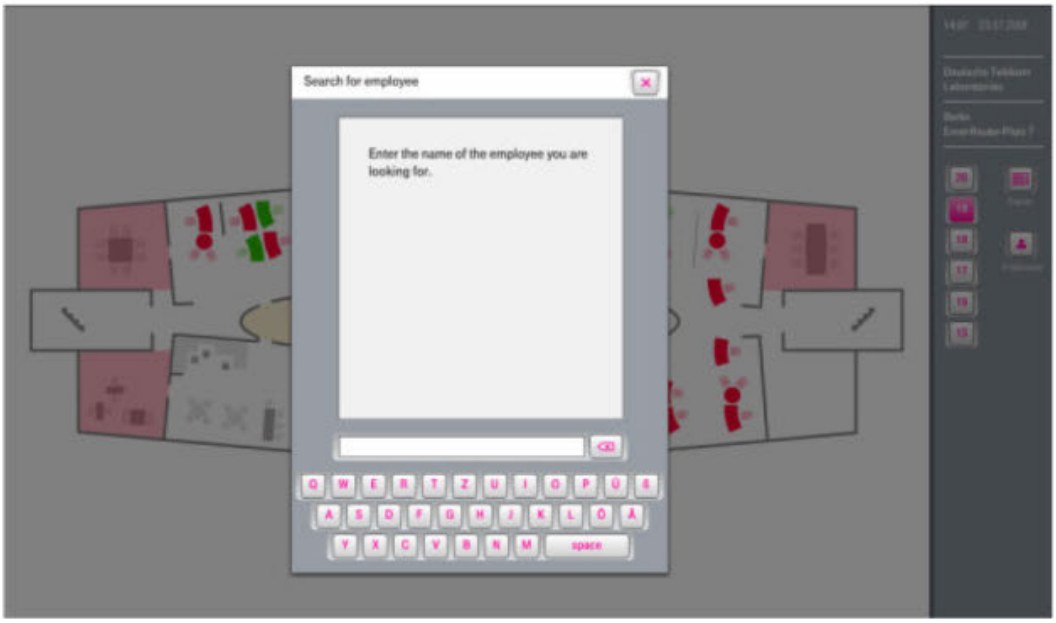
Dies ist der Eventscreen, auf dem alle gebuchten Events der nächsten Stunden aufgelistet sind. Hier können Sie herausfinden, in welchem Raum ein bestimmtes Event stattfindet. Durch Tippen auf den Raum bzw. nennen des Raumnamens gelangen Sie zur Stockwerksanzeige mit dem entsprechenden Raum. Durch Berühren einer beliebigen Stelle des Eventscreens gelangen sie zum Mainscreen, bei dem es sich um die Anzeige der 15. Etage handelt.

Mainscreen



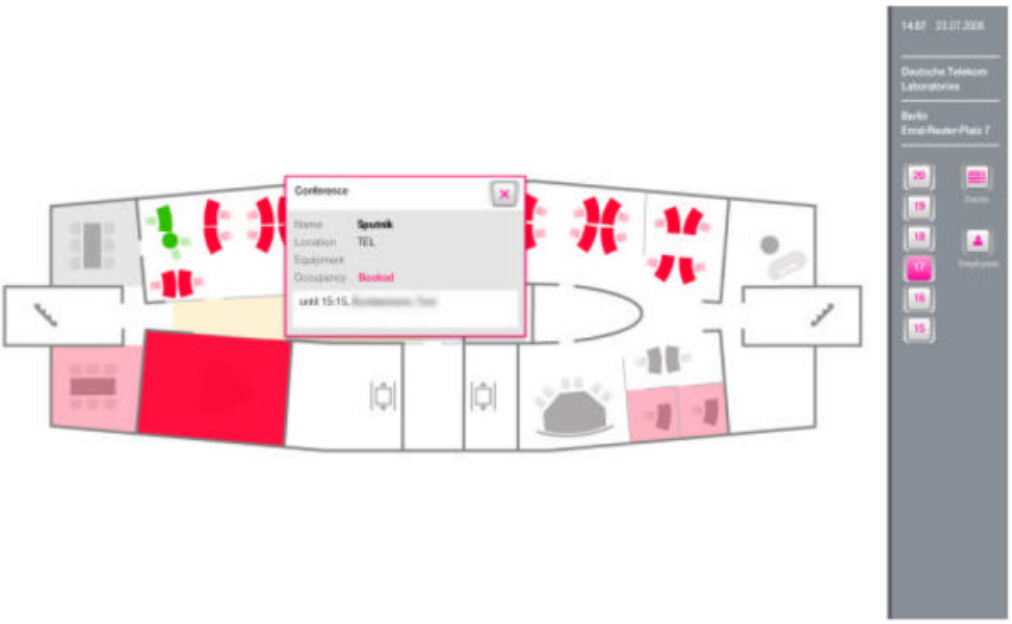
Dies ist der Mainscreen, die Stockwerksanzeige der 15. Etage. Hier können Sie beispielsweise durch Berühren des entsprechenden Buttons bzw. nennen der Zahl zur Anzeige einer anderen Etage gelangen. Der Button „Employees“ bzw. ein entsprechender Sprachbefehl bringt Sie in die Mitarbeitersuche.

Mitarbeitersuche



Dies ist die Mitarbeitersuche, in der Sie die Möglichkeit haben, den Namen des Mitarbeiters, nach dem Sie suchen, per Sprache einzugeben oder einzutippen. Wurde ein Mitarbeiter gefunden, können Sie sich dessen Buchungen anzeigen lassen. Das Suchfenster können Sie beispielsweise mit Hilfe des Sprachbefehls „Close“ oder durch berühren des Kreuzes (oben rechts) schließen.

Raumansicht



Dies ist die Anzeige der Details zu einem bestimmten Raum. Hierher gelangen Sie, indem Sie auf den Raum Ihrer Wahl tippen, bzw. seinen Namen nennen. Im Fenster, das sich darauf hin öffnet (Tooltip), erhalten Sie Informationen über den Buchungsstatus und Details zur einer eventuell vorhandenen Buchung.

Und nun: Viel Spaß beim Versuch!

A.3 Personenbezogener Fragebogen

Allgemeine Angaben

VP-Nr: Datum: Uhrzeit:

Geschlecht: ○ weiblich ○ männlich Alter:

Haben Sie Erfahrung mit Systemen die über Sprachsteuerung bedient werden?

○ ja Wenn Ja, welche?: _____

○ nein _____

Haben Sie Erfahrung mit Systemen die über Bewegungssteuerung bedient werden?

○ ja Wenn Ja, welche?: _____

○ nein _____

Welche Sprachkenntnisse in Englisch besitzen Sie?

keine	geringe	gute	sehr gute	Muttersprache, perfekt
○	○	○	○	○

Beruf _____

(wenn Student, bitte mit Fachrichtung)

A.4 Aufgaben

Aufgaben

1. Veranlassen Sie das System vom Eventscreen zum Mainscreen (Stockwerkanzeige) zu wechseln.

2. Veranlassen Sie das System zur Anzeige der 18. Etage zu wechseln.

3. Sie möchten den Arbeitsplatz Ihrer Kontaktperson, Sebastian Möller, finden.
   Suchen Sie Sebastian Möller mit Hilfe des Systems.
   Wie viele Buchungen hat Sebastian Möller? _____
   Lassen Sie sich Buchung 2 anzeigen.
   Lassen Sie sich den Ort der zweiten Buchung anzeigen.

4. Sie möchten wissen, ob der Raum Sputnik gebucht ist. Suchen Sie den Raum mit Hilfe des Systems.
   Ist der Raum gebucht? _____
   Schließen Sie das Detailfenster, nachdem Sie die gewünschten Informationen abgefragt haben.

5. Veranlassen Sie das System den Eventscreen anzuzeigen.

6. Sie möchten wissen wo sich der Raum zu einem bestimmten Event befindet.
   Veranlassen Sie das System den Raum anzuzeigen.

A.5 SASSI

SASSI Speech

Datum:_____ Uhrzeit:_____ VpNr.:_____

Dieser Fragebogen dient der Erfassung Ihres Urteils zu dem von Ihnen getesteten System. Bitte kreuzen Sie zu jedem der folgenden Sätze an, wie stark Sie mit dieser Aussage übereinstimmen, bzw. sie ablehnen.
Im Anschluss haben Sie die Möglichkeit, zusätzliche Kommentare abzugeben.

Vielen Dank!

1. Das System ist genau.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

2. Die Interaktion mit dem System war unvorhersehbar.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

3. Das System ist verlässlich.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

4. Die Interaktion mit dem System ist konsistent.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

5. Die Interaktion mit dem System ist effizient.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

6. Das System ist nützlich.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

7. Es ist klar wie mit dem System zu sprechen ist.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

8. Ich würde das System benutzen.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

9. Ich hatte das Gefühl die Interaktion mit dem System zu kontrollieren.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

10. Ich fühlte mich sicher beim Nutzen des Systems.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

11. Ich fühlte mich angespannt beim Nutzen des Systems.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

12. Die Interaktion mit dem System ist monoton.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

13. Die Interaktion mit dem System ist langweilig.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

14. Die Interaktion mit dem System ist irritierend.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

15. Die Interaktion mit dem System ist frustrierend.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

16. Ich fragte mich manchmal, ob ich das richtige Wort benutze.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

17. Ich war mir nicht immer sicher, was das System tat.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

18. Es kann schnell passieren, dass man nicht weiß an welcher Stelle man sich in der Interaktion mit dem System befindet.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

19. Die Interaktion mit dem System ist schnell.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

20. Das System tat nicht immer das, was ich wollte.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

21. Das System ist unzuverlässig.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

22. Ich wusste zu jeder Zeit, was ich dem System sagen konnte.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

23. Das System reagierte zu langsam.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

24. Das System reagierte nicht immer wie erwartet.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

25. Das System machte wenig Fehler.

stimme stark zu	stimme zu	unentschieden	lehne ab	lehne stark ab
❍	❍	❍	❍	❍

26. Ich konnte auftretende Fehler leicht beheben.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

27. Ich fühlte mich entspannt beim Nutzen des Systems.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

28. Ich musste mich sehr auf die Interaktion mit dem System konzentrieren.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

29. Ich hatte Spaß bei der Benutzung des Systems.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

30. Das System ist einfach zu bedienen.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

31. Die Benutzung des Systems lässt sich leicht erlernen.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

32. Das System ist zu unflexibel.

lehne stark ab	lehne ab	unentschieden	stimme zu	stimme stark zu
❍	❍	❍	❍	❍

33. Haben Sie weitere Hinweise, Kommentare oder Fragen?

_____

_____

_____

_____

_____

_____

Literaturverzeichnis

[1199] 11, ISO 9241 T.: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs). Part 11: Guidance on Usability. In: *International Organization for Standardization, Genf.* Part 11: (1999)

[BD06] Bortz, Jürgen ; Döring, Nicola: *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler.* Springer Medizin Verlag, 2006

[BEG+96] Boros, M. ; Eckert, W. ; Gallwitz, F. ; Hanrieder, G. ; Niemann, H.: Towards understanding spontaneous speech: Word accuracy vs. concept accuracy. In: *In Proceedings of the Fourth International Conference on Spoken Language Processing (ICSLP 96)*, 1996, S. 1009–1012

[Ber02] *Kapitel* From Theory to Design Support Tool. In: Bernsen, N. O.: *Multimodality in Language and Speech Systems.* Dordrecht, Kluwer Academic Publishers, 2002, S. 93–148

[BG00] Bazzi, Issam ; Glass, James R.: Modeling Out-Of-Vocabulary Words for Robust Speech Recognition. In: *Sixth International Conference on Spoken Language Processing* Bd. 1, 2000, S. 401–404

[BKL+02] Beringer, N ; Kartal, U. ; Louka, K. ; Schiel, F. ; Türk, U: PROMISE: A Procedure for Multimodal Interactive System Evaluation. In: *Proceedings of the Workshop 'Multimodal Resources and Multimodal Systems Evaluation'*, 2002, S. 77–80

[Bol80] Bolt, Richard A.: "Put-that-there": Voice and gesture at the graphics interface. Proceedings of the 7th annual conference on Computer graphics and interactive techniques. In: *International Conference on Computer Graphics and Interactive Techniques*, ACM, 1980, S. 262–270

[BS03] Burtschnik, H.-J. ; Siefkes, D.: *Theoretische Grundlagen der Informatik 1. Skriptum zur Vorlesung am Institut für Softwaretechnik und*

theoretische Informatik der TU Berlin. Theor. Informatik / Formale Spezifikation, 2003

[CAS97] Churcher, Gavin E. ; Atwell, Eric S. ; Souter, Clive: Dialogue management systems: a survey and overview. 1997. – Forschungsbericht

[CCC+96] Cook, G. ; Christie, J. ; Clarkson, P. ; Hochberge, M. ; Logan, B. ; Robinson, A. ; Seymour, C.: Real-Time Recognition of Broadcast Radio Speech. In: *Proc. ICASSP '96*. Atlanta, GA, 1996, 141–144

[Cha94] Charwat, Hans J.: *Lexikon der Mensch-Maschine-Kommunikation*. Oldenbourg, 1994

[Coh97] *Kapitel* Dialogue Modeling. In: Cohen, Phil: *Survey of the State of the Art in Human Language Technology*. Cambridge University Press and Giardini, 1997, S. 204–210

[Cos05] Costabile, Maria F.: *Human computer interaction*. Springer, 2005

[DA05] Delago, Ramón López-Cózar ; Araki, Masahiro: *Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment*. Wiley, 2005. – 54–85 S.

[Dah06] Dahm, Markus: *Grundlagen der Mensch-Computer-Interaktion*. Pearson Studium, 2006

[DHM07] *Kapitel* Spoken Dialogue Systems Evaluation. In: Dybkjær, Laila ; Hemsen, Holmer ; Minker, Wolfgang: *Evaluation of Text and Speech Systems*. Springer, 2007, S. 187–219

[DLW04] Dybkjær L., Bernsen N. O. ; W., Wolfgang M.: Evaluation and usability of multimodal spoken language dialogue systems. In: *Speech Communication* 43 (2004), S. 33–54

[DMS00] Dale, Robert ; Moisl, Hermann ; Somers, Harold ; Dale, Robert (Hrsg.): *Handbook of Natural Language Processing*. Marcel Dekker Inc, 2000. – 602–603 S.

[ED81] Everitt, B.S. ; D.J., Hand: *Finite mixture distributions*. Chapman & Hall, 1981

[Eul06] Euler, Stephen ; Euler, Stephen (Hrsg.): *Grundkurs Spracherkennung*. Vieweg, 2006

[FGH+97] Finke, M. ; Geutner, P. ; Hild, H. ; Kemp, T. ; Ries, K. ; Westphal, M.: The Karlsruhe-Verbmobil Speech Recognition Engine. In: *Proc. ICASSP '97*. Munich, Germany, 1997, 83–86

[Fra97] Fraser, N.: Assessment of Interactive Systems. In: Gibbon, D. (Hrsg.) ; Moore, R. (Hrsg.) ; Winski, R. (Hrsg.): *Handbook on Standards and Resources for Spoken Language Systems*, Mounton de Gruyter, Berlin, 1997, S. 564–615

[Gal99] Gales, M. J. F.: Semi-tied covariance matrices for hidden Markov models. In: *IEEE Transactions on Speech and Audio Processing* 7 (1999), S. 272–281

[GY93] Gilbert, J. ; Yang, W.: A real-time face recognition system using custom VLSI hardware. In: *IEEE Workshop on Computer Architectures for Machine Perception*, 1993, 58–66

[HBK03] *Kapitel* AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In: Hassenzahl, M. ; Burmester, M. ; Koller, F.: *Mensch & Computer 2003. Interaktion in Bewegung*. B.G. Teubner, 2003, S. 187–196

[HPGL00] Hone, Kate S. ; Ph, Ub ; Graham, Robert ; Link, Alencon: Towards a tool for the subjective assessment of speech system interfaces (SASSI). In: *Natural Language Engineering* 6 (2000), S. 287–305

[HU69] Hopcroft, John E. ; Ullman, Jeffrey D.: *Formal languages and their relation to automata*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1969

[Hun00] Hunt, Andrew ; Hunt, Andrew (Hrsg.): *JSpeech Grammar Format*. Sun Microsystems, 2000 `http://www.w3.org/TR/jsgf/`

[IPA05] *International Phonetic Association*. 2005 `http://www2.arts.gla.ac.uk/IPA/IPA_chart_(C)2005.pdf`

[Jek00] Jekosch, Ute: Sprache hören und beurteilen: Ein Ansatz zur Grundlegung der Sprachqualitätsbeurteilung. In: *Habilitation thesis, Universität/Gesamthochschule Essen, D–Essen* (2000)

[Jek04] Jekosch, Ute: Basic Concepts and Terms of "Quality", Reconsidered in the Context of Product-Sound Quality. In: *Acta Acustica united with Acustica* 90 (2004), S. 999–1006

[KC93] Kirakowski, Jurek ; Corbett, Mary: SUMI: the Software Usability Measurement Inventory. In: *British Journal of Educational Technology* 24 (1993), S. 210–212

[Kol08] Kolossa, Dorothea ; Berlin, TU (Hrsg.): *Einführung in die automatische Spracherkennung. Skriptum zur Vorlesung am Institut für Energie- und Automatisierungstechnik*. Fachgebiet Elektronik und medizinische Signalverarbeitung, 2008

[KWL99] Kamm, Ace ; Walker, Marilyn ; Litman, Diane: *Evaluating Spoken Language Systems*. 1999

[Lar03a] Larsen, Lars B.: Assessment of Spoken Dialogue System Usability - What are We really Measuring? In: *EUROSPEECH 2003*, 2003

[Lar03b] Larsen, Lars B.: *On the Usability of Spoken Dialogue Systems*, Aalborg University Denmark, Diss., 2003

[Lar08] Larsson, Staffan: *Dialogue systems and projects*. `http://www.ling.gu.se/~sl/dialogue_links.html`. Version: 08 2008

[LBG+02] Lamel, L. ; Bennacef, S. ; Gauvain, J. L. ; Dartiguesy, H. ; Tememy, J. N.: User Evaluation of the Mask Kiosk. In: *Speech Communication,* 38 (2002), S. 131–139

[Lik32] Likert, Rensis: A Technique for the Measurement of Attitudes. In: *Archives of Psychology* 140 (1932), S. 1–55

[McT98] McTear, Michael F.: Modelling spoken dialogues with state transition diagrams: Experiences with the CSLU toolkit. In: *Proc 5th International Conference on Spoken Language Processing*, 1998, S. 1223–1226

[Met05] Metze, Florian: *Articulatory Features for Conversational Speech Recognition*, Universität Fridericiana zu Karlsruhe (TH), Diss., 2005

[Met07] Metze, F.: *Computergestützte Interaktion. Skriptum zur Vorlesung am Institut für Softwaretechnik und theoretische Informatik der TU Berlin*. Quality and Usability Lab, 2007

[MFP+04] Metze, Florian ; Fügen, Christian ; Pan, Yue ; Schultz, Tanja ; Yu, Hua: The ISL RT-04S Meeting Transcription System. (2004). `http://isl.ira.uka.de/fame`

[Möl07] Möller, S. ; Berlin, TU (Hrsg.): *Sprachkommunikation - Sprachsignalverarbeitung und Sprachtechnologie. Skriptum zur Vorlesung am Institut Softwaretechnik und Theoretische Informatik der TU Berlin*. Quality and Usability Lab, 2007

[Möl08a] Möller, S.: Evaluating Interactions with spoken dialogue telephone services. In: *Recent Trends in Discourse and Dialogue*. L. Dybkjœr and W. Minker, Eds., 2008, S. 69–100

[Möl08b] Möller, S.: *Usability Engineering. Skriptum zur Vorlesung am Institut Softwaretechnik und Theoretische Informatik der TU Berlin der TU Berlin*. Quality and Usability Lab, 2008

[MM76] McGurk, Harry ; MacDonald, John: Hearing lips and seeing voices. In: *Nature* 264 (1976), S. 746 – 748

[MS00] Manning, Christopher D. ; Schütze, Hinrich ; Cambridge, MA: The MIT P. (Hrsg.): *Foundations of statistical natural language processing*. Stanford University and Xerox Palo Alto Research Center, 2000

[Nie93] Nielsen, Jakob: *Usability Engineering*. Kaufmann, 1993

[Nie94] *Kapitel* Heuristic Evaluation. In: Nielsen, Jakob: *Usability Inspection Methods*. Wiley and Sons, 1994, S. 25–62

[NW08] Naumann, A. B. ; Wechsung, I.: Developing Usability Methods for Multimodal Systems: The Use of Subjective and Objective Measures. In: *Proceedings of the International Workshop on Meaningful Measures: Valid Useful User Experience Measurement*, 2008

[OLC05] Oviatt, Sharon ; Lunsford, Rebecca ; Coulston, Rachel: Individual Differences in Multimodal Integration Patterns: What Are They And Why Do They Exist. In: *Proc. CHI'05*, 2005, S. 241–249

[Ovi99] Oviatt, S. L.: Ten myths of multimodal interaction. In: *Communications of the ACM* 42 (1999), S. 74–81

[PHS07] Peissner, M. ; Hipp, C. ; Steimel, B.: *Qualitätskriterien, Maße und Verfahren für Sprachapplikationen*. Initiative VOICE BUSINESS, Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, 2007

[PRS02] Preece, Jenny ; Rogers, Yvonne ; Sharp, Helen: *Interaction Design: Beyond Human-Computer Interaction*. New York, NY, USA : John Wiley & Sons, Inc., 2002

[PS84] Polanyi, R. ; Scha, R.: A syntactic approach to discourse semantics. In: *Proceedings of the 10th International Conference on Computational Linguistics*. California, 1984, S. 413–419. – ACL

[SMFW01] Soltau, H. ; Metze, F. ; Fügen, C. ; Waibel, A.: A One Pass-Decoder Based on Polymorphic Linguistic Context Assignment. In: *ASRU*, 2001

[SSB01] Schiel, Florian ; Steininger, Silke ; Beringer, Nicole: Off-talk - a problem for human-machine interaction. In: *In Proc. European Conf. on Speech Communication and Technology*, 2001, S. 2197–2200

[ST95] Schukat-Talamazzini, E.G.: Automatische Spracherkennung - Grundlagen, statistische Modelle und effiziente Algorithmen. Braunschweig : Vieweg, 1995

[Stu00] *Kapitel* Training und Übung. In: Stumpf, H.: *Determinanten individueller Unterschiede*. M. Amelang, 2000 (Enzyklopädie der Psychologie : Themenbereich C, Theorie und Forschung : Ser. 8, Differentielle Psychologie und Persönlichkeitsforschung ; Bd. 4), S. 487–538

[Stu05] Sturm, Janienke: *On the Usability of Multimodal Interaction for Mobile Access to Information Services*, Radboud University Nijmegen, Diss., 2005

[Tcl08] *Tcl Developer Xchange site*. 2008 `http://www.tcl.tk/`

[TE07] Truckenbrodt, Hubert ; Eichler, Kathrin: *Einführung in die Allgemeine Sprachwissenschaft*. Universität Tübingen und DFKI Berlin, 2007

[TSK03] Totzke, I. ; Schmidt, G. ; Krüger, H.-P.: Mentale Modelle von Menüsystemen - Bedeutung kognitiver Repräsentationen für den Kompetenzerwerb. In: Grandt, M. (Hrsg.): *Entscheidungsunterstützung für die Fahrzeug- und Prozessführung*. Deutsche Gesellschaft für Luft- und Raumfahrt e.V., 2003, S. 133–158

[Ull06] Ullenboom, Christian ; Computing, Galileo (Hrsg.): *Java ist auch eine Insel*. Galileo Computing, 2006

[WLKA97] Walker, Marilyn A. ; Litman, Diane J. ; Kamm, Ace A. ; Abella, Alicia: PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In: *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 1997, S. 271–280

[WRLP94] *Kapitel* The Cognitive Walkthrough Method: A Practioner's Guide. In: Wharton, C. ; Rieman, J. ; Lewis, C. ; Polson, P: *Usability Inspection Methods*. Wiley and Sons, 1994, S. 105–140

[WWL00] Walker, Marilyn ; Wright, Jerry ; Langkilde, Irene: Using Natural Language Processing and Discourse Features to Identify Understanding Errors in a Spoken Dialogue System. In: *Proc. 17th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA, 2000, 1111–1118

[YEG+06] Young, S.J. ; Evermann, G. ; Gales, M.J.F. ; Kershaw, D. ; Moore, G. ; Odell, J.J. ; Ollason, D.G. ; Povey, D. ; Valtchev, V. ; Woodland, P.C.: The HTK Book version 3.4 / Cambridge University Engineering Department, Cambridge, UK. 2006. – Forschungsbericht

Erklärung

Die selbständige und eigenhändige Ausfertigung versichert an Eides statt

Berlin, den 7. Januar 2009

. . . . . .

Unterschrift