Institut für Theoretische Informatik
Fakultät für Informatik
Universität Karlsruhe (TH)

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, USA

Statistische Modellierung des Dialogverlaufes in natürlichsprachlichen Dialogsystemen

Diplomarbeit

von

cand. Inform. Ulf Krum

betreut von
Dipl.-Inform. Hartwig Holzapfel
Dr.-Ing. Thomas Schaaf
Prof. Dr. Alexander Waibel

Oktober 2006

Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 31. Oktober 2006

Ulf Krum

Kurzfassung

Diese Arbeit schlägt ein statistisches Modell zur Abbildung eines kompletten Dialogverlaufes auf einer hohen Abstraktionsebene vor. Das Modell erweitert die Diskursgrenzen einzelner Dialogziele und setzt die Dialogziele in einen Kontext zueindander. Dieser Kontext ermöglicht beispielsweise eine bessere Fehlerbehandlung in hochinteraktiven Systemen, Disambiguierung auf einer höheren Ebene und die Verbesserung der Spracherkennerleistung.

Zunächst wurden Schwächen des Basissystems analysiert und ein „Wizard of Oz"-Experiment durchgeführt, um Daten für den Dialogverlauf in der gewählten Domäne zu sammeln. Mit den Daten wurden Analysen hinsichtlich der möglichen strukturellen Eigenschaften des Modells durchgeführt. Es wurde ein Entwicklungsprozeß definiert, der Schritte zur Gewinnung des Modells und Implementierung eines Dialogsystems festlegt. Abschließend wurde das Modell exemplarisch für die Behandlung von Fehlersituationen in der Interaktion mit einem humanoiden Roboter evaluiert.

Danksagung

Diese Arbeit wurde am Language Technologies Institute der Carnegie Mellon University, Pittsburgh, USA und am Institut für Theoretische Informatik der Universität Karlsruhe durchgeführt. Für die Möglichkeit, in internationalem Umfeld zu forschen, möchte ich mich bei Prof. Dr. Alex Waibel bedanken.

Besonderer Dank geht an meine Betreuer Hartwig Holzapfel und Tomas Schaaf, die mich mit vielen interessanten Diskussionen unterstützten und immer Zeit für Fragen und wertvolle Beiträge fanden. Thomas unterstützte mich außerdem in vielen organisatorischen Belangen während meines Aufenthaltes in Pittsburgh. Dies schließt insbesondere seine tatkräftige Hilfe, die der Einrichtung meines Appartements half, mit ein.

Besonderen Dank geht auch an Tamim Asfour und Pedram Azad. Sie gaben mir die einzigartige Möglichkeit, auf dem humanoiden Robotersystem ARMAR zu arbeiten und unterstützten mich bei der Integration des Dialogsystems und der Durchführung des abschließenden Experimentes maßgeblich.

Bedanken möchte ich mich bei allen Probanden in Pittsburgh und Karlsruhe, die eine Durchführung der Experimente erst ermöglichten. Außerdem geht mein besonderer Dank an Heidrun Weber, Marina Schramm und Susanne Peterson für ihre Unterstützung bei der Korrektur dieser Ausarbeitung.

Schließlich möchte ich mich bei Peter Krum für seine unendliche Geduld und materielle Unterstützung bedanken, sie ermöglichten erst meinen Aufenthalt in Pittsburgh.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Man schätzt, daß sich der Mensch seit ca. 2,5 Millionen Jahren Werkzeuge zunutze macht, um die Bewältigung des Alltages zu erleichtern. Zu Beginn dienten die meisten Werkzeuge dazu, das Überleben zu sichern. Sie waren primitiv, hatten jedes für sich nur eine Funktion, und die Schnittstelle zur Interaktion war vergleichsweise intuitiv. Im Laufe der Zeit wurden die Werkzeuge komplexer und trugen mehr und mehr dazu bei, auch die übrigen Facetten des Alltages zu vereinfachen. Die Komplexität der Werkzeuge nahm mit der Komplexität der Aufgaben zu, die mit ihnen bewältigt wurden. Folgerichtig nahm auch die Komplexität der Schnittstelle zur Bedienung der Werkzeuge zu. Mehr und mehr Werkzeuge, die in den letzten 100 Jahren entwickelt wurden, haben mehr als eine Funktion, mehr als einen Betriebsmodus und können an die Bedürfnisse des Benutzers angepaßt werden. Der Gebrauch des Werkzeuges muß erst erlernt werden, um seinen optimalen Nutzen zu erreichen. Eine komplizierte, klassische Schnittstelle mit Knöpfen, Tastatur oder graphischer Anzeige ist nicht nur wegen der nicht intuitiven Bedienung ein Problem, sie ist in manchem Umfeld auch nicht einsetzbar oder gar ein Sicherheitsrisiko. Man denke dabei an automatische Dienstleistungen, die über das Telefon angeboten werden, oder Informationssysteme für das Auto. Beispielsweise stellt die Bedienung eines Autonavigationssystems mit Tasten und Bildschirm während der Fahrt ein nicht zu unterschätzendes Risiko dar.
Mit der Entwicklung und Verbesserung der automatischen Spracherkennung wird Sprache als Eingabemodalität immer häufiger in einer Mensch-Maschine-Schnittstelle angeboten. In der Praxis werden diese Möglichkeiten aber selten genutzt oder sie sind umständlich zu handhaben. Viele sind kommandoorientiert; die Bedienung muß auch hier erst wieder erlernt werden. Abhilfe können Dialogsysteme schaffen, die es erlauben, in natürlicher, spontaner Sprache mit einem Gerät zu interagieren. Das Erlernen einer solchen Schnittstelle ist nicht notwendig bzw. wurde bereits in den ersten Lebensjahren eines Menschen absolviert. Desweiteren kann mit dem Gerät interagiert werden, ohne die visuelle Aufmerksamkeit von anderen Aktivitäten abwenden zu müssen oder manuelle Tätigkeiten zu unterbrechen. Die Vorteile sind offensichtlich, dennoch sind natürlichsprachliche Dialogsysteme in der Praxis noch nicht weit verbreitet. Ein wesentlicher Grund hierfür ist sicher die enorme Komplexität der menschlichen

Interaktion. Es ist viel kognitive Leistung nötig, um diese Form der Interaktion maschinell zumindest in seinen grundlegenden Facetten zu beherrschen. Die Akzeptanz beim Benutzer spielt eine besonders große Rolle; ist der Markt nicht vorhanden, fehlen die Mittel zur Entwicklung. Neben psychologischen Hürden bei einer Unterhaltung mit einem an sich leblosen Apparat hängt sie vor allem von der Natürlichkeit ab. Wir sind sicher noch sehr weit davon entfernt, philosophische Gespräche mit Maschinen wie „HAL 9000"[1] oder „Marvin"[2] führen zu können. Auch ein autonomes humanoides Robotersystem, das im Haushalt helfen soll, ist bis heute bestenfalls als eine Vision anzusehen. Allerdings bietet es einen gut geeigneten Forschungsrahmen für natürlichsprachliche Schnittstellen. Insbesondere bietet es weitreichenden Spielraum für die Forschung, da hier enger verzahnte Interaktionen möglich sind als mit den sonst in diesem Bereich üblichen Informationssystemen.

1.2 Problematik

Der Einsatz eines natürlichsprachlichen Dialogsystems als Schnittstelle zu einem autonomen humanoiden Robotersystem stellt eine besondere Herausforderung dar, denn der Benutzer kann ihm Aufträge unterschiedlicher Komplexität erteilen. Ein Beispiel: Der Roboter soll die Spülmaschine einräumen. Dieser Auftrag erfordert viele Teilaktionen; er muß umherstehendes Geschirr detektieren, Teile einzeln greifen und an eine passende, freie Stelle in der Spülmaschine stellen. Auf der anderen Seite muß der Roboter auch einfache, dirigierende Aufträge verstehen, wie beispielsweise eine Anzahl Schritte in eine genannte Richtung zu gehen. Jeder Auftrag kann für sich genommen mit den vorhandenen Dialogkonzepten geklärt werden. Für jeden Auftrag existiert ein kontextueller Rahmen, in dem der zugehörige Dialog gesteuert werden kann. Je komplexer der Auftrag, desto mehr Fehlersituationen können auftreten, die mit Hilfe des Benutzers durch einfache, dirigierende Aufträge aufgelöst werden können. Das Problem dabei besteht darin, daß nun der ursprüngliche kontextuelle Rahmen verlassen wird und das Dialogsystem keinen Zusammenhang zwischen dem ursprünglichen Auftrag und dem zur Lösung beitragenden Teilauftrag herstellen kann, da eine übergeordnete Kontextebene fehlt. Der ursprüngliche Auftrag kann infolgedessen nicht fortgeführt werden.
Zur Lösung des Problems könnte ein statistisches Modell beitragen, das den Dialogverlauf auf dieser höheren Ebene abbildet. In diesem Zusammenhang stellen sich die folgenden Fragen:

> Welchen Mehrwert bietet ein statistisches Modell über den Dialogverlauf auf einer höheren Kontextebene?
>
> Wie könnte ein solches statistisches Modell aussehen, welche Konzepte sind involviert?
>
> Wie kann ein konkretes statistisches Modell für ein Dialogsystem erstellt werden?

[1] Der Bordcomputer aus „2001: Odyssee im Weltraum"
[2] Der manisch-depressive Roboter aus „Hitchhikers Guide to the Galaxy"

1.3 Beiträge

Der Gesamtbeitrag dieser Arbeit ist die Entwicklung, Analyse und Evaluation einer statistischen Struktur zur Modellierung des Dialogverlaufes. Sie liegt in der Kontexthierarchie oberhalb der bisher angewandten Auftragsrahmen und stellt diese in einen Sinnzusammenhang. Die Arbeit zeigt durch ein Benutzerexperiment, daß mit Hilfe der Struktur Fehlersituationen aufgelöst werden können, ohne den Kontext zum gegebenen Gesamtauftrag zu verlieren. Weiterhin schlägt sie vor, die Struktur zur Verbesserung der Spracherkennerleistung über die Grenzen der Auftragsrahmen einzusetzen. Bisher war eine Verbesserung nur innerhalb eines Auftragsrahmens möglich. Damit beantwortet sie die Frage nach dem Mehrwert eines statistischen Modells über den Dialogverlauf.
Schließlich definiert sie einen Entwicklungsprozeß für Dialogsysteme. Als wesentlichen Bestandteil des Prozesses schließt das die Entwicklung einer konkreten Dialogstruktur für eine gegebene Domäne mit ein. Damit beantwortet sie die Frage danach, wie ein statistisches Modell über den Dialogverlauf eines Dialogsystems erstellt werden kann.
Zur Beantwortung der Frage nach dem Aufbau der Struktur und den involvierten Konzepten beschreibt die Arbeit die Durchführung verschiedener Analysen. Diese basieren auf einem zuvor durch ein „Wizard of Oz"-Experiment erstellten Datenkorpus.
Neuartig an dem beschriebenen Ansatz ist die Modellierung von kompletten Dialogverläufen von der Aufnahme des Dialoges über die Bearbeitung ein oder mehrere Aufträge bis hin zum Abschluß des Dialoges. Bisher war eine Steuerung nur innerhalb einzelner Teildialoge möglich. Als Basis dienen reale Dialoge aus der Vergangenheit.

1.4 Überblick

In Kapitel 2 werden Grundlagen zu Dialogsystemen, humanoide Roboter und verwendete statistische Modelle beschrieben. Danach werden in Kapitel 3 weitere Arbeiten kurz vorgestellt, die sich allgemein mit statistischen Modellen im Bereich des Forschungsgebietes beschäftigen. Das Dialogsystem bildet die Schnittstelle zu einer Applikation. Kenntnisse über die Applikation und Anwenderszenarien sind wesentlich für das Verständnis dieser Arbeit, daher werden sie gesondert in Kapitel 4 beschrieben. Kapitel 5 beschreibt das Basissystem, auf dem diese Arbeit aufbaut. Das entwickelte statistische Modell wird in Kapitel 6 beschrieben. Außerdem wird ein Entwicklungsprozeß für natürlichsprachliche Dialogsysteme vorgestellt. Es folgen die durchgeführten Experimente und Analysen in Kapitel 7. Schließlich werden die Beträge und Ergebnisse dieser Arbeit zusammen mit einem Ausblick in Kapitel 8 diskutiert. In den Anhängen finden sich Materialien zu den Versuchen sowie Beschreibungen einiger Dialogziele und Beispieldialoge des Systems.

Kapitel 2

Grundlagen

Dieses Kapitel gibt einen Überblick über Dialogsysteme zur Mensch-Maschine-Interaktion. Es werden gängige Applikationstypen sowie der prinzipelle Aufbau eines Dialogsystems vorgestellt. Weiterhin werden ein paar grundlegende Bemerkungen über Humanoide Roboter gemacht und die verwendeten statistischen Modelle beschrieben.

2.1 Mensch-Maschine-Dialog

Ein Dialog (von altgriech. *dialégomai:* sich unterhalten) im klassischen Sinn ist eine Form der Kommunikation, bei der zwei oder mehr Personen im Wechsel Beiträge zu einem Thema leisten. Das Ziel eines Dialoges kann sehr unterschiedlicher Natur sein. In der Literatur wird er oft neben dem Monolog als Stilmittel eingesetzt.
Bei einem Dialog zwischen Mensch und Maschine besteht das Ziel meist in einem Nutzen oder Mehrwert für den Menschen. Der Nutzen wird durch die Funktion der Maschine bzw. die Kooperation zwischen Mensch und Maschine erreicht. In dieser Arbeit wird Dialog wie folgt definiert:

> **Definition 1 (Dialog):**
> Ein Dialog ist eine Interaktion zwischen zwei Agenten, basierend auf sequentiellen Äußerungen.

Diese Definition wird von Wissenschaftlern dieses Gebietes sehr häufig verwendet, so auch beispielsweise von Pietquin (vgl. [Pie04]).
Der natürliche Dialog zwischen Mensch und Maschine dient, analog zu grafischen Benutzeroberflächen, als Schnittstelle zu einer Applikation. Die Applikation hält, in Form ihrer Funktionalitäten, den eigentlichen Nutzen für den Menschen bereit.
Es kann mehrere Gründe für eine natürlichsprachliche oder multimodale Benutzerschnittstelle geben: Sie verspricht intuitiv bedienbar zu sein. Der Benutzer kommuniziert mit der Maschine fast ebenso wie mit einem Menschen und es müssen nicht erst Kommandos erlernt werden, die von der Maschine verstanden werden.

Bei Telefonhotlines steht nur die Sprache als Modalität zur Verfügung. Beim so genannten *call routing* wird mit Hilfe von Dialogsystemen der Grund des Anrufes thematisch identifiziert, um den Anrufenden automatisch mit dem entsprechenden Spezialisten zu verbinden.
Bei Servicelines können Dienste abgerufen werden. UPS beispielsweise ermöglicht so das nachträgliche Ändern der Lieferadresse für ein Paket.
Um einen autonomen, humanoiden Roboter zu steuern wären Maus, Tastatur und Monitor kontraproduktiv. Einen Roboter, der beispielsweise im Haushalt helfen soll, möchte man durch Sprache und Gestik steuern können.

2.1.1 Applikationen und Dialogtypen

Anhand der zugrundeliegenden Applikation und der Komplexität der damit verbundenen Aufgabe können Dialogsysteme unterschiedlich klassifiziert werden. Diese Arbeit folgt grob der Klassifikation von Olivier Pietquin (vgl. [Pie04]), die im folgenden kurz erläutert werden soll.

Formularbasiert

Formularbasierte oder rahmenbasierte Systeme sind die einfachsten Anwendungen, die in der Literatur zu finden sind. Der Benutzer muß eine Reihe von Informationen geben, die in einer Art elektronischem Formular als Felder definiert sind. Sind alle Informationen vorhanden, kann ein Dienst ausgeführt werden, beispielsweise die Abfrage von Gebrauchtwagen in einer Datenbank [GMP+96].
Es kann eine vom System präferierte Reihenfolge definiert werden, in der die Informationen gegeben werden sollen. Das System leitet dann den Benutzer durch den Dialog, indem es für jede Information eine entsprechende Anfrage an den Benutzer stellt. Die Dialogführung ist dennoch so flexibel gestaltet, daß der Benutzer die Möglichkeit hat, eine andere Information zu liefern, ohne daß diese verloren geht. Voraussetzung dafür ist, daß sie im Formular auch vorgesehen ist.
Der Dialog ist meist sehr einfach und hat nicht den Anspruch auf Natürlichkeit.

Informationsgewinnend

Bei dieser Art Systeme geht es um die Abfrage von spezifischen Daten aus einer Datenbank, die dem Benutzer dann präsentiert werden. Die meisten Dialogsysteme, die wir im täglichen Leben nutzen, sind von dieser Bauart.
An den Philips-Forschungslaboratorien in Aachen wurde beispielsweise ein Informationssystem für Zugverbindungen entwickelt (vgl. [AOSS95]). Basierend auf diesem System wurde am gleichen Institut die automatische Telefonvermittlung PADIS realisiert und vorgestellt. Das System kann ein Telefonat an eine genannte Person weiterleiten. Außerdem kann es Auskunft über Telefonnummern, Email-Adresse oder Raumnummer der betreffenden Person erteilen, vgl. [KRS96].
Ähnlich der Zugauskunft, wurde am Language Technologies Institute der Carnegie Mellon University das System „Let's Go!" entwickelt und in der Öffentlichkeit eingesetzt (vgl. [RLBE05]).

Es bietet für Pittsburgh eine Fahrplanauskunft der öffentlichen Verkehrsmittel an, erreichbar unter der Telefonnummer +1 412-268-3526. Zum Zeitpunkt des Entstehens dieser Arbeit war es in Betrieb. Ein wenig Ortskenntnis vorausgesetzt, kann es getestet werden.

Problemlösend

Ein ganz anderes Ziel verfolgen Dialogsysteme, die zur Lösung von Problemen dienen. Der Mensch verfügt über leistungsfähige Sensorik und Haptik. Der Computer kann zuverlässig und schnell relevante Informationen aus einer sehr großen Wissensbasis abrufen. Diese beiden Eigenschaften versucht man geschickt zu kombinieren, um quasi gemeinsam das Problem lösen zu können. Dieses Vorgehen ist wohl das wichtigste Unterscheidungsmerkmal zu anderen Systemen. Außerdem ist der geführte Dialog kontinuierlicher als bei der Abfrage von Daten und die Zusammenarbeit zwischen Mensch und Maschine enger.
Als Beispiel sei hier das „Circuit Fix-it Shoppe"-System von R. Hipp und R. Smith [HS93, BGHS93] genannt. Mit ihm können Fehlerursachen in elektronischen Baugruppen detektiert und behoben werden. Das System besitzt nicht nur eine Wissensbasis über den Aufbau der Baugruppen, sondern pflegt auch ein dynamisches Benutzermodell. Dadurch kann das System entscheiden, welche Informationen der Benutzer zur Fehlerdetektion und -behebung benötigt (vgl. [SHB92]). Bekannte Informationen kann es übergehen. Das steigert die Effizienz bei der Problemlösung und erhöht die Akzeptanz beim Benutzer.

Tutorsysteme

Tutorsysteme sind den Systemen zur Problemlösung ähnlich. In einem kontinuierlichen, auf starke Interaktion ausgelegten Dialog vermitteln sie dem Benutzer Wissen zu bestimmten Themen. Gute Systeme erkennen dabei Schwächen und Lücken des Benutzers, um ihm die fehlenden Informationen geeignet präsentieren zu können. Das von Litman und Silliman an der Carnegie Mellon University entwickelte ITSPOKE ist ein Beispiel für diese Art Dialogsysteme (vgl. [LS04]).

Soziale Konversation

Diese Kategorie vereint Dialogsysteme, die zu reinen Forschungszwecken in der KI oder zur Unterhaltung entwickelt wurden. Sie ermöglichen einen Dialog von Mensch und Maschine zu einem bestimmten Themenbereich.
Der berühmteste Vertreter ist das von Joseph Weizenaum am MIT entwickelte Programm ELIZA (vgl. [Wei66]). Es mimt einen Psychotherapeuten, dem der Benutzer seine Probleme anvertrauen kann. Zum Erschrecken seines Entwicklers unterhielten sich Mitarbeiter des MIT über Stunden ernsthaft mit dem System. ELIZA kann wohl als Urvater moderner Chat-Bots bezeichnet werden.
Im World Wide Web dienen solche Chat-Bots oft dazu, ein Chat-Forum interessanter zu gestalten und Benutzer zu binden.
Alan M. Turing schlug 1950 einen Test vor, mit dessen Hilfe beurteilt werden kann, ob eine Maschine intelligent sei. Kann eine Testperson nicht unterscheiden, ob sich hinter einem Sy-

stem ein menschlicher Operator oder eine künstliche Intelligenz verbirgt, kann das System als intelligent bezeichnet werden (vgl. [A. 50]).

Andere Systeme

Pietquin faßt alle anderen Systeme in dieser Rubrik zusammen. Insbesondere erwähnt er solche, bei denen der Benutzer per Dialog eine Applikation steuert und bedient. Dies kann zum Beispiel notwendig sein, weil er freie Hände und Blick für andere Aufgaben benötigt. Navigationssysteme für Autos sind ein Beispiel hierfür (vgl. [BD01]).
Die klassischen Bedienformen zur Steuerung, wie Maus, Tastatur und Monitor, können aber ganz einfach zu umständlich und der Applikation nicht angemessen sein. Dies ist beispielsweise bei einem kooperierenden, autonomen Robotersystem der Fall, wie es in dieser Arbeit beschrieben wird und das als zugrundeliegende Applikation dient (vgl. Abschnitt 4.1).

2.1.2 Aufbau eines Dialogsystems

Da nun verschiedene Anwendungen und Dialogtypen diskutiert wurden, soll ein Überblick über den Aufbau einer dialoggesteuerten Anwendung und ihrer Komponenten gegeben werden. In dieser Arbeit wird nur die Sprache als Modalität für einen Dialog berücksichtigt. Für einen allgemeinen Überblick sei auf die oben schon genannte Dissertation von Olivier Pietquin [Pie04] verwiesen. Abbildung 2.1 zeigt die einzelnen Komponenten eines natürlichsprachlichen Dialogsystems (Spoken Dialogue System, SDS) und den Datenfluß zwischen ihnen.

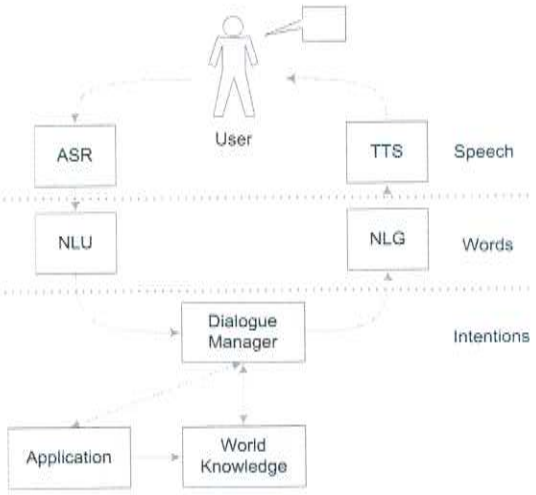

Abbildung 2.1: Übersicht über die Komponenten eines Dialogsystems und ihren Datenfluß

Im einzelnen sind dies der automatische Spracherkenner (Automatic Speech Recognition, ASR), die Sprachverstehenskomponente (Natural Language Understanding, NLU), der Dialogmana-

ger (DM), eine Modellierung des benötigten Weltwissens (World Knowledge, WK), Generator für natürliche Sprache (Natural Language Generator, NLG) und die Sprachsynthese (Text To Speech, TTS).
Die Verarbeitung der Daten findet auf verschiedenen Ebenen statt. ASR und TTS arbeiten auf der reinen Signal- und Sprachebene. NLU und NLG interpretieren auf Wortebene Intentionen. Der DM steuert aufgrund der Intentionen den Dialog.
Schließlich wird die zugrundeliegende Applikation durch den DM gesteuert. Dieser präsentiert wiederum dem Benutzer bestimmte Informationen der Applikation. Die Applikation selbst kann das Weltwissen des Dialogsystems beeinflussen.

Automatische Spracherkennung (ASR)

In der ASR wird das aufgenommene akkustische Signal zunächst digitalisiert. Ein Segmentierer bestimmt den Teil des Signals, der Sprache enthält (Voice Activity Detector, VAD), und gibt diesen zur weiteren Verarbeitung. Es werden Merkmale extrahiert, die hinterher als akkustische Vektoren vorliegen. Durch akkustische Modelle und Sprachmodelle werden den beobachteten Merkmalsfolgen Wörter zugeordnet. In modernen Spracherkennern kommen meist HMMs (Hidden Markov Models) und N-Gram-Modelle zum Einsatz. Statt der N-Gram-Modelle können auch Phrasenstrukturgrammatiken eingesetzt werden. Dies geschieht beispielsweise, um die ASR und NLU bzw. DM enger zu koppeln [FHW04].
Als Ergebnis der Verarbeitung wird eine Wortsequenz oder eine N-Besten-Liste von Wortsequenzen an die Sprachverstehenskomponente weitergegeben. Der Fokus dieser Arbeit liegt nicht auf der Spracherkennung, daher sei hier auf [WL90] und [HH01] für detaillierte Informationen verwiesen.

Sprachverstehenskomponente (NLU)

Die vom Spracherkenner erhaltenen Wordsequenzen werden in drei Schritten verarbeitet, um bedeutungstragende Elemente zu erhalten. Dabei sollte eine NLU-Komponente beachten, daß die Ergebnisse der Spracherkennung fehlerbehaftet sein können.

Syntaktische Analyse Im ersten Schritt, der syntaktischen Analyse, wird die Funktion jedes einzelnen Wortes bestimmt. Außerdem werden die Relationen der Wörter zueinander analysiert, wie sie gruppiert sind und wie sie sich gegenseitig modifizieren. So können beispielsweise Ambiguitäten, die durch Homophone auftreten, aufgelöst werden: Im Englischen kann das Wort *flies* der Plural des Nomens *the fly* oder die dritte Person singular des Verbs *to fly* bedeuten. Durch die Position im Satz kann die Bedeutung des Wortes bestimmt werden: *"The fly flies."* oder *"The flies fly."*

Semantische Analyse Im zweiten Schritt, der semantischen Analyse, wird die kontextunabhängige Bedeutung der Wortfolge bestimmt. In den meisten Fällen wird es sich bei der vom Spracherkenner übergebenen Wortfolge um eine zusammenhängende Äußerung des Benutzers handeln, also einem mehr oder weniger vollständigen Satz. Diesem kann, für sich isoliert, eine Bedeutung zugeordnet werden. Zum Beispiel ist der Satz *"The cup is red"* eine Aussage

über einen Gegenstand. Der Satz *"The red cup"* referenziert einen Gegenstand. Das Ergebnis dieses Schrittes ist eine semantische Repräsentation des Satzes. In dieser Arbeit werden sog. typisierte Merkmalsstrukturen (Typed Feature Structure, TFS) verwendet, die in Abschnitt 5.1 beschrieben werden. Der obige Beispielsatz ist als TFS in Abblidung 2.2 dargestellt.

$$\begin{bmatrix} obj\_cup \\ \text{NAME}\left[\, the\ red\ cup \,\right] \end{bmatrix}$$

Abbildung 2.2: Die semantische Repräsentation des Satzes *"The red cup."* als TFS.

Interpretation im Kontext Im dritten Schritt wird die semantische Repräsentation im Kontext interpretiert. Dazu werden Informationen aus dem Diskurs verwendet. Der Diskurs repräsentiert alle bis zu diesem Zeitpunkt im Dialog gesammelten Informationen. Nach der ersten Benutzeräußerung im Dialog aus Abbildung 2.4 hätte der Diskurs beispielsweise die Form wie sie in Abbildung 2.3 gezeigt wird.

$$\begin{bmatrix} act\_bring \\ \text{OBJ}\left[\, obj\_cup \,\right] \end{bmatrix}$$

Abbildung 2.3: Diskurs nach der ersten Äußerung des Benutzers (vgl. Abbildung 2.4)

Die zweite Äußerung entspricht in ihrer semantischen Repräsentation wieder der aus Abbildung 2.2. Die Interpretation im Kontext besteht hier nun darin, die zusätzliche Information passend in den Diskurs einzufügen, und hat das in Abbildung 2.5 gezeigte Ergebnis.

U1:	"Bring me a cup."
R1:	"Which cup do you want me to bring?"
U2:	"The red cup."
R2:	"Going to bring you the red cup."

Abbildung 2.4: Dialog eines Benutzers (U) mit dem Roboter ARMAR (R).

Natürliche Sprachgenerierung (NLG)

Aufgabe der natürlichen Sprachgenerierung ist es, natürliche Sprache aus informationstragenden Konzepten zu generieren. Soll dies dynamisch erfolgen, kann die NLG-Komponente sehr komplex werden. In den meisten Fällen reicht es aber aus, fest vorgeschriebene Systemanfragen oder -aussagen zu verwenden. Um dennoch eine gewisse Flexibilität zu erreichen, werden Platzhalter eingesetzt. Diese können dann durch die jeweils passenden Werte ersetzt werden. Eine Vorlage *"Going to bring you the <obj_takeable>"* wäre im Beispiel (vgl. Abbildung 2.4) direkt durch den Wert *"the red cup"* aus dem Diskurs ersetzt worden.

$$\begin{bmatrix} act\_bring \\ \text{OBJ} \begin{bmatrix} obj\_cup \\ \text{NAME} [\, the\ red\ cup \,] \end{bmatrix} \end{bmatrix}$$

Abbildung 2.5: Diskurs nach Interpretation der zweiten Äußerung im Kontext (vgl. Abbildung 2.4)

Sprachsynthese (TTS)

Die Sprachsysthese-Komponente wandelt den in Zeichenketten vorliegenden Text in synthetische akkustische Sprach-Signale um. Dies hat den Vorteil gegenüber aufgenommenen Systemäußerungen, daß das Gesamtsystem flexibler in der Entwicklung und Wartung ist. Schließlich müßten Erweiterungen mit demselben Sprecher neu aufgezeichnet werden. Bekanntermaßen sind synthetische Stimmen aber noch weit entfernt von einem natürlichen Klang. Probleme stellt beispielsweise die korrekte Intonation dar. Dies kann wiederum zu Akzeptanzproblemen bei Benutzern führen.

Weltwissen (WK)

Das Weltwissen stellt insbesondere für aufgabenorientierte Dialogsysteme eine wichtige Komponente dar. Der Dialogmanager benötigt Kenntnis seiner Umgebung, um sinnvoll agieren zu können. Die Komponente kann in die Bereiche Aufgabenmodell, Diskursmodell und Umgebungsmodell unterteilt werden.

Aufgabenmodell: Das Aufgabenmodell ist eine formale Repräsentation der Aufgaben, die das System bearbeiten kann. Es wird für die weiterführende Bearbeitung und das Ziehen von Schlußfolgerungen aus Benutzeräußerungen benötigt. Aus den Konzepten, die das NLU-Subsystems liefert, müssen s Zustandsrepräsentationen des Dialogmanagers abgeleitet werden. Intention und Ziele des Benutzers müssen in Konzept-Sequenzen erkannt werden.

Diskursmodell: Der Diskurs ist für das Speichern von Informationen über den Dialogverlauf hinweg verantwortlich, wie es zuvor schon angedeutet wurde. Mit Hilfe des Diskursmodells können Ambiguitäten in einem pragmatischen Kontext interpretiert und meist aufgelöst werden.
Eine andere wichtige Aufgabe des Diskursmodells ist es, den Konsens über die Dialogbeiträge der beteiligten Agenten zu verwalten. In der engl. Literatur wird der Konsens als *common ground* bezeichnet. Angenommen, ein Dialogbeteiligter trägt eine bestimmte Information zum Dialog bei, so muß die Information von allen anderen Beteiligten so interpretiert werden, wie der Beitragende sie gemeint hat. Vor allem aber müssen alle Beteiligten darüber übereinstimmen, die Information auf dieselbe Weise interpretiert zu haben (*grounding criterion*, vgl. [CS89]).

Umgebungsmodell: Das Umgebungsmodell verwaltet von den Aufgaben des Systems unabhängige Informationen, die den Zustand oder das Verhalten des Dialogsystems beeinflussen können.
Beispielsweise haben verschiedene Benutzer einen unterscheidlichen Wissensstand der Aufgaben-Domäne. Das System sollte sich dementsprechend anpassen.

Dialogmanager:

Der Dialogmanager stellt das Gehirn eines Dialogsystems dar. In ihm laufen die Kontrollen über alle Teilsysteme zusammen. Er hält den Diskurs aktuell und entscheidet über die nächsten Schritte. Dies sind meist Anfragen an den Benutzer oder Dienstanforderungen an die zu steuernde Applikation. Dabei ist zu beachten, daß der Dialogmanager nicht auf Wortebene, sondern auf Intentionsebene arbeitet.
Ziel ist es, einen schnellen und effektiven Informationsfluß zwischen Applikation und Benutzer sicherzustellen. Bei einer Mensch-Roboter-Kommunikation kann dies von entscheidender Bedeutung sein, um die Roboteraktionen konsistent zum Dialogzustand zu halten.
Desweiteren ist der DM für die Auswahl der verwendeten Dialogstrategie verantwortlich:
Bei schlechter Leistung des ASR-Subsystems kann es beispielsweise erforderlich sein, von einer impliziten Bestätigungsstrategie (*confirmation strategy*) zu einer expliziten Strategie zu wechseln.
Oder es kann bei einem drohenden Kontextwechsel möglicherweise nicht entschieden werden, ob er durch einen Fehler im ASR oder durch Absicht des Benutzers hervorgerufen wird. In diesem Fall kann der Dialogmanager einen, im Vergleich zum bisherigen Dialog, sehr einfach gehaltenen Klärungsdialog beginnen (vgl. [KHW05]).

Applikation

Die bisher erläuterten Subsysteme bilden das eigentliche Dialogsystem. Ihr Zusammenspiel dient als Schnittstelle zwischen Mensch und Maschine. Der Nutzen der Maschine für den Menschen wird durch ihre Dienste und ihre Funktionalität bestimmt.
Verschiedene Beispiele solcher Applikationen wurden schon in Abschnitt 2.1.1 beschrieben. Die in dieser Arbeit zu steuernde Applikation ist der humanoider Roboter ARMAR und wird in Kapitel 4 ausführlich beschrieben.

2.1.3 Dialogmodellierungen

Die Steuerung des Dialogverlaufes wird maßgeblich durch die verwendeten Dialogmodelle bestimmmt. Teilweise sind sie aus den Anwendungstypen abgeleitet, wie beispielsweise formularfüllende Ansätze (vgl. Abschnitt 2.1.1).
Pietquin [Pie04] und McTear [McT02] geben einen Überblick über bisher verwendete Modelle. Exemplarisch sollen hier drei Modelle kurz erläutert werden.

Endliche Automaten

Der Dialogablauf wird durch einen vordefinierten, endlichen Automaten bestimmt. Jede Benutzereingabe führt zu einem Zustandsübergang. Für jeden Zustand gibt es nur eine bestimmte Menge an Eingabeformulierungen, die in einen neuen Zustand führen. Der Benutzer hat also sehr wenig Spielraum für die Formulierung seiner Eingaben.
Ein oder mehrere Finalzustände repräsentieren die Dienste der zugrundeliegenden Anwendung. Zu jedem Finalzustand führen ein oder mehrere Pfade, entlang derer die benötigten Informationen vom Benutzer angesammelt werden.
Meist erlauben diese Systeme nur einzelne Wörter oder Phrasen als Benutzereingabe, insbesondere nur eine Information innerhalb einer Eingabe. Diese Systeme sind also sehr restriktiv. Die meisten kommerziell verfügbaren Dialogsysteme verwenden diese Form der Dialogkontrollstrategie.
Abbildung 2.6 zeigt einen Dialogablauf mit solch einem System, welches zudem die Eingabe des Benutzers in jedem Zustand verifiziert.

S1:	"What is your destination?"
U1:	"London."
S2:	"Was that London?"
U2:	"Yes."
S3:	"What day do you want to travel?"
U3:	"Friday."
S4:	"Was that Sunday?"
U4:	"No."
S5:	"What day do you want to travel?"
U5:	...

Abbildung 2.6: Beispieldialog mit einem System basierend auf endlichen Automten. Entnommen aus [McT02]

Rahmenbasierte Systeme

Bei einem rahmen- oder vorlagenbasierten (framebased, templatebased) System wird der Benutzer so befragt, daß freie Felder mit Informationen gefüllt werden können. Der Dialogfluß ist in diesem Fall nicht vorherbestimmt, sondern hängt vom Informationsgehalt der jeweiligen Äußerung ab. In Abbildung 2.7 verhält sich das System wie in Abschnitt 2.1.3 beschrieben. Sind mehr Informationen in der Äußerung des Benutzers als verlangt wurden, gehen diese nicht verloren. Und in den nächsten Fragen werden diese nicht explizit angefordert, wie in Abbildung 2.8 dargestellt.

Agentenbasierte Systeme

Agentenbasierte Dialogsysteme sind darauf ausgelegt, komplexe Kommunikation zwischen dem Benutzer und der zugrundeliegenden Anwendung zu ermöglichen, um ein Problem oder

S1:	"What is your destination?"
U1:	"London."
S2:	"Was that London?"
U2:	"Yes."
S3:	"What day do you want to travel?"
U3:	"Friday."

Abbildung 2.7: Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02]

S1:	"What is your destination?"
U1:	"London on Friday around 10 in the morning."
S2:	"I have the following connection ..."

Abbildung 2.8: Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02]

eine Aufgabe zu lösen. In Abbildung 2.9 ist ein beispielhafter Dialog gezeigt. Die Antwort des Systems ist zwar negativ, jedoch werden dem Benutzer Alternativen angeboten.
In [McT02] wird ein System vorgestellt, das dem Benutzer beim Beheben eines Fehlers in einem elektronischen Schaltkreis assistiert. Dabei geht das System auf die Fachkenntnis des Benutzers ein und gibt, falls nötig Hilfestellung.

U1:	"I'm looking for a job in the Calais area. Are there any servers?"
S1:	"No, there aren't any emloyment servers for Calais. However, there is an employment server for Pas-de-Calais and an employment server for Lille. Are you interested in one of these?"

Abbildung 2.9: Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02]

2.2 Humanoide Roboter

Der Begriff *Roboter* hat seinen Ursprung im Slawischen und kann auf das Wort *robota*: Arbeit, Fronarbeit, Zwangsarbeit zurückgeführt werden (vgl. [Wikc]). Die tschechischen Schriftsteller Josef und Karel Čapek haben den Begriff *robota* in der Science-Fiction-Literatur geprägt. In einem Theaterstück von Karel Čapek werden menschnähnliche künstliche Arbeiter in Tanks gezüchtet. Die Vorstellung, sich das Leben durch einen unermüdlichen, freundlichen Helfer einfacher zu gestalten, beflügelt nicht nur Autoren der erzählenden Literatur. Zeichnungen, die ungefähr um 1495 entstanden, belegen, daß schon Leonardo da Vinci eine menschenähnliche Maschine erdachte (vgl. [Wikb]).
Heute sehen viele Wissenschaftler in der Entwicklung humanoider Roboter die Grundlage für die Erschaffung einer anthropomorphen künstlichen Intelligenz. Nach ihrer Auffassung kann

eine solche KI nicht vollständig[1] programmiert werden, sondern resultiert überwiegend aus einem Lernprozeß (vgl. [Wika]). Humanoide Roboter werden ihn ihren Abmessungen, Gewichtsverhältnissen und Bewegungsabläufen dem Menschen möglichst genau nachempfunden (vgl. [MKOB02]). Für den Einsatz eines derartigen Robotersystems hat diese Vorhegehensweise einige Vorteile. Es wird ein universeller Helfer für den Menschen geschaffen, der zum Beispiel schwere oder gefährliche Arbeit erledigen kann. Dabei ist der Roboter an Umgebungen angepaßt, die eigentlich auf den Menschen optimiert sind. Eine Serienfertigung könnte den Roboter ökonomischer als jeden Spezialroboter werden lassen.
Das Forschungsgebiet ist sehr weit gefächert. Es beginnt beim dynamisch stabilen Laufen auf zwei Beinen. Der Roboter muß seine Umgebung warnehmen können und wissen, wo er sich gerade befindet. Kollisionen mit statischen und dynamischen Hindernissen müssen vermieden werden. Der Roboter muß über seinen aktuellen Zustand Bescheid wissen und entscheiden können, welche Aktionen schadensfrei ausführbar sind. Beispielsweise darf er mit einer Tasse in der Hand nicht winken. Natürlich muß der Roboter auch auf natürliche Weise mit dem Menschen kommunizieren können. Das schließt beispielsweise Sprache und Gestik mit ein.
Das wohl prominenteste Beispiel eines humanoiden Roboters dürfte „Asimo" der Firma Honda sein (vgl. [Asi]). Der Roboter des Sonderforschungsbereiches „SFB 588 Humanoide Roboter - Lernende und kooperierende multimodale Roboter" wird in Kapitel 4 detaillierter vorgestellt.

2.3 Statistische Modelle

In dieser Arbeit werden Methoden des Maschinellen Lernens untersucht, mit deren Hilfe der Dialogverlauf vorhergesagt werden soll. Es werden der 1R-Klassifikator, Bayes-Netzwerke und N-Gram-Modelle verglichen. Diese werden in diesem Abschnitt kurz beschrieben. Als Vergleichsbasis dient die Gleichverteilung, die hier nicht separat beschrieben wird[2].

2.3.1 1R

Der 1R-Klassifikator (*1-rule*) ist sehr simpel, erzielt aber in typischen Datenkorpora eine Leistung, die nicht signifikant unter der komplizierter Klassifikatoren steht (vgl. [Rob93]). 1R entspricht praktisch einem Entscheidungsbaum mit einer Ebene. Das Training erfolgt folgendermaßen: Gegeben seien die Attribute $V_1, \ldots, V_n$ und die Klassen $a_1, \ldots, a_m$. Für jedes Attribut V_i wird für jeden möglichen Wert von V_i gezählt, wie oft jede Klasse a_j im Trainingsdatensatz vorkommt. Es wird die am häufigsten auftretende Klasse a_k für diesen Wert ausgewählt. Es wird eine Regel erzeugt, die die Klasse a_k diesem Wert zuordnet. Für jede so entstandene Regel wird die Fehlerrate errechnet und diejenige mit dem geringsten Fehler gewählt.
1R erzeugt eine Wahrscheinlichkeitsverteilung, die für die vorhergesagte Klasse eine Wahrscheinlichkeit von 1,0 vorsieht und 0,0 für alle anderen. Da dies für die Berechnung der Perplexität Probleme mit sich bringt [3], wurde für diese Arbeit eine Variante *Smoothed-1R* verwendet:

[1] Im Gegensatz zum zitierten Artikel der Wikipedia geht der Autor dieser Arbeit davon aus, daß eine Basis der KI immer programmiert werden muß.
[2] Jede Klasse erhält die gleiche Wahrscheinlichkeit, gegeben eine Beobachtung.
[3] Division durch 0!

Die Wahrscheinlichkeitsverteilungen wird mit der Gleichverteilung zu gleichen Gewichten interpoliert. Eine Beschreibung des 1R-Klassifikators ist in [IE05] zu finden.

2.3.2 Bayes-Netzwerk

Ein naiver Bayes-Klassifikator macht die Annahme, daß alle Attributewerte statistisch unabhängig von der zugehörigen Klasse sind. Bayes-Netzwerke sind nicht so restriktiv und stellen, unter diesem Aspekt gesehen, eine Verallgemeinerung des naiven Bayes-Klassifikators dar. Sie repräsentieren die gemeinsame Wahrscheinlichkeitsverteilung über die Attribute. Statistische Abhängigkeiten werden durch einen gerichteten azyklischen Graphen repräsentiert. Die Knoten des Graphen entsprechen den Attributen. Ist ein Attribut V_i statistisch abhängig von einem Attribut V_j, so existiert eine Kante von V_j nach V_i. Außerdem gibt es für jedes Attribut eine Tabelle, die die bedingten Wahrscheinlichkeiten für jeden möglichen Wert des Attributes, gegeben die Werte der direkten Vorgängerattribute. Die gemeinsame Wahrscheinlichkeit für jede beliebige diskrete Belegung $v_1, \ldots, v_n$ der Variablen $V_1, \ldots, V_n$ kann wie folgt berechnet werden:

$$P(v_1, \ldots, v_n) = \prod_{i=1}^{n} P(v_i | \mathit{Parents}(V_i))$$

Für das Training eines Bayes-Netzwerkes ergeben sich nun zwei Probleme: Die Struktur des Graphen muß aus den Trainingsdaten erlernt werden, falls sie nicht vorgegeben ist. Und die Attribute können teilweise im Trainingsdatensatz nicht beobachtet werden. Sind die Struktur bekannt und alle Attribute beobachtbar, erfolgt das Training analog zum Training eines naiven Bayes-Klassifikators. Zum Erlenen der nicht beobachtbaren Variablen kann der EM-Algorithmus eingesetzt werden (vgl. [HGC95]). Das Erlernen der Struktur ist schwierig. Cooper und Herskovits haben den heuristischen K2-Algorithmus vorgestellt (vgl. [CH92])
Bayes-Netzwerke und Algorithmen zum Training werden in [Mit97] beschrieben. In [Was04] werden Graphen und statistische Unabhängigkeit im allgemeinen diskutiert.

2.3.3 N-Gram

N-Gram-Modelle werden oft als statistische Sprachmodelle verwendet. Mit ihnen kann das Vorkommen eines Wortes w_n, gegeben eine Historie $w_1, \ldots, w_{n-1}$ von Wörtern, vorhergesagt werden. Dazu wird die Wahrscheinlichkeit $P(w_n)$ jedes Wortes aus dem zugrundeliegenden Alphabet wie folgt berechnet:

$$P(w_n) = P(w_n | w_1, \ldots, w_{n-1})$$

Das wahrscheinlichste Wort ergibt sich dann aus:

$$\hat{w} = \mathrm{argmax}_W P(w_n | w_1, \ldots, w_{n-1})$$

Ein Klassifikator, der auf N-Grame basiert, wird durch einfaches Zählen der vorkommenden Wörter und Wortfolgen trainiert. Weitere Informationen über statistische Sprachmodelle sind in [JM00] zu finden.

Kapitel 3

Verwandte Arbeiten

Dieses Kapitel gibt einen Überblick über zuvor veröffentlichte und im Kontext dieser Arbeit stehende Beiträge. Die Beiträge lassen sich in drei Bereiche einordnen: automatische Detektion von Diskurselementen, maschinelles Lernen von Dialogstrategien sowie enge Koppelung von Spracherkenner und Dialogmanager.

3.1 Detektion von Sprech-Akten und anderen Diskurselementen

In der Arbeit von Levin et al. (vgl. [LRTGL99]) wird ein mehrschichtiger Klassifikator für das automatische Markieren von Sprechakten und Dialogspielen vorgestellt. Ein Dialogspiel (*dialogue game*) wird wie folgt charakterisiert:

> *"A set of utterances starting with an intention and encompassing all utterances up until the purpose of the game has been either fulfilled (e.g. the requested information has been transferred) or abandoned."*

Es handelt sich dabei um ein Konzept des Diskurses, das zwischen der Ebene der Dialog-Akte und der Ebene der Dialogziele liegt[1]. Sprechakte können durch N-Gram Modelle auf Wortebene detektiert werden. Für die Detektion von Dialogspielen haben Levin et al. im wesentlichen zwei Ansätze untersucht: Zum einen kamen wieder N-Grame auf Wortebene zum Einsatz. Es stellte sich heraus, daß dies nicht möglich ist[2]. Zum anderen wurden N-Gram Modelle auf Sprechakt-Ebene eingesetzt. Dieser Ansatz zeigte eine bessere Leistung. Levin et al. sehen die statistische Erkennung von Dialog-Akten als robuste Alternative zu den traditionellen, auf Grammatiken beruhenden Analysen an. Außerdem verwenden sie die Erkennung von Dialog-Akten in der maschinellen Übersetzung. Die gefundenen Dialog-Akte dienen hierbei als flache zwischensprachliche Repräsentation.
Levin et al. verwenden die untersuchten Modelle zur automatischen Markierung der genannten Diskurselemente in Datenkorpora aus nicht aufgabenorientierten Dialogen. Die Modelle sind also für die Analyse von Dialogen im allgemeinen gedacht. Vorstellbar wäre aber auch der Einsatz in einem aufgabenorientierten Dialogsystem. Die Bestimmung des Dialog-Aktes einer

[1] In dieser Arbeit werden *Dialogakt* und *Sprech-Akt* synonym gebraucht.
[2] Mit Einschränkungen.

Äußerung wird dann mit Hilfe der Modelle durchgeführt, statt mit Phrasenstrukturgrammatiken, wie sie in dieser Arbeit verwendet werden. Außerdem können sie zur Verbesserung der Spracherkennerleistung beitragen in dem, mit der Vorhersage der nächsten Benutzeräußerung, der Suchraum eingeschränkt wird (vgl. Abschnitt 6.4).
Die Arbeit von Wang (vgl. [WW97]) geht in eine ähnliche Richtung; es werden ebenfalls statistische Modelle verwendet, um Segmente in Äußerungen zu detektieren und diese mit einem zugehörigen Sprechakt zu annotieren. Ein Segment kann dabei ein ganzer Satz, ein Halbsatz oder Phrasen sein. Die Annotationen werden zur Reduzierung von Ambiguitäten und zur Verbesserung der Leistung maschineller Übersetzer verwendet.
Im Vergleich zu den Modellen der genannten Arbeiten kommt die, in der vorliegenden Arbeit entwickelte Dialogstruktur auf einer höheren, kontextuellen Ebene zum Einsatz.

3.2 Maschinelles Lernen von Dialogstrategien

Dialogstrategien können auf Klassifikatoren beruhen, die die Auswahl einer Systemantwort oder -aktion zu einem bestimmten Zeitpunkt im Dialogverlauf vornehmen.
Levin und Pieraccini abstrahieren dazu über Dialogsysteme und beschreiben sie mit drei Konzepten (vgl. [LP97]): Zustandsraum, Aktionsmenge und Strategie. Ein Dialogzustand beschreibt das Wissen, das das System zu einem bestimmten Zeitpunkt hat. Alle möglichen Zustände über den Dialogverlauf bilden den Zustandsraum. Die Aktionsmenge beschreibt die Aktionen, die dem System zur Verfügung stehen. Dies können Interaktionen mit dem Benutzer oder externen Wissensquellen sein. Die Strategie bestimmt welche Aktion durch das System als nächstes ausgeführt wird, gegeben einen bestimmten Zustand.
Bei den betrachteten Dialogsystemen handelt es sich um einfache, rahmenbasierte Systeme. Ein Zustand s_t zu einem zeitpunkt t kann als Tupel dargestellt werden:

$$s_t = (k_0, \ldots, k_N)$$

N ist die Anzahl der Informations-Slots k_i, die gefüllt werden müssen. Die Aktionsmenge A besteht dann aus

$$A = \{a_0, \ldots, a_N\}$$

Aktionen, wobei eine Aktion a_i eine zum Slot k_i gehörige Anfrage an den Benutzer darstellt. Ziel eines Dialoges ist es, alle Slots zu füllen, um anschließend den eigentlichen Dienst des Dialogsystems bzw. der zu steuernden Applikation ausführen zu können. Dies wiederum erfordert eine Folge der Aktionen. Sie kann als Markov-Prozeß modelliert werden. Gesucht sind also die Übergangswahrscheinlichkeiten zwischen den Zuständen und damit die Auswahl einer Aktion a_t zum Zeitpunkt t. Die Wahrscheinlichkeiten werden mit Hilfe des sog. Reinforcement Learnings berechnet. Jeder Aktion wird in einem bestimmten Zustand ein bestärkender oder bestrafender Wert zugeordnet. Auf diese Weise können die Kosten für einen bestimmten Dialogverlauf errechnet werden. Eine optimale Dialogstrategie minimiert die Kosten des Dialogverlaufes durch Auswahl der jeweils besten Aktion a_t.
Thomas Prommer legt in seiner Diplomarbeit einen Vorschlag vor, das Reinforcement Learning mit Hilfe eines Simulationsmodells des Benutzers durchzuführen (vgl. [Pro06] und [PHW06]).

Dies ermöglicht eine schnelle, prototypische Entwicklung des Dialogsystems und ermöglicht eine angemessene Anzahl von Trainingsschritten[3]. Es reicht ein initiales „Wizard of Oz"-Experiment aus, um Daten über das Benutzerverhalten zu gewinnen. In mehreren Iterationen können dann Simulation und die Strategie trainiert werden, um die gewünschte Systemleistung zu erhalten. Außerdem kann die Domäne bei Bedarf relativ einfach erweitert werden.

In einer weiteren Arbeit besteht die Dialogstrategie aus einem Neuronalen Netz (vgl. [LGSS06]). Bei dem Dialogsystem handelt es sich um ein Fahrplaninformationssystem, das telefonisch Auskunft erteilt.
Der Dialog wird auf Basis von Dialog-Akten gesteuert. Darin eingeschlossen sind sowohl Dialog-Akte von Benutzer-Turns (U_t) als auch von System-Turns (A_t). Der Dialogverlauf wird als Sequenz von korrespondierenden (A_t, U_t)-Paaren modelliert:

$$(A_1, U_1), \ldots, (A_t, U_t), \ldots, (A_n, U_n)$$

Zu einem Zeitpunkt t hat die Dialogstrategie die Aufgabe, die beste Systemantwort $\hat{A}_t$ zu finden:

$$\hat{A}_t = \text{argmax} P(A_t | (A_1, U_1), \ldots, (A_{t-1}, U_{t-1}))$$

Die Folge der vergangenen Interaktionen $(A_1, U_1), \ldots, (A_{t-1}, U_{t-1})$ wird durch ein Dialogregister repräsentiert. Das Dialogregister ist der in dieser Arbeit verwendeten Interpretation des Diskurses gleichzusetzen, daher ist fortan vom Diskurs D_{t-1} die Rede. Damit kann das Auswahlproblem wie folgt dargestellt werden:

$$\hat{A}_t = \text{argmax} P(A_t | D_{t-1}, (A_{t-1}, U_{t-1}))$$

Es wird also der Diskurs und die letzte Interaktion berücksichtigt. Für das Neuronale Netz dienen die $D_{t-1}, (A_{t-1}, U_{t-1})$ als Eingabe. Als Netz kommt ein mehrschichtiges Perzeptron zum Einsatz. Es wurde mit Hilfe eines annotierten Datenkorpus, der aus einem „Wizard of Oz"-Experiment gewonnen wurde, trainiert.

In den genannten Arbeiten wird jeweils die beste Systemantwort bzw. -aktion aufgrund eines beobachteten Ereignisses – in den meisten Fällen eine Benutzeräußerung – ausgewählt. Es findet jedoch keine vollständige, statistische Modellierung des erwarteten Dialogverlaufes statt, wie es in dieser Arbeit vorgeschlagen wird.
Die Modelle agieren auf der Ebene der Dialog-Akte, Entscheidungen werden innerhalb des kontextuellen Rahmens eines Dialogziels getroffen. Im Gegensatz dazu werden in der vorgelegten Arbeit die Entscheidungen auf dieser Ebene regelbasiert getroffen. Das statistische Modell über den Dialogverlauf erlaubt Entscheidungen in einem Dialogziel übergreifenden Rahmen. Es handelt sich also um eine Mischung von regelbasiertem und statistischem Ansatz für die Dialogsteuerung.

[3] Ca. 10^7 Dialoge.

3.3 Kopplung von Dialogmanager und Spracherkenner

Die Analyse des Dialogverlaufes hinsichtlich einer Folge von Dialog-Akten ermöglicht nicht nur die Auflösung von Ambiguitäten oder Ellipsen auf dieser Kontextebene, sondern ermöglicht auch die Verbesserung der Spracherkennerleistung. Dazu wird der Suchraum eingeschränkt, indem der Dialogmanager dem ASR-Subsystem mitteilt, in welchem Unterraum eine zukünftige Benutzeräußerung erwartungsgemäß liegt. Fordert das Dialogsystem den Benutzer beispielsweise auf, die Farbe eines Gegenstandes zu nennen, der vom Roboter gebracht werden soll, kann der Suchraum auf plausible Antworten eingeschränkt werden.
Zu diesem Zewck verwenden Holzapfel und Fügen (vgl. [FHW04]) für das ASR-Subsystem eine semantische Grammatik anstatt eines statistischen N-Gram Sprachmodells. Dieselbe Grammatik wird auch im NLU-Subsystem des Dialogsystems verwendet. Die Dialogstrategie kann während der Laufzeit des Systems kontextabhängig einzelne Grammatikregeln aktivieren, deaktivieren oder gewichten und dies dem Spracherkenner mitteilen. Die Entscheidung darüber, wie die einzelnen Regeln behandelt werden, ist in der domänenabhängigen Deklaration des Dialogsystems festgeschrieben. Auf diese Weise wird eine Subgrammatik für den aktuellen Kontext gebildet, die den Suchraum letztendlich verkleinert. In der Arbeit wurde eine Verbesserung der Erkennungsleistung sowohl für Nahaufnahmen als auch für Distanzaufnahmen der Sprache experimentell nachgewiesen werden.
In einer weiterführenden Arbeit schlägt Holzapfel einen domänenunabhängigen Ansatz vor (vgl. [Hol06]). Wie in der vorliegenden Arbeit wird der Diskurs in Form einer TFS repräsentiert. Insbesondere sind die benötigten Informationen für ein bestimmtes Dialogziel in ähnlicher Weise deklariert. Anhand des aktuellen Dialogziels und einer noch fehlenden Information ermittelt ein Algorithmus rekursiv alle Elternknoten in der TFS. Aus der Liste dieser Knoten werden dann Grammatikregeln ausgewählt, deren Ableitungen in den geforderten semantischen Kontext konvertiert werden können.
In beiden genannten Arbeiten kann der Suchraum nur innerhalb des kontextuellen Rahmens eines Dialogziels eingeschränkt werden. Es besteht keine Möglichkeit, über die Grenzen dieses Rahmens hinaus Vorhersagen über eine zukünftige Benutzeräußerung zu treffen. Da die in dieser Arbeit vorgeschlagene Dialogstruktur einen kontextuellen Rahmen über die Ebene der Dialogziele legt, sind diese Vorhersagen nun möglich. Im Gegensatz zu den genannten Arbeiten werden sie nicht regelbasiert, sondern auf statistischer Basis getroffen.

Kapitel 4

Applikation und Szenarien

Der Sonderforschungsbereich „SFB 588 Humanoide Roboter - Lernende und kooperierende multimodale Roboter" ist seit 1. Juli 2001 bei der Deutschen Forschungsgemeinschaft (DFG) etabliert. Ziel des SFB 588 ist es, Konzepte und Methoden für ein Robotersystem zu entwickeln, das seinen Arbeitsbereich mit dem Menschen teilt. Ergebnis der Arbeiten wird das Demonstrationssystem ARMAR sein (vgl. [SFB, ARA+06]).
In diesem Kapitel wird kurz auf das Demonstrationssystem eingegangen, da es für diese Arbeit als zu steuernde Applikation dient. Außerdem werden die Szenarien beschrieben, auf deren Grundlage eine Wizard of Oz-Studie und der abschließende Benutzertest durchgeführt wurden. Letzteres ist auch Teil der im SFB 588 definierten Demonstrationsszenarien.

4.1 Der humanoide Roboter ARMAR

ARMAR soll dem Menschen bei seiner täglichen Arbeit als Assistent unterstützen können. Dazu muß sich der Roboter im Arbeitsbereich seines menschlichen Benutzers bewegen und diesen mit ihm teilen. Anders als bei Industrierobotern entfällt hier also die Arbeitszelle, die Maschine und Mensch voneinander trennen. Es müssen nicht nur einige sicherheitsrelevante Maßnahmen unternommen werden, der Roboter muß sich auch an den Arbeitsbereich anpassen. Alle manipulierbaren Objekte, Werkzeuge und Einrichtungen berücksichtigen die Anatomie des Menschen und sind für die Manipulation durch ihn optimiert.
Eine Adaption des Roboters an den Arbeitsbereich des Menschen kann mit einem teilanthropomorphen System erreicht werden. Das bedeutet, ARMAR II und sein Nachfolgemodell ARMAR III sind ihm ähnlich aufgebaut. Als weiteren Vorteil erhofft man sich eine höhere Akzeptanz beim Benutzer, da der Roboter sowohl ein vertrautes Aussehen als auch ein vertrautes Verhalten hat.
Die Gesamtsysteme entsprechen den geometrischen Gegebenheiten einer etwa 1,65 m großen Person. Sie bestehen aus einem Kopf, Rumpf mit zwei Armen und einer fahrbaren Plattform. Fotos sind in Abbildung 4.1 zu sehen. In den folgenden Abschnitten werden die Teile der Roboter mit ihren Funktionen kurz beschrieben.

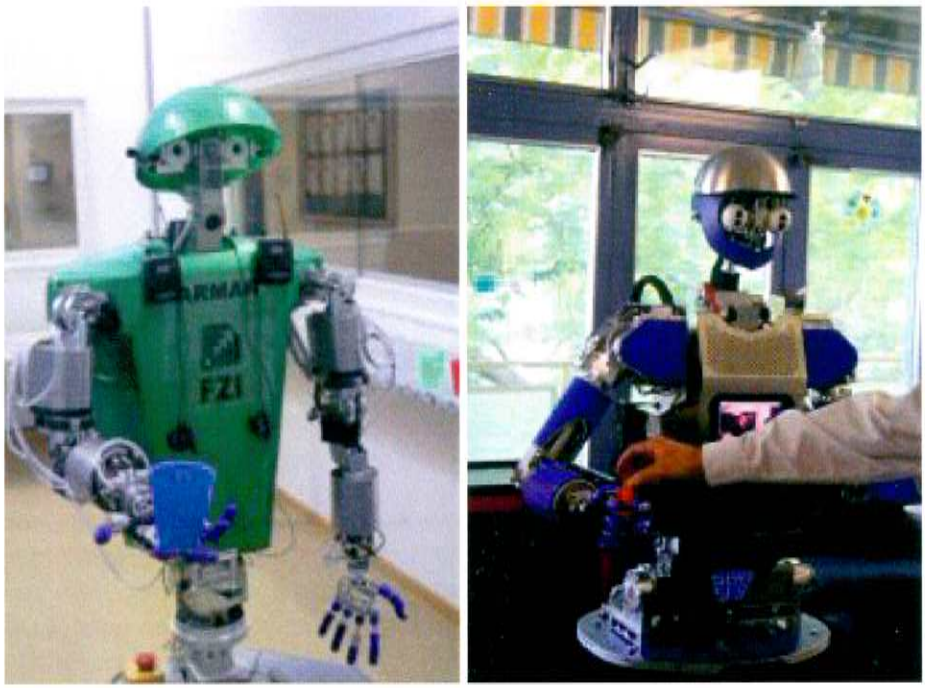

Abbildung 4.1: Der humanoide Roboter ARMAR II (links) und sein Nachfolger ARMAR III (rechts)

4.1.1 Kopf und Hals

Der Kopf ist Träger der kognitiven Sensorik des Roboters. Mit ihrer Hilfe nimmt er seine Umwelt wahr. Zum großen Teil ermöglicht sie die multimodale Steuerung des Systems.
Der Kopf ruht auf einem Halselement mit vier Freiheitsgraden. Sie ermöglichen ein Drehen, Nicken, Kippen und Neigen des Kopfes (vgl. [MKOB02]).
Zwei Mikrophone sind rechts und links außen am Kopf angebracht, die zusammen als Stereomikrophon arbeiten. Bei ARMAR III kommen zusätzliche Mikrophone zum Einsatz. Sie alle ermöglichen die allgemeine akkustische Wahrnehmung der Umwelt.
Eine Stereokamera kommt für die visuelle Wahrnehmung zum Einsatz. Dazu gehören das Erkennen von Objekten sowie die Benutzerdetektion und -identifikation. Bei ARMAR II sind die Kameras starr im Kopf integriert. ARMAR III besitzt zwei Augen, die unabhängig von einander bewegt werden können. In jedem Auge sind zwei Kameras integriert. Damit verfügt der Roboter über ein Stereo-Kamerapaar für Weitwinkel- und eines für Teleaufnahmen.

4.1.2 Rumpf

Der Rumpf ist direkt auf die Plattform montiert und dient als Träger für Kopf und Arme. Er ist neigbar und drehbar. Außerdem ist ein Teleskopelement integriert, mit dem die Höhe angepaßt

werden kann (vgl. [SBD02, ABD02]).

4.1.3 Arme und Hände

ARMAR soll dem Benutzer bei Tätigkeiten als hilfreicher Assistent zur Seite stehen. Daher besitzt er zwei redundante anthropomorphe Hände als Endeffektoren. Diese können durch zwei, ebenfalls redundante Arme positioniert werden und werden bei ARMAR III inzwischen pneumatisch bewegt. Die Arme besitzen jeweils sieben Freiheitsgrade: drei im Schultergelenk, einen im Ellenbogengelenk, zwei im Handgelenk und einen im Unterarm.

4.1.4 Plattform

Die jeweiligen Plattformen von ARMAR II und ARMAR III sorgen für die Mobilität des Roboters auf ebenen Flächen. Dabei kommt für Armar II ein Differentialantrieb zum Einsatz. Dieser ermöglicht Drehungen um die Längsachse des Roboters sowie Vorwärtsbewegungen. Die Plattform für ARMAR III ist mit einem omnidirektionalen Antrieb mit drei Freiheitsgraden ausgestattet. Sie ermöglicht die longitudinale und transversale Fortbewegung sowie Rotationen.
Zur Kollisionsdetektion und -vermeidung werden auf beiden Plattformen Laserscanner eingesetzt. ARMAR II verfügt über einen Scanner, der nach vorne ausgerichtet ist. Für ARMAR III kommen drei kleine Scanner zum Einsatz, die um die Plattform herum angebracht sind.

4.2 Demo-Szenario des SFB 588

Das Szenario „Tasse holen" ist ein im SFB 588 definiertes Demoszenario. Für alle Demoszenarien wurde als Arbeitsumgebung die Küche gewählt. Beherrscht ARMAR erst einmal diesen Bereich, ist ein Transfer in andere Bereiche ein vergleichsweise kleiner Schritt.
In diesem Szenario bekommt das System vom Benutzer den Auftrag, ihm eine bestimmte Tasse zu bringen. Es wurde für die abschließende Benutzerevaluation herangezogen (vgl. Abschnitt 7.5). Der prinzipielle Ablauf und mögliche Fehlersituationen werden in den folgenden Abschnitten erläutert.

4.2.1 Ablauf

Es wird davon ausgegangen, daß der Roboter in der Küche steht und bereit ist, Aufträge entgegenzunehmen. Abbildung 4.2 zeigt einen Zustandsautomaten des Ablaufes.

Erkennen Es kommen zwei Möglichkeiten der „Kontaktaufnahme" zwischen Benutzer und ARMAR in Frage: Entweder wird ein Benutzer über die visuellen Komponenten erkannt, sobald er sich im Sichtbereich des Roboters befindet, oder der Benutzer spricht ihn an.

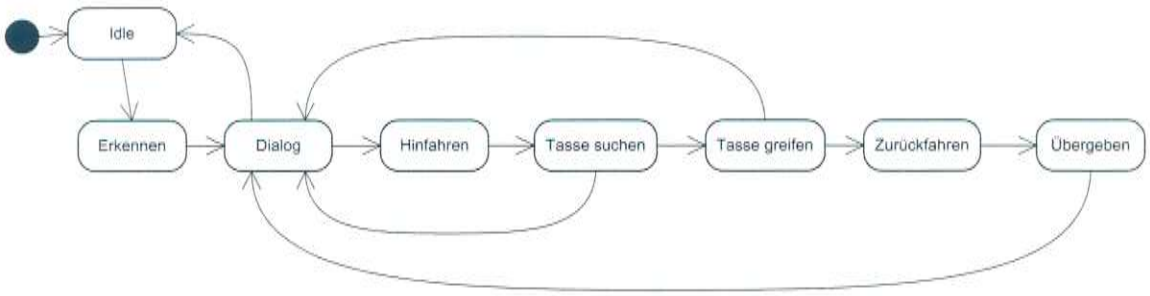

Abbildung 4.2: Zustandsautomat der Ablaufsteuerung für das "Tasse holen"-Szenario.

Dialog In einem initialen Dialog wird der Auftrag des Benutzers an den Roboter geklärt. Prinzipiell kann dieser Zustand zu jeder Zeit erreicht werden. In Abbildung 4.2 sind nur diejenigen Rück-Transitionen eingezeichnet, die durch Fehlersituationen zum erneuten Dialog führen. Durch die Unterstützung des Benutzers kann die Situation aufgelöst werden. Mögliche Fehlersituationen werden in Abschnitt 4.2.2 diskutiert.

Hinfahren ARMAR fährt in die Nähe der Tasse. Die Position des Roboters ist durch die Koordinaten der Tasse vorgegeben. Diese wiederum sind in einer Objektdatenbank gespeichert, die Teil des Weltwissens darstellt (vgl. Abschnitt 2.1.2).

Tasse suchen Der Roboter versucht die Tasse mit Hilfe seiner visuellen Komponenten zu detektieren. Dazu muß sich die Tasse in der Reichweite der Komponenten befinden.

Tasse greifen Es werden Griffart und Handposition berechnet. Die Hand wird in eine Greifposition gefahren. Die Tasse wird gegriffen und angehoben.

Zurückfahren Der Roboter muß den Benutzer detektieren und lokalisieren. Danach wird ein Weg berechnet, der in einer geeigenten Position vor dem Benutzer endet. Schließlich fährt er auf diese Position.

Übergeben Die Tasse wird dem Benutzer übergeben. Dazu muß der tragende Arm eine entsprechende Trajektorie abfahren. Das System wartet, bis der Benutzer die Tasse ergriffen hat. Danach kann es die Tasse loslassen.

4.2.2 Fehlersituationen

Während der Interaktion zwischen Mensch und Maschine können zahlreiche Fehlersituationen auftreten. In dieser Arbeit werden sie in drei Klassen unterteilt:

Systeminterne Fehler treten meist auf mittleren und unteren Kontrollebenen auf. Sie können durch geeignete Maßnahmen gelöst werden. Oder sie führen in einen inkonsistenten Zustand des Systems, sodaß die weitere Ausführung der Applikation nicht sinnvoll ist. Da sie keinen Dialog mit dem Benutzer erfordern, werden sie in dieser Arbeit nicht berücksichtigt.

Systemdetektierte Fehler können nur mit Hilfe des Benutzers gelöst werden. Zu diesem Zweck wird ein (Sub-)Dialog mit dem Benutzer gestartet.

Benutzerdetektierte Fehler können vom System nicht selbst erkannt werden. Der Benutzer muß das System selbständig korrigieren bzw. in dessen Handlungen eingreifen.

Es sind viele verschiedene Fehler in den einzelnen Kategorien denkbar. In den zwei folgenden Abschnitten werden diejenigen diskutiert, die für diese Arbeit und das vorliegende Szenario relevant sind.

Systemdetektierte Fehler

Wird durch die Applikation eine Fehlersituation erkannt, muß der Benutzer darüber informiert werden. Die Applikation leitet den Fehler an den Dialogmanager weiter. Dieser gibt eine einfache Mitteilung an den Benutzer, zum Beispiel: *"I do not see the blue cup"*.

Hindernis Die Sensoren der Plattform haben ein Hindernis detektiert, und es wird kein alternativer Weg gefunden. Dies kann geschehen, wenn die Plattform zu nahe am Hindernis steht oder von Hindernissen umgeben scheint. Befindet sich ARMAR II zu tief in einer Ecke des Raumes, detektiert der frontale Laserscanner in seinem gesamten Sichtbereich ein Hindernis. Die Plattform bewegt sich infolgedessen nicht mehr. Der Benutzer kann um Hilfestellung gebeten werden, um eine drohende Kollision zu vermeiden. Dieser kann den Roboter sicher aus der Gefahrensituation dirigieren.

Tasse nicht detektierbar Aus der angefahrenen Position und der Blickrichtung des Roboters ist die Tasse möglicherweise nicht detektierbar. Sie könnte durch einen Gegenstand verdeckt sein oder sich einfach nicht im Gesichtsfeld des Roboters befinden. Der Benutzer kann hier wiederum Abhilfe schaffen, indem er einfache, die Position betreffende Befehle gibt. Beispielsweise *"Look to the right."* oder *"Move one step forward"*.

Tasse nicht greifbar Die Tasse kann aus zwei Gründen nicht greifbar sein:
Ein Anrücken der Hand in eine Greifposition ist nicht möglich, weil ein oder mehrere Gegenstände im Weg stehen.

Es können weitere Gegenstände auf der Tasse gestapelt sein. Ein Anheben und Wegtragen der Tasse wäre unmöglich. Zumindest müßte evtl. ein Verlust zerbrechlicher Gegenstände durch Herunterfallen in Kauf genommen werden. Hier benötigt der Roboter die Entscheidung des Benutzers, ob eine andere Tasse gebracht werden soll oder die betreffenden Gegenstände aufgeräumt werden.

Benutzer nicht gefunden Um dem Benutzer die Tasse aushändigen zu können, muß seine Position bestimmt werden. Die Position, die der Benutzer während des Dialoges hatte, muß nicht notwendigerweise bekannt gewesen sein. Oder aber der Benutzer kann sie geändert haben, weil er unterdessen andere Tätigkeiten aufgenommen hat.
Wie die Tasse kann auch der Benutzer verdeckt sein, beispielsweise durch eine offene Kühlschranktür. Der Benutzer kann ganz einfach zu weit weg stehen oder der Roboter schaut in die falsche Richtung.

Fehler bei Übergabe Bei der Übergabe ist es wichtig, daß die Tasse jederzeit von mindestens einem Beteiligten gehalten wird. Das System muß also sicher sein, daß sich die Hand des Benutzers fest um die Tasse schließt, bevor es seine Hand öffnet. Ist diese Situation nicht eindeutig visuell identifizierbar, muß sie sich ARMAR verbal bestätigen lassen. So wie es auch bei der Übergabe von Objekten zwischen zwei Menschen geschehen kann.

Benutzerdetektierte Fehler

Die folgenden Fehlersituationen können nur durch den Benutzer erkannt werden. Sie treten vor allem auf pragmatischer Ebene auf und basieren auf fehlerhafter Kongnition des Roboters. Der Benutzer muß in die Handlung des Roboters korrigierend eingreifen oder alternative Aufträge erteilen.

ARMAR verfährt sich Auf dem Weg zur Tasse oder zum Benutzer kann sich der Roboter schlicht verfahren, zum Beispiel weil die Postion nicht korrekt „Wizard of Oz"-Experimentoder eine alternative Route um ein Hindernis herum falsch berechnet wurde.

Falsche Tasse Durch Fehler in der Kognition bringt der Roboter die falsche Tasse oder gar ein ganz anderes Objekt. Dieser Fehler kann schon im Dialog auftreten, wenn beispielsweise das ASR-Subsystem die falsche Farbe erkannt hat. Oder aber er tritt durch falsche Detektion der Tasse auf. In diesem Fall liegt die Ursache in der visuellen Komponente.

Tasse fällt aus der Hand Die Tasse fällt dem Roboter beim Transport aus der Hand. Ursachen hierfür können falsch verstandene Kommandos des Benutzers sein oder einfach Fehlfunktionen in der Handsteuerung.

4.3 „Wizard of Oz"-Szenario

Zu Beginn dieser Arbeit wurde das Basissystem auf Schwächen und Einschränkungen in Bezug auf Mensch-Mensch-Dialoge analysiert. Später werden die gefundenen Eigenschaften und ihre Konsequenzen genauer erläutert (vgl. Abschnitt 5.4). Das Szenario für das „Wizard of Oz"-Experiment sollte die betreffenden Situationen in einem möglichst natürlichen Umfeld untersuchen (vgl. 7.2.1). Die üblicherweise bisher untersuchten Szenarien bieten nicht die Komplexität für das Auftreten der untersuchten Situationen.
Daher wird für das „Wizard of Oz"-Szenario das gerade beschriebenen Demo-Szenario erweitert: Ein zweiter Benutzer wird als dritter Dialogagent zum bisherigen Benutzer und dem Roboter hinzugezogen. Der prinzipielle Ablauf des Szenarios entspricht weiterhin dem SFB-Szenario (vgl. Abschnitt 4.2). Ablaufsteuerung und Fehlersituationen bleiben dadurch erhalten. Durch den zweiten Benutzer sind Benutzerwechsel während der Interaktion mit dem Roboter möglich (vgl. Abschnitt 5.4.2). Außerdem werden zwei Informationsfelder definiert, die über den Rahmen des aktuellen Dialogkontextes hinaus verwendet werden:

- Der Name des Benutzers
- Die Lieblingstasse eines Benutzers

Benutzername

Auf die Anweisung eines Benutzers wird der Dialogmanager feststellen, ob der Benutzer im Weltwissen gespeichert ist. Ist dies nicht der Fall, wird ein Subdialog initiiert, der den Namen ermitteln soll.

Lieblingstasse

Der Benutzer soll die Möglichkeit haben, sich seine Lieblingstasse bringen zu lassen. Dazu muß im Weltwissen einem bestimmten Benutzer eine Tasse zugeordnet sein. Ist dies nicht der Fall, wird ein entsprechender Subdialog initiiert, um die fehlende Information zu ermitteln.

4.4 Stand der Technik

Auf der CeBIT 2006 wurden ARMAR II und ARMAR III im Rahmen des Sonderstandes „Mensch-Technik-Interaktion" des Bundesministeriums für Bildung und Forschung (BMBF) präsentiert (vgl. [RG06]). Der aktuelle Stand der Technik entspricht demjenigen auf der CeBIT und wird im folgenden kurz erläutert.
ARMAR III ist zum Zeitpunkt des Entstehens dieser Arbeit in der Entwicklungsphase. Die Hardware ist zwar nahezu vollständig, aber es sind noch nicht alle Teilsysteme integriert. Eine Steuerung ist nur mit klassischen Mitteln möglich. Es handelt sich dabei um ein Gamepad und eine graphische Benutzeroberfläche auf einem Steuerungsrechner.
ARMAR II kann dagegen multimodal gesteuert werden. Die Möglichkeiten beschränken sich dabei allerdings auf Basisaktionen. Im einzelnen sind dies:

Relative Position anfahren: Die Plattform fährt relativ zur aktuellen Position um die Anzahl der angegebenen Halbmeter und dreht sich anschließen in die gegebene Richtung um 90°.

Drehen der Plattform: Die Plattform dreht sich im Umfang der angegebenen Grade in die angegebene Richtung.

Drehen des Kopfes: Drehen des Kopfes in eine bestimmte Richtung.

Arm-Trajektorie abfahren: Die Arme können einzeln entlang einer zuvor gespeicherten Trajektorie bewegt werden.

Erkennen von Objekten: Bekannte Objekte, die auf Tischhöhe stehen, werden detektiert.

Verfolgen des Benutzers: ARMAR verfolgt den Benutzer mit Kopfbewegungen.

Das bedeutet, ein Auftrag, wie ihn das SFB-Szenario vorsieht, müßte in einzelnen, kleinen Schritten erteilt werden. Der Benutzer müßte also dem Roboter jede Aktion, nach dem Zustand „Dialog" (vgl. Abschnitt 4.2), in Auftrag geben.
Diese Vorgehensweise wäre trivial und wenig von Nutzen. Außerdem wäre eine Evaluation des entwickelten Konzeptes, mit dem daraus entstehende Dialog, nicht möglich. Daher wurde eine Ablaufsteuerung implementiert, die im nächsten Abschnitt erläutert werden soll.
Zur eingeschränkten Funktionalität kommt erschwerend hinzu, daß ARMAR II kein Umgebungsmodell besitzt. Eine Kollisionsdetektion ist nur auf unmittelbare Bodennähe und die Plattform beschränkt. Der Roboter kann also bspw. nicht entscheiden, ob einer seiner Arme an eine Tischplatte stößt.

4.5 Implementierung einer Ablaufsteuerung

Zur Realisierung der Ablaufsteuerung war es zunächst notwendig, Rückmeldungen über den Status einer bestimmte Teilaktion von der Robotersteuerung zu erhalten (vgl. Abbildung 4.2 und Abschnitt 4.4). Für das erfolgreiche Anfahren einer Position, das Drehen der Plattform, das Drehen des Kopfes und das Abfahren einer Arm-Trajektorie wurde jeweils eine einfache Erfolgsmeldung an die Ablaufsteuerung implementiert. Es war leider nicht möglich, Fehlersituationen weiterzuleiten. Nach der Detektion von Objekten wurde eine Liste mit den erkannten Objekten an die Ablaufsteuerung gesendet.
Die Ablaufsteuerung selbst wurde als zentrales Bindeglied zwischen Dialogmanager und Robotersteuerung implementiert. Abbildung 4.3 zeigt die Einbettung der Ablaufsteuerung und die Nachrichten, die zwischen den einzelnen Softwarekomponenten versendet werden.
Alle Dienstanforderungen an die Roboterhardware, die nichts mit dem Auftrag „Tasse bringen" zu tun haben, werden direkt an die Robotersteuerung gesendet. Hat der Dialogmanager den Auftrag „Tasse bringen" vollständig identifiziert, wird die Ablaufsteuerung benachrichtigt (*bring cup*). In einer Fehlersituation benachrichtigt der Dialogmanager die Ablaufsteuerung nach Abschluß eines Korrekturauftrages fortzufahren (*go ahead*). Die Ablaufsteuerung sendet gemäß Aublaufdiagramm den aktuellen Teilbefehl an die Robotersteuerung (vgl. Abbildung

Abbildung 4.3: Einbettung der Ablaufsteuerung

4.2). Diese teilt der Ablaufsteuerung entweder den Erfolg mit oder liefert eine Liste detektierter Objekte. In letzterem Fall überprüft die Ablaufsteuerung, ob die gewünschte Tasse dabei ist, und sendet den nächsten Teilbefehl an die Robotersteuerung oder meldet einen Fehler an den Dialogmanager. Außerdem meldet die Ablaufsteuerung die betreffenden Übergänge von Dialogzielzuständen (vgl. Abschnitte 5.2.1 und 6.1).

Kapitel 5

TAPAS - Werkzeuge für Dialogsysteme

TAPAS wurde auf Basis des Dialogsystems ARIADNE (vgl. [Den02]) entwickelt. Es bietet ein Rahmenwerk und Werkzeuge zur schnellen, prototypischen Entwicklung von Dialogsystemen für den akademischen Gebrauch. Die Dialogsysteme setzen sich aus der domänenunabhängigen Dialogsteuerung und der domänenabhängigen Dialogstrategie zusammen.
In diesem Kapitel werden zunächst die grundlegenden Konzepte des Systems erläutert. Anschließend werden Eigenschaften diskutiert, die zu Einschränkungen im Dialogablauf führen.

5.1 Semantische Repräsentation

Das NLU-Subsystem muß zu Beginn seiner Verarbeitungskette eine textuelle Hypothese in eine semantische Repräsentation überführen (vgl. 2.1.2). Diese Aufgabe wird mit Hilfe eines Parsers und kontextfreier Grammatiken erledigt. Alle weiteren Interpretations- und Verarbeitungsschritte werden auf dieser Repräsentation durchgeführt.
In TAPAS kommen sowohl für die Repräsentation der Eingabesemantik als auch für die des Diskurses typisierte Merkmalstrukturen zum Einsatz. Welche Informationen aus der Hypothese einem bestimmten Element in der TFS entspricht, ist in den Regeln der Grammatiken annotiert. Beispiele einer TFS wurden bereits im Abschnitt 2.1.2 gezeigt (vgl. Abbildungen 2.2, 2.3 und 2.5).
Die Typisierten Merkmalstrukturen wurden von Bob Carpenter entwickelt (vgl. [Car92]). Carpenter definiert eine Typisierte Merkmalstruktur wie folgt:

> **Definition 2 (Typisierte Merkmalstruktur):**
> Eine Typisierte Merkmalstruktur (*Typed Feature Structure*, TFS) über Typ und Merkmal ist ein Tupel $F = (Q, \bar{q}, \theta, \delta)$ mit einer Menge von Knoten Q, dem Wurzelknoten $\bar{q} \in Q$, einer Funktion zur Typisierung der Knoten θ und einer Funktion δ, die einem Merkmal einen Typ zuweist:
>
> $$\begin{aligned} \theta : Q &\rightarrow \mathit{Type} \\ \delta : \mathit{Feature} \times Q &\rightarrow Q \end{aligned}$$

und es gilt:

$$\begin{array}{lcll} \delta(\epsilon, q) & = & q & \text{mit leerem Pfad } \epsilon \\ \delta(f\pi, q) & = & \delta(\pi, \delta(f, q)) & \text{mit } f, q, \pi \in \mathit{Feature} \end{array}$$

Carpenter definiert auf den TFS eine Relation $\sqsubseteq$ (*Subsummation*): Gegeben seien die TFSen $F = (Q, \bar{q}, \theta, \delta)$ und $F' = (Q', \bar{q}', \theta', \delta')$. $F \sqsubseteq F'$ (F subsummiert F'), wenn q und $\bar{q}'$ zueinander äquivalent sind und es äquivalente Knoten $q \in Q$ zu allen $q' \in Q'$ gibt. Mit anderen Worten: F' ist eine Spezialisierung von F.

Carpenter definiert weiterhin die Unifikations-Operation. In TAPAS wird die Unikifation zur Aufnahme von Informationen aus der Eingabesemantik in den Diskurs herangezogen. Die *Unifikation* $(F \sqcup F') = F \sqcup F'$ kann wie folgt erklärt werden: $(F \sqcup F')$ ist eine Spezialisierung zu F und F', in der alle Informationen aus F und F' enthalten sind. Es gilt: $(F \sqcup F') \sqsubseteq F \wedge (F \sqcup F') \sqsubseteq F'$ und $(F \sqcup F')$ ist bezüglich der Subsummation die kleinste obere Schranke zu F und F'. Formale Definitionen für die Subsummation und die Unifikation sind in [Car92] zu finden.

Um den Sachverhalt zu verdeutlichen, werden Eingabesemantiken und Diskursverlauf für den Dialog in Abbildung 5.1 erläutert: Initial ist der Diskurs leer. Zum Zeitpunkt der Äußerung U1

U1:	"Bring me a cup of coffee with sugar."
R1:	"Do you want milk?"
U2:	"Bring me a coffee with milk."
R2:	"Going to make you a cup of coffee with milk and sugar."

Abbildung 5.1: Beispiel-Dialog zur Unifikation von Eingabesemantik und Diskurs

des Benutzers sehen demnach Diskurs und Eingabesemantik wie folgt aus:

$$F_{discourse,0} = \emptyset,\ F_{input,1} = \begin{bmatrix} act\_make \\ \text{OBJ} \begin{bmatrix} obj\_coffee \\ \text{NAME} [\, coffee \,] \end{bmatrix} \\ \text{SUGAR} \begin{bmatrix} prp\_sugar \\ \text{BOOL} [\, true \,] \end{bmatrix} \end{bmatrix}$$

Da $F_{discourse,0}$ leer ist, wird die Eingabesemantik als neuen Diskurs übernommen.
Nach der folgenden Antwort des Benutzers (Äußerung U 2) liegen die folgenden Merkmalstrukturen vor:

$$F_{discourse,1} = \begin{bmatrix} act\_make \\ \text{OBJ} \begin{bmatrix} obj\_coffee \\ \text{NAME} [\, coffee \,] \end{bmatrix} \\ \text{SUGAR} \begin{bmatrix} prp\_sugar \\ \text{BOOL} [\, true \,] \end{bmatrix} \end{bmatrix},\ F_{input,2} = \begin{bmatrix} act\_make \\ \text{OBJ} \begin{bmatrix} obj\_coffee \\ \text{NAME} [\, coffee \,] \end{bmatrix} \\ \text{MILK} \begin{bmatrix} prp\_milk \\ \text{BOOL} [\, true \,] \end{bmatrix} \end{bmatrix}$$

Daraus folgt der neue Diskurs:

$$F_{discourse,1} \sqcup F_{input,2} = \begin{bmatrix} act\_make \\ \text{OBJ} \begin{bmatrix} obj\_coffee \\ \text{NAME} [\, coffee \,] \end{bmatrix} \\ \text{MILK} \begin{bmatrix} prp\_milk \\ \text{BOOL} [\, true \,] \end{bmatrix} \\ \text{SUGAR} \begin{bmatrix} prp\_sugar \\ \text{BOOL} [\, true \,] \end{bmatrix} \end{bmatrix}$$

5.2 Dialogsteuerung

Die Dialogsteuerung stellt den sprach- und domänenunabhängigen Teil eines Dialogsystemes dar, das auf TAPAS basiert. Zentrale Bestandteile der Dialogsteuerung sind das Konzept des Dialogzieles und der Abstrakte Dialogzustand. Mit Hilfe dieser Konzepte kann die Dialogsteuerung in bestimmten Situationen Entscheidungen über den Dialogverlauf treffen. Diese Entscheidungen sind über die Sprache und die Domäne des Dialogsystems abstrahiert und daher allgemein gültig. Ein Beispiel hierfür ist die Holdstrategie (vgl. [HG04]). Das Design des TAPAS-Rahmenwerkes begünstigt einen einfachen Austausch der Dialogsteuerung.

5.2.1 Dialogziel und Dialogzielzustand

Ein Dialogziel definiert einen Rahmen für eine Aktion des Dialogsystems. Alle in diesem Rahmen definierten Informationen müssen vorhanden sein, um die Aktion ausführen zu können. Aktionen betreffen den Dialog an sich oder sind mit Funktionalitäten der zugrundeliegenden Applikation verknüpft.
Ein Dialogziel kann verschiedene Zustände annehmen, je nachdem, welche Informationen im Diskurs bereits vorhanden sind:

deselected: Keine der definierten Informationen ist im Diskurs vorhanden.

selected: Ein Teil der definierten Informationen ist im Diskurs vorhanden.

finalized: Alle benötigten Informationen sind im Diskurs vorhanden.

Der zugehörige Zustandsübergangsgraph ist in Abbildung 5.2 zu sehen.
Mit den definierten Dialogzielen und den Zuständen kann der Dialogzielzustand definiert werden:

> **Definition 3 (Dialogzielzustand):**
> Seien G die Menge aller Dialogziele eines Dialogsystems und S die Menge aller Zustände, die Dialogziele erreichen können. Dann ist das Tupel $d = (g, s)$ mit $g \in G$ und $s \in S$ ein Dialogzielzustand.

Abbildung 5.2: Übergänge der Zustände von Dialogzielen(*deselected* ist Startzustand und *finalized* ist Endzustand).

5.2.2 Abstrakter Dialogzustand

Der Abstrakte Dialogzustand charakterisiert die Zustände des Dialogverlaufes. Beispielsweise ist es durch das sprach- und domänenunabhängige Konzept des Dialogzieles möglich, auch sprach- und domänenunabhängig zu beurteilen, ob alle Informationen für die Erbringung eines bestimmten Dienstes der zugrundeliegenden Applikation im Diskurs vorhanden sind.
Es gibt die unterschiedlichsten, auch orthogonal zueinander liegenden, Aspekte die den Dialogverlauf unabhängig von Sprache und Domäne charakterisieren können. Daher wird der Abstrakte Dialogzustand (*abstract dialogue state*, (ADS)) als Tupel aus verschiedenen Zustandsvariablen definiert. Das Tupel ist dabei um beliebige Variablen erweiterbar, je nachdem, welche Aspekte durch die Dialogsteuerung berücksichtigt werden sollen. Ein konkreter Dialogzustand wird durch die Belegung der Variablen definiert. Die vier folgenden Variablen sind die wichtigsten :

INTENTION Gibt den Fortschritt des Dialoges bzgl. eines Dialogziels an.

SPEECHACT Der Sprechakt bzw. Dialog-Akt

SELECTEDGOALS Die Menge G_s der Dialogziele, die sich im Zustand *selected* befinden (vgl. 5.2.1).

FINALIZEDGOALS Die Menge G_f der Dialogziele, die sich im Zustand *finalized* befinden (vgl. 5.2.1).

Die Variablen SELECTEDGOALS und FINALIZEDGOALS sind soweit selbsterklärend.

Die ADS-Variable SPEECHACT

Der Sprechakt charakterisiert die aktuelle Äußerung des Benutzers. Die Begriffe „Sprechakt" und „Dialog-Akt" werden in dieser Arbeit synonym verwendet, daher wird im folgenden nur noch vom „Dialog-Akt" die Rede sein. Der Dialog-Akt findet sich aus dem Wurzel-Knoten der Eingabesemantik ab. Er entspricht einem Nichtterminal der Grammatik. Die Sprechakte können in eine Taxonomie eingebettet werden (vgl. [Den02]) und werden in der Ontologie der Applikation deklariert. Im Unterschied zu linguistisch motivierten Taxonomien soll die hier verwendete Taxonomie beschreiben, welche Aktion das Dialogsystem am besten ausführen soll.

Die ADS-Variable INTENTION

Die Variable INTENTION ist aus den Variablen SELECTEDGOALS und FINALIZEDGOALS abgeleitet. Sie kann die Werte *selected, determined, finalized und deselected* annehmen und wird nach folgender Vorschrift belegt:

$$\text{INTENTION} = \begin{cases} \textit{selected} & : \ |G_s| > 1 \wedge G_f = \emptyset \\ \textit{determined} & : \ |G_s| = 1 \wedge G_f = \emptyset \\ \textit{finalized} & : \ |G_f| = 1, g = \textit{finalized}, g \in G_f \\ \textit{deselected} & : \ G_s = \emptyset \wedge G_f = \emptyset \end{cases}$$

Sie stellt ein Maß dafür da, wie weit der Dialog von der Finalisierung eines Dialogzieles entfernt ist. Abbildung 5.3 zeigt mögliche Zustandsübergänge für INTENTION .

Abbildung 5.3: Übergänge der ADS-Variable INTENTION (*selected* ist Startzustand, *deselected* und *finalized* sind Endzustände).

5.3 Dialogstrategie

In der Dialogstrategie werden die sprach- und domänenabhängigen Konzepte deklariert. Insbesondere wird hier festgelegt, welche Anfragen und Antworten das System auf bestimmte Benutzeräußerungen stellt bzw. gibt. Die Systemäußerungen werden als Vorlagen in den unterstützten Sprachen deklariert. Dadurch entfällt eine kompliziertes NLG-Subsystem (vgl. Abschnitt 2.1.2).
Eine Ontologie beschreibt nicht nur die oben genannten Dialog-Akte, sondern beschreibt im Zusammenspiel mit einer Objekt-Datenbank das Weltwissen des Dialogsystems.
Die deklarierten Dialogziele (*goals*) beschreiben die Funktionalität der Applikation und definieren in einer Vorbedingung, welche Informationen im Diskurs gespeichert sein müssen, damit sie ausgeführt werden können. Dazu wird die Typ-Struktur der TFS vorgegeben. Weiterhin werden die betreffenden Aufrufe der Funktionalitäten und bestätigende Systemantworten deklariert.
Die eigentliche Strategie wird durch die deklarierten Dialogschritte (*moves*) festgelegt. Ein Dialogschritt setzt sich – analog einem Dialogziel – aus Vorbedingungen und Aktionen zusammen. In die Vorbedingungen fließen insbesondere Zustände des Abstrakten Dialogzustandes ein. Genauer gesagt, werden Werte für die einzelnen Variablen festgelegt. Aktionen bestehen dann beispielsweise aus dem Erfragen fehlender Informationen, der Aktivierung von Regeln

für die Interpretation der Benutzeräußerung im Kontext oder der Gewichtung von Grammatikregeln. Zur Erläuterung wird das Kaffee-Beispiel aus Abschnitt 5.1 etwas in der zweiten Benutzeräußerung abgewandelt (vgl. Abbildung 5.4): Der Benutzer antwortet nun mit *"Yes."*, danach liegen zunächst die TFS $F_{input,2}$ für als Eingabesemantik und $F_{discourse,1}$ als Diskurs vor.

U1:	"Bring me a cup of coffee with sugar."
R1:	"Do you want milk?"
U2:	"Yes."
R2:	"Going to make you a cup of coffee with milk and sugar."

Abbildung 5.4: Beispiel-Dialog zur Kontext-Transformation

$$F_{discourse,1} = \begin{bmatrix} act\_make \\ \text{OBJ}\begin{bmatrix} obj\_coffee \\ \text{NAME}[\,coffee\,] \end{bmatrix} \\ \text{SUGAR}\begin{bmatrix} prp\_sugar \\ \text{BOOL}[\,true\,] \end{bmatrix} \end{bmatrix}, \quad F_{input,2} = \begin{bmatrix} yes \end{bmatrix}$$

$F_{input,2}$ und $F_{discourse,1}$ sind nicht unifizierbar, weil die Wurzelknoten nicht übereinstimmen. $F_{input,2}$ muß also erst transformiert werden:

$$F_{input,2} \longrightarrow F'_{input,2} = \begin{bmatrix} act\_make \\ \text{MILK}\begin{bmatrix} prp\_milk \\ \text{BOOL}[\,true\,] \end{bmatrix} \end{bmatrix}$$

Die entsprechende Transformationsregel wurde zusammen mit der Systemäußerung R 1 in einem Dialogschritt deklariert, der genau dann ausgeführt wird, wenn das Dialogziel *MakeDrink* selektiert ist, aber die Information über den Zucker fehlt.

5.4 Grenzen des Basissystems

Nachdem nun das Basissystem kurz beschrieben wurde, werden die Grenzen der verwendeten Konzepte erörtert. Diese Grenzen und die daraus resultierenden Konsequenzen motivieren die danach folgenden Überlegungen zu einer übergeordneten Dialogstruktur.

5.4.1 Eigenschaften

E1 Der Diskurs des Basissystems ist auf ein Dialogziel beschränkt.

Die Sammlung von Informationen im Diskurs ist darauf ausgerichtet, genau ein Dialogziel bestimmen zu können und auszuführen. Das Dialogziel stellt einen Rahmen dar, innerhalb dessen die Informationen in einen Sinnzusammenhang gebracht werden können. Informationen, die über den Rahmen hinausgehen, sind nicht zuordenbar bzw. führen zu einem neuen Diskurs.

E2 Zwei verschiedene Diskurse sind unabhängig voneinander.

Informationen, die in zwei verschiedenen Diskursen gesammelt wurden, können nicht in Beziehung zueinander gebracht werden. Die jeweils finalisierten Dialogziele und deren zugeordenter Dienste sind ebenfalls unabhängig von einander.

E3 Das Ergebnis der Ausführung eines Dienstes ist nicht bekannt.

Nach dem Finalisieren eines Dialogziels wird der zu steuernden Applikation eine Nachricht gesendet, den zum Ziel gehörigen Dienst auszuführen. Auf der Seite des Dialogmanagers ist das betreffende Dialogziel abgeschlossen. Es gibt keine Rückmeldung der Applikation, ob der Dienst erfolgreich ausgeführt wurde.

5.4.2 Konsequenzen

Die oben beschriebenen Eigenschaften führen zu einem weniger natürlichen Dialog und münden möglicherweise in einem Fehlverhalten der Applikation. Es bedarf mehr Äußerungen, bis der Benutzer an sein Ziel gelangt. Eventuell müssen ganze Dialogabschnitte wiederholt werden, um ein Dialogziel erneut zu finalisieren.

Neustart

Angenommen, ein Dialogziel Z mit den Argumenten bzw. Informationen, $A = \{a_1, \ldots, a_n\}$ wurde finalisiert. Der zugehörige Dienst wird gerade ausgeführt, ist aber noch nicht beendet. Eine Äußerung, die zum selben Dialogziel Z mit denselben Argumenten A führt, wird als neue Anforderung des Dienstes interpretiert. Verantwortlich dafür ist die Eigenschaft E3. Ein Beispiel ist in Abbildung 5.5 gegeben. Ein erneutes Ausführen des Dienstes führt zu einer Fehlersituation, da der Roboter die Tasse bereits in der Hand hält. Durch die in dieser Arbeit

U 1:	"Please bring me the cup."
R 1:	"Which cup do you want me to bring?"
U 2:	"The green cup." *(Roboter fährt zur Position an der die Tasse stehen soll.)*
R 2:	"I do not see the green cup."
U 3:	"Move one step to the right, Robbi." *(Roboter kann nun die Tasse detektieren und ergreift sie. Er bleibt dann aber stehen.)*
U 4:	"Robbi, please bring me the cup."
R 4:	"Which cup do you want me to bring?"

Abbildung 5.5: Beispiel-Dialog für einen Neustart

vorgeschlagene Erweiterung der Dialogzielzustände um den Zustand *executed* kann der Dialogmanager entscheiden, ob der Benutzer den Dienst erneut anfordert.

Korrektur

Zu jeder Zeit kann es der Benutzer für notwendig erachten, seinen Auftrag an die zu steuernde Applikation zu ändern. Denkbar sind Stornierung oder Korrektur einzelner Argumente. Ein Beispiel ist in Abbildung 5.6 gegeben. Der Benutzer entscheidet sich für die grüne Tasse anstelle der blauen. Der Dialogmanager interpretiert die Äußerung wegen Eigenschaft E1 und E3 als neuen Auftrag. Der Roboter wird womöglich zuerst die blaue und dann die grüne Tasse bringen.

U 1:	Please, bring me the blue cup.
	(Der Roboter fährt los, um die blaue Tasse zu holen.)
U 2:	Well, take the green one.
R 2:	Which one do you want me to take?

Abbildung 5.6: Beispiel-Dialog für eine Korrektur

Kooperation

Eine Kooperation kann es erfordern, dem zu steuernden System einzelne Teilschritte in Auftrag zu geben. Wegen Eigenschaft E2 ist dies nicht möglich. Der Beispieldialog in Abbildung 5.7 verdeutlicht den Sachverhalt. Die Referenz auf die Tasse in der zweiten Äußerung kann nicht aufgelöst werden, da sie auf ein Objekt im vorherigen Diskurs weist.

U 1:	Take the blue cup.
	(Der Roboter ergreift die blaue Tasse.)
U 2:	Put it into the cabinet.
R 2:	Which object should I take?

Abbildung 5.7: Beispiel-Dialog für eine Kooperation

Ambige Intention

Mit der Äußerung eines Benutzers können Ambiguitäten auftreten, die durch die bisherigen Konzepte des NLU-Subsystems und des Dialogmanagers nicht aufgelöst werden können. Wegen E1 und E2 ist der Kontext, in dem jeweils eine Ambiguität aufgelöst werden kann, auf den Rahmen eines Dialogziels beschränkt. Es können also insbesondere dann Ambiguitäten auftreten, wenn ein Übergang von der Behandlung eines Dialogziels zur Behandlung eines anderen Ziels stattfindet. In Abschnitt 7.2.4 wird ein Beispiel erläutert.

Fehlerbehandlung

Bei der Ausführung eines Dienstes können zwei Fehlertypen unterschieden werden: Solche, die durch das System detektiert werden, aber ohne Hilfe des Benutzers nicht gelöst werden können. Und solche, die nur durch den Benutzer erkannt werden können.

In beiden Fällen wird die Fehlersituation dem Benutzer mitgeteilt, damit dieser das System aus der Situation führen kann. Dies ist mit der Ausführung weiterer Dialogziele verbunden. Ist die Situation gelöst, kann das System aber nicht die Ausführung des ursprünglichen Dienstes wieder aufnehmen. Nach Eigenschaft E3 ist dem System gar nicht bekannt, daß ein Dienst nicht erfolgreich ausgeführt werden konnte. Nach Eigenschaft E2 kann nicht entschieden werden, daß die Ausführung der problemlösenden Dialogziele die Fortsetzung des ursprünglichen zur Folge haben sollte.
Der Benutzer muß den Vorgang komplett von vorne wiederholen.

Informationen

Dialogzielübergreifende Informationen werden nach dem Erwerb „vergessen" (Eigenschaft E1). Im „Wizard of Oz"-Szenario muß beispielsweise der Name des Benutzers bekannt sein. Außerdem soll einem Benutzer eine Lieblingstasse zugeordnet werden können. Diese Informationen sollen natürlich nur erfragt werden, wenn sie noch nicht bekannt sind [1]. Aus diesem Grund können sie nicht in einem Dialogziel als benötigte Informationen verankert sein.

U 1:	Please bring my favorite cup.
R 1:	Which one is your favorite cup?
U 2:	The blue one.
	Das System speichert die Information.
	Nichts weiter passiert.
U 3:	Please bring my favorite cup.
	Der Roboter bringt die blaue Tasse.

Abbildung 5.8: Beispiel-Dialog für eine dialogzielübergreifende Information

Im Beispiel-Dialog (vgl. Abbildung 5.8) verlangt der Benutzer nach seiner Lieblingstasse. Da diese noch nicht nicht bekannt ist, eröffnet das System einen entsprechenden Subdialog. Nach dem Subdialog kann auf die Information nicht über den Diskurs zugeriffen werden, der Benutzer muß erneut nach seiner Lieblingstasse fragen.

Benutzerwechsel

Der Wechsel des aktiven Benutzers ist ein Sonderfall, da er nicht aufgrund der genannten Eigenschaften auftritt. Zu Beginn der Arbeit wurde er als einer der Aspekte identifiziert, die mit den bisherigen Konzepten des Dialogsystems nicht behandelt werden können. Daher wird er hier erwähnt.
Es gibt hauptsächlich zwei Gründe, warum der Benutzer während einer Interaktion mit dem Roboter wechseln kann. Der erste ist sehr banal: Ein zweiter Benutzer unterbricht den Dialog zwischen dem ersten Benutzer und dem System. Falls sich die Intentionen der beiden Benutzer unterscheiden, findet ein Kontextwechsel statt. Der bis zu diesem Zeitpunkt geführte Dialog

[1] Wenn sie also noch nicht persistent in einer Datenbank auffindbar sind.

und der entsprechende Diskurs gehen verloren. Sind die Intentionen gleich, würde der Dialog einfach fortgeführt.
Der zweite Grund ist, der Roboter wendet sich dem zweiten Benutzer zu. Dies kann beispielsweise passieren, wenn der Roboter den Auftrag hatte, dem Benutzer einen Gegenstand zu bringen. Nach Ergreifen des Gegenstandes muß der Benutzer detektiert werden, um ihm den Gegenstand aushändigen zu können. Dieser Vorgang ist natürlich einer gewissen Fehlerwahrscheinlichkeit unterworfen.
In allen Fällen ist sich das Dialogsystem nicht bewußt, daß ein Benutzerwechsel stattfand.

Kapitel 6

Dialogstruktur

Die definierten Dialogziele einer Dialogapplikation definieren die höchste strukturelle Ebene. Jedes Dialogziel stellt einen Rahmen dar, innerhalb dessen der Dialogmanager Entscheidungen über den Dialogverlauf fällt. Im vorherigen Kapitel wurden Eigenschaften und Konsequenzen des Basissystem diskutiert. In diesem Kapitel wird eine übergeordnete Dialogstruktur vorgestellt, die zur Lösung der geschilderten Probleme beitragen kann. Außerdem wird ein Entwicklungsprozess für die Implementierung von Dialogsystemen vorgeschlagen.

6.1 Erweiterung der Dialogzielzustände

Bevor die eigentliche Dialogstruktur beschrieben wird, sollen die bisherigen Dialogzielzustände des Basissystems erweitert werden.
Dialogziele können im Basissystem die Zustände *deselected*, *selected*, *determined* und *finalized* erreichen (vgl. Abschnitt 5.2.1). Nach Erreichen des Zustandes *finalized* ist die Ausführung des Dialogziels aus der Sicht des Dialogmanagers abgeschlossen. Hierin begründet sich die Einschränkung E3 (vgl. Abschnitt 5.4) mit den daraus resultierenden Problemen. Um die eigentliche Ausführung eines Dienstes besser in der Dialogsteuerung berücksichtigen zu können, werden weitere Dialogzielzustände definiert:

aborted Die Ausführung des Dienstes wurde aufgrund eines Fehlers abgebrochen.

executed Der Dienst wurde erfolgreich ausgeführt.

Abbildung 6.1 zeigt den aktualisierten Übergangsgraphen für alle Dialogzielzustände. Im Falle einer erfolgreichen Ausführung des betreffenden Dienstes wird der Zustand *executed* erreicht. Der Dialogmanager kann neue Aufträge des Benutzers bearbeiten.
Tritt eine Fehlersituation ein, die mit Hilfe des Benutzers gelöst werden kann, wird der Zustand *aborted* erreicht. Die Abhandlung der folgenden Dialogziele kann der Dialogmanger im Kontext der Fehlersituation durchführen. An geeigenten Stellen kann der Dialogmanger die Applikation auffordern, die Ausführung des Dienstes fortzuführen.

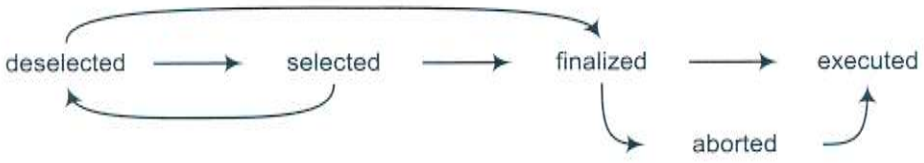

Abbildung 6.1: Übergänge für Dialogzielzustände im erweitertem Zustandsraum (*selected* ist Startzustand, *aborted* und *executed* sind Endzustände).

6.2 Höhere Kontextebene

Fehlerbehandlung, Korrekturen und Umgangsformen in der Kommunikation erfordern selbst für eine einfache Aufgabe, wie „Tasse bringen" (vgl. Abschnitt 4.2), die Ausführung mehrerer Dialogziele. Dabei läßt sich für jede Aufgabe ein mehr oder weniger festgelegtes Auftreten dieser Dialogziele und deren Reihenfolge definieren. Ein ähnliches Vorgehen ist bereits aus der Softwaretechnik bekannt: die Formulierung von Anwendungsfällen (vgl. [Bal99]). Tatsächlich läßt sich das Konzept auf die Entwicklung von dialoggesteuerten Anwendungen übertragen. Später wird bei der Beschreibung eines Entwicklungsprozesses genauer darauf eingegangen (vgl. Abschnitt 6.5).

Die meisten Benutzer beginnen ihren Dialog mit einer Begrüßung. Danach formulieren sie ihren Auftrag an den Roboter. Nach einer erfolgreichen Ausführung bedanken sie sich und verabschieden sich eventuell. Diese Partitionierung des gesamten Dialoges in drei Teile wurde durch das Ergebnis der Wizard of Oz Studie (vgl. Abschnitt 7.2.4) bestätigt. Namentlich sind dies die folgenden Partitionen:

Einleitung: Begrüßung, evtl. Austausch von Höflichkeitsfloskeln.

Hauptteil: Formulierung des Auftrages, klären fehlender Informationen und Fehlerbehandlung. Eventuell Höflichkeitsfloskeln, wie Bedanken und Bearbeitung weiterer Aufträge.

Dialogende: Bedanken und Verabschieden.

Dialogziele aus dem Dialogende werden sehr unwahrscheinlich in der Einleitung auftreten. Die Vertreter der einzelnen Partitionen werden sehr wahrscheinlich in der Reihenfolge Einleitung, Hauptteil, Dialogende bearbeitet.
Vorsicht: Dialogziele können in mehr als einer Partition auftreten. Das Dialogziel *Thanks* ist ein Beispiel dafür: Es wird sowohl im Hauptteil als auch am Dialogende bearbeitet, sofern der Benutzer einen weiteren Auftrag vergibt. Diese Einteilung ist also nicht rein mathematisch zu verstehen. Sie soll vielmehr verdeutlichen, daß bestimmte Dialogziele mehr oder weniger wahrscheinlich aufeinander folgen können.

Aufgrund der vorangegangenen Diskussion kann nun eine Kontextebene definiert werden, die über derjenigen der Dialogziele gelegt wird.
Das Szenario „Tasse bringen" (vgl. Abschnitt 4.2) entspricht, wie oben schon angedeutet, einem Anwendungsfall. Für jeden Anwendungsfall der dialoggesteuerten Applikation kann ein gerichteter Graph definiert werden, der Auftreten und Reihenfolge der möglichen Dialogziele beschreibt. Genauer gesagt, werden Auftreten und Reihenfolgen von *Dialogzielzuständen* modelliert.

Definition 4 (Dialogstruktur):
Sei D die Menge aller möglichen Dialogzielzustände für einen Anwendungsfall. Seien weiter $t = (D \times D, P)$ eine Transition mit Wahrscheinlichkeit P und T die Menge aller Transitionen, dann ist der gerichtete Graph $S = (G, T)$ eine Dialogstruktur für diesen Anwendungsfall.

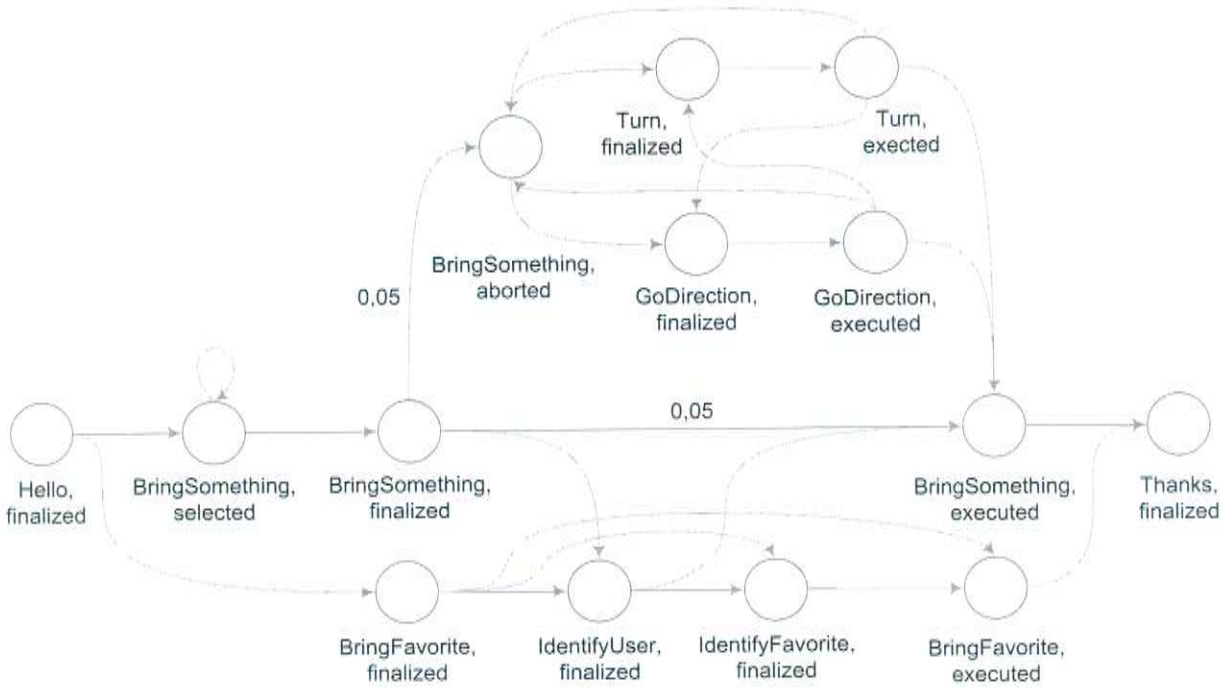

Abbildung 6.2: Ausschnitt einer übergeordneten Dialogstruktur für den Anwendungsfall „Tasse bringen" (Transitionen und Wahrscheinlichkeiten sind nicht vollständig, sondern nur exemplarisch angegeben).

Abbildung 6.2 zeigt einen Ausschnitt des Graphen für den Anwendungsfall „Tasse bringen". Die Wahrscheinlichkeiten der ausgehenden Kanten eines jeden Knotens bilden eine normierte Wahrscheinlichkeitsverteilung.

Der Graph bildet den erwarteten Dialogverlauf ab. Die Zustandsübergänge kommen entweder durch eine Zustandsänderung der Applikation oder durch Benutzeräußerungen zustande. Die einzelnen Wahrscheinlichkeiten werden in drei Schritten bestimmt:

1. Annotation von Hand
2. Training eines Klassifikators
3. Lineare Interpolation

Annotation von Hand

Mit der Modellierung der Anwendungsfälle können die initialen Übergangswahrscheinlichkeiten der einzelnen Dialogzielzustände bestimmt werden. Das entspricht einem initialen Dialogdesign, das direkt aus der Analyse hervorgeht. Zunächst werden alle vorgesehenen Transitionen zwischen den einzelnen Dialogzielzuständen bestimmt. Für die ausgehenden Transitionen eines Zustandes d folgt dann die Wahrscheinlichkeitsverteilung der Gleichverteilung:

$$f_H(i) = \begin{cases} \frac{1}{n} & t_i \in T \text{ ist ausgehende Transition von } d \\ 0 & \text{sonst} \end{cases}$$

Dabei ist n die Anzahl der ausgehenden Transitionen von d. Der Wert von n kann maximal so groß sein, wie die Anzahl der Dialogzielzustände. Im Umkehrschluß wäre also ein Übergang zu jedem Dialogzielzustand im Design vorgesehen und wird wohl eher selten sinnvoll sein. Dieser Schritt ist einerseits nötig, um initiale Werte für eine Benutzerevaluation zu bekommen. Andererseits können so in der Evaluation ungesehene Übergänge berücksichtigt werden.
Tabelle 6.1 zeigt in der ersten Zeile die annotierten Wahrscheinlichkeiten für das Beispiel aus Abbildung 6.2.

Training eines Klassifikators

Mit den Daten aus einer Benutzerevaluation wird ein Klassifikator trainiert. Für einen beobachteten, aktuellen Zustand liefert er eine Wahrscheinlichkeitsverteilung für alle Transitionen zu allen existierenden Dialogzielzuständen.
Tabelle 6.1 zeigt in der zweiten Zeile die trainierten Wahrscheinlichkeiten für das Beispiel aus Abbildung 6.2. Als Klassifikator wurde ein Bayes-Netz verwendet.

Lineare Interpolation

Im letzten Schritt werden die zwei Verteilungen, die jeweils zu einem Zustand gehören, linear interpoliert. Dabei werden die zwei Übergangswahrscheinlichkeiten, die zu einer Transition gehören, nach Gleichung 6.1 addiert.

$$P = \lambda P_H + (1 - \lambda) P_B \tag{6.1}$$

Tabelle 6.1 zeigt das Ergebnis der Interpolation für das Beispiel aus Abbildung 6.2 in der dritten Zeile. Die beiden Verteilungen wurden gleichwertig gewichtet ($\lambda = 0,5$).

	(*BringSomething, aborted*)	(*BringSomething, executed*)
Hand	0,1	0,1
Klassifikator	0,001	0,001
Interpolation	0,0505	0,0505

Tabelle 6.1: Übergangswahrscheinlichkeiten für den Dialogzielzustand (*BringSomething, finalized*) nach (*BringSomething, aborted*) und (*BringSomething, executed*). Die vollständigen Verteilungen werden später in Abbildung 6.4 dargestellt.

Mit Hilfe der Dialogstruktur können Aktionen des Dialogmanagers zu bestimmten Zeitpunkten im Dialogverlauf ausgewählt werden. Die Aktionen führen dann zu den erwarteten Übergängen zwischen den Dialogzielzuständen. Im obigen Beispiel wäre es die Fortführung des abgebrochenen Dienstes, der durch das Dialogziel *BringSomething* repräsentiert wird. Angenommen, der Dialogverlauf ist im Zustand (*GoDirection, executed*) angelangt. Eine Aufforderung an die Applikation, den unterbrochenen Dienst wieder aufzunehmen, führt zum Zustand (*BringSomething, executed*). Zumindest mit einer sehr großen Wahrscheinlichkeit.

6.3 Aktionen des Dialogmanagers

Wie schon zuvor beschrieben, bildet die Dialogstruktur den erwarteten Dialogverlauf ab. Dieser kann nun eingesetzt werden, um zu einem bestimmten Zeitpunkt im Dialog eine adäquate Aktion des Dialogmanagers auswählen zu können. Dazu werden die Übergänge zwischen den Dialogzielzuständen mit den auszuführenden Aktionen annotiert. Die möglichen Aktionen werden im folgenden genauer erläutert.

Ausführen der Dialogstrategie

Der Dialogmanager stellt Anfragen an den Benutzer oder startet Dienste der Applikation, so wie es im domänenabhängigen Teil des Dialogsystems deklariert wurde. Die zugehörigen Übergänge der Dialogzielzustände sind meist der Form:

$$\begin{aligned}
(g, \mathit{selected}) &\rightarrow (g, \mathit{selected}) \\
(g, \mathit{selected}) &\rightarrow (g, \mathit{finalized}) \\
(g_{t-1}, \mathit{executed}) &\rightarrow (g_t, \mathit{selected}) \\
(g_{t-1}, \mathit{executed}) &\rightarrow (g_t, \mathit{finalized})
\end{aligned}$$

Im Beispiel aus Abschnitt 6.2 ist ein entsprechender der Übergang zu sehen:

$$(\mathit{BringSomething}, \mathit{selected}) \rightarrow (\mathit{BringSomething}, \mathit{finalized})$$

Fortführen eines Dialogziels

Korrekturen, Kontextwechsel und fehlende Informationen, die über den Rahmen des Dialogziels hinausgehen, erfordern Subdialoge. Nach der Beendigung eines Subdialoges muß der vorherige Dialog wieder aufgenommen werden. Genauer gesagt, die Finalisierung des vorangegangenen Dialogzieles wird wieder aufgenommen. Die zugehörigen Zustandsübergänge können sein:

$$(g_{t-1}, s) \quad \rightarrow \quad (g_t, s), \text{ mit } s \in \{selected, finalized\}$$

Fortführen eines Dienstes

Nach der Unterbrechung eines Dienstes der zugrundeliegenden Applikation soll dieser Dienst wieder aufgenommen werden. Dies ist insbesondere nach der Auflösung von Fehlersituationen der Fall. Daher sind die meisten korrespondierenden Übergänge der Form:

$$(g_{Lösung}, executed) \quad \rightarrow \quad (g_{Dienst}, executed)$$

Im Beispiel (vgl. Abschnitt 6.2) wäre dies:

$$(GoDirection, executed) \quad \rightarrow \quad (BringSomething, executed)$$

Mitteilung an den Benutzer

Es wird eine Mitteilung an den Benutzer ausgegeben, die meist den Systemzustand betrifft. Insbesondere in den Fehlersituationen (vgl. Abschnitt 4.2.2) wird diese Aktion gewählt. Die Mitteilung soll den Benutzer implizit zur Hilfestellung auffordern.
Wird die Tasse nicht detektiert (vgl. Abschnitt 4.2), meldet der Roboter: *"I do not see the blue cup"*. Der Benutzer kann dann den Roboter so dirigieren, daß sie in sein Gesichtsfeld rückt. In Abbildung 6.2 entspricht dies dem Übergang

$$(BringSomething, aborted) \rightarrow (GoDirection, finalized)$$

Anfrage an den Benutzer

Das Dialogsystem benötigt Informationen oder Entscheidungen, die über den Rahmen eines Dialogziels hinaus gehen. Bei einem Kontextwechsel bzw. Wechsel des behandelten Dialogzieles, kann das System eine Klärungsfrage stellen (vgl. [Kru05, KHW05]). Ist der Wechsel beabsichtigt oder liegt ein Fehler des ASR-Subsystems vor?
Ein anderer Grund kann eine Information sein, die über den Rahmen eines Dialogzieles hinaus verwendet wird. Dies sind beispielsweise der Name eines Benutzers oder seine Präferenzen bezüglich einer Tasse.

6.4 Verbessern der Spracherkennerleistung

Holzapfel und Fügen haben gezeigt, daß durch eine enge Koppelung des Dialogmanagers mit dem ASR-Subsystem eine Verbesserung der Spracherkennerleistung möglich ist (vgl. Abschnitt

3.3). Durch die Verwendung der gleichen linguistischen Wissensbasen für Dialogmanager und ASR-Subsystem kann der Suchraum, bezüglich einer Äußerung, für den Spracherkenner eingeschränkt werden. Praktisch geschieht dies durch Gewichtung von bestimmten Regeln der Grammatik.
Der in den vorgestellten Arbeiten berücksichtigte Kontext beschränkt sich aber auf den Rahmen eines Dialogzieles. Ausnahme ist die Disambiguierung mehrerer Dialogziele. In diesem Falle beschränkt sich der Kontext aber auf die Disambiguierung. Mit der Dialogstruktur kann die Spracherkennungsleistung auch an den Übergängen zwischen zwei Dialogzielen verbessert werden: Angenommen, für einen Dialogzielzustand g_{t-1} existieren die Folgezustände $g_{t,1} \dots g_{t,n}$ mit der korrespondierenden Wahrscheinlichkeitsverteilung $f_t(i)$ über die Transitionen. Für jeden $g_{t,i}$ kann eine Menge von Benutzeräußerungen A_i bestimmt werden, die zum Übergang in $g_{t,i}$ führen. Wird der Übergang nicht durch eine Äußerung initiiert, so ist $A_i = \emptyset$. Zu jedem A_i kann eine Regel r_i aller Grammatikregeln bestimmt werden, deren Ableitung die Äußerungen in A_i ergibt. Anhand der Wahrscheinlichkleitsverteilung $f_t(i)$ kann die Regel r_i gewichtet werden.
Tabelle 6.2 zeigt beispielhaft Nachfolgezustände, Regeln und Übergangswahrscheinlichkeiten für einen Dialogzielzustand (*BringSomething, aborted*).

Dialogzielzustand	Nachfolgezustand	W-keit	Grammatikregel
(*BringSomething, aborted*)	(*GoDirection, selected*)	0,20	*act_godirect*
	(*GoDirection, finalized*)	0,25	*act_godirect*
	(*Turn, selected*)	0,20	*act_turn*
	(*Turn, finalized*)	0,25	*act_turn*
	(*GoodBye, finalized*)	0,10	*act_goodbye*

Tabelle 6.2: Übergangswahrscheinlichkeiten und Grammatikregeln für (*BringSomething, finalized*)

Hier würden die Regeln *act_godirect* und *act_turn* stark gewichtet. Neutral bleibt *act_goodbye*, während alle anderen, nicht genannten Regeln „bestraft" würden.

6.5 Entwicklungsprozess

Die Entwicklung von Software jeder Art erfordert einen reproduzierbaren Prozeß, um eine gewisses Maß an Qualität sichern zu können. Dies trifft natürlich auch auf die Entwicklung von Dialogsystemen zu. Die Komplexität und Vielfalt der gültigen Eingaben für ein natürlichsprachliches Dialogsystem kann erfahrungsgemäß nicht mit einem „Großen Wurf" abgedeckt werden, wie es beispielsweise in der Arbeit von Matthias Denecke (vgl. [Den02]) vorgeschlagen wird. Jeder Praxiseinsatz bringt neue Formulierungen und Verhaltensweisen der Benutzer mit sich. Sie sind daher nur schwer vorhersehbar und beeinträchtigen die Gesamtleistung des Systems. Wird die Entwicklung des Dialogsystems iterativ nach dem in Abbildung 6.3 gezeigten Schema durchgeführt, kann dieses Problem umgangen werden.
Aufgrund der Erfahrungen aus dieser Arbeit wird der folgende Entwicklungsprozeß für natürlichsprachliche Dialogsysteme vorgeschlagen:

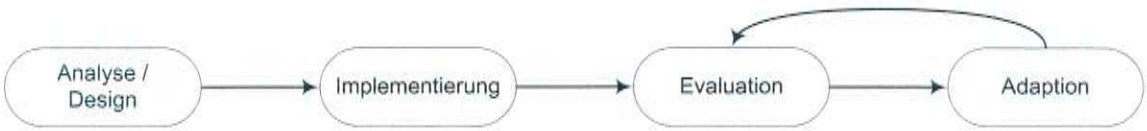

Abbildung 6.3: Schema für einen iterativen Entwicklungsprozeß

1. Definition von Anwendungsfällen
2. Erstellen der initialen Dialogstruktur, Ontologie und Grammatik
3. Durchführung eines „Wizard of Oz"-Experimentes
4. Adaption von Dialogstruktur, Ontologie und Grammatik
5. Durchführung einer Benutzerevaluation
6. Bei schlechter Systemleistung die Schritte 4 bis 6 wiederholen

6.5.1 Definition von Anwendungsfällen

Zu Beginn muß die Funktionalität des Systems definiert werden, wie sie nach außen für den Benutzer wahrnehmbar ist. Zu diesem Zweck werden sog. Anwendungsfälle (*use cases*) identifiziert. Das Konzept der Anwendungsfälle ist aus der objektorientierten Softwareentwicklung bekannt (vgl. [Jac95]).
Der Begriff „Anwendungsfall"kann auf unterschiedlichen Abstraktionsebenen definiert werden (vgl. [Bal99]). Hier wird der Anwendungsfall auf der Ebene des Systems definiert:

> **Definition 5 (Anwendungsfall):**
> Ein Anwendungsfall ist eine Sequenz zusammengehöriger Aktionen, die von einem Akteur im Dialog mit einem System ausgeführt werden, um für den Akteur ein Ergebnis von meßbarem Wert zu erstellen.

Ein wichtiger Aspekt der Anwendungsfälle ist, daß sie aus der Sicht des Benutzers identifiziert und beschrieben beschrieben werden. Dies geht deutlich aus der Definition hervor. Sämtliche Details zu Design und Implementierung werden dabei nicht berücksichtigt. D.h., die Funktionalität des Dialogsystems wird aus der Sicht des Benutzers beschrieben.

Die Aktionen können in zwei Klassen eingeteilt werden: charakteristische Aktionen und Aktionen zur Behandlung von Fehlersituationen. Damit der Anwendungsfall erfolgreich abgeschlossen werden kann, müssen bestimmte charakteristische Aktionen – mindestens jedoch eine – ausgeführt werden.
Um die Aktionen zur Behandlung von Fehlersituationen bestimmen zu können, werden die Fehlersituationen selbst identifiziert.

Zuletzt werden aus den Aktionen Dialogziele und ihre Zustände abgeleitet. Die Wahrscheinlichkeitsverteilungen $P_{H,i}$ der Transitionen werden für jeden Dialogzielzustand von Hand bestimmt.
Weiterhin werden eine initiale Ontologie und Grammatik erstellt.

6.5.2 Durchführung „Wizard of Oz"-Experiment und Analysen

Nachdem die Anwendungsfälle definiert wurden, wird für jeden ein „Wizard of Oz"-Experiment durchgeführt. Das Experiment gibt Aufschluß darüber, wie sich die Benutzer dem System gegenüber verhalten und wie gut Ontologie und Grammatik bereits dieses Verhalten widerspiegeln. Verschiedene Analysen beurteilen die qualitative Leistung des Systems und decken Schwachstellen auf.
Anhand der Analysen werden Dialogziele, Ontologie und Grammatik adaptiert. Meist kann die initial erstellte Grammatik nicht annähernd alle Benutzeräußerungen erzeugen. Das Experiment stellt also die erste reale Grundlage für die Entwicklung einer nahezu vollständigen Grammatik dar[1].
Die Dialogstruktur für jeden Anwendungsfall wird durch die Analyse der Dialogverläufe bestimmt. Dabei werden die auftretenden Dialogzielzustände und ihre Transitionen untersucht und in der Struktur abgebildet: Die Wahrscheinlichkeitsverteilungen $P_{B,i}$ der Transitionen werden für jeden Dialogzielzustand bestimmt. Anschließend werden die $P_{H,i}$ und $P_{B,i}$ interpoliert (vgl. Abschnitt 6.2).
Abbildung 6.4 zeigt die vollständigen Wahrscheinlichkeitsverteilungen, die durch Annotation von Hand, Training eines Bayes-Netzes und Interpolation entstanden sind.

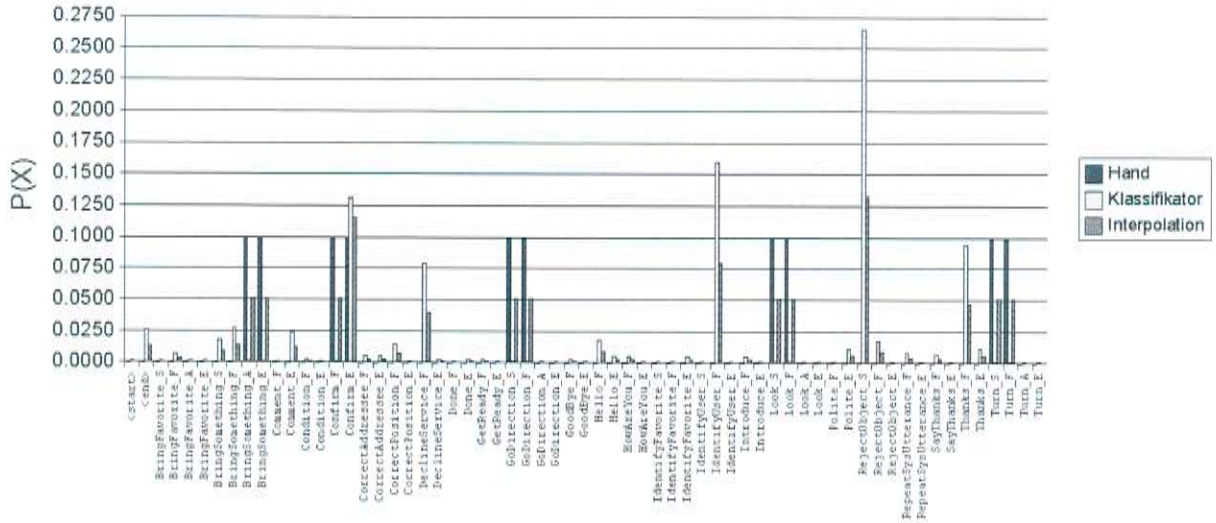

Abbildung 6.4: Wahrscheinlichkeitsverteilungen für den Dialogzielzustand (*BringSomething, finalized*)

[1]Eine vollständige Grammatik für spontane, natürliche Sprache zu erstellen ist sehr resourcenintensiv und wird daher wohl nie angestrebt werden.

Die Durchführung eines „Wizard of Oz"-Experimentes wird in Abschnitt 7.2 detaillierter beschrieben. Details zu den Analysen sind in Abschnitt 7.4 zu finden.

6.5.3 Durchführung von Benutzerevaluation und Analysen

Alle benötigten Quellen des Dialogsystems stehen nun als Artefakte des letzten Schrittes zur Verfügung. Das System kann jetzt einer Benutzerevaluation in seiner Zielumgebung unterzogen werden. Die gesammelten Daten werden wieder unter dem Aspekt der Systemleistung analysiert, Ontologie und Grammatik werden nach Bedarf erweitert. Die Wahrscheinlichkeitsverteilungen für die Dialogstruktur werden erneut berechnet und gewinnen an statistischer Relevanz.
Benutzerevaluation und Adaption des Systems können iterativ erfolgen, bis die gewünschte Systemleistung erreicht ist oder keine weitere Verbesserung erzielt werden kann.

Kapitel 7

Experimente und Analysen

Im Rahmen dieser Arbeit wurden zwei Benutzerexperimente sowie einige Analysen durchgeführt. In diesem Kapitel werden zunächst Metriken erläutert, die in den Analysen und Experimenten zur Auswertung der Daten herangezogen wurden. Danach werden das „Wizard Of Oz"-Experiment, die Benutzerevaluation und die Analysen in Zielsetzung, Aufbau und Durchführung beschrieben.

7.1 Metriken

7.1.1 Wortfehlerrate

Die Wortfehlerrate (*word error rate*, WER) wird oft bei der Spracherkennung als Maß für die Leistung des Spracherkenners herangezogen. Dabei wird die berechnete Hypothese einer Referenz (Transkription) gegenübergestellt. Beim Vergleich von Hypothese und Referenz können drei Fehlertypen unterschieden werden: Eingefügte Wörter (*insertions*, i), substituierte Wörter (*substitutions*, s) und fehlende Wörter (*deletions*, d), bezogen auf die Referenz. Die Wortfehlerrate wird nach Gleichung 7.1 berechnet (vgl. [Wikd]):

$$WER = \frac{C(i) + C(s) + C(d)}{C(n)} \tag{7.1}$$

Dabei sind $C(i)$, $C(s)$, $C(d)$ die Anzahl der jeweiligen Fehler und $C(n)$ die Anzahl der Wörter der Referenz. Abbildung 7.1 zeigt ein Beispiel zur Berechnung der Wortfehlerrate. Man sieht

Referenz:	***	******	turn	*ninety*	*degrees*	*to*	*the*	*right*
Hypothese:	*and*	*please*	turn	*around*	*******	**	***	*****
Korrekt:	16,7% (1)							
WER (Fehler):	116,7% (7)							

Abbildung 7.1: Beispiel für die Berechnung der Wortfehlerrate

sofort, daß zwei Wörter eingefügt wurden ("and", "please"), ein Wort substituiert wurde ("ni-

nety") und vier Wörter fehlen ("degrees", "to", "the", "right"). In Summe ergibt dies sieben Fehler. Da die Referenz nur sechs Wörter lang ist, fällt die Wortfehlerrate großer als 100% aus.

7.1.2 Wortakkuratheit und Satzakkuratheit

Zur Berechnung von WER und Wortakkuratheit und Satzakkuratheit werden in dieser Arbeit Werkzeuge aus dem Sphinx-4 Projekt verwendet. Diese implementieren einen Teil des „NIST align/scoring"-Algorithmus (vgl. [WLK+04, Sph]). Wortakkuratheit und Satzakkuratheit werden wie folgt berechnet:

$$Accurracy = \frac{C(\text{korrekte Wörter})}{C(\text{Wörter der Referenz})} \tag{7.2}$$

$$Sent.Acc. = \frac{C(\text{korrekte Sätze})}{C(\text{Sätze})} \tag{7.3}$$

7.1.3 Konzeptfehlerrate

Bei Dialogsystemen muß eine Benutzeräußerung nicht notwendigerweise Wort für Wort korrekt erkannt worden sein. Im Vordergrund steht das richtige Konzept bzw. die Frage, ob die Intention des Benutzers richtig erkannt wurde. [1]
Zur Berechnung werden für die Hypothese und ihre zugehörige Referenz jeweils die Semantik berechnet. Die beiden Semantiken werden auf Übereinstimmungen verglichen. Stimmen in beiden die Dialog-Akte überein, wurde zumindest das Konzept richtig erkannt. Darüber hinaus können auch die restlichen Teile der Semantiken übereinstimmen. Dementsprechend werden die Konzeptfehlerrate (*concept error rate*, CER) und die Semantikfehlerrate (*semantic error rate*, SER) nach den beiden folgenden Gleichungen berechnet.

$$CER = 1 - \frac{C(\text{korrekte Dialog-Akte})}{C(n)} \tag{7.4}$$

$$SER = 1 - \frac{C(\text{korrekte Semantiken})}{C(n)} \tag{7.5}$$

$C(n)$ steht für die Anzahl der Äußerungen insgesamt. Abbildung 7.2 zeigt ein Beispiel in dem die Semantik – und damit auch das Konzept – korrekt erkannt wurden. In den Abbildungen 7.3 und 7.4 sind Beispiele mit korrektem Dialog-Akt bzw. Konzept, zu sehen.

Referenz:	bring	me	*the*	blue	cup	$\begin{bmatrix} act\_bring \\ \text{OBJ} \begin{bmatrix} obj\_cup \\ \text{OPROP}[blue] \\ \text{NAME}[the\ blue\ cup] \end{bmatrix} \end{bmatrix}$
Hypothese:	bring	me	*that*	blue	cup	

Abbildung 7.2: Beispiel für ein korrekt erkanntes Konzept

[1] Selbstverständlich können auf dieser Ebene auch Ambiguitäten auftreten.

Referenz:	bring	me	the	*blue*	cup	$\begin{bmatrix} act\_bring \\ \text{OBJ} \begin{bmatrix} obj\_cup \\ \text{OPROP}[green] \\ \text{NAME}[the\ green\ cup] \end{bmatrix} \end{bmatrix}$
Hypothese:	bring	me	the	*green*	cup	

Abbildung 7.3: Beispiel für einen korrekt erkannten Dialog-Akt bei falscher Semantik

Referenz:	bring	me	*the*	*blue*	cup	$\begin{bmatrix} act\_bring \\ \text{OBJ} \begin{bmatrix} obj_drinkableCold \\ \text{NAME}[coke] \end{bmatrix} \end{bmatrix}$
Hypothese:	bring	me	*coffee*	****	***	

Abbildung 7.4: Beispiel für einen korrekt erkannten Dialog-Akt bei falscher Semantik

7.1.4 Erkennungsrate

Die Erkennungsrate (*detection rate*, DR) kann zur Beurteilung der Erkennungsleistung eines Klassifikators herangezogen werden. Sie ist der Quotient aus den im Testdatensatz richtig erkannten Klassen und der Gesamtzahl der Datensätze:

$$DR = \frac{C(\text{Anzahl korrekt erkannter Klassen})}{C(\text{Anzahl Daten im Testdatensatz})} \tag{7.6}$$

7.1.5 Quadratic Loss

Die Erkennungsrate beurteilt quasi schwarz-weiß, ob eine Wahrscheinlichkeitsverteilung zum richtigen Ergebnis führt. Will man die Qualität der Wahrscheinlichkeitsverteilung selbst beurteilen, kann man beispielsweise die Quadratic Loss Funktion anwenden (vgl. [IE05]): Für eine bestimmte Belegung der Merkmale im Testdatensatz berechnet der zu beurteilende Klassifikator eine Wahrscheinlichkeitsverteilung $p_1, \ldots, p_k$[2]. Es wird eine weitere Verteilung $a_1, \ldots, a_k$ bestimmt. Das zur erwarteten j-ten Klasse gehörige a_j erhält den Wert 1, alle anderen den Wert 0. Nach folgender Formel wird dann ein Wert für diese Belegung berechnet:

$$\sum_{i}^{k} (p_i - a_i)^2 \tag{7.7}$$

Beinhaltet der Testdatensatz mehrere Instanzen, werden alle Werte aufsummiert und eventuell normiert. Je kleiner der Wert ausfällt, desto zuverlässiger kann die Wahrscheinlichkeitsverteilung die erwartete Klasse schätzen.

7.1.6 Entropie

Die Entropie $H(X)$ ist ein Maß für den mittleren Informationsgehalt eines Zeichens, das von einer diskreten gedächtnislosen Quelle X über einem Alphabeth Z emmittiert wird. Sie ist

[2] Die Wahrscheinlichkeitsverteilung muß normiert sein.

definiert als:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{7.8}$$

Dabei bezeichnet p_i die Wahrscheinlichkeit, mit der das i-te Zeichen emmitiert wird und n ist die Anzahl der Zeichen im Alpahet. Je kleiner die Entropie ausfällt, um so mehr Redundanzen enthält ein Text, der durch die Quelle erzeugt wird. Man kann auch sagen, er enthält mehr statistische Regelmäßigkeiten.
Die Entropie $H(X)$ kann mit der maximalen Entropie $H_{max}(X)$ normiert werden. Die maximale Entropie wird bei einer Gleichverteilung der Emmisionswahrscheinlichkeiten erreicht und berechnet sich durch:

$$H_{max}(X) = -\sum_{i} \frac{1}{n} \log_2 \frac{1}{n} = -\log_2 n \tag{7.9}$$

Durch die Normierung

$$\bar{H}(X) = \frac{H(X)}{H_{max}(X)}$$

können auch Quellen über verschiedene Alphabete miteinander verglichen werden. In dieser Arbeit werden Vorhersagemodelle untersucht, die sich in der Art der Beobachtungen und der Vorhersagen unterscheiden. Die Leistungen der Vorhersagemodelle werden durch die normierte Entropie gegenüber derjenigen Leistung verglichen, die bei einfachem Raten der jeweils vorherzusagenden Klasse erzielt werden kann.

7.2 „Wizard of Oz"-Experiment

Das „Wizard of Oz"-Experiment war für diese Arbeit aus zwei Sichten notwendig: aus der Sicht einer wissenschaftlichen Arbeit und aus der Sicht eines Entwicklungsprozesses für Dialogsysteme. Das Experiment liefert charakteristische Daten für das gewählte Szenario.
Das Szenario des Experimentes wurde schon in Abschnitt 4.3 erläutert. Streng genommen handelte es sich bei diesem Experiment um kein reines „Wizard of Oz"-Experiment (vgl. [Kel84]). Die Teilnehmer waren sich der Tatsache bewußt, daß Teile des Gesamtsystems durch menschliche Operatoren simuliert wurden. Im Einzelnen waren das der Roboter und das Dialogsystem. Der Operator des Dialogsystems (der Wizard) trat dabei so weit wie möglich in den Hintergrund. So wurde eine Situation geschaffen, die der eines „echten" „Wizard of Oz"-Experimentes sehr nahe kam.
Das Experiment wurde am Language Technologies Institute (LTI) der Carnegie Mellon University, Pittsburgh, USA durchgeführt.

7.2.1 Zielsetzung

Das „Wizard of Oz"-Experiment hatte zum Ziel, die folgenden Fragestellungen – aus der Sicht einer wissenschaftlichen Arbeit – zu erörtern:

- Ist das gewählte Szenario generell als analytische Grundlage für die geplanten, zu entwickelnden Modelle geeignet?

- Welches sind die grundlegenden Erkenntnisse, die bzgl. der in Abschnitt 5.4 beschriebenen Probleme gewonnen werden können?
- Wie sehen die charakteristischen Daten eines Basissystems aus?

Außerdem wurde das Experiment zur Erörterung folgender Fragestellungen – aus der Sicht eines Entwicklungsprozesses – herangezogen:

- Wie verhalten sich Benutzer während der Interaktion mit dem System?
- Wie sehen Ontologie und Grammatik für das gewählte Szenario aus?
- Wie ist die Dialogstruktur für das gewählte Szenario real aufgebaut? Welche Dialogzielzustände und Transitionen sind enthalten?

Eignung

Anhand der Dialogverläufe und der gewonnenen Daten konnte das gewählte Szenario in seiner Komplexität beurteilt werden. Bei zu niedriger oder zu hoher Komplexität der Dialoge kann das Szenario entsprechend korrigiert werden. Beispielsweise könnten häufig Situationen auftreten, die nach dem aktuellen Forschungsstand nicht gehandhabt werden können oder nicht zum geplanten Forschungsziel gehören.

Erkenntnisse

In Abschnitt 5.4 wurden Grenzen des Basissystems diskutiert. Um die damit verbundenen Probleme lösen zu können, müssen möglichst alle beschriebenen Situationen in ausreichender Anzahl vorkommen. Ein „Wizard of Oz"-Experiment gibt Aufschluß über das Auftreten der Situationen in einem realen Umfeld.
Möglicherweise ist eine Situation nicht relevant, weil sie nur sehr selten oder gar nicht auftritt. Eine Verbesserung des Dialogverlaufes durch das zu entwickelnde Modell ist dann schwer nachweisbar, da die statistische Signifikanz fehlt. Andererseits wäre der entsprechende Forschungsaufwand unnötig betrieben worden, da die Natürlichkeit des Dialogverlaufes nicht nachhaltig durch die Situation beeinträchtigt wird.

Basissystem

Die gewonnenen Daten charakterisieren ein mögliches Dialogsystem, das Dialoge mit einem sehr hohen Grad an Natürlichkeit erzeugt. Es ist daher als Vergleichssystem für das zu entwickelnde System geeignet. Der Begriff „Basissystem" mag an dieser Stelle irreführend sein, da das System nicht verbessert werden soll. Im Gegenteil stellt es ein Ziel dar, dem das zu entwickelnden System möglichst nahe kommen soll. Aus den gewonnenen Daten können Referenzmaße für die qualitative Beurteilung der Modelle berechnet werden.

Benutzerverhalten

Eine vollständige Vorhersage des Benutzerverhaltens ist bei der Entwicklung natürlich- und spontansprachlicher Dialogsysteme nahezu unmöglich. Aus der Erfahrung heraus bringt jedes Experiment – das neue und systemunerfahrene Benutzer einschließt – auch neue Erkenntnisse über deren Verhalten. Schon allein aus diesem Grund ist ein „Wizard of Oz"-Experiment aus der Sicht eines Entwicklungsprozesses unabdingbar. Aber auch aus wissenschaftlicher Sicht sind diese Erkenntnisse sehr wertvoll.

Ontologie und Grammatik

Ein wichtiger Teil des Benutzerverhaltens spiegelt sich in seinen Äußerungen wider. Diese bestimmen neben den Anwendungsfällen maßgeblich Ontologie und Grammatik eines Dialogsystems. Da diese die wichtigsten Konzepte eines Dialogsystems darstellen, werden sie gesondert hervorgehoben.
Viele Äußerungen sind als Grundstock vorhersehbar. Natürliche, spontane Sprache bietet allerdings eine Fülle von Variationsmöglichkeiten. Gründe hierfür sind Füllwörter, Wörter aus der Umgangssprache, "laxer" Umgang mit der Grammatik und Ellipsen.

Dialogstruktur

Die gewonnenen Daten über die Dialogverläufe fließen in die Dialogstruktur ein. Es können die Übergänge zwischen den einzelnen Dialogzielzuständen bestimmt werden sowie deren Wahrscheinlichkeiten.

7.2.2 Aufbau

Wie eingangs schon erwähnt, wurde das „Wizard of Oz"-Experiment in Pittsburgh durchgeführt. Das Robotersystem ARMAR wird an der Universität Karlsruhe entwickelt und gewartet, daher stand es nicht zur Verfügung. Um dennoch dem gewählten Szenario so gut wie möglich gerecht zu werden, wurde der Roboter durch einen Menschen simuliert. Er war durch eine Brille mit VGA-Monitor mit dem Dialogsystem verbunden. Seine Aufgabe war es, nur auf die Kommandos zu reagieren, die auf er auf seinem Monitor empfing. Diese waren sehr einfach und grundlegend gehalten, wie beispielsweise *"go to the left table"* oder *"take the blue cup"*. Abbildung 7.5 zeigt eine Nahaufnahme der Brille.
Neben Stellwänden wurden zwei Tische so aufgebaut, daß ein virtueller Raum entstand, der die Küche darstellen sollte. Auf den Tischen waren drei Tassen in unterschiedlichen Farben platziert: weiß, gelb und blau. Innerhalb des Raumes sollten sich die Benutzer und der Roboter aufhalten. Abbildung 7.6 zeigt ein Photo. In der Mitte ist der "Roboter" zu sehen und am rechten Rand der Proband, die Rolle des Benutzers übernahm.
Für die Sprachaufnahme wurde jeder Benutzer mit einem Funkmikrophon ausgestattet. Die Signale wurden über ein Mischpult, das auch als Vorverstärger diente, auf den linken und rechten Kanal einer Stereoaufnahme mit Hilfe einer handelsüblichen Soundkarte gesplittet. Unabhängig dazu wurde die gesamte Szene mit einer Videokamera aufgenommen, um die

Abbildung 7.5: Video-Brille mit VGA-Monitor

spätere Analyse der Daten zu vereinfachen. Ein Lautsprecher, platziert in einer Ecke des abgegrenzten Raumes, diente als Ausgabe für das TTS-Subsystem.
Zentraler Bestandteil des Aufbaus war ein PC, der gleichzeitig für die Sprachaufzeichung, Sprachsynthese und Steuerung des Dialoges genutzt wurde. Um den Operator (den Wizard) bei der Steuerung des Dialoges zu unterstützen, wurde ein Werkzeug implementiert. Dieses stellte Systemäußerungen als Vorlagen bereit, die per Mausklick einfach ausgewählt werden konnten und daraufhin an das TTS-Subsystem gesendet wurden. Analog dazu konnten aus einer weiteren Liste die Kommandos für den Roboter gewählt und an die Video-Brille gesendet werden. Gleichzeitig wurden alle Aktionen in einer Log-Datei festgehalten, um später eine vollständige Auswertung der Dialogverläufe zu ermöglichen.
Die Vorlagen unterstützten ein konsistentes Systemverhalten. So hatten die Benutzer den Eindruck, tatsächlich mit einem Dialogsystem zu sprechen. Andererseits war das Werkzeug so implementiert, daß einfache Ergänzungen möglich waren. Bei der Durchführung des Experimentes hat sich diese Vorgehensweise ausgezahlt. Abbildung 7.7 zeigt den Aufbau mit den gerade beschriebenen Komponenten schematisch.

7.2.3 Durchführung

Während der Durchführung des Experimentes wurden insgesamt neun Probanden involviert. Darunter waren sowohl Muttersprachler als auch Teilnehmer, deren Muttersprache nicht Englisch war. Die Dialoge wurden alle auf Englisch geführt.
Damit alle Probanden das gleiche Verständnis davon hatten, um was es sich bei dem Experiment drehte und was sie zu tun hatten, wurden sie gebeten, eine schriftliche Einweisung still durchzulesen. Dadurch waren für jeden Durchgang die gleichen Voraussetzungen gegeben. Die Einweisung enthielt Informationen über das Szenario und eine Rollen- bzw. Aufgabenbeschreibung (vgl. Anhang A). Die Probanden wurden mit Headset-Mikrophonen ausgestattet

Abbildung 7.6: Aufbau des „Wizard of Oz"-Experimentes mit "Roboter" (Mitte) und einem der Benutzer (rechts)

und die Signale ausgepegelt. Anschließend wurden drei verschiedene Varianten des Szenarios durchgespielt. Um mehr Dialoge mit der gleichen Anzahl Probanden zu erhalten, wurden die Rollen getauscht und nochmals drei Varianten durchgespielt.
Die zwei beteiligten Benutzer bekamen jeweils eine Rolle zugeteilt:

Benutzer (*User*, U): startet den Dialog bzw. wird als aktiver Benutzer vom System erkannt. Er hat die Aufgabe, sich eine der Tassen vom Roboter bringen zu lassen.

Unterbrechender Benutzer (*Interferring User*, IU): hält sich passiv im Hintergrund, unterbricht die Interaktion des anderen Benutzers mit dem System oder wird als aktiver Benutzer vom System detektiert.

Um verschiedene Situationen im Dialog zu simulieren, wurden Varianten des Benutzer- und Systemverhaltens vorgeschrieben:

Unbekannter Benutzer: Das System kennt den Benutzer nicht namentlich. Es erfragt den Namen, bevor der erste Dienst ausgeführt wird. Falls der Benutzer um seine Lieblingstasse gebeten hat, muß auch diese Information erfragt werden.

Falscher Benutzer: Der Roboter übergibt die Tasse an den falschen Benutzer. In den meisten Fällen war dies der IU.

Unterbrechung: Der IU unterbricht die Interaktion zwischen Roboter und Benutzer.

Roboter verfährt sich: Der Roboter fährt zum falschen Tisch.

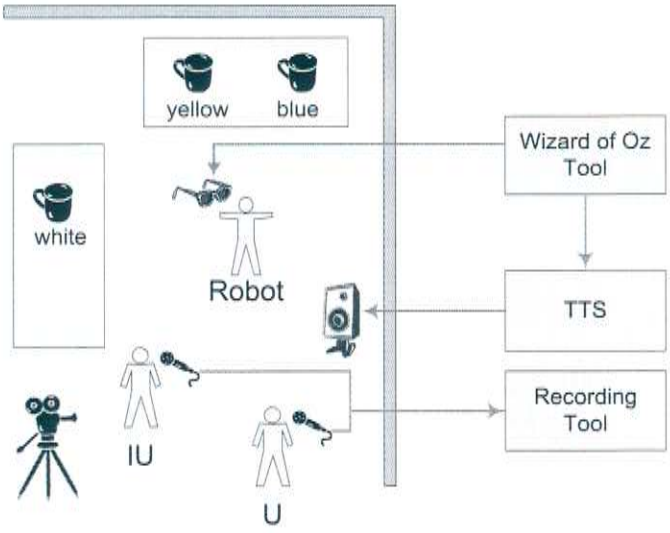

Abbildung 7.7: Schema für den Aufbau des Wizard-of-Oz Experimentes

7.2.4 Ergebnisse

Die Probanden hatten teilweise Erfahrung mit natürlichsprachlichen Dialogsystemen. Insgesamt wurden 21 Dialoge durchgespielt. Davon konnten nur 19 ausgewertet werden, da die Aufnahmen nicht vollständig waren. In den auswertbaren Dialogen kamen 272 Äußerungen der Benutzer mit insgesamt 1567 Wörtern vor.
Das Szenario eignet sich prinzipiell gut als analytische Grundlage, obgleich sich später herausstellte, daß es für die gegebenen Hardware-Eigenschaften zu kompliziert ist (vgl. auch Abschnitt 4.4). Eine der wichtigsten Vorraussetzungen, um zwei Benutzer gleichzeitig berücksichtigen zu können, ist nicht erfüllt: Auf dem Roboter ist keine Komponente integriert, die die Benutzer erkennen und unterscheiden, geschweige denn den aktuell aktiven Benutzer identifizieren kann. Aus diesem Grund wurde später das Szenario für die Evaluation auf einen Benutzer reduziert.
Die Dialoge wurden auf Ambiguitäten untersucht, die durch das Basissystem nicht aufgelöst werden können (vgl. Abschnitt 5.4.2). Es konnte keine statistische Relevanz festgestellt werden, da derartige Ambiguitäten kaum auftraten und unterschiedliche Ausprägungen hatten. Ein Beispiel ist in Abbildung 7.8 zu sehen. Äußerung U1 führt zum finalisierten Dialogziel *Ho-*

U1:	"Hello Robbi, how are you feeling today?"
R1:	"Fine, thanks"
U2:	"Excellent."
R2:	"Thank you."

Abbildung 7.8: Beispiel einer ambigen Benutzeräußerung

wAreYou. In Äußerung U2 kommentiert der Benutzer die Systemantwort S1. Das Basissystem interpretiert sie als Lob – dies entspricht dem Dialogzielzustand (*SayThanks,finalized*) – und antwortet brav mit einem *"Thank you"*.
Die Dialoge wurden nicht auf das Problem „Neustart" analysiert, da zum Zeitpunkt des „Wizard of Oz"-Experimentes der Fokus auf den Ambiguitäten lag.
Die beschriebenen Probleme „Korrektur", „Fehlerbehandlung", „Benutzerwechsel" wurden durch die Vorgaben für den jeweiligen Versuchsdurchgang provoziert.
Die in Abschnitt 6.2 angesprochene Partitionierung der Dialogziele auf einer zeitlichen Achse hat sich, wie zu erwarten war, bestätigt. Die gesamte Dialogstruktur ist für eine lesbare Darstellung zu komplex, da sie über 50 Dialogzielzustände enthält, wobei die Zustände *selected* und *finalized* bereits zusammengefaßt sind. Abbildung 7.9 zeigt quasi eine Skizzierung der Dialogstruktur als gerichteten Graphen. Die Dialogzielzustände sind als Knoten repräsentiert, jede der Kanten stellt eine vorkommende Transition dar. Anhand der Anzahl der Kanten, die die selben Ausgangs- und Endknoten haben, lassen sich tendentielle Pfade durch die möglichen Dialogverläufe erkennen.
Die Ergebnisse zu Ontologie und Grammatik werden in Abschnitt 7.4 diskutiert.

7.3 Analysen zu statistischen Modellen

Die Dialoge des „Wizard of Oz"-Experimentes dienten als Grundlage für eine Untersuchung, inwieweit Methoden des Maschinellen Lernens den Dialogverlauf im gegebenen Szenario vorhersagen können. Es wurde die Vorhersageleistung unterschiedlicher Methoden auf der Grundlage verschiedener Merkmalskombinationen untersucht.

7.3.1 Zielsetzung

Die Analysen sollten Aufschluß darüber geben, unter welchen Rahmenbedingungen der Dialogverlauf am besten vorhergesagt werden kann. Es sollte die Frage beantwortet werden, welche Konzepte des Dialogmanagers sich als Merkmale eignen und welche Methoden des Maschinellen Lernens eine gute Vorhersageleistung erbringen.

7.3.2 Durchführung

Um die Analysen durchführen zu können, mußten zunächst die Dialoge mit Konzepten des Dialogmanagers annotiert werden: Jeder Äußerung wurde der entsprechende Dialog-Akt und das entprechende Dialogziel zugeordnet. Systemaktionen, die ein Dialogziel in den Zustand *executed* versetzten, wurden ebenfalls annotiert.
Zur Berechnung der verschiedenen Modelle kam das WEKA der Universität of Waikato, NZ zu Einsatz (vgl. [IE05]). WEKA ist in Java geschrieben und bietet bereits eine Fülle von Klassifikatoren. Dennoch wurde es um einen N-Gram-Klassifikator und einem Gleichverteilungs-Klassifikator erweitert, da diese nicht vorhanden waren. Außerdem wurde der 1R-Klassifikator (vgl. Abschnitt 2.3.1) derart erweitert, daß er Wahrscheinlichkeitsverteilungen liefert, die keine Null-Werte aufweisen. Insgesamt kamen neben den eben genannten Klassifikatoren noch ein

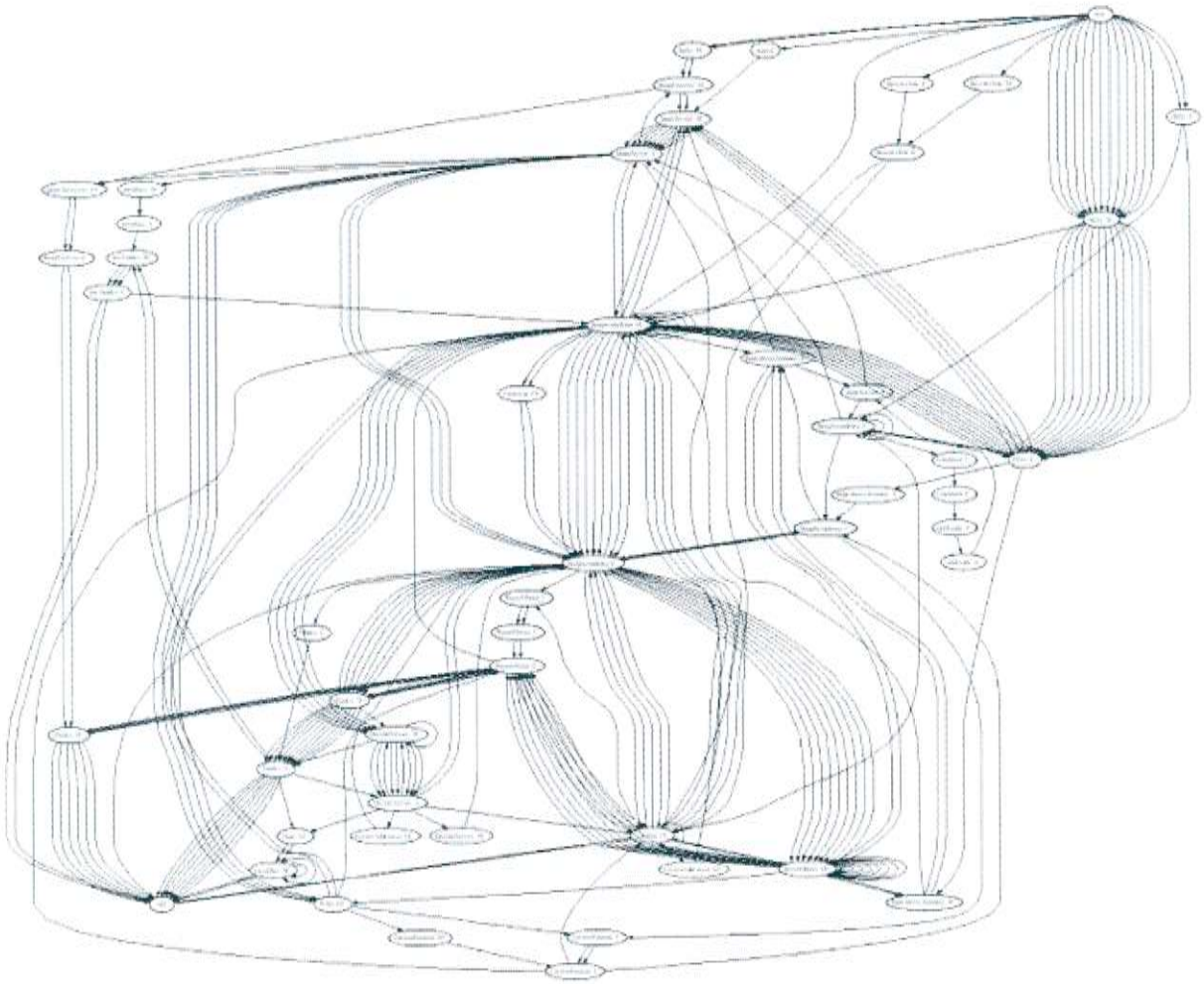

Abbildung 7.9: Zustandsübergänge für Dialogzielzustände

Bayes-Netzwerk-Klassifikator zum Einsatz. Klassifiziert wurde nach dem aktuellen Dialogziel (*actual goal*, AG), nach dem folgenden Dialogziel (*next goal*, NG), nach dem folgenden Dialogzielzustand (*next goal state*, NGS) und nach dem folgenden Dialogakt (*next dialogue act*, NDA) bezogen auf die aktuelle Benutzeräußerung. Als Eigenschaften wurden das vorherige Dialogziel (*last goal*, LG), das aktuelle Dialogziel, der aktuelle Dialogzielzustand (*actual goal state*, AGS), der aktuelle Dialogakt (*actual dialogue act*, AGA), der vorherige Dialogakt (*last dialogue act*, LDA) und Kombinationen daraus verwendet.
Da der Datenkorpus des „Wizard of Oz"-Experimentes nicht sehr groß aus der Sicht der Methoden des Maschinellen Lernens ist, wurden Training und Test ähnlich der k-fachen Kreuzvalidierung durchgeführt (vgl. [Mit97, IE05]). Dazu wurde der Datenkorpus so partitioniert,

daß sich ungefähr 10% der Dialoge im Testdatensatz befanden, dies entspricht zwei Dialogen. Es wurde darauf geachtet, daß die zwei Dialoge nicht die gleiche Belegung der Benutzerrollen hatten, weil davon auszugehen ist, daß diese sehr ähnlich im Verlauf sind. Pro Klassifikator wurden für jedes Trainings- und Testdatensatz-Paar Erkennungsrate, Quadratic-Loss und Entropie, berechnet. Die Ergebnisse wurden dann über alle Paare gemittelt.

7.3.3 Ergebnisse

Von den 21 Dialogen wurden jeweils zwei als Testdatensatz verwendet, die restlichen bildeten den Trainigsdatensatz. So waren im Schnitt der Trainingsdatensatz 152,5 Äußerungen und der Testdatensatz 16,9 Äußerungen groß. Die Ergebnisse sind in den Tabellen 7.1, 7.2 und 7.4 zu sehen. Für einen qualitativen Vergleich wurden die korrekte Vorhersage (*detection rate*, DR), Quadratic Loss (QL) und die Perplexität (PP) berechnet.

Merkmale	Klassenmerkmal	Modell	DR	QL	$\bar{H}(X)$
LG	AG	Gleichverteilung	0,0%	0,95	1,00
		1R	36,3%	0,87	0,70
		Bayes-Netzwerk	35,6%	0,78	0,69
		N-Gram	5,2%	1,15	0,66

Tabelle 7.1: Vergleich der Modelle für die Vorhersage des aktuellen Dialogziels

In Tabelle 7.1 wurden Vorhersagemodelle für das aktuell zu behandelnde Dialogziel beurteilt. Mit diesen Modellen können Ambiguitäten aufgelöst werden. Leider konnte dies nicht empirisch nachgewiesen werden, da die Zahl der Ambiguitäten in den Daten des „Wizard of Oz"-Experiments statistisch nicht relevant sind (vgl. Abschnitt 7.2.4).

Merkmale	Klassenmerkmal	Modell	DR	QL	$\bar{H}(X)$
AG	NG	Gleichverteilung	0,0%	0,95	1,00
		1R	37,9%	0,85	0,70
		Bayes-Netzwerk	37,2%	0,74	0,63
		N-Gram	5,2%	1,17	0,64
AGS	NGS	Gleichverteilung	0,0%	0,98	1,00
		1R	32,1%	0,93	0,66
		Bayes-Netzwerk	30,1%	0,83	0,70
		N-Gram	0,0%	1,23	0,55

Tabelle 7.2: Vergleich der Modelle für die Vorhersage des folgenden Dialogziels bzw. den folgenden Dialogzielzustand

In Tabelle 7.2 werden Vorhersagemodelle für das folgende Dialogziel bzw. des folgenden Dialogzielzustandes beurteilt. Mit ihnen kann der Dialogmanager passende Aktionen auswählen (vgl. Abschnitt 6.3). Damit der Dialogmanager mit Hilfe der Dialogstruktur sinnvoll Entscheidungen über den Dialogverlauf treffen kann, ist eine ausreichende Granularität nötig, die durch die verwendeten Symbole der Struktur gegeben wird. Dialogziele sind nicht dafür geeignet, da

mit ihnen allein keine Fehlersituationen modelliert werden können. Wie in Kapitel 6 schon beschrieben, werden Dialogzielzustände als Symbole verwendet. Dennoch wurde die Vorhersage des nächsten Dialogziels, gegeben das aktuelle Dialogziel, mit der Vorhersage des nächsten Dialogzielzustandes, gegeben den aktuellen Dialogzielzustand, verglichen. Natürlich fällt die durchschnittliche Leistung über alle statistischen Klsassifikationsmehtoden für die Dialogzielzustände schlechter aus, als für die Dialogziele (vgl. Tabelle 7.3). Das betrifft Erkennungsrate und Quadratic-Loss, und liegt vermutlich an der fast dreifachen Menge von möglichen Klassen[3]. Ein echter Vergleich ist also nur über die normierten Entropien möglich. Danach ist eine Verhersage des Dialogzielzustandes mit Hilfe der Klassifikatoren besser möglich als die des Dialogziels. Das paßt natürlich gut in die Vorgaben. Als Klassifikator wurde der 1R-Klassifikator gewählt, da mit ihm die beste Erkennungsrate erzielt wurde. Der 1R-Klassifikator ist zwar einfach, liefert aber auf den meisten Korpora sehr gute Resultate (vgl. [Rob93]).

Merkmale	Klassenmerkmal	Anzahl Klassen	DR	QL	$\tilde{H}(X)$
AG	NG	23	20,1%	0,93	0,74
AGS	NGS	63	15,5%	0,99	0,72

Tabelle 7.3: Durchschnittliche Leistung der Modelle für verschiedene Merkmale

In Tabelle 7.4 wurde die zukünftige Benutzeräußerung vorhergesagt. Mit ihr kann die Leistung

Merkmale	Klassenmerkmal	Modell	DR	QL	$\tilde{H}(X)$
AG	NDA	Gleichverteilung	0,0%	0,97	1,00
		1R	27,6%	0,96	0,69
		Bayes-Netzwerk	29,0%	0,85	0,75
		N-Gram	1,8%	1,15	0,66
ADA	NDA	Gleichverteilung	0,0%	0,97	1,00
		1R	27,2%	0,97	0,69
		Bayes-Netzwerk	27,9%	0,86	0,75
		N-Gram	0,9%	1,18	0,64
LG, ADA	NDA	Gleichverteilung	0,0%	0,97	1,00
		1R	27,2%	0,96	0,69
		Bayes-Netzwerk	32,7%	0,85	0,65
		N-Gram	1,5%	1,36	0,46
LDA, ADA	NDA	Gleichverteilung	0,0%	0,97	1,00
		1R	28,3%	0,95	0,69
		Bayes-Netzwerk	27,6%	0,90	0,59
		N-Gram	0,9%	1,20	0,62

Tabelle 7.4: Vergleich der Modelle für die Vorhersage des folgenden Dialog-Aktes

des Spracherkenners gesteigert werden, indem die korrespondierenden Grammatik-Regeln ge-

[3]Für jedes Dialogziel sind vier Zustände möglich, die aber nicht alle verwendet wurden.

wichtet werden. Der Bayes-Netzwerk-Klassifikator mit vorherigem Dialogziel und aktuellem Dialog-Akt als Entscheidungsmerkmale scheint für diese Aufgabe am besten geeignet, vergleicht man Erkennungsrate und den Wert der Quadratic Loss Funktion mit den übrigen Werten.

7.4 Analysen zu Ontologie und Grammatik

Für den humanoiden Roboter ARMAR existierte zu Beginn dieser Arbeit schon ein Dialogsystem, das unter anderem auch das gewählte Szenario zum großen Teil abdeckte (vgl. Abschnitt 4.4). Die Daten des „Wizard of Oz"-Experiment wurden dazu verwendet, die vorhandene Ontologie und die Grammatik zu adaptieren.

7.4.1 Zielsetzung

Zunächst sollten die Schwachstellen der vorhandenen Ontologie und der Grammatik ermittelt werden, um danach gezielt eine Adaption vornehmen zu können. Nach der Adaption sollte eine erneute Analyse eine Leistungsverbesserung des Systems zeigen. Eventuell können Ontologie und Grammatik erneut angepaßt werden. Die Analysen wurden auf syntaktischer Ebene (Wortebene) und auf semantischer Ebene (Semantische Repräsentation des NLU-Subsystems) durchgeführt.

7.4.2 Analyse der Spracherkennerleistung

Die Benutzeräußerungen wurden während des Wizard-of-Oz Experimentes nur aufgezeichnet. Es fand keine weitere Verarbeitung durch ein ASR-Subsystem statt. Die Aufgabe der Spracherkennung und -interpretation wurde durch den Wizard erledigt.
Die vorliegenden Audiodaten wurden zunächst von Hand segmentiert und dann mit dem Spracherkenner Janus (vgl. [IW95]) mit Decoder Ibis (vgl. [Sol01]) und der vorhandenen Grammatik in Text übersetzt. Ibis kann neben N-Gram-Modellen auch Phrasenstrukturgrammatiken als Sprachmodelle verwenden (vgl. [FHW04]). In diesem Falle verwenden Dialogsystem und ASR-Subsystem dieselbe Grammatik. Als Ergebnis lag für jede Äußerung eine Hypothese vor. Die Hypothesen wurden mit ihren korrespondierenden Transkriptionen verglichen. Die verwendeten Metriken sind in Abschnitt 7.1 beschrieben. Tabelle 7.5 zeigt die Ergebnisse der Analyse. Die erste Zeile gibt die Wortakkuratheit und die Fehler an. Die Gesamtfehlerzahl von 975 setzt sich aus 560 substituierten, 55 eingefügten und 360 fehlenden Wörtern zusammen. Die zweite Zeile gibt die Wortfehlerrate (WER) und die dritte Zeile die Satzakkuratheit (Sent.Acc) an.

Accurracy:	41,33%	Errors:	975	(560, 55, 360)	
WER:	62,18%	Words:	1568	Matches:	648
Sent.Acc.:	21,69%	Sent.:	272	Matches:	59

Tabelle 7.5: Initialer Vergleich von Hypothesen und Transkriptionen

Im Kontext der Entwicklung eines Dialogsystems gibt es drei mögliche Ursachen für die auftretenden Fehler:

1. Die Äußerung ist „out of domain".
2. Die Grammatik ist unvollständig.
3. Es liegt ein Fehler des ASR-Subsystems vor.

Zunächst wurden kleinere Schreibfehler in den Transkriptionen korrigiert und die Grammatik erweitert. Beispielsweise fehlte in der Grammatik die Objektbeschreibung *"yellow"*. Äußerungen wie *"Please bring me the yellow cup."* konnten dadurch nicht korrekt erkannt werden. Ebenso waren relative Ortsangaben für greifbare Objekte nicht vorgesehen sowie Namen der Benutzer. Abbildung 7.10 zeigt ein Beispiel.

Transkription:	i	*wanted*	the	*yellow*	cup	*on*	*the*	*right*	*table*
Hypothese:	i	*want*	the	******	cup	*four*	*there*	*as*	*well*
Korrekt:	33,3% (3)								
Fehler:	66,7% (6)								

Abbildung 7.10: Beispiel für eine fehlerhaft erkannte Äußerung

Die segmentierten Audiodaten wurden, mit der adaptierten Grammatik, erneut mit Janus in Text übersetzt und die resultierenden Hypothesen mit den Transkriptionen verglichen. Tabelle 7.6 zeigt das Ergebnis. Wort- und Satzakkuratheit wurden deutlich um 11,7 bzw. 12,9 Prozent-

Accurracy:	53,03%	Errors:	770	(447, 34, 289)	
WER:	49,14%	Words:	1567	Matches:	831
Sent.Acc.:	34,56%	Sent.:	272	Matches:	94

Tabelle 7.6: Vergleich von Hypothesen und Transkriptionen nach Korrekturen in Grammatik und Transkriptionen

punkte gesteigert. Die Tatsache, daß im zweiten Durchlauf ein Wort weniger gezählt wurde, kann mit den Korrekturen der Transkriptionen erklärt werden.
Von den 272 Äußerungen wurden 40 als „out of domain"eingestuft. D.h. diese Äußerungen muß das Dialogsystem nicht interpretieren können, weil sie über die geplanten Funktionalitäten des Roboters hinausgehen. Zum Teil sind die betroffenen Äußerungen sehr lang, und beeinflussen die gemessene Systemleistung sehr stark negativ.
Ein Beispiel hierfür ist der folgende Satz: *"I would like you to actually place this cup next to the eyeglasses on the table and bring me the cup that says 'Maddison Square Garden' instead."* Die Aufgabenstellung ist für die korrekte Interpretation und Ausführung zu komplex; die Brille auf einem der Tische müßte – aus größerer Entfernung – detektiert werden und die relative Ortsangabe aufgelöst werden. Außerdem war die Brille nicht Teil des Szenarios. Sie gehörte dem Operator, der den Roboter simuliert hat, und lag nur versehentlich auf dem Tisch. [4]

[4]Dieses Beispiel zeigt auch, welche engen Vorgaben bislang notwendig sind, damit ein Robotersystem, wie ARMAR, sinnvoll arbeitet.

Die Systemleistung wurde nochmals ohne „out of domain"-Äußerungen gemessen. Tabelle 7.7 zeigt das Resultat. Mit 64,10% ist die Wortakkuratheit noch recht niedrig.

Accurracy:	64,10%	Errors:	444	(272, 20, 157)	
WER:	37,60%	Words:	1181	Matches:	757
Sent.Acc.:	42,67%	Sent.:	232	Matches:	99

Tabelle 7.7: Vergleich von Hypothesen und Transkriptionen ohne „out of domain"-Äußerungen

7.4.3 Semantische Analyse

Nicht alle fehlerhaft erkannten Äußerungen führen auch zu einer fehlerhaften Interpretation durch das NLU-Subsystem. Abbildung 7.11 zeigt ein typisches Beispiel dafür[5]. Um die Kon-

Transkription:	i	want	to	have	my	favorite	cup	it's	*the*	blue	one
Hypothese:	i	want	to	have	my	favorite	cup	it's	*a*	blue	one
Korrekt:	90,9% (10)										
Fehler:	9,1% (1)										

$$\begin{bmatrix} act\_bring \\ \text{OBJ} \begin{bmatrix} obj\_cup \\ \text{OPROP}[\,blue\,] \\ \text{NAME}[\,the\ blue\ cup\,] \end{bmatrix} \end{bmatrix}$$

Abbildung 7.11: Beispiel für eine fehlerhaft erkannte, aber semantisch korrekte Äußerung

zeptfehlerrate messen zu können, wurden Transkription und Hypothese einer Äußerung mit dem NLU-Subsystem von Tapas interpretiert. Unterschieden wurde dabei, ob die komplette Semantik korrekt war oder nur der Dialog-Akt. Ist beispielsweise die Farbe der Tasse falsch erkannt worden, ist die Semantik nicht korrekt. Tabelle 7.8 zeigt das Ergebnis einmal mit und einmal ohne „out of domain"-Äußerungen (OOD).

	mit OOD		ohne OOD	
Korrekter Dialog-Akt:	58,8%	160	65,1%	151
Korrekte Semantik:	49,3%	134	53,4%	124
Äußerungen gesamt:		272		232

Tabelle 7.8: Semantischer Vergleich von Hypothesen und Transkriptionen

7.4.4 Ergebnisse

In Tabelle 7.9 ist eine Übersicht der Ergebnisse der vorherigen Abschnitte zu sehen. In den

[5] Im Beispiel ist die Semantik für natürlichsprachliche Dialogsysteme korrekt. Kommt Gestik als Modalität hinzu, muß zwischen einer blauen und einer *bestimmten* blauen Tasse unterschieden werden.

	WER	UER	CER	SER
Initial	62,2%	78,3%	–	–
Adaptierte Grammatik	49,1%	65,4%	–	–
In-Domain	37,6%	57,3%	34,9%	46,6%

Tabelle 7.9: Übersicht der Analyseergebnisse

Spalten sind der Reihe nach die Wortfehlerrate (WER), Satzfehlerrate (*utterance error rate*, UER), die Konzeptfehlerrate (*concept error rate*, CER) und die Semantikfehlerrate (SER) notiert.
Die erste Zeile gibt die initialen Werte an. Zeile zwei zeigt die Fehlerkorrekturen in den Transkriptionen. In der letzten Zeile sind die Werte ohne „out of domain"-Äußerungen angegeben. Die Werte zeigen, daß die Leistung des Dialogsystems mit Hilfe der Daten eines „Wizard of Oz"-Experimentes verbessert werden kann und das System an eine leicht veränderte Domäne angepaßt werden kann. Werden die Analysen in einen Entwicklungsprozeß für Dialogsysteme integriert, kann eine iterative, systematische Verbesserung der Systemleistung erreicht werden.

7.5 Abschließende Evaluation

Die abschließende Evaluation wurde als „proof of concept"auf dem Roboter ARMAR II in Karlsruhe durchgeführt. Das gewählte Szenario wurde in Abschnitt 4.2 bereits beschrieben. Es wurde übernommen, da es sich um ein Demo-Szenario für den SFB 588 Humanoide Roboter handelt. Der Roboter konnte zum Zeitpunkt des Experimentes weder den Benutzer detektieren noch zu ihm hinfahren. Er konnte auch nicht signalisieren, zu welchem Benutzer er spricht, z.B. durch Zuwenden. Diese Fakten sprachen gegen das Szenario des „Wizard of Oz"-Experimentes mit zwei Benutzern.

7.5.1 Zielsetzung

Die abschließende Evaluation wurde durchgeführt um nachzuweisen, daß die Dialogstruktur zur Lösung der in Abschnitt 5.4 erörterten Probleme beitragen kann. Exemplarisch wurde die Auflösung von systemdetektierten Fehlersituationen herangezogen. Bei Durchführung der Experimente war nur eine Fehlersituation vom System detektierbar: In Schritt „Tasse suchen"der Ablaufsteuerung (vgl. Abschnitt 4.2.1) kann die visuelle Komponente die Tasse nicht erkennen. Andere Fehlersituationen hätten nur durch einen „Timeout"als solche erkannt werden können. Dies hätte einen Zeitraum zur Folge, in dem der Roboter keinerlei beobachtbares Verhalten bzw. Antwortverhalten, zeigt. Die Benutzer können in diesem Zeitraum sehr unvorhersehbar reagieren, daher wurde auf dieses Vorgehen verzichtet.

7.5.2 Aufbau

Die Benutzerervaluation wurde in der Demo-Küche des SFB 588 durchgeführt. Wieder standen dem Benutzer drei farblich unterschiedliche Tassen zur Auswahl. Sie standen alle auf einem seitlichen Board and der Wand. Abbildung 7.12 zeigt eine schematische Darstellung.

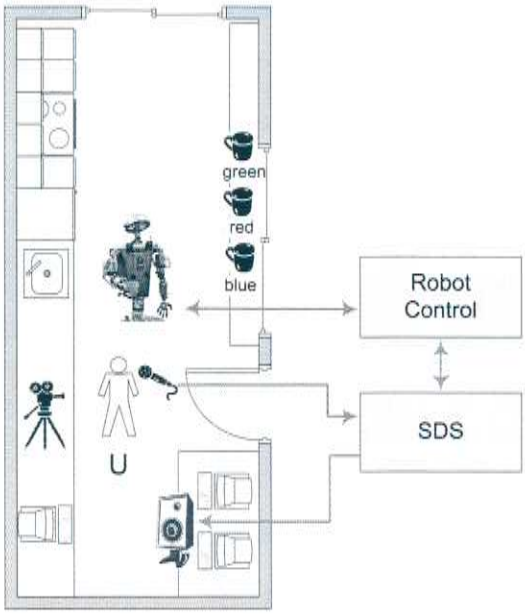

Abbildung 7.12: Aufbau für die abschließende Evaluation

Zur Steuerung waren zwei Laptops im Einsatz. Auf dem ersten lief die Robotersteuerung; er war direkt mit dem Roboter über ein Firewire-Kabel und Netzwerkkabel verbunden. Auf dem zweiten Laptop lief das komplette Dialogsystem, die Ablaufsteuerung (vgl. Abschnitt 4.5), sowie eine Kommunikationsplattform. Über die Plattform tauschten die einzelnen Softwarekomponenten Nachrichten aus.
Wie auch schon beim „Wizard of Oz"-Experiment war der Benutzer über ein Headsetmikrophon und Funk mit dem Dialogsystem verbunden. Die Benutzeräußerungen wurden direkt an den Spracherkenner Janus gesendet und nach dem Segmentieren gespeichert.
Zusätzlich zum Closetalk-Signal wurde auch ein Stereo-Distant-Speech-Signal aufgezeichnet. Dies diente vor allem der Datensammlung von Sprache im authentischen Umfeld des Roboters für Zwecke außerhalb des Rahmens dieser Arbeit. Der Spracherkenner soll an die Betriebsgeräusche des Roboters adaptiert werden.
Das gesamte Experiment wurde, zur leichteren Analyse, per Videokamera aufgezeichnet.

7.5.3 Durchführung

Insgesamt waren 10 Teilnehmer am Experiment involviert, jeder absolvierte zwei bis fünf Durchgänge. Wie beim „Wizard of Oz"-Experiment wurden alle Dialoge in Englisch geführt. Außer-

dem wurden die Probanden vor Versuchsbeginn wieder schriftlich instruiert, um ein einheitliches Verständnis der Aufgabe zu schaffen (vgl. Anhang A).
Es war nicht notwendig, die Dialoge durch Vorgaben zu beeinflussen; Fehlersituationen kamen sehr häufig zustande, sodaß es nicht notwendig war, diese zu simulieren. Das lag vor allem an den Limitierungen des Roboters (vgl. Abschnitt 4.4). Zum Beispiel mußte der Weg des Roboters zu den Tassen fest und relativ zur Startposition einprogrammiert werden. Dadurch konnte das visuelle Subsystem sehr oft die Tassen nicht detektieren.
Da auch eine "echte" Übergabe der Tasse an den Benutzer nicht möglich war, galt die gestellte Aufgabe als erfüllt, wenn der Roboter mit der Tasse in Reichweite des Benutzers stand. Ein Durchgang – und damit ein Dialog – war also mit der Übergabe der Tasse oder einem Abbruch beendet. Wenn eine verfahrene Situation so aussichtslos war, daß eine Übergabe nicht mehr möglich war, wurde der Durchgang abgebrochen. Meistens mußte der Roboter mittels "Not-Aus" abgeschalten werden, um eine Kollision mit der Umwelt zu vermeiden. Der Roboter verfügte nur durch die Plattform über eine Kollisionsdetektion, sodaß die Arme durchaus gefährdet waren. Der zweite Grund für einen Abbruch lag dann an der Kollisitonsdetektion der Plattform (vgl. Abschnitt 4.2.2).

7.5.4 Ergebnisse

Die zehn Probanden hatten, bis auf einen, keine Erfahrung mit natürlichsprachlichen Dialogsystemen oder dem Roboter. Es wurden 34 Dialoge mit insgesamt 369 Äußerungen und 1533 Wörtern Durchgeführt. Zunächst wurde die Spracherkennerleistung analsysiert, wie auch schon bei den Analysen zu Ontologie und Grammatik des „Wizard of Oz"-Experimentes geschehen (vgl. Abschnitt 7.4). Tabelle 7.10 zeit die errechneten Werte über alle Äußerungen. Tabelle 7.11 zeigt die errechneten Werte ohne „out of domain"-Äußerungen. Die Leistung der semantischen Interpretation der Benutzeräußerungen ist in Tabelle 7.12 zu sehen. Die verwendeten Metriken sind in Abschnitt 7.1 beschrieben.

Accurracy:	59,95%	Errors:	678	(396, 64, 218)	
WER:	44,23%	Words:	1533	Matches:	919
Sent.Acc.:	36,60%	Sent.:	369	Matches:	135

Tabelle 7.10: Vergleich von Hypothesen und Transkriptionen

Accurracy:	61,49%	Errors:	623	(367, 53, 203)	
WER:	42,10%	Words:	1480	Matches:	910
Sent.Acc.:	38,03%	Sent.:	355	Matches:	135

Tabelle 7.11: Vergleich von Hypothesen und Transkriptionen ohne „out of domain" Äußerungen

Im Vergleich zum „Wizard of Oz"-Experiment fiel die Spracherkennerleistung in der Evaluation schwächer aus (vgl. Tabelle 7.13). Für die Wortfehlerrate kann eine relative Verschlechterung von 10,7% und für die Konzeptfehlerrate von 3,3% festgestellt werden.

	mit OOD		ohne OOD	
Korrekter Dialog-Akt:	61,5%	227	63,9%	227
Korrekte Semantik:	46,6%	172	48,5%	172
Äußerungen gesamt:		369		355

Tabelle 7.12: Semantischer Vergleich von Hypothesen und Transkriptionen

Für beide Experimente wurde die gleiche Grammatik verwendet. In beiden Experiemten war die Muttersprache fast aller Probanden nicht Englisch. Ausnahme war ein Proband im „Wizard of Oz"-Experiment. Auch wurden in beiden Experimenten baugleiche Mikrophone verwendet. Ein Grund könnte in der Segmentierung des Sprachsignals liegen: Die Daten des „Wizard of Oz"-Experimentes wurden per Hand segmentiert, in der Evaluation wurde online ein automatischer Segmentierer verwendet.
Der wesentlich Grund dürfte aber in einer falschen Gewichtung von Grammatikregeln während der Evaluation liegen: Es wurden Grammatikregeln aktiviert, wie in Abschnitt 3.3 beschrieben. Insgesamt waren für die 355 Äußerungen in 167 Fällen die falschen Regeln aktiviert. Das entspricht einer Fehlerrate von 47% für die Aktivierung. Eine Gewichtung der Grammatikregeln sollte also folgerichtig nur dann vorgenommen werden, wenn die nächste Benutzeräußerung hinreichend zuverlässig vorhergesagt werden kann.
Interessant ist der Vergleich der „out of domain"-Äußerungen; hier fällt der Anteil in der Evaluation deutlich niedriger aus als im „Wizard of Oz"-Experiment (vgl. Tabelle 7.13). Die Probanden haben sich offensichtlich recht stark daran orientiert, daß sie es mit einer menschlichen Person zu tun haben, die die Rolle des Roboters im „Wizard of Oz"-Experiment übernommen hatte. Teilweise haben sie sich zu fast philosophischen Äußerungen hinreißen lassen (vgl. Dialog 5 in Anhang C). Die Erwartungshaltung bzgl. der Systemleistung orientierte sich bei der Evaluation dagegen stark an der vorgegebenen Domäne und war dadurch deutlich herabgesetzt.

	WOz	Eval.
WER (ohne OOD):	37,6%	42,1%
CER (ohne OOD):	34,9%	36,1%
OOD Äußerungen:	14,7%	3,8%

Tabelle 7.13: Vergleich des Wizard of Oz Experimentes mit der Evaluation

Tabelle 7.14 zeigt die Rahmendaten des Experimentes aufgeschlüsselt nach Probanden. Es fällt auf, daß bei den Probanden 2, 9 und 10 die ASR-Leistung mit eine CER von über 50% sehr schlecht ist. Da das ASR-Subsystem nicht Thema dieser Arbeit ist, können über die Ursachen nur Vermutungen angestellt werden. Bei den Probanden 2 und 10 kann leise Aussprache ein Grund gewesen sein. Leider konnten mit dem vorhandenen Versuchsaufbau die Audio-Signale nicht optimal für jeden Probanden ausgepegelt werden. Bei dem anderen Proband war mögli-

cherweise ein ausgeprägter Akzent ausschlaggebend.

Proband:	1	2	3	4	5	6	7	8	9	10	Σ
Dialoge:	3	2	3	2	2	4	4	5	5	4	34
mittl. Dialoglänge:	15,7	8,0	19,0	9,0	11,0	9,5	7,3	7,0	9,0	15,5	10,9
erf. Dialoge (Tasse):	1	0	2	1	1	2	4	2	1	0	14
mittl. Erfolg (Tasse):	0,4	0,0	0,7	0,5	0,5	0,5	1,0	0,4	0,2	0,0	0,4
erf. Dialoge:	1	0	2	1	1	2	2	1	0	0	10
mittl. Erfolg:	0,4	0,0	0,7	0,5	0,5	0,5	0,5	0,2	0,0	0,0	0,3
CER (in %):	25,5	56,3	33,4	33,4	18,2	34,2	27,6	25,7	52,6	72,6	38,8

Tabelle 7.14: Rahmendaten der Evaluation nach Probanden aufgeschlüsselt

Von den insgesamt 34 Dialogen konnten nur 10 erfolgreich abgeschlossen werden (vgl. Tabelle 7.15). Als „erfolgreich" gelten Dialoge, an deren Ende der Benutzer die gewünschte Tasse in den Händen hielt.

Anzahl Dialoge	34	100,0%
Erfolgreiche Dialoge	10	29,4%
Abbruch durch Hardware	15	44,1%
Abbruch durch SDS	9	26,5%

Tabelle 7.15: Erfolgsbilanz der Dialoge

Für Abbrüche, die auf die Hardware zurückzuführen sind, gab es folgende Gründe:

Drohende Kollision: Der Durchlauf wurde durch Drücken des Not-Aus-Knopfes am Roboter abgebrochen, da eine Kollision von Teilen des Roboters mit der Umwelt drohte. Das konnte z.B. der Arm mit der Arbeitsfläche sein.

Stillstand der Plattform: Der Roboter ließ sich durch kein Kommando mehr bewegen. Die war meist der Fall, wenn die Plattform keinen kollisionsfreien Weg mehr berechnen konnte.

Fehlfunktion des Kopfes: Der Kopf hat sich nicht bewegt, dadurch konnte die gewünschte Tasse nicht detektiert werden.

Die folgenden Gründe führten zu Abbrüchen, die dem Dialogsystem zuzuschreiben waren:

Benutzerabbruch: Der Benutzer hat den Dialog abgebrochen.

Abbruch durch den Versuchsleiter: Der Versuchsleiter hat den Dialog abgebrochen, weil sich das System in einer Situation befand, die nicht mehr zum Ziel führen konnte.

Fehler der Spracherkennung: In einem Fall führte ein Fehler der Spracherkennung dazu, daß der Roboter die Tasse fallen ließ. Hier wäre eigentlich eine übergeordnete Kognitionsebene schuld, die das Fehlverhalten hätte verhindern müssen. Eine solche Ebene gibt es (noch) nicht.

Hauptverantwortlich für die hardwarebedingten Abbrüche war das Fehlen einer geeigneten Strategie zur Kollisionsvermeidung auf der Seite des Roboters (vgl. Abschnitt 4.4). Insgesamt 9 der 15 Abbrüche gingen auf dieses Konto. Dies wirkte sich auch sehr stark auf einen erfolgreichen Abschluß eines Dialoges aus. Da in einigen Fällen zumindest die gewünschte Tasse detektiert werden konnte, der Roboter also in der Lage gewesen wäre, die Tasse zu greifen, wurde dies als Teilerfolg akzeptiert. hardwarebedingt mußte der Vorgang des Greifens simuliert werden (vgl. Abschnitt 4.4).

Tabelle 7.16 vergleicht die Erfolgsraten der Dialoge bei Einsatz der Dialogstruktur gegenüber den Erfolgsraten, falls die Dialogstruktur nicht verwendet worden wäre. In der ersten Zeile sind die komplett und erfolgreich durchgeführten Dialoge angegeben. In der zweiten Zeile sind diejenigen Dialoge angegeben, die wenigstens zu einer erfolgreichen Detektion der gewünschten Tasse führten.

	mit Struktur		ohne Struktur	
erf. Dialoge	10	29,4%	1	2,9%
detekt. Tasse	14	41,2%	4	11,8%
Dialoge gesamt	34	100%	34	100%

Tabelle 7.16: Vergleich der Dialogleistung mit und ohne Dialogstruktur

Tabelle 7.17 gibt die gleichen Werte wie Tabelle 7.16 an, jedoch ohne Dialoge, deren Abbruch durch die Hardware herbeigeführt wurde.

	mit Struktur		ohne Struktur	
erf. Dialoge	10	52,6%	1	5,3%
detekt. Tasse	11	57,9%	4	21,1%
Dialoge gesamt	19	100%	19	100%

Tabelle 7.17: Vergleich der Dialogleistung ohne hardwarebedingte Abbrüche

Anhand dieser Zahlen läßt sich eine Relative Verbesserung durch die Dialogstruktur von 89,9% im Falle der komplett erfolgreichen Dialoge feststellen. Für die Dialoge mit wenigstens detektierter Tasse läßt sich immerhin noch eine Verbesserung von 63,6% verzeichnen.
Durch die Dialogstruktur konnte die Dialogsteuerung Entscheidungen über den Dialogverlauf treffen, die über den Rahmen eines Dialogzieles hinausgingen. War der aktuelle Dialogzielzustand $(BringSomething, selected)$ oder $(BringSomething, finalized)$, wurden die entsprechenden Dialogschritte bzw. das Dialogziel ausgeführt. War der aktuelle Dialogzielzustand $(BringSomething, aborted)$, konnte die Fehlersituation genauer spezifiziert werden und eine entsprechende Fehlermeldung an den Benutzer weitergegeben werden. Im Falle von $(GoDirection, executed)$ oder $(Turn, executed)$ konnte die Dialogsteuerung entschieden, ob die Ablaufsteuerung mit ihrem Auftrag, ein Objekt zu bringen, fortfahren sollte. Praktisch hat die Dialogsteuerung hier die Entscheidungen getroffen, wann ein Subdialog zur Behebung einer Fehlersituation eingeleitet und wann er abgeschlossen wird.

In den meisten Fällen liefen die Dialoge nach dem gleichen Schema ab: Der Benutzer begrüßte den Roboter und fragte ihn anschließend nach einer Tasse. Der Roboter fuhr an eine Position in der Näher der Tassen und versuchte sie zu detektieren. Der Benutzer dirigierte ihn näher zur Tasse, weil sie meist nicht erkannt wurde. Der Grund dafür war, daß sich die Tassen nicht im Blickfeld des Roboters befanden. Er befand sich entweder zu weit weg oder in einem falschem Winkel zur Tasse. Die Plattform hatte die Eigenschaft nach Drehungen leicht weiter zu driften, sodaß die Sollwinkel meist nicht erreicht wurden. Nach erfolgreicher Korrektur und Detektion trug der Roboter durch simuliertes Greifen[6] die Tasse auf der Hand. Danach fuhr er zum Benutzer zurück. Oft mußte der Benutzer dabei auch helfend eingreifen, da der Roboter nicht den vorgegebenen Weg zum Benutzer zurück fuhr. Der Grund dafür waren Versorgungskabel, die die Plattform als Hindernis erkannte. Die Dialoge 1 und 4 im Anhang C sind Beispiele dafür.

Abgesehen von Abbrüchen, die durch die Benutzer oder den Versuchsleiter vorgenommen wurden[7], konnte in einem Fall die Dialogstruktur den Dialogabbruch nicht verhindern. Der Grund lag in einem fatalen Fehler der Spracherkennung, wie in Abbildung 7.13 zu sehen ist (Äußerung U6). Das ASR-Subsystem hat statt *"Step forward"* verstanden *"That's it, robot "*. Zu diesem Zeitpunkt waren die Grammatik-Regeln ⟨robbi:obj_cup,N,_⟩ gewichtet. Der komplette Dialog ist in Anhang C als Dialog 2 zu finden.

	...
U6:	"Turn twenty degrees to the left."
R6:	"Turning twenty degrees to the left."
U6:	"Step forward."
R6:	"Good bye."
	Roboter winkt und läßt dabei Tasse fallen.

Abbildung 7.13: ASR-Fehler führt zu Tassenverlust, Ausschnitt aus Dialog 2

Einerseits handelt es sich hier um einen Fehler des ASR-Subsystems, der vielleicht auf zwei Arten hätte verhindert bzw. ausgeglichen werden können: Erstens durch eine korrekte Gewichtung der Grammatik-Regeln. Zweitens durch die Überlegung, daß ein Übergang der Form

$$(GoDirection, executed) \rightarrow (Thanks, finalized)$$

wahrscheinlicher ist als

$$(GoDirection, executed) \rightarrow (GoodBye, finalized)$$

Andererseits hätte sich der Roboter quasi dessen bewußt sein müssen, daß er noch eine Tasse in der Hand hält und deswegen nicht winken darf. Dieser Fall zeigt deutlich, daß dem Roboter eine übergeordnete Kognitions- und Planungsebene fehlt, die die Verantwortung über das Zusammenspiel aller Subsysteme übernimmt.

[6] Der Versuchsleiter hat die Tasse auf die ausgestreckte Hand des Roboters gestellt.

[7] Die Abbrüche wurden beispielsweise vorgenommen, weil eine schlechte Spracherkennerleistung keine sinnvolle Möglichkeit der Fortführung des Dialoges vermuten ließ.

Insgesamt konnte in nur einem einzigen Dialog das Szenario ohne Fehlersituation absolviert werden (vgl. Abbildung 7.14). Abgesehen von den Höflichkeitsfloskeln ist der Benutzer hier

U1:	"Hello Robbi."
R1:	"Hello, what can I do for you?"
U2:	"Bring me the green cup, please."
R2:	"Going to to bring you the green cup."
U3:	"Thank you, Robbi."
R3:	"You are welcome."

Abbildung 7.14: Dialog ohne Fehlersituation

mit einer minimalen Anzahl von Äußerungen ausgekommen. In allen anderen Fällen, also 97,1% bzw. 94,7%[8], traten Fehlersituationen auf. Die Benutzer hatten in einem solchen Fall den Auftrag, den Roboter mit weniger komplexen Aufträgen aus der Situation zu dirigieren. Ohne die Dialogstruktur wäre ein „Zurückschalten" aus diesem kommandoähnlichen Modus nicht möglich gewesen. Darum hätten die Dialoge nur in dieser weniger benutzerfreundlichen Art, nämlich Schritt für Schritt, erfolgreich abgeschlossen werden können.

[8] Alle Dialoge bzw. Dialoge ohne Hardware-bedingte Abbrüche

Kapitel 8

Zusammenfassung und Ausblick

8.1 Beiträge

Die wesentlichen Beiträge dieser Arbeit sind eine übergeordnete Dialogstruktur, die bisherigen Einschränkungen im Dialogverlauf entgegen wirkt, und ein Entwicklungsprozess für Dialogsysteme, die diese Struktur nutzen.

8.1.1 Dialogstruktur

Die bisher im Basissystem verwendeten Konzepte und Strukturen erlauben die Steuerung eines natürlichsprachlichen Dialoges nur in relativ kleinen Sinnzusammenhängen. Diese Sinnzusammenhänge werden jeweils durch ein Dialogziel beschränkt. Verläßt der Dialogverlauf einen Rahmen und tritt in einen anderen ein, findet automatisch ein Kontextwechsel statt. Daher wurde eine übergeordnete Dialogstruktur entwickelt, die Entscheidungen über den Dialogverlauf auf einer höheren Kontextebene und damit über die Rahmen der Dialogziele hinaus ermöglicht.
Dazu wurden zunächst die Schwachstellen des Basissystems analysiert (vgl. Abschnitt 5.4). Diese resultieren hauptsächlich aus der eben genannten Einschränkung. Als Symbole der übergeordneten Struktur wurden Dialogzielzustände gewählt. Der Zustandsraum der Dialogziele wurde erweitert, um den Verlauf der Verarbeitung eines Dialogzieles vollständig abbilden zu können (vgl. Abschnitt 6.1).
Die Dialogstruktur kann als gerichteter Graph aufgefaßt werden: Die Dialogzielzustände werden durch Knoten repräsentiert, die Transitionen zwischen den Dialogzielzuständen werden durch Kanten mit Übergangswahrscheinlichkeiten repräsentiert. Für jeden Dialogzielzustand wird eine Wahrscheinlichkeitsverteilung über die ausgehenden Transitionen berechnet, die wiederum aus Beispieldaten mit Methoden des Maschinellen Lernens ermittelt wird. Die Beispieldaten wurden durch ein „Wizard of Oz"-Experiment gewonnen (vgl. Abschnitt 7.2). Sie dienten auch als Datenkorpus für eine Analyse, die verschiedene Klassifikatoren miteinander verglich. Es wurden verschiedene Konzepte des Dialogsystems als Eigenschaften zur Vorhersage des Dialogverlaufes getestet (vgl. Abschnitt 7.3). Dazu wurden 1R-, Bayes-Netzwerk- und N-Gram-Klassifikatoren trainiert und deren Leistung getestet. Die ersten beiden genannten

schnitten dabei meist akzeptabel ab, wenn man die geringe Größe des Trainingskorpus berücksichtigt. Nicht zu gebrauchen waren die N-Gram-Klassifikatoren. Bei ihnen handelte es sich um Bigram-Modelle, da für höhere N zu wenig Trainingsdaten zur Verfügung standen.
Die Dialogstruktur wurde in einer Benutzerevaluation auf dem humanoiden Roboter ARMAR II getestet (vgl. Abschnitt 7.5). Aufgrund der technischen Einschränkungen des Roboter-Systems wurde exemplarisch der Nutzen der Struktur in Fehlersituationen beurteilt. Es konnte festgestellt werden, daß die meisten Dialoge ohne Einsatz der Dialogstruktur nicht zum dem, vom Benutzer gewünschten, Resultat geführt hätten.

8.1.2 Entwicklungsprozeß

Es konnte gezeigt werden, daß die übergeordnete Dialogstruktur zur Verbesserung des Dialogverlaufes beiträgt. Um ein reproduzierbares Ergebnis bei der Entwicklung von Dialogsystemen zu erhalten, wurde ein Entwicklungsprozeß vorgeschlagen (vgl. Abschnitt 6.5). Dieser unterscheidet sich insbesondere durch das iterative Vorgehen von den durch Matthias Denecke vorgeschlagenen acht Schritten für die Implementierung eines schnellen, prototypischen Dialogsystems (vgl. [Den02]). Denecke sieht für jede Quelle des Dialogsystems, wie die Deklaration der verfügbaren Dienste, die Datenbank, die Ontologie, die Grammatik und ähnliches, einen Schritt vor[1]. Die Erfahrung hat aber gezeigt, daß eine vollständige Spezifizierung des Dialogsystems mit all seinen linguistischen und sonstigen Quellen in einer Iteration nicht möglich ist.
Der in dieser Arbeit vorgelegte Entwicklungsprozeß ist zwar auch für einen schnellen prototypischen Entwurf gedacht, orientiert sich aber mehr an realem Benutzerverhalten in der gegebenen Domäne. Die erste Iteration läuft grob nach einem Schema der Form: Analyse, Design, Implementierung, Evaluation ab. Weitere Iterationen enthalten die Schritte Adaption und Evaluation. Die Anzahl der weiteren Interationen wird durch die Ergebnisse der jeweils vorangegangenen Evaluation bestimmt.
Tatsächlich konnte die Leistung des Basissystems durch die Analysen auf den Daten des „Wizard of Oz"-Experimentes und die anschließende Adaption wesentlich verbessert werden. Die Wortfehlerrate wurde von 62,2% auf 49,1% reduziert. Dies entspricht einer relativen Verbesserung von 21,1%. Die Satzfehlerrate betrug vor der Adaption 78,3%, danach 65,4%. Das ist immerhin noch eine relative Verbesserung von 16,5%.

8.2 Diskussion

Die Experimente haben gezeigt, daß die Interaktion zwischen Mensch und Roboter wesentlich enger verzahnt ist, als bei Informationssystemen. Ein Grund hierfür ist sicher, daß der Benutzer die Aktionen des Roboters verfolgen kann und in der Regel sofort eingreift wenn sich abzeichnet, daß das resultierende Ergebnis nicht in seinem Sinne ist. Bei Informationssystemen kann der Benutzer nur das Endergebnis einer Aktion beobachten.
Auch ohne Fehlersituationen ist die natürlichsprachliche Kommunikation sehr komplex. Schon um einen einfachen Auftrag – wie eine bestimmte Tasse zu bringen – zu erteilen, wird in fast

[1] Alle Schritte zusammen bilden letztendlich eine einzige Iteration.

allen Fällen mehr als ein zu bearbeitendes Dialogziel benötigt. Die Benutzer verhalten sich gegenüber dem Robotersystem so höflich wie bei einem menschlichen Gegenüber. Im Idealfall werden daher drei Dialogziele bearbeitet: Eines zur Begrüßung, möglicherweise auch, um die Aufmerksamkeit des Systems für sich sicherzustellen. Danach wird das Dialogziel bearbeitet, das den Auftrag repräsentiert. Zum Schluß bedanken sich die Benutzer (vgl. Dialog 3 in Anhang C). Die drei Dialogziele repräsentieren auch gleichzeitig drei Partitionen, die den gesamten Dialogverlauf unterteilen: Einleitung, Hauptteil und Dialogende (vgl. Abschnitt 6.2). Der Auftrag „Tasse holen" muß seitens des Robotersystems in mehreren Teilschritten ausgeführt werden, in denen Fehlersituationen auftreten können. Sind dies Situationen, die mit Unterstützung durch den Benutzer aufgelöst werden können, muß der Dialogmanager durch eine entsprechende Rückmeldung einen Subdialog initiieren. Der Benutzer kann dann mit einfachen Aufträgen den Roboter dirigieren. Diese Aufträge werden auch durch Dialogziele repräsentiert. Es findet also ein Kontextwechsel auf der Ebene der Dialogziele statt. Dennoch gehören die bearbeiteten Dialogziele auf einer höheren Kontextebene zusammen: Sie dienen letztendlich dazu, den Auftrag „Tassse holen" erfolgreich abzuschließen. Im Basissystem fehlt eine solche Kontextebene.

	mit Struktur		ohne Struktur	
erf. Dialoge	10	52,6%	1	5,3%
detekt. Tasse	11	57,9%	4	21,1%
Dialoge gesamt	19	100%	19	100%

Tabelle 8.1: Vergleich der Dialogleistung ohne Hardware-bedingte Abbrüche

Tabelle 8.1 zeigt neben der Anzahl der erfolgreich abgeschlossenen Dialoge[2] auch die Anzahl der Dialoge, in denen die gewünschte Tasse zumindest detektiert werden konnte. Die Dialoge wurden im Rahmen der Benutzerevaluation geführt (vgl. Abschnitt 7.5). Die Dialogstruktur führt zu einer relativen Verbesserung von 89,9% für die erfolgreich abgeschlossenen Dialoge. Für das Detektieren der Tasse wird eine relative Verbesserung von 63,6% erreicht. In nur einem Dialog hätte ohne die Dialogstruktur der Dialog erfolgreich abgeschlossen werden können. Dies lag daran, daß keine Fehlersituation aufgetreten ist (vgl. Dialog 3 in Anhang C). In der Tabelle werden diejenigen Dialoge nicht berücksichtigt, die durch hardwarebedingte Abbrüche nicht abgeschlossen werden konnten. Die entsprechenden Fehlersituationen konnten nicht an das Dialogsystem gemeldet werden oder waren auch mit der Unterstützung des Benutzers nicht lösbar. Dies war beispielsweise der Fall, wenn sich die Plattform nicht mehr bewegte, weil die Kollisionsdetektion es nicht zuließ.
Im Falle der Fehlersituationen liegt der Nutzen der Dialogstruktur in der Entscheidung, wann ein Subdialog initiiert werden soll, welche Dialogziele noch zum Kontext des Gesamtauftrages gehören und wann er wieder ohne Hilfe des Benutzers fortgeführt werden kann. Die Interaktion wechselt von einer auftragsorientierten zu einer schrittweise dirigierenden und wieder zurück. Stünde nur die erste Interaktionsform zur Verfügung, wäre ein erfolgreicher Abschluß eines Auftrages in den meisten Fällen nicht möglich. Andererseits wäre eine reine, schrittweise dirigierende Interaktionsform für den Benutzer zu umständlich.

[2]Der Benutzer hält die gewünschte Tasse in der Hand.

8.3 Ausblick

Durch die hardwarebedingten Einschränkungen konnte die Dialogstruktur nicht in einem vollständig realen Szenario evaluiert werden. Beispielsweise konnte nur eine Fehlersituation berücksichtigt werden. Daher wäre eine Evaluation mit einem vollständig funktionsfähigen Roboter interessant. Mit dieser Evaluation geht auch die Möglichkeit einher, den Datenkorpus für die Trainingsdaten der Dialogstruktur zu vergrößern. Damit wäre eine größere Robustheit erreicht.
Für die Entscheidungen über die Aktionen des Dialogmanagers an einem bestimmten Punkt im Dialogverlauf wurde nur der aktuelle Zustand des Dialoges berücksichtigt. In weiteren Experimenten könnte evaluiert werden, inwiefern eine größere Historie die Entscheidungen beeinflußt.
Auch werden die Entscheidungen nur anhand der lokalen Wahrscheinlichkeitsverteilungen getroffen. Daher wird nur lokal die beste Aktion ausgewählt, nicht aber in einem größeren Kontext. Mit Hilfe eines A^*-Algorithmus wäre es beispielsweise möglich, den besten Pfad durch die Dialogstruktur vom aktuellen Zustand zum Dialogabschluß zu berücksichtigen. An dieser Stelle wäre der Einsatz von Reinforcement-Learning denkbar, da hier nach der besten Aktion zu einem bestimmten Zeitpunkt bzgl. eines Gesamtergebnisses gesucht wird (vgl. [Pro06] und Abschnitt 3.2). Insgesamt könnte dadurch die Robustheit des Systems weiter erhöht werden.
Auch kann möglicherweise ein vom Benutzer beabsichtigter Kontextwechsel detektiert werden. Der Dialogmanager kann sich dann über Klärungsfragen den Wechsel bestätigen lassen. Die Dialogstruktur würde dann als Alternative zu den in [Kru05] und [KHW05] vorgeschlagenen Triggern dienen.
Weiterhin bietet die Dialogstruktur die Möglichkeit, über den Einsatz des Systems dynamisch adaptiert zu werden, da sie auf statistischen Modellen beruht. Mit Mitteln des Basissystems ist dies nicht möglich, da Entscheidungen über den Dialogverlauf deklarativ festgelegt ist.

Anhang A

Versuchsdokumente

Die Probanden der beiden Benutzer-Experimente wurden jeweils vor einem Durchgang schriftlich über zu ihrer Aufgabe instruiert. Dies hatte den Zweck, eine definierte, gemeinsame Basis zu schaffen. Für das „Wizard of Oz"-Experiment wurden für die jeweiligen Benutzerrollen die Dokumente aus den Abbildungen A.1 und A.2 erstellt. In der abschließenden Evaluation wurde das Dokument aus Abbildung A.3 verwendet.

Pilot study for the Cup-Setting

Participant: _____
Role: Interfering User

You are participating in this experiment as an *interfering user*. We have a room with a robot and three cups located on two tables.
You are already in the room. Please follow the pattern of behavior as marked below:

A) Just stay where you are and wait until the robot does address you.

B) Interrupt the interaction between user and robot once, whenever you want. You may do this by just saying "hello" to the robot or by assigning the robot to bring you a cup.

In any case please interact with the system, in the way that you think is appropriate.

Abbildung A.1: Instruktionen an den Benutzer für das Wizard-of-Oz Experiment

Pilot study for the Cup-Setting

Participant: _____
Role: User

You are participating in this experiment as a *user*. We have a room with a robot and three cups located on two tables. As a user, your desire is to get a cup served by the robot.

Please enter the room and ask the robot for a cup. You may choose any cup and specifiy it for example as follows:

- By it's color,
- Indirectly by it's relative position (e.g. "... on the left Table", "... the left cup") or
- As "... my cup", "my favorite cup".

Please interact with the robot, if he does not satisfy your assignments.
Now, have fun and thank you very much for your participation.

Abbildung A.2: Instruktionen an den unterbrechenden Benutzer für das Wizard-of-Oz Experiment

Evaluation zur Diplomarbeit
„Statistische Modellierung des Dialogverlaufs in natürlichsprachlichen Dialogen"

Sehr geehrter Teilnehmer,

vielen Dank, Daß Sie sich diesem Benutzertest zu Verfügung stellen.

Aufgabenbeschreibung:
Ihre Aufgabe besteht darin, sich von ARMAR, dem Haushaltsroboter, eine der drei Tassen bringen zu lassen.
Sprechen Sie mit ARMAR ihrer Intuition entsprechend.
Sollte ein Fehler auftreten, oder ARMAR Hilfe benötigen, versuchen Sie bitte, ihn mit einfachen Anweisungen zu dirigieren, um die Situation zu lösen. ARMAR kann Vorwärts und zur Seite fahren. Außerdem kann er den Kopf drehen.

ARMAR versteht in diesem Experiment nur Englisch.

Hier einige Beispielsätze, die ARMAR versteht:
„Hello Robbi"
„Bring me the blue cup"
„I want the red cup"
„Move one step to the right"
„Look left"

Dialog-Nr.	*Fehlersituation*

SFB 588 Humanoide Roboter - Lernende und kooperierende multimodale Roboter

Abbildung A.3: Instruktionen für die abschließende Evaluation

Anhang B

Dialogziele

Einige der Dialogziele der zugrundeliegenden Applikation werden hier exemplarisch zusammenfaßt und erläutert. Sie sind nach unterschiedlichen Aufgaben unterteilt. Es werden die erforderlichen Informationen aufgelistet. Außerdem werden eine oder mehrere Äußerungen als Beispiel gegeben, die mindestens zur Selektion des betreffenden Dialogziels führt.

B.1 Robotersteuerung

Die folgenden Dialogziele stellen die eigentlichen Fuktionalitäten der Applikation dar.

BringSomething Der Roboter bringt dem Benutzer ein Objekt.
Parameter: Spezifikation des Objektes
Äußerung: *"Bring me the red cup."*
Dialog-Akt: act_bring

BringFavorite Der Roboter bringt dem Benutzer seine Lieblingstasse. Dieses Dialogziel wird nur im „Wizard of Oz"-Experiment verwendet, daher die Beschränkung auf eine Tasse.
Parameter: Richtung und Umfang in Grad.
Äußerung: *"Turn to the left."*

GoDirection Der Roboter fährt in eine Richtung. Zum Zeitpunkt der Versuchsdurchführung drehte sich der Roboter zuerst in die angegebene Richtung und fuhr dann die angegebene Anzahl Schritte (ca. 0,5m)
Parameter: Richtung und Anzahl der Schritte.
Äußerung: *"Move forward."*

Turn Der Roboter dreht sich komplett in die angegebene Richtung.
Parameter: Richtung und Umfang in Grad.
Äußerung: *"Turn to the left."*

Look Der Roboter dreht seinen Kopf in die angegebene Richtung.
Parameter: Richtung und Umfang in Grad.
Äußerung: *"Look to the left."*

B.2 Konversation

Die folgenden Dialogziele dienen hauptsächlich der Konversation und unterstützen die Natürlichkeit des Dialoges.

Hello Begrüßung des Roboters. Das System antwortet und fragt den Benutzer nach seinem Begehren.
Parameter: keine
Äußerung: *"Hello Robbi."*

SayThanks Der Roboter bedankt sich für ein Lob.
Parameter: keine
Äußerung: *"Good work."*

HowAreYou Der Roboter teilt dem Benutzer sein Befinden mit.
Parameter: keine
Äußerung: *"How are you?"*

Anhang C

Dialoge

Hier sind einige Beispieldialoge aus den Experimenten aufgelistet.

Dialog 1

U1:	"Hello Robbi."
R1:	"Hello, what can I do for you?"
U2:	"Bring me the red cup."
R2:	"Going to to bring you the red cup." *Roboter fährt Richtung Tasse.*
R3:	"I do not see the red cup."
U4:	"Move one step to the left."
R4:	"Going one step to the left."
R5:	"I do not see the red cup."
U6:	"Move ten degrees to the left."
R6:	"Turning ten degrees to the left."
R7:	"I do not see the red cup."
U8:	"Move one step forward."
R8:	"Going one step forward."
R9:	"I do not see the red cup."
U10:	"Turn eighty degrees to the right."
R10:	"Turning eighty degrees to the right." *Roboter bewegt Arm in Halteposition, Versuchsleiter stellt die Tasse auf die Hand (simuliertes Greifen).* *Roboter dreht sich in Richtung des Benutzers.*
R11:	"Here you are."
U12:	"Can you see me."
R12:	"Which object do you mean? Which cup do you mean, the red cup, the blue cup or the green cup?"

U13:	"Move one step forward."
R13:	"Going one step forward."
U14:	"Turn ten degrees to the left."
R14:	"Turning ten degrees to the left."
U13:	"Move one step forward."
R13:	"Going one step forward."
	Roboter steht vor dem Benutzer und „übergibt" die Tasse.
U14:	"Thank you, Robbi."
R14:	"You are welcome."

Dialog 2

In diesem Dialog verliert der Roboter die Tasse durch einen Fehler im Spracherkenner:

U1:	"Could you bring me the cup, please."
R1:	"Which cup do you want me to take? The red cup, the blue cup or the green cup?"
U2:	"The red one"
R2:	"Which cup do you want me to take? The red cup, the blue cup or the green cup?"
U3:	"I would like to have the red cup."
R3:	"Going to bring you the red cup."
	Roboter fährt zur Tasse.
R4:	"I do not see the red cup."
U5:	"Turn ten degrees to the right."
R5:	"Turning ten degrees to the right."
	Roboter bewegt Arm in Halteposition, Versuchsleiter stellt die Tasse auf die Hand (simuliertes Greifen).
	Roboter bewegt sich in Richtung des Benutzers.
R6:	"Here you are."
U7:	"Turn twenty degrees to the left."
R7:	"Turning twenty degrees to the left."
U8:	"Step forward."
R8:	"Good bye."
	Roboter winkt und läßt dabei Tasse fallen.

Dialog 3

„Goldener Dialog" mit minimalem Benutzeraufwand:

U1:	"Hello Robbi."
R1:	"Hello, what can I do for you?"
U2:	"Bring me the green cup, please."
R2:	"Going to to bring you the green cup." *Roboter fährt zur Tasse und bewegt Arm in Halteposition, Versuchsleiter stellt die Tasse auf die Hand (simuliertes Greifen).* *Roboter fährt zurück zum Benutzer und „übergibt" die Tasse .*
R3:	"Here you are."
U4:	"Thank you, Robbi."
R4:	"You are welcome."

Dialog 4

U1:	"Hello."
R1:	"Hello, what can I do for you?"
U2:	"Please bring me the red cup."
R2:	"Going to to bring you the red cup." *Roboter fährt zur Tasse und bewegt Arm in Halteposition, Versuchsleiter stellt die Tasse auf die Hand (simuliertes Greifen).* *Roboter dreht sich in Richtung des Benutzers.*
R3:	"Here you are."
U4:	"Go two steps forward, please."
R4:	"Going two steps forward."
U5:	"Go one step forward."
R5:	"Going one step forward." *Roboter steht vor dem Benutzer und „übergibt" die Tasse.*
U6:	"Thank you."
R6:	"You are welcome."

Dialog 5

	...
U9:	"I would like you to actually place this cup next to the eyeglasses on the table and bring me the cup that says maddison square garden, instead."
R9:	"I can bring you a cup, sorry thats all for now."
U10:	"Ohh." *(lacht)* Are you sure, robbi?
R10:	"Yes."
U11:	"You are after all a robot.
U12:	"Wow."
U13:	"I payed a good amount of money for you."
U14:	"You robot arent cheap."
U15:	"Nothing to say for your self, hugh?"
R15:	"Sorry, thats all I can do for you."
U16:	"Do you know the number for technical support?"
R16:	"No."
U17:	"No? Wow."
U18:	"Do you remember where I put the manual?"
R18:	"Sorry, I did not understand."
U19:	"So, I am looking for the instructions they came with you."
R19:	"Sorry, I did not understand."
U20:	*(lacht)* "My view, Robbi is, that you are malfuntioning."
	...

Literaturverzeichnis

[A. 50] A. M. Turing. *Computing Machinery and Intelligence. Mind*, 1950.

[ABD02] T. Asfour, K. Berns, and R. Dillmann. *The Humanoid Robot ARMAR: Design and Control*, 2002.

[AOSS95] Harald Aust, Martin Oerder, Frank Seide, and Volker Steinbiss. *The Philips Automatic Train Timetable Information System. Speech Commun.*, 17(3-4):249–262, 1995.

[ARA+06] T. Asfour, K. Regenstein, P. Azad, J. Schröder, and R. Dillmann. *ARMAR-III: A Humanoid Platform for Perception-Action Integration.* In *HCRS, Second international workshop on Human-Centred Robotic Systems*, München, Germany, Oktober 2006.

[Asi] *Asimo: Humanoider Roboter.* http://www.honda-robots.com/german/html/asimo/frameset2.html.

[Bal99] Heide Balzert. Lehrbuch der Objektmodellierung - Analyse und Entwurf. Spektrum Akademischer Verlag, 1999.

[BD01] N. O. Bernson and L. Dybkjær. *Exploring Natural Interaction in the Car.* In *Proceedings of the International Workshop on Information Presentation and Natural Multimodal Dialogue*, pages 75–79, Verona, December 2001.

[BGHS93] Alan W. Biermann, Curry I. Guinn, D. Richard Hipp, and Ronnie W. Smith. *Efficient collaborative discourse: a theory and its implementation.* In *HLT '93: Proceedings of the workshop on Human Language Technology*, pages 177–181, Morristown, NJ, USA, 1993. Association for Computational Linguistics.

[Car92] Bob Carpenter. The Logic of Typed Feature Structures. Cambridge University Press, Cambridge, England, 1992.

[CH92] Gregory F. Cooper and Edward Herskovits. *A Bayesian Method for the Induction of Probabilistic Networks from Data. Mach. Learn.*, 9(4):309–347, 1992.

[CS89] Herbert H. Clark and Edward F. Schaefer. *Contributing to Discourse. Cognitive Science*, 13(2):259–294, 1989.

[Den02] Matthias Denecke. Generische Interaktionsmuster für Aufgabenorientierte Dialogsysteme. Dissertation, University of Karlsruhe, 2002.

[FHW04] C. Fügen, H. Holzapfel, and A. Waibel. *Tight Coupling of Speech Recognition and Dialog Management Dialog-Context Dependent GrammarWeighting for Speech Recognition.* In *In Proceedings of the International Conference on Spoken Language Processing, ICSLP,* 2004.

[GMP+96] David Goddeau, Helen Meng, Joe Polifroni, Stephanie Seneff, and Senis Busayapongchai. *A Form-Based Dialogue Manager for Spoken Language Applications.* In *Proceedings of the International Conference on Spoken Language Processing,* 1996.

[HG04] H. Holzapfel and P. Gieselmann. *A Way Out of Dead End Situations in Dialogue Systems for Human-Robot Interaction.* In *Humanoids 2004,* Los Angeles, 2004.

[HGC95] D. Heckerman, D. Geiger, and D. M. Chickering. *Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. Machine Learning,* 20(3):197–243, September 1995.

[HH01] Xuedong Huang and Hsiao-Wuen Hon. Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[Hol06] Hartwig Holzapfel. *A Multilingual Expectations Model for Contextual Utterances in Mixed-Initiative Spoken Dialogue,* 2006.

[HS93] R. Hipp and R. Smith. *A Demonstration of the Circuit Fix-it Shoppe.* National Conference on Artificial Intelligence, 1993.

[IE05] Ian H. Witten and Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco, 2005.

[IW95] R. Ivica and A. Waibel. *The Janus speech recognizer.* In ARPA SLT Workshop., 1995.

[Jac95] Ivar Jacobson. Object-Oriented Software Engineering: a Use Case driven Approach. Addison–Wesley, Wokingham, England, 1995.

[JM00] Daniel Jurafsky and James H. Martin. Speech and language processing. Prentice Hall, 2000.

[Kel84] J. F. Kelley. *An iterative design methodology for user-friendly natural language office information applications. ACM Trans. Inf. Syst.,* 2(1):26–41, 1984.

[KHW05] Ulf Krum, Hartwig Holzapfel, and Alex Waibel. *Clarification Questions to Improve Dialogue Flow and Speech Recognition in Spoken Dialogue Systems.* Interspeech 2005, September 2005.

[KRS96] A. Kellner, B. Rueber, and F. Seide. *A Voice-Controlled Automatic Telephone Switchboard and Directory Information System,* 1996.

[Kru05] Ulf Krum. *Klärungsfragen und Subdialoge durch Anomalieanalyse für natürlichsprachliche Dialogsysteme.* Studienarbeit, 2005.

[LGSS06] Lluís F. Hurtado, David Griol, Encarna Segarra, and Emilio Snachis. *A Stochastic Approach for Dialogue Management based on Neural Networks.* Interspeech 2006, September 2006.

[LP97] Esther Levin and Roberto Pieraccini. *A Stochastic Model of Computer-Human Interaction for Learning Dialogue Strategies.* In *Proc. Eurospeech '97*, pages 1883–1886, Rhodes, Greece, 1997.

[LRTGL99] Lori Levin, Klaus Ries, Ann Thymé-Gobbel, and Alon Levie. *Tagging of Speech Acts and Dialogue Games in Spanish Call Home.* In Marilyn Walker, editor, *Towards Standards and Tools for Discourse Tagging: Proceedings of the Workshop*, pages 42–47. Association for Computational Linguistics, Somerset, New Jersey, 1999.

[LS04] Diane J. Litman and Scott Silliman. *ITSPOKE: An Intelligent Tutoring Spoken Dialogue System.* In *Proceedings of the Human Language Technology Conference: 4th Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL) (Companion Proceedings)*, Boston, MA, May 2004.

[McT02] Micheal F. McTear. *Spoken Dialogue Technology: Enabling the Conversational User Interface*, 2002.

[Mit97] Tom Mitchell. Machine Learning. McGraw Hill, 1997.

[MKOB02] J. Martin, R. Keppler, D. Osswald, and W. Burger. *Mechatronische Konzepte zur Verbesserung der MenschMaschineInteration*, 2002.

[PHW06] Thomas Prommer, Hartwig Holzapfel, and Alex Waibel. *Rapid Simulation-Driven Reinforcement Learning of Multimodal Dialog Strategies in Human-Robot Interaction.* Interspeech 2006, September 2006.

[Pie04] Olivier Pietquin. A Framework for Unsupervised Learning of Dialogue Strategies. Dissertation, Faculté Polytechnique de Mons, 2004.

[Pro06] Thomas Prommer. *Rapid Simulation-Driven Reinforcement Learning of Multimodal Dialog Strategies in Human-Robot Interaction.* Diplomarbeit, 2006.

[RG06] B. Reuse and U. Grote.
Dokumentation BMBF-Sonderstand "Mensch-Technik-Interaktion" CeBIT 2006, Halle 9.
http://www.dlr.de/pt_it/sw/CeBIT_2006/Dokumente/MTI_CeBIT_2006_Dokumentation, 2006.

[RLBE05] A. Raux, B. Langner, A. Black, and M. Eskenazi. *Let's Go Public! Taking a Spoken Dialog System to the Real World.* Interspeech 2005, September 2005.

[Rob93] Robert C. Holte. *Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Mach. Learn.*, 11(1):63–90, 1993.

[SBD02] P. Steinhaus, R. Becher, and R. Dillmann. *SFB 588 Humanoide Roboter Lernende und kooperierende multimodale Roboter.* http://www.sfb588.uni-karlsruhe.de/textdateien/ziele_frame.html, 2002.

[SFB] *SFB 588 Humanoide Roboter Lernende und kooperierende multimodale Roboter.* http://www.sfb588.uni-karlsruhe.de.

[SHB92] R. W. Smith, D. R. Hipp, and A. W. Biermann. *A Dialog Control Algorithm and Its Performance.* In *Proc. of the Third Conference on Applied Natural Language Processing*, pages 9–16, Trento, Italy, 1992.

[Sol01] A one-pass decoder based on polymorphic linguistic context assignment, 2001.

[Sph] *Sphinx-4 – A speech recognizer written entirely in the Java(TM) programming language.* http://cmusphinx.sourceforge.net/sphinx4.

[Was04] Larry Wasserman. All of Statistics : A Concise Course in Statistical Inference (Springer Texts in Statistics). Springer, September 2004.

[Wei66] Joseph Weizenbaum. *ELIZA: A Computer Program for the Study of Natural Language Communication between Man and Machine. Communciations of the ACM*, 9(1), 1966.

[Wika] *Wikipedia: Humanoide Roboter.* http://de.wikipedia.org/wiki/Humanoide_Roboter.

[Wikb] *Wikipedia: Leonardo's Robot.* http://en.wikipedia.org/wiki/Leonardo's_robot.

[Wikc] *Wikipedia: Roboter.* http://de.wikipedia.org/wiki/Roboter.

[Wikd] *Wikipedia: Word Error Rate.* http://en.wikipedia.org/wiki/Word_error_rate.

[WL90] Alex Waibel and Kai-Fu Lee, editors. Readings in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[WLK+04] W. Walker, P. Lamere, P. Kwok, B. Raj, E. Gouvea, P. Wolf, and J. Woelfel. *Sphinx-4: A Flexible Open Source Framework for Speech Recognition.* http://cmusphinx.sourceforge.net/sphinx4/doc/Sphinx4Whitepaper.pdf, 2004.

[WW97] Ye-Yi Wang and Alex Waibel. *Statistical Analysis of Dialogue Structure.* In *Proc. Eurospeech '97*, pages 2703–2706, Rhodes, Greece, 1997.