Diplomarbeit

Emotionen als Parameter der Dialogverarbeitung

Hartwig Holzapfel

betreut von
Prof. Dr. Alexander Waibel
Dipl.-Inform. Christian Fügen
Dipl.-Inform. Matthias Denecke

März 2003

Institut für Logik Komplexität
und Deduktionssysteme
Fakultät für Informatik

Universität Karlsruhe (TH)
Interactive Systems Labs

Institut für Logik Komplexität und Deduktionssysteme
Fakultät für Informatik
Universität Karlsruhe (TH)

Interactive Systems Labs

Emotionen als Parameter der Dialogverarbeitung

Diplomarbeit

von

Hartwig Holzapfel

betreut von

Prof. Dr. Alexander Waibel
Dipl.-Inform. Christian Fügen
Dipl.-Inform. Matthias Denecke

März 2003

Erklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Quellen verwendet zu haben.

Karlsruhe, 31.03.2003

Hartwig Holzapfel

Kurzfassung

Emotionen haben in letzter Zeit vermehrt Beachtung im Bereich der Mensch-Maschine Interaktion gefunden. Verschiedene Arbeiten beschreiben, dass Benutzer im Dialog mit dem Rechner emotional reagieren. Um diese Informationen im Dialog sinnvoll nutzen zu können, müssen Emotionen erfasst und modelliert werden. Bei dieser Aufgabe spielen verschiedene Disziplinen, von der Signalverarbeitung bis zur Psychologie, eine Rolle.

Die vorliegende Arbeit schlägt ein Rahmenwerk für emotionssensitive Dialogverarbeitung vor. Das Rahmenwerk basiert auf dem Dialogmanager *ARIADNE*, der mit der Fähigkeit erweitert wurde, den emotionalen Zustand des Benutzer zu erfassen und als Parameter in der Dialogverarbeitung zu berücksichtigen. Dazu werden vier Punkte untersucht, erstens wie werden Emotionen sinnvoll modelliert, zweitens wie können Emotionen erkannt und klassifiziert werden, drittens wie lässt sich eine Integration in das Dialogsystem bewerkstelligen, viertens welche Vorteile kann das Dialogsystem aus den neuen Parametern ziehen.

Danksagungen

Bedanken möchte ich mich bei Christian Fügen und Matthias Denecke, die meine Arbeit inhaltlich betreut und mir viel Wissen vermittelt haben. Weiterhin möchte ich mich bei Professor Waibel für die Möglichkeit bedanken, an diesem spannenden Thema zu arbeiten, sowie für die vielen hilfreichen und visionären Ideen. Dank geht auch an Petra Gieselmann, die mich bei der Durchführung der Datensammlung unterstützt hat und an Kornel Laskowski, mit dem ich viele interessante Diskussionen über Emotionserkennung geführt habe. Weiterer Dank geht an Thomas Schaaf und Klaus Ries, die mich intensiv bei der Suche nach dem Thema meiner Arbeit unterstützt haben und an Rubino Geiß für viele hilfreiche Kommentare. Natürlich möchte ich mich auch bei allen Mitarbeitern des Lehrstuhls bedanken, die im Umfeld der Arbeit jederzeit für Fragen und Diskussionen zur Verfügung standen. Der wohl größte Dank geht an meine Verlobte Patrycja Piernicka, die mich vor allem in den arbeitsintensivsten Phasen unterstützt und motiviert hat.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Heute sind Computer in fast alle Bereiche des alltäglichen Lebens vorgedrungen. Mit immer größerer Rechenleistung können neue Aufgaben gelöst werden. Gleichzeitig wird Software auch den Eigenschaften des Menschen so gut wie möglich angepasst. Ein Faktor, der dabei bisher allerdings weitgehend unberücksichtigt blieb, sind Emotionen des Benutzers. Studien zufolge, beschimpfen oder beschädigen über 80% der Benutzer ihren Computer (Concord Communications USA, Mori). Reeves und Nass beschreiben in ihrem Buch "The Media Equation", dass Benutzer emotional im Umgang mit technischen Geräten agieren. Sie beschreiben, dass Benutzer einen natürlichen Umgang mit Medien erwarten und die Anforderung stellen, dass Medien sozialen Anforderungen genügen und den Regeln des sozialen Lebens unterliegen. Diese Erkenntnisse gelten insbesondere für die Kommunikation und Interaktion durch Sprache, da Sprache eine der wichtigsten natürlichen Modalitäten ist, um Informationen auszutauschen. Die vorliegende Arbeit betrachtet, im Feld der Mensch-Maschine-Kommunikation, Emotionen des Benutzers im Dialog. Die Arbeit beschreibt ein Rahmenwerk zur Modellierung der Emotionen und darauf basierende Parameter, die als Zusatzinformation in der Dialogverarbeitung verwendet werden.

1.2 Beiträge und Abgrenzung

Diese Arbeit schlägt ein Rahmenwerk für emotionssensitive Dialogverarbeitung vor. Das Rahmenwerk basiert auf dem Dialogmanager *ARIADNE* und stellt eine Architektur zur Verfügung, mit der auf einfache Art und Weise bestehende Dialog-Anwendungen mit emotionaler Intelligenz erweitert werden können. Das System modelliert Emotionen als Eigenschaften der Spracheingabe. Aus der Spracheingabe werden Parameter extrahiert, die dazu verwendet werden können, die Spracheingabe unterschiedlich zu interpretieren, die Dialogstrategie zu verändern oder Einfluss auf die Antwortgenerierung zu nehmen. Hierzu werden Emotionsmodellierungen vorgeschlagen und verschiedene Verfahren aufgezeigt, Emotionsparameter im Dialog zu nutzen. Diese Arbeit beschreibt weiterhin die Implementierung des Rahmenwerks, zeigt Beispiele für die Übertragbarkeit des Konzeptes und beschreibt die durchgeführte Benutzerstudie. Im Anhang befindet sich ein Kapitel über Emotionserkennung, das Versuche mit einem einfachen Klassifikationsschema beschreibt.

Diese Arbeit setzt auf Arbeiten zur Emotionserkennung und -modellierung auf. Kapitel 2 stellt einige Modelle vor und beschreibt, welche für das vorgeschlagene System besonders geeignet sind. In der Literatur konnte kein vergleichbares System gefunden werden, das ebenso eine generische Architektur verwendet, um Emotionsparameter in eine bestehende Dialoganwendung zu integrieren, um damit die Dialogstrategie zu verändern, Dialogziele zu definieren und unterschiedliche Interpretationen der semantischen Eingabe zu ermöglichen.

Die Art und Weise, wie Emotionen im Dialog eingesetzt werden, lässt sich am ehesten mit der von Tsukahara und Ward [TW01] vergleichen. Sie beschreiben ein System, das Emotionen des Benutzers verwendet, um in einem computergesteuerten Memory-Spiel verschiedenartige Bestätigungen zu generieren. Im Gegensatz zu unserem System bieten sie aber nicht die Möglichkeit, unterschiedliche Strategien zu benutzen, um komplexe Dialogziele zu erreichen oder unterschiedliche Interpretationen der semantischen Eingabe zu ermöglichen. Darüber hinaus kann die hier vorgeschlagene Architektur auch das Emotionsmodell von Tsukahara und Ward verwenden (siehe Kapitel 2.2.4).

Das vorgeschlagene System grenzt sich von anderen ab, die Emotionen nur zur Darstellung benutzen, oder Emotionen dazu verwenden, um die "Glaubhaftigkeit" des Systems zu erhöhen. Der Schwerpunkt unseres Systems liegt darauf, erkannte Emotionen zu benutzen.

1.3 Aufbau der Arbeit

Kapitel 2 und 3 sind Grundlagenkapitel, in denen Emotionsmodelle und das, der Arbeit zugrunde liegende Dialogsystem *ARIADNE* beschrieben werden. Kapitel 4 beschreibt die architekturelle Integration von emotionalen Parametern in das Dialogsystem und Implementierungen der unterschiedlichen Konzepte. Kapitel 5 beschreibt einige Dialog-Anwendungen, die erstellt wurden, um die Umsetzbarkeit und Übertragbarkeit des vorgestellten Konzeptes zu zeigen. Kapitel 6 beschreibt eine Benutzerstudie, die mit einer Roboter-Simulation durchgeführt wurde, darauf basierende Analysen und Konzepte des Benutzerverhaltens, einschließlich Emotionen. Weiterhin wird beschrieben, wie die erfassten Merkmale im Dialog genutzt werden oder genutzt werden können. Kapitel 7 schließlich, beschreibt Erkenntnisse, die aus der Arbeit gezogen wurden, enthält eine Zusammenfassung der Arbeit und gibt einen Ausblick auf weitere Einsatzmöglichkeiten des Systems und weiterführende Forschungsthemen.

1.4 Vorbemerkungen

Verwendung englischer und deutscher Begriffe

Im Allgemeinen wurde versucht, Begriffe aus dem Englischen ins Deutsche zu übersetzen. Bei feststehenden Begriffen wie z.B. Namen von Emotionsmodellen werden in dieser Arbeit meistens die originalen Bezeichnungen beibehalten, um eine bessere Übersichtlichkeit zu erhalten. Es ist dann jeweils eine deutsche Übersetzung beigefügt, um das Verständnis zu fördern. Bei Emotionsnamen wurde in den meisten Fällen auf eine Übersetzung verzichtet, da die exakte Bedeutung bei der Übersetzung in eine andere Sprache in vielen Fällen verloren geht.

Zum Wort "emotional"

Die meisten Arbeiten zur Emotionsverarbeitung in Verbindung mit Computern sind in englisch abgefasst. Picard definiert den Begriff "affective computing" als Berechnungen und Programme, die sich auf Emotionen beziehen, aus Emotionen hervorgehen, oder absichtlich Emotionen beeinflussen [Pic97]. Die meisten Arbeiten verwenden eine ähnliche Terminologie und benutzen vorrangig das Wort "affect" (wie affective speech, affective intent,

affective agents, u.a.). Das Deutsche bietet hierzu kein passendes Äquivalent, ohne dass die Bedeutung des Wortes "affect" stark in Mitleidenschaft gezogen wird. In dieser Arbeit wird deshalb das Wort "emotional" verwendet und Begriffe wie "affective speech" als "emotionale Sprache" übersetzt. Diese Übersetzung ist dementsprechend in etwa als emotional erzeugte Sprache, Emotionen auslösende Sprache, oder Sprache mit sonstigem Bezug zu Emotionen zu verstehen.

Kapitel 2

Grundlagen: Emotionen

Dieses Kapitel beschreibt Grundlagen der Emotionstheorie, Emotionsmodelle und deren Anwendung in Software. Es dient dazu, Grundlagen zu erläutern und einen Überblick über das Gebiet der Emotionen zu geben. Der Hauptfokus des Autors liegt auf Kommunikation im Dialog und dabei insbesondere auf sprachbasierter Kommunikation.

Ein Dialogsystem, dem nur das Sprachsignal als Eingabe zur Verfügung steht, kann zur Emotionserkennung Merkmale des Sprachsignals und linguistische Merkmale der Sprache verwenden. Allerdings geben, auch bei der Kommunikation zwischen Menschen, nicht nur Merkmale der Sprache Auskunft über den emotionalen Zustand des Gesprächspartners, sondern auch andere Modalitäten, wie Gestik oder Mimik.

2.1 Grundlagen der Emotionstheorie

2.1.1 Eigenschaften von Emotionen

Bevor wir beginnen, verschiedene Modelle und deren Einsatz in künstlichen Systemen zu beschreiben, soll zuerst darauf eingegangen werden, was Emotionen sind und wie sie zustande kommen. Schon über das Wort *Emotionen* selbst gibt es unterschiedliche Ansichten.

Es findet sich in der psychologischen Literatur eine Menge Grundlagenarbeit über die Definition von Emotionen. Ekman und Davidson [ED94] beschreiben grundlegende Eigenschaften über Emotionen, deren Ursachen und damit verbundene kognitive Abläufe beim Menschen. Neben verschiedenen Arbeiten, die Emotionen als einfache Reaktionen des Körpers beschreiben und anderen Arbeiten, die den Ursprung eher im kognitiven Bereich sehen [LKF80], stimmt ein Großteil der heutigen Literatur darin überein, dass Emo-

tionen komplexer Natur sind und als Kombination von physischen und kognitiven Faktoren angesehen werden müssen. Verschiedene Ansätze gibt es aber darin, ob man Emotionen besser durch Merkmale des Körpers modelliert, z.B. [SB96], oder eher kognitive Modelle verwendet. Eng an die kognitiven Modelle sind Systeme wie der *Affective Reasoner* angelehnt (siehe auch Kapitel 2.3), die versuchen, über Schlussfolgerungsprozesse Emotionszustände zu inferieren. Picard [Pic97] definiert verschiedene Bezeichnungen und bezeichnet den physischen Teil als *körperliche Emotionen* (engl. *bodily emotions*) oder *primäre Emotionen* (engl. *primary emotions*); der kognitive Teil wird als *geistige Emotionen* (engl. *mental emotions*) bezeichnet; Auswirkungen der mentalen Zustände auf den Körper wird als *Sentic Modulation* bezeichnet.

2.1.2 Dauer von Emotionen

[MA93] und [CDCT+01] beschreiben Zusammenhänge zwischen Emotionen und dem Sprachsignal. Der Hauptfokus liegt dabei auf Emotionszustände, die über mehrere Minuten oder länger andauern. Eine Ausnahme bildet Cowie [CDCR99], der sich explizit mit der Veränderung von Emotionen im Sekundenbereich beschäftigt. Für die Veränderung im Sekundenbereich werden wir in Abschnitt 2.2.4 ebenfalls ein Emotionsmodell kennen lernen.

Wir werden später sehen, dass es in dem vorgeschlagenen System sinnvoll ist, nur einen Emotionswert für die gesamte Dauer einer Äußerung zu berechnen. Nicht außer Acht gelassen wurde dabei, dass ein Sprecher Affekt gegenüber Objekten [Pic97] ausdrücken kann, was sich in der Sprache wiederspiegelt. Das Dialogsystem ist in der Tat in der Lage, emotionale Schwankungen auch innerhalb einer Äußerung zu modellieren (siehe Kapitel 3). Allerdings bringt dies für das Gesamtsystem keine Vorteile, da das Benutzermodell, auf dem die Dialogstrategien aufsetzen, den emotionalen Zustand des Benutzers während seines Gesprächsbeitrags (Turn) modelliert.

2.2 Emotionsmodelle und Kategorisierungen

Es gibt verschiedene Modelle zur Repräsentation von Emotionen. In diesem Kapitel werden die wichtigsten vorgestellt.

Zwischen den verschiedenen Emotionsmodellen gibt es grundlegende Unterschiede. Einige Modelle verwenden kontinuierliche, andere diskrete Werte. Die Werte können entweder einem ein- oder mehrdimensionalen Raum, oder einer abstrakten Klasseneinteilung oder Kategorisierung entstammen.

2.2.1 Kontinuierliche Modelle

In einem kontinuierlichen Modell, z.B. [LBC90], werden Emotionen in einem kontinuierlichen, multidimensionalen Raum dargestellt. Ein solches Modell, mit zwei Dimensionen, ist das *Arousal-Valence* Modell (Abbildung 2.1). Das *Arousal-Valence* Modell wird von vielen Autoren benutzt, um Emotionen einzuteilen oder zu organisieren [Plu94]. *Valence* (dt. *Wertigkeit* oder *Valenz*) beschreibt wie positiv oder negativ eine Emotion ist. *Arousal* (dt. *Aktivierung* oder *Intensität*) beschreibt die innere Erregung der Person oder die Intensität der Emotion. Abbildung 2.1 enthält erstens die Einordnung emotionaler Zustände (normale Schrift, Beispiel *Freude*) und zweitens die Einordnung von Bildern bzw. Fotographien (Schrift in eckigen Klammern, Beispiel *[Skisprung]*) nach den Reaktionen von Betrachtern (aus [Pic97]).

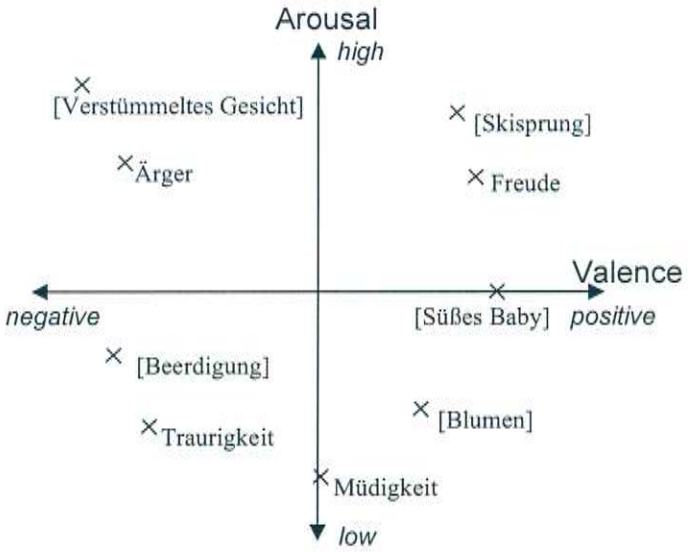

Abbildung 2.1: Arousal-Valence Ebene

2.2.2 Basic Emotions

Ein weiteres Emotionsmodell ist unter dem Namen *Basic Emotions* [Ekm92] bekannt (zu dt. als *Elementare Emotionen* übersetzbar). Es handelt sich um ein diskretes Emotionsmodell, das aus fünf, sechs oder mehr Werten besteht, wie z.B. *happiness*, *sadness*, *anger*, *fear*, *surprise*, *disgust* und *neutral*. Der emotionale Zustand wird im allgemeinen durch einen dieser Werte charakterisiert.

Entsprechend Ekman [Ekm99] steht das Wort "basic" für drei grundlegende Bedeutungen und Eigenschaften dieses Modells:

1. Als erstes trennt es verschiedene Emotionen, die sich durch wichtige Eigenschaften unterscheiden. Aus dieser Sicht unterscheiden sich die negativen Emotionen *fear*, *anger*, *disgust*, *sadness* und *contempt* in ihrer Taxierung, auslösenden Ereignissen, möglicher Reaktion, Physiologie und weiteren Charakteristika. Ebenso unterscheiden sich die positiven Emotionen *amusement*, *pride in achievement*, *satisfaction*, *relief* und *contentment* von einander.

2. Zweitens soll das Wort "basic" aufzeigen, dass Emotionen durch anpassungsfähige Werte Einfluss auf lebenswichtige Aufgaben haben, und damit eine grundlegende Bedeutung für den menschlichen Organismus darstellen.

3. Bisher wurde der Begriff "basic" auch dafür verwendet, Basiselemente zu beschreiben, die zusammen komplexe Emotionen bilden [Ekm92]. Entsprechend [Ekm99], wird an dieser Eigenschaft aber inzwischen offiziell nicht mehr festgehalten.

2.2.3 OCC-Modell

Ein verbreitetes Modell für Emotionen ist das *OCC-Modell* [OCC88]. Das *OCC-Modell* benutzt eine Wertemenge, die sich aus 22 diskreten Emotionswerten zusammensetzt. Die Emotionswerte sind nicht nach einem Modell wie die *Basic Emotions* geordnet. Die Emotionen werden entsprechend verschiedenen Bedingungen, die unterschiedliche kognitive Auswirkungen haben, in Kategorien eingeteilt. Die Emotionen lassen sich damit nicht in einem Raum, der durch verschiedene Dimensionen aufgespannt wird, definieren, wie es beim *Arousal-Valence* Modell der Fall ist. Die Kategorien sind abstrakte Klassen, wobei eine Klasse eine Gruppe von Zuständen repräsentiert, die gleiche oder ähnliche Eigenschaften haben. Die Eigenschaften werden durch Reaktionen und Situationen definiert. Die Werte und Einteilungen in die Klassen sind in Tabelle 2.1 abgebildet. Verschiedene Beispiele, wie das *OCC-Modell* eingesetzt wird um Emotionen zu beschreiben, klassifizieren oder zu synthetisieren, können in der Literatur, z.B. [Ell92] [AKG+99] [Pic97], gefunden werden.

Das *OCC-Modell* benötigt ein gutes Kontext- bzw. Weltmodell. In den meisten Situationen des echten Lebens, in denen sich z.B. humanoide Roboter zurechtfinden müssen, sind die Einflussfaktoren zu komplex und vielfältig,

Gruppe	Beschreibung	Name und Emotionstyp
Well-Being	appraisal of a situation as an event	**joy**: pleased about an event **distress**: displeased about an event
Fortunes-of-Others	presumed value of a situation as an event affecting another	**happy-for**: pleased about an event desirable for another **gloating**: pleased about an event undesirable for another **resentment**: displeased about an event desirable for another **sorry-for**: displeased about an event undesirable for another
Prospect-based	appraisal of a situation as a prospective event	**hope**: pleased about a prospective desirable event **fear**: displeased about a prospective undesirable event
Confirmation	appraisal of a situation as confirming or disconfirming an expectation	**satisfaction**: pleased about a confirmed desirable event **relief**: pleased about a disconfirmed undesirable event **fears-confirmed**: displeased about a confirmed undesirable event **disappointment**: displeased about a disconfirmed desirable event
Attribution	appraisal of a situation as an accountable act of some agent	**pride**: approving of one's own act **admiration**: approving of another's act **shame**: disapproving of one's own act **reproach**: disapproving of another's act
Attraction	appraisal of a situation as containing an attractive or unattractive object	**liking**: finding an object appealing **disliking**: finding an object unappealing
Well-being/ Attribution	compound emotions	**gratitude**: admiration + joy **anger**: reproach + distress **gratification**: pride + joy **remorse**: shame + distress
Attraction/ Attribution	compound emotion extensions	**love**: admiration + liking **hate**: reproach + disliking

Tabelle 2.1: Emotionswerte und Typisierung nach dem OCC-Modell aus [OCC88]

um die Emotionen des Benutzers durch kognitive Modelle genau ermitteln zu können. In einer solchen Umgebung scheint es sinnvoll, ein Modell zu verwenden, das auf physiologischen Merkmalen basiert oder zumindest durch eine Kombination mit einem solchen Modell ein robusteres Benutzermodell zu erstellen.

Um verschiedenen Anforderungen gerecht zu werden und auch Erweiterungsmöglichkeiten für neue Emotionsmodelle zu bieten, stellt diese Arbeit eine Architektur zur Verfügung, die von dem tatsächlichen Modell zur Emotionserkennung weitgehend unabhängig ist (siehe Kapitel 4.2.3).

2.2.4 Ephemeral Emotions

Das Modell der *Ephemeral Emotions* wurde von Tsukahara und Ward eingeführt [War00] und bedeutet in etwa *Flüchtige Emotionen* oder *Kurzlebige Emotionen*. Die Tabelle 2.2 stammt aus [TW01] und beschreibt verschiedene Emotionskategorien.

Ich will meine Gedanken ausdrücken (und etwas sagen)
Ich fühle mich unbehaglich (mit diesem Thema)
Ich amüsiere mich (über deine Geschichte)
Ich bin frustriert (dass ich dich nicht überzeugen konnte)
Es gefällt mir (dass du die Ironie meiner Worte schätzt)
Ich verstehe etwas nicht (also musst du deutlicher werden)
Ich brauche einen Moment (um die Aussage zu verdauen)
Ich weiß worüber ich spreche (also hör mir einfach kurz zu)
Ich kann mich nicht festlegen (also darfst du gerne weiter sprechen)
Ich bin gelangweilt (also lass uns über etwas anderes sprechen)
Ich bin besorgt (dass ich mich nicht gut genug ausdrücke)
Ich bin wirklich interessiert (an deiner Meinung darüber)
Ich weiß, wie du dich fühlst (und fühle mit dir)
Das ist mit schon bewusst (also können wir jetzt über anderes sprechen)
Ich werde unruhig (also lass uns diese Konversation beenden)
Ich bin stark verärgert (über den Ton deiner letzten Äußerung)

Tabelle 2.2: Beispiele von Gefühlen, die als *Flüchtige Emotionen* im Dialog auftreten, übersetzt aus [TW01].

Die Tabelle basiert auf Beobachtungen und Analyse der Prosodie, Rückmeldungspartikeln, Disfluenzmarkern und Gesten, wie sie in tutoriellen Dialogen, Gesprächen und Erzählungen auftreten. Details hierzu sind in [WK99] und [War00] zu finden.

Dieses Modell unterscheidet sich von den anderen Emotionsmodellen vor allem durch die Bezeichnungen der Zustände, die nicht direkt mit dem allgemeinen Verständnis von "Emotionen" vergleichbar sind. Da es sich hier auch um ein diskretes Modell handelt, können die Zustände dieses Modells ebenfalls als Eingabe für das unser Rahmenwerk verwendet werden (siehe Kapitel 3.3).

2.2.5 Affective Intent

Das letzte Modell, das hier vorgestellt werden soll, modelliert emotionale Zustände der Sprache von Müttern gegenüber ihren vorsprachlichen Kindern. Betrachtet werden dabei Sprechakte, die Kinder beim Lernen unterstützen sollen, bzw. benutzt werden, um Kindern etwas beizubringen. Da die Kinder noch nicht in der Lage sind, den linguistischen Inhalt des Gesprochenen zu erfassen, erfolgt die Information über Merkmale des Sprachsignals wie tonale und prosodische Merkmale. Diese Informationen werden in [BA02] als *Affective Intent* bezeichnet (zu dt. als emotionale Absichten übersetzbar). Als Merkmale werden vier grundsätzlich unterschiedliche prosodische Muster identifiziert: Lob (*praise*), Verbot (*prohibition*), Vorsicht (*attention*) und Trost (*comfort*).

Diese Eigenschaften der Kommunikation passen ebenso auf interaktive Lernphasen des Roboters. Ziel dieses Modells ist es, diese Eigenschaften und Lernmethoden auch in Kombination mit Robotern zu verwenden. Bei dieser Lernmethode wird ausschließlich Verhalten in Betracht gezogen, wie es Mütter gegenüber vorsprachlichen Kindern zeigen. Tonale und prosodische Merkmale, sowie Intensität der Aussprache sind dabei typischerweise stark übertrieben. Diese besondere Sprechweise ist auch unter dem Begriff *Motherese* [Sno72] bekannt.

Im Unterschied zu den bisher vorgestellten Emotionsmodellen, ist dieses Modell zwar auf die Kommunikation mit vorsprachlichen Kindern eingeschränkt. Der Vorteil ist aber, dass alle emotionalen Zustände dieses Modells für Kommunikation intendiert sind und der Gesprächspartner (das vorsprachliche Kind oder der Roboter) in der Lage sein soll, den emotionalen Zustand zu erfassen.

2.3 Affektive Agenten

Verschiedene Arbeiten beschäftigen sich damit, einem Agenten ein typisches Verhaltensmuster zu geben und darüber, wie dieses Verhalten vom Menschen aufgefasst und interpretiert wird. Was sind aber typische Verhaltensmuster?

Dazu gehören vor allem Ansätze, in einem Agenten menschliches Verhalten zu simulieren oder ihm Persönlichkeit zu verleihen. Menschliche Verhaltensmuster sind allerdings sehr komplex und dadurch auch sehr schwer zu simulieren. Immerhin zeigen Untersuchungen, dass Verhaltensmuster von Systemen, wie z.B. dominantes Verhalten, von Menschen als solches erkannt und von anderen Verhaltensmuster unterschieden werden kann [NMF+95]. Weiterhin haben verschiedene Menschen auch oft Präferenzen für unterschiedliche Verhaltensmuster [NMF+95].

Persönlichkeitsmodelle

Vor allem virtuelle Tutoren und Avatare profitieren von einer eigenen Persönlichkeit. Sie soll die Interaktion interessanter und lebhafter gestalten und dabei auch den Benutzer in Betracht ziehen. Das am häufigsten verwendete Persönlichkeitsmodell ist das *Fünf-Faktoren-Modell*. Dieses Modell stammt ursprünglich aus der Psychologie, wird aber auch für Persönlichkeitsmodelle von Software eingesetzt (s.u.). Es setzt sich aus fünf Faktoren zusammen, deren genaue Bezeichnungen bei verschiedenen Autoren schwanken. In den letzten Jahren konnte Konsens über folgende Bezeichnungen gefunden werden [Hei00]:

- Extrovertiertheit (*extraversion*)
- Liebenswürdigkeit (*agreeableness*)
- Gewissenhaftigkeit (*conscientousness*)
- Emotionalität (*neuroticism* oder *emotionality*)
- Offenheit (*openness*)

Glaubhaftigkeit

Es existieren verschiedene Arbeiten, die sich damit beschäftigen, die *Glaubhaftigkeit* (im Sinne menschenähnlichen Charakters) von künstlichen Systemen zu erhöhen. In einer der grundlegenden Arbeiten zu diesem Thema, beschreibt Bates, dass einer der wichtigsten Faktoren, um künstlichen Figuren "Leben" einzuhauchen, Emotionen sind [Bat97]. Die weiteren Hauptfaktoren sind *Persönlichkeit*, *Interaktivität* und *Soziales Verhalten* [Rei96]. Ein Modell, das Emotionen und auslösende Faktoren modelliert, ist der *Affective Reasoner* [AKG+00]. Er verwendet das *Fünf-Faktoren-Modell* zur Modellierung der Persönlichkeit und das *OCC-Modell* [OCC88] zur Modellierung der Emotionen.

Affektive Benutzermodellierung

Eine Implementierung, die ebenfalls auf dem *Affective Reasoner* aufbaut, ist der animierte Tutor *Steve* [RJ97]. Neben eigener Persönlichkeit und *Affektiver Benutzermodellierung* liegen die Schwerpunkte des Systems auch auf der Synthese von Emotionen und Planung der Interaktion entsprechend pädagogischen Aspekten, siehe [ELR97]. *Steve* zeigt eigene Emotionen, die durch den Kontext der Simulation beeinflusst werden. Das genaue Emotionsmodell wird in [Ell97] weiter beschrieben, es hat aber in etwa die gleichen Werte, wie das zugrunde liegende *OCC-Modell* (in Tabelle 2.1 dargestellt). *Steve* benutzt verschiedene Techniken, um den Benutzer zu motivieren. Diese Techniken nutzen ein affektives Benutzermodell, um zu erkennen, wann sich der Benutzer langweilt. [Geb01] versucht, Präsentationsstrategien für öffentliche Informationssysteme durch *Affektive Benutzermodellierung* (engl. *Affective User Modeling*) mit dem *Affective Reasoner* zu verbessern.

Die oben beschriebenen Systeme beziehen sich auf Software-Agenten. Im Gegensatz dazu beschreibt [BS99] die Integration von Wahrnehmung, Aufmerksamkeit, (An-)Trieb, Emotionen, Unterscheidung von Verhalten und Ausdrucksstärke in ein Robotersystem (*Kismet*) mit Sozialverhalten.

2.4 Emotionsbasierte Entscheidungsfindung

Ein weiterer Aspekt von Emotionen ist die Auswirkung auf das Entscheidungsverhalten beim Menschen. Dieser Aspekt soll bei dieser Arbeit nicht im Vordergrund stehen, wird hier aber der Vollständigkeit wegen erwähnt. Emotionen sind - ob nun als Auslöser oder eher als eine Begleiterscheinung - ein wichtiger Bestandteil bei Entscheidungsprozessen und der Planung von Aktionen. In Stress- und Gefahrensituationen muss der Mensch in der Lage sein, schnell und effektiv zu handeln. Der Mensch lenkt seine gesamte Aufmerksamkeit auf die Aufgabe, die er durchführen muss. Alle anderen Prozesse werden dabei in den Hintergrund gedrängt. Neben diesen offensichtlichen Eigenschaften beeinflussen Emotionen, nach dem aktuellen Stand der Forschung, auch Entscheidungen, die traditionell als rein rationale Entscheidungen vermutet wurden. Diese Arbeiten, z.B. [Dam95], beschreiben, dass die meisten Entscheidungen im Alltag "emotional" und nicht "rational" getroffen werden. Viele Erkenntnisse auf diesem Gebiet stammen von Untersuchungen mit autistischen Personen, denen einfachste Alltagsentscheidungen extrem schwer fallen, wenn nicht sogar unlösbare Probleme darstellen. Sie verfügen zwar über hervorragende logische Fähigkeiten und ein ausgeprägtes Gedächtnis, allerdings ist deren emotionales System gestört.

Diese Forschungen könnten von großer Relevanz für das Gebiet der künstlichen Intelligenz sein, wo bisher hauptsächlich versucht wurde, durch rein logische Schlussfolgerungen Entscheidungen zu treffen. Emotionsmodelle zur Integration von Emotionen in Entscheidungsprozesse werden in [Vel97] und [Bre98] vorgestellt. Ein Ansatz zur Umsetzung der *Emotionsbasierten Entscheidungsfindung* für eine Roboter-Steuerung wird in [Vel98a] und [Vel99] beschrieben. Weitere Auswirkungen von Emotionen sind interessant bei der Betrachtung von Lernverhalten oder Zielsetzung von Menschen, siehe z.B. [BTM96] und [Wri96].

2.5 Ausdruck und Erkennen von Emotionen

Dieser Abschnitt beschreibt den Ausdruck von Emotionen beim Menschen und damit verbunden Systeme, sowohl zur Emotionserkennung, als auch zur Synthese von Emotionen. Diese Analyse ist, z.B. für Sprache, für Emotionserkennung und -synthese gleichermaßen relevant, da beide Disziplinen die Auswirkungen von emotionalen Zuständen auf die Signaleigenschaften modellieren. Zudem können Emotionen über verschiedene Modalitäten kommuniziert werden. Hier soll auf Emotionen in Sprache, in Text und in Gestik und Mimik jeweils kurz eingegangen werden, da diese Modalitäten für diese Arbeit am interessantesten sind. Zusätzlich gibt es Arbeiten, die sich mit anderen Modalitäten befassen und z.B. Drucksensoren für haptische Kontrollgeräte benutzen. Weiterhin gibt es auch Arbeiten für künstliche Systeme, die nicht versuchen, den Menschen zu imitieren, sondern eigene Emotionen entwickeln, die auf ihre eigenen Ressourcen, wie z.B. Batteriestatus oder Rechenkapazität, bezogen sind [Pic97].

2.5.1 Synthese von Emotionen

Mit der Synthese von Emotionen wird hier Bezug auf die Darstellung von Emotionen in einem künstlichen System genommen, mit der Zielsetzung, sie einem Menschen gegenüber zu kommunizieren. Dies beinhaltet auf der einen Seite, dass es eine Modalität geben muss, über die ein Mensch Informationen von der Maschine empfangen kann und dass die Emotionen so dargestellt werden können, dass sie für den Menschen intuitiv verständlich sind. Das OCC-Modell [OCC88] ist gut geeignet als Basismodell und wird z.B. von Reilly und Bates [RB92] verwendet und von Picard [Pic97] ebenfalls für diesen Einsatz beschrieben. Das OCC-Modell dient hierbei als Kategorisierung der Emotionen. Zusätzlich sind weitere Modelle für die entsprechenden Modalitäten notwendig, die entsprechende Darstellungsformen beschreiben, wie

z.B. die Form der Lippen für die Emotionen *happy*, *sad* usw.

Sprachsynthese

Emotionssynthese in Sprache wird z.B. von Murray und Arnott [MA96] beschrieben. Sie beschreiben eine prototypische Implementierung, für eine regelbasierte Simulation von Emotionen in Sprache, genannt *HAMLET* (Helpful Automatic Machine for Language and Emotional Talk). Dem System zugrunde liegt das *DECtalk* Spracherzeugungssystem. HAMLET enthält Regeln, die für die sechs Emotionen (anger, happiness, sadness, fear, disgust, relief) verschiedene Einstellungen der Sprachsynthese definieren. Die Parameter der Sprachsynthese sind Sprachqualität, bezogen auf die gesamte Äußerung, Grundfrequenz (Pitch) und Sprechgeschwindigkeit der Phoneme. Experimente in [MA95] haben gezeigt, dass Zuhörer immerhin in der Lage waren, den emotionalen Zustand des Systems in den meisten Fällen zu erkennen.

Visuelle Darstellung

Mehr Arbeiten, als zur Simulation von Emotionen in Sprache, gibt es zur visuellen Darstellung von Emotionen. Die einfachste Darstellung sind *Emoticons* oder *Smileys*, die es sowohl als Kombinationen von Textsymbolen, wie auch als Graphiken gibt. *Emoticons* in Textform, wie z.B. :-) oder :-(erfreuen sich einer großen Beliebtheit in Internet-Chats oder für elektronische Briefe, da sie einfach zu verwenden, aber dennoch ausdrucksstark sind. Neben der reinen Textform sind solche kleinen "Anhängsel" die einzige Möglichkeit dem Konversationspartner einen Hinweis darauf zu geben, ob der Text z.B. ernst oder ironisch zu verstehen ist. Eine weitere, komplexere Darstellungsform sind *Avatare*. Sie haben inzwischen besonders im Internet eine weite Verbreitung gefunden. Die meisten dieser Systeme können kleine Animationen abspielen, fast alle können den einen oder anderen emotionalen Zustand darstellen. Allerdings haben nur wenige Systeme ein umfangreiches Repertoire an emotionalen Zuständen.

2.5.2 Emotionen in Sprache

[PW98b] [SLS84] und [SB96] beschreiben Grundlagen über Emotionen im Sprachsignal und über Emotionserkennung auf Sprache. In der Literatur findet man verschiedene Verfahren, Emotionserkennung zu betreiben. Grundsätzlich unterscheidet man zwischen akustischen Merkmalen der Sprache und dem linguistischen Inhalt. Einfache Erkenner verwenden als Merkmale Grundfrequenz (Pitch) und Energie des Sprachsignals. Andere verwenden Konturen

der Grundfrequenz. [Sch86], [DPW96] und [PW98a] beschreiben verschiedene Verfahren mit mehreren Kombinationen aus (Sprach-)Merkmalen und Lernverfahren. Im Anhang D befindet sich eine Beschreibung unserer Experimente zur Emotionserkennung mit einfachen Merkmalen.

2.5.3 Linguistischer Ausdruck von Emotionen

Abschnitt 2.5.2 behandelte Auswirkungen von Emotionen auf die Art, *wie* etwas gesprochen wird. Dieser Abschnitt behandelt nun Auswirkungen auf das, *was* gesprochen wird. Das kann zum einen bedeuten, dass Worte selber Träger dieser Information sind. [KM01] beschreibt hierzu einen Ansatz für geschriebenen Text, der mit Verknüpfungen und Synonymklassen von *WordNet*[1] arbeitet. Zum anderen geben auch, über die lexikalische Analyse hinaus, semantische oder pragmatische Bedeutung von Phrasen oder ganzen Sätzen Informationen über die Absichten und den emotionalen Zustand des Benutzers.

Kombination der Modelle

Neben Arbeiten, die nur auf der Akustik oder nur auf Text basieren, können kombinierte Modelle von beidem profitieren. Eine Kombination aus Akustik und Sprachmodellen wird in [Pol99] beschrieben. Die akustischen Merkmale können direkt aus dem Sprachsignal extrahiert werden. Um Merkmale mit einem Sprachmodell zu extrahieren, wird ein Spracherkenner verwendet, der die gesprochene Sprache in Text umwandelt.

Generierte Sprache

Weiterhin ist auch relevant, welche Effekte die Sprachausgabe des Systems hat. In [dRG99] wird *Affective Natural Language Generation* als Sprachgenerierung mit Einfluss auf Emotionen des Hörers beschrieben.

2.5.4 Emotionen in Gestik und Mimik

Bei der Kommunikation zwischen Menschen nehmen wir nicht nur allein die Sprache wahr, sondern sehen auch, wie der Gesprächspartner sich bewegt und nehmen Gestik und Veränderung der Gesichtszüge wahr. Diese Eigenschaften haben wir im letzten Abschnitt, bei Eigenschaften der Sprache, nicht berücksichtigt. Erkennung von emotionalen Zuständen mit Bildverarbeitung wird z.B. in [PC95] beschrieben. Vermutlich lassen sich auch verschiedene

[1] http://www.cogsci.princeton.edu/~wn/

Modalitäten kombinieren und Emotionserkennung über Sprache und Emotionserkennung mit Bildverarbeitung kombinieren.

2.6 Auswahlkriterien

In diesem Kapitel wurden verschiedene Emotionsmodelle, das Auftreten von Emotionen, Methoden zur Erkennung und Methoden zur Synthese von Emotionen vorgestellt.

Für diese Arbeit ist das Arousal-Valence Modell von besonderer Relevanz. Dieses Modell hat die wenigsten Einschränkungen und ist besonders für Emotionserkennung auf physiologischen Signalen geeignet. Systeme, wie Affektive Agenten (Abschnitt 2.3), sind durch ihre Emotionsmodelle (das OCC-Modell und Verwendung des Affective Reasoners) in ihrem Einsatz auf eine Domäne beschränkt. Die zugrunde liegenden Merkmale sind nicht so leicht übertragbar, wie es bei Verwendung von physiologischen Merkmalen, Merkmalen des Sprachsignals und dem Arousal-Valence Modell der Fall ist. Der Dialogmanager selber verwendet Emotionsnamen, um einzelne, diskrete Zustände zu bezeichnen. Auf dem Arousal-Valence Modell aufbauend, ist eine Diskretisierung definiert, die die Werte des kontinuierlichen Modells in eine Kategorisierung überführt. Neben dem Arousal-Valence Modell, können aber auch andere diskrete Modelle, wie z.B. die Ephemeral Emotions, als Eingabe für den Dialogmanager verwendet werden.

Kapitel 3

Grundlagen: Dialogsystem

Dieses Kapitel erläutert kurz die Grundlagen der Dialogverarbeitung und geht auf den verwendeten Dialogmanager *ARIADNE* ein. Da die theoretischen Hintergründe zu *ARIADNE* bereits in [Den02] beschrieben sind, sollen hier nur die Techniken erklärt werden, die für die weitere Ausführung wichtig oder grundlegend sind.

ARIADNE wurde von Matthias Denecke an der Carnegie Mellon Universität in Pittsburgh entwickelt und wird zur Zeit als OpenSource-Projekt veröffentlicht.

3.1 Grundlagen

3.1.1 Dialogverarbeitung

Zu Dialogmodellierung und Dialogmanagement gibt es eine ganze Reihe von Arbeiten mit unterschiedlichen Ansätzen. Diese reichen von sehr einfachen Systemen wie *ChatBots*, deren erster Vertreter *ELIZA* von Weizenbaum [Wei66] Anfang der 60er Jahre war, bis zu sehr komplexen Systemen. Da ChatBots aufgrund ihrer einfachen Struktur keine komplexen oder umfangreichen Dialoge modellieren können, werden sie hauptsächlich zur Unterhaltung eingesetzt. Komplexere Systeme besitzen mehr Möglichkeiten, Dialoge zu modellieren. Das System *ARTIMIS* [SBP97] von France Télécom verwendete eine Modallogik, um generelle Prinzipien rationalen Verhaltens (*Rational Agency*) zu modellieren. Allgemein wird Kommunikation als zielbasiert angenommen [HR83]. Aussagen dienen dazu, Zielen näher zu kommen, Unterziele zu verfolgen oder auf neue Ziele umzuschwenken. Dementsprechend arbeiten die meisten Dialogsysteme auch zielbasiert und ermöglichen *Subdialoge*, wobei abzuarbeitende (übergeordnete) Dialogziele auf einem Stapel gespeichert

werden [GS86]. Fast alle Dialogsysteme arbeiten *aufgabenorientiert*.

3.1.2 Wissensrepräsentation

Die meisten Systeme verwenden *Rahmenbasierte Repräsentationen* [BGS90] [Bil91] um Informationen darzustellen und damit zu arbeiten. Diese können sehr einfach aus den Zerteilungsbäumen semantischer Grammatiken erstellt werden, wie in den Systemen *PHOENIX* [War94] und *SOUP* [Gav00]. Im *SUNDIAL* (Speech Understanding and Dialogue) System [Pec91] wird weiterhin auch Unsicherheit in semantischer Repräsentation [HMY92] modelliert.

3.2 Komponenten von ARIADNE

ARIADNE ist aufgabenorientiert, zielbasiert und turnbasiert. Als semantische Repräsentation verwendet das Systeme *multidimensionale typisierte Merkmalsstrukturen*. Es nimmt weiterhin eine Trennung von aufgabenabhängigem Wissen und interaktionsspezifischem Wissen vor, was durch Modularisierung des Systems unterstützt wird.

Dieses Kapitel beschreibt den Aufbau von *ARIADNE* und die Trennung in aufgabenabhängiges Wissen und interaktionsspezifisches Wissen. Das Gesamtsystem, in das *ARIADNE* eingebettet wird, besteht aus Spracherkennung, Sprachverarbeitung (semantische Analyse und Konvertierung), Dialogmanagement, Generierung, Spracherzeugung und Sprachausgabe. *ARIADNE* deckt hierbei den Bereich von der Sprachverarbeitung bis zur Generierung ab. Das System kann, über eine definierte Schnittstelle, Methoden einer Anwendung aufrufen, um Aktionen auszuführen. Wann diese Aktionen ausgeführt werden, wird unter Zuhilfenahme der Wissensquellen durch das Dialogmanagement ermittelt. Die verschiedenen Wissensquellen werde im folgenden beschrieben.

3.2.1 Aufgabenabhängiges Wissen

Das aufgabenabhängige Wissen lässt sich in 5 Wissensquellen unterteilen:

- **Ontologie**: beschreibt die Konzepte, die das Dialogsystem 'versteht'
- **Dialogzielbeschreibungen**: beschreiben die Aufgaben, die das Dialogsystem ausführen kann

- **Satzanalyse-Grammatiken**: welche die Spracherkennerausgabe mit Hilfe struktureller Beschreibungen in eine semantische Repräsentation einer typisierten Merkmalslogik umwandeln; und auch vom Spracherkenner zur Dekodierung benutzt werden
- **Datenbankbeschreibungen**: welche die Referenz von Nominalphrasen auflösen
- **Generierungsgrammatiken**: welche Konzepte der typisierten Merkmalslogik zurück in natürlichsprachliche Ausdrücke umformen

Das Zusammenspiel der Wissensquellen ist in Abbildung 3.1 dargestellt.

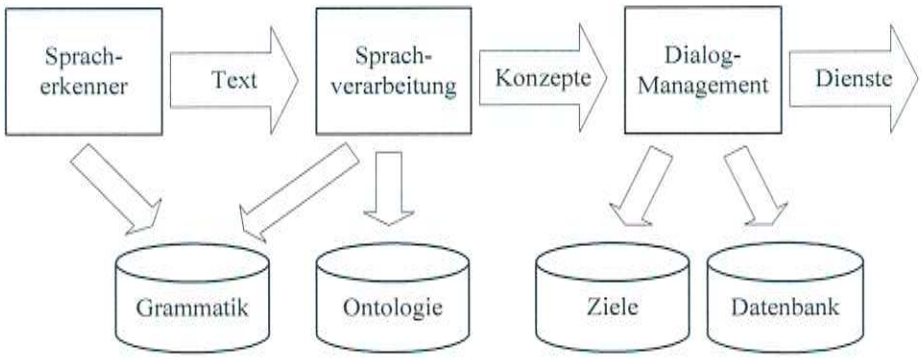

Abbildung 3.1: Verwendung der Wissensquellen im Dialogmanager

Ontologie und Domänenmodell

Das Domänenmodell legt fest, welche Konzepte das System kennt und wie sie zusammengehören. Es ist als Ontologie von Objekten, Aktionen und Eigenschaften aufgebaut. Die Elemente der Ontologie können voneinander erben (siehe Abbildung 3.2).

Eingabe-Grammatiken

Die Eingabe (Hypothese vom Spracherkenner) wird entsprechend den Regeln einer semantischen Grammatik zerteilt. Im Gegensatz zu einer reinen syntaktischen Grammatik, enthält eine semantische Grammatik syntaktische und semantische Informationen. Während syntaktische Information, z.B. zur Zerlegung einer Nominalphrase, domänenunabhängig ist, ist die semantische Information domänenspezifisch. Um beide Teile gemeinsam nutzen zu

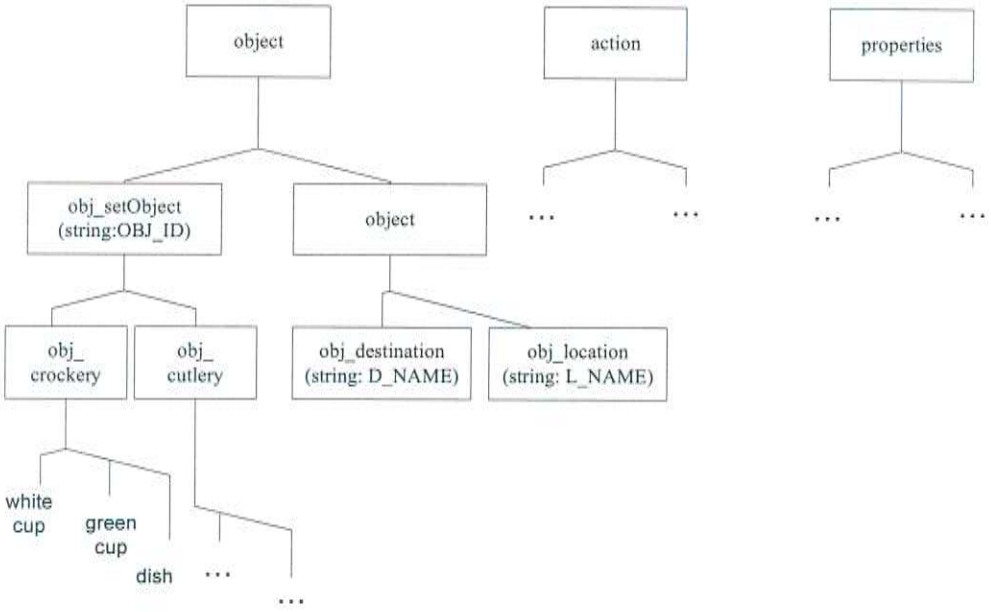

Abbildung 3.2: Teil einer Ontologie. Die Vererbung erfolgt von allgemeinen Konzepten zu anwendungsspezifischen

können, ohne dass die domänenunabhängige Information für jede Anwendung neu geschrieben werden muss, können *vektorisierte kontextfreie Grammatiken* [Den00b] verwendet werden. Entsprechend diesem Konzept, kann syntaktische Information für neue Anwendungen wiederverwendet werden. Der semantische Teil, welcher in den meisten Projekten allerdings den größeren Umfang hat, muss neu erstellt werden.

3.2.2 Semantische Repräsentation

Das *ARIADNE*-System verwendet zur semantischen Repräsentation spezielle Graphstrukturen, die auf *typisierten Merkmalsstrukturen* [Car92] aufbauen. Die Graphstrukturen heißen *multidimensionale Merkmalsstrukturen* und [Den00a] erlauben es, verschiedene Modalitäten der Eingabe zu modellieren. Die multidimensionale Merkmalsstrukturen sind eine Erweiterung der typisierten Merkmalsstrukturen [Den02]. Semantische Einträge werden als Knoten der Graphstrukturen repräsentiert. Die Knoten der Merkmalsstrukturen haben ein Typsymbol und entstammen einer endlichen Menge, auf der eine Halbordnung definiert ist und sich *Typenhierarchie* nennt. Im Dialog wird Information über mehrere Äußerungen oder Gesprächsbeiträge (Turns)

gesammelt. Daher muss die semantische Repräsentation (i) einen Vergleich bezüglich des Informationsgehaltes (ii) Unifikation von Repräsentationen (iii) Konsistenzprüfungen und (iv) Zurücknahme von Repräsentationen zulassen. Dies ist für typisierte Merkmalsstrukturen und auch für multidimensionale typisierte Merkmalsstrukturen erfüllt [Den02].

3.2.3 Interaktionsspezifisches Wissen, Modularisierung und Aufteilung in Schichten

Das Dialogsystem ist grob in drei Schichten unterteilt (siehe hierzu auch Abbildung 3.3). Die unterste Schicht enthält die fünf Wissensquellen des aufgabenabhängigen Wissens. Die zweite und dritte Schicht enthält Informationen zur Dialog-Ablaufplanung. Die zweite Schicht enthält die Interaktionsmuster

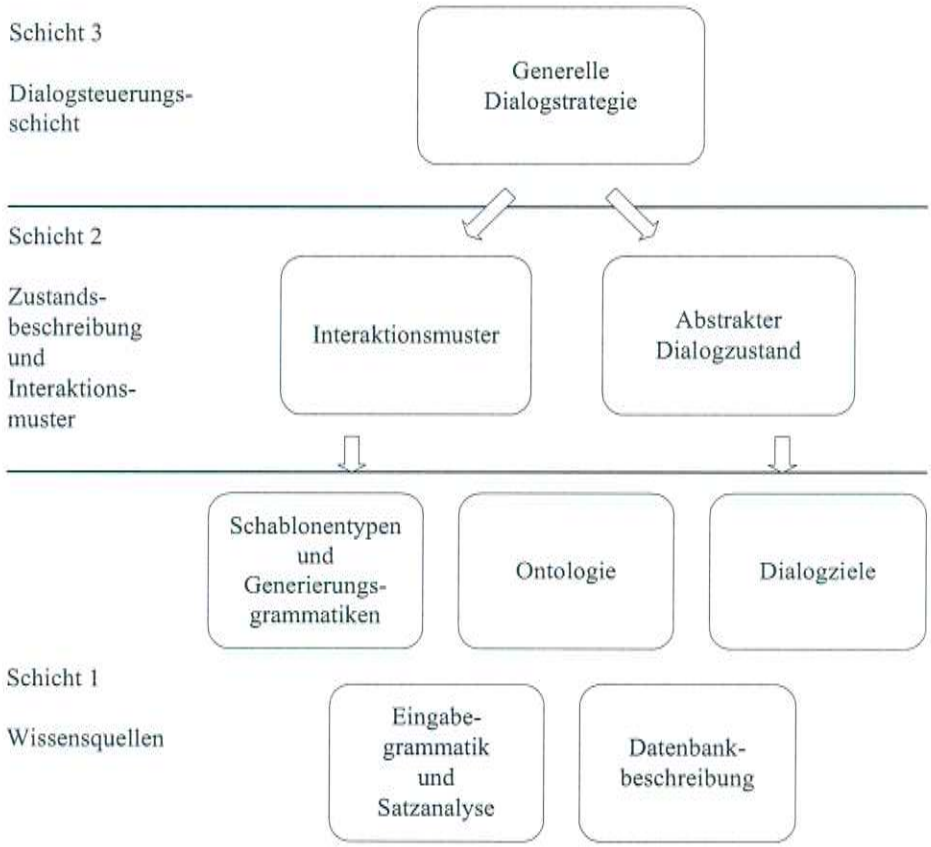

Abbildung 3.3: Aufteilung des Dialogmanagers in drei Schichten

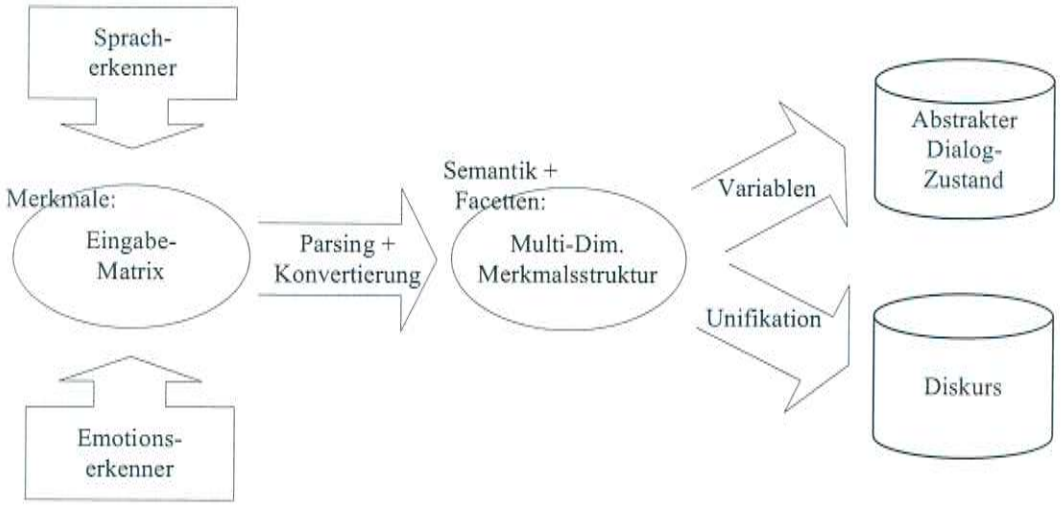

Abbildung 3.4: Abfolge der Eingabeverarbeitung

und den abstrakten Dialogzustand, während die dritte Schicht (die Dialog-Steuerungs-Schicht) die Beschreibung der abstrakten Dialogstrategie enthält.

Die Art und Weise, wie das System Wissen erlangt, Wissen vermittelt und Information im Diskurs anpasst, wird durch *Interaktionsmuster* und die Dialog-Steuerungs-Schicht definiert. Diese sind in 3.5 und 3.6 weiter beschrieben.

3.3 Eingabeverarbeitung, Facetten und Merkmalsmatrizen

Jeder Eintrag einer multidimensionalen Merkmalsstruktur ist ein $n-$ dimensionaler Vektor. Beim Start des Dialogsystems wird n fest definiert und damit die Anzahl der Elemente, deren Typ und deren Reihenfolge festgelegt. Die Elemente des Vektors heißen Facetten und werden in der Anwendungskonfiguration deklariert (siehe Abschnitt 3.7). Sei der Vektor $\vec{v}$ ein Knoten der Merkmalsstruktur mit n Elementen. Jedes $\vec{v_f}$, $1 < f \leq n$ bezeichnet eine Facette. Dann gibt es auch eine Eingabematrix mit n Zeilen, aus der die Facetten berechnet werden. Die Komponenten sind in Abbildung 3.4 gezeigt.

3.3.1 Eingabe

Die Eingabe erfolgt in folgendem Format: Sei $w = w_1, .., w_k$ die Hypothese des Spracherkenners. Dann ist das Eingabeformat eine Matrix M der Dimension

$k \times n$. Die erste Zeile $M_{1,i}$ enthält die Hypothese des Spracherkenners. Jede weitere Zeile f enthält die Werte $M_{f,i}$, $1 \leq i \leq n$, wobei der Wert mit Index i auf das i-te Wort der Hypothese w_i bezogen ist und zur Berechnung der Facette f: $\vec{v_f}$ verwendet wird.

Zum Zerteilen der Eingabe (Parsen) wird der Top-Down Parser SOUP [Gav00] verwendet. Da der Parser reine kontextfreie Grammatiken verwendet, werden die vektorisierten Grammatiken des Dialogmanagers in reine kontextfreie Grammatiken umgewandelt [Den02]. Der Parser berechnet einen Zerteilungsbaum auf w, auf dem dann die Konvertierung in die Merkmalsstruktur F durchgeführt wird.

3.3.2 Konvertierung

Die Konvertierung erzeugt aus dem Zerteilungsbaum anhand der Konvertierungsregeln die semantische Repräsentation in Form von Merkmalsstrukturen. Die Abbildung lässt sich in folgender Definition beschreiben: Sei s ein Knoten im Zerteilungsbaum, der auf den Merkmalsknoten t abgebildet wird. Desweiteren sei s in der Hierarchie des Zerteilungsbaum eine Abstraktion über die Wörter $w_a, \ldots, w_b$; $a \leq b$, $w_a, \ldots, w_b$ ist eine Subsequenz von w . Dann ist t_1 die Semantische Repräsentation des Knotens s und die Facette $\vec{v_f}, f > 1$ wird berechnet aus der Unifikation von $\{M_{f,a}, \ldots, M_{f,b}\}$.

3.4 Abstrakter Dialogzustand

Ein Ziel des Dialogsystems *ARIADNE* sind sprach- und domänenunabhängige Dialogstrategien. Hierfür wird eine Abstraktion des aktuellen Dialogzustandes im Dialogmanager berechnet. Diese informationelle Klassifikation des Dialogzustandes ist spezifisch für dieses Dialogsystem. Die Beschreibung des Dialogzustandes setzt sich aus verschiedenen Zustandsvariablen zusammen, von denen jede einzelne einen Aspekt des Dialogverlaufs beschreibt und berechnet. Die Zustandsvariablen sind (*i*) *TurnQuality*, (*ii*) *OverallQuality*, (*iii*) *SpeechAct*, (*iv*) *Reference*, (*v*) *ReferringExpressions*, (*vi*) *Intention* (siehe auch Tabelle 3.1). Die Variablen *Intention*, *TurnQuality*, und *OverallQuality* sind für die weitere Arbeit kurz erklärt. Die anderen Variablen sind in [Den02] genauer beschrieben. In Kapitel 4.4 werden weitere Variablen beschrieben, die für die emotionssensitive Dialogverarbeitung eingeführt wurden.

Intention - Variable

Die Variable Intention beschreibt den Fortschritt des Dialogs bezüglich der akquirierten Information. Die Variable modelliert somit, in wieweit die im

Zustandsvariable	Werte	Bedeutung
TurnQuality	*good*, *inconsistent*, *lowconfidence*, *poor*	Qualität der Repräsentation der aktuellen Äußerung
OverallQuality	*good*, $intermediate_1$, $intermediate_2$, *poor*	Qualität der Gesamt-Dialogs
SpeechAct	*sa_answer*, *sa_yesnoquestion*, ...	Sprechakt der aktuellen Äußerung
Reference	*neutral*, *selected*, *determined*, *finalized*	Zustand des Prozesses des Auflösens von Referenzen
ReferringExpressions	*NoReference*, *ImperfectReference*, *UniqueReference*, *AmbiguousReference*	Zustand der referierenden Ausdrücke
Intention	*neutral*, *selected*, *determined*, *finalized*, *deselected*	Grad zu dem die Intention des Anwenders bestimmt werden konnte

Tabelle 3.1: Überblick über die Zustandsvariablen des Dialogsystems *ARIADNE*, siehe [Den02]

Diskurs vorhandene Information die Absicht des Benutzers beschreibt. Es gibt fünf Zustände [Den02]:

1. *neutral*: Im Diskurs ist keine Information repräsentiert.

2. *selected*: Die im Diskurs vorhandene Information ist kompatibel mit der Dialogzielbeschreibung, und es gibt wenigstens eine weitere Dialogzielbeschreibung, die mit der Information im Diskurs kompatibel ist. Die Absicht des Benutzers kann nicht genau bestimmt werden, sie muss durch weitere Information disambiguiert werden.

3. *deselected*: Die Dialogzielbeschreibung ist inkompatibel mit der Information im Diskurs. Die Absicht des Benutzers kann nicht bestimmt werden, es existieren widersprüchliche Informationen.

4. *determined*: Die Information im Diskus ist kompatibel mit der Dialogzielbeschreibung, und es gibt keine weiteren Dialogzielbeschreibungen, die mit der Information im Diskurs kompatibel sind. Die Absicht des Benutzers wird damit zumindest unverwechselbar bestimmt, weitere

Informationen sind aber nötig um die verlangten Informationen der Dialogzielbeschreibung zu bekommen.

5. *finalized*: Die Absicht des Benutzers kann eindeutig bestimmt werden, und die mit dem Dialogziel verbundenen Aktionen können ausgeführt werden.

Geht man von dem Fall aus, dass im Dialog monoton neue Information hinzu kommt und keine Information aus dem Diskurs gelöscht wird, so sind für die Variable Intention die in Abbildung 3.5 dargestellten Zustandsübergänge möglich.

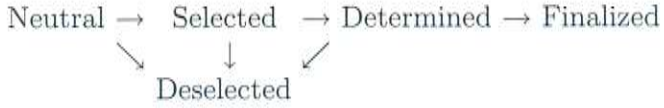

Abbildung 3.5: Zustände und Zustandsübergänge der Variable Intention bei monoton wachsender Spezifizität [Den02]

TurnQuality - Variable

Die Variable *TurnQuality* beschreibt die Qualität der aktuellen Eingabe, genauer gesagt, die Qualität der semantischen Repräsentation der Eingabe, nach einem intern definierten Maß. In ihrer ursprünglichen Beschreibung [Den02] bezieht sich die Variable auf Eigenschaften der Spracherkennung, die die Eingabe produziert. Die Definition der Zustände ist in ihrer Beschreibung jedoch so allgemein gehalten, dass sie auch für weitere Modalitäten gelten kann. Die Werte der Variablen sind (*i*) *good_quality*, (*ii*) *low_confidence*, (*iii*) *partially_inconsistent_representation* und (*iv*) *poor_quality*.

OverallQuality - Variable

Im Gegensatz zur Variablen TurnQuality, die nur Qualität der aktuellen Eingabe beschreibt, beschreibt die Variable OverallQuality die Gesamtqualität des fortschreitenden Dialogs. Die Variable kann die folgenden Werte, sortiert von schlecht nach gut annehmen: (*i*) *poor*, (*ii*) *intermediate*1, (*iii*) *intermediate*2 (*iv*) *perfect*.

3.5 Dialogziele und Dialogstrategie

Dialogziele beschreiben, welche Information nötig ist, um bestimmte Aufgaben zu erfüllen. Bei einfachen Aufgaben, kann die Information in einer Äußerung des Benutzers angegeben werden. Wenn die existierende Information nicht ausreicht, regeln Dialogstrategien, wie die fehlende Information angesammelt wird [DW97].

Die Dialogstrategie gibt an, welche Aktionen in Abhängigkeit des Dialogzustandes getroffen werden. Sie besteht aus einem allgemeinen Steuerungsteil, der bestimmt, wann welche Variablen neu berechnet werden, wie die Eingabe des Benutzers interpretiert wird und welche Aktionen ausgeführt werden. Dies beinhaltet, den Diskurs entsprechend den neuen Informationen anzupassen, Datenbankanfragen durchzuführen, Aktionen eines finalisierten Dialogziels auszuführen und eventuell weitere Klärungsfragen zu stellen. Um den Diskurs entsprechend der Interaktion anzupassen, hat die Dialogstrategie verschiedene Interaktionsmuster und Instantiierungen von Interaktionsmustern zur Auswahl. Interaktionsmuster definieren, welche Information ausgetauscht wird. Instantiierungen von Interaktionsmuster werden durch die Anwendung definiert und beschreiben, wie diese Information ausgetauscht wird. Klärungsfragen sind Fragen des Dialogsystems, die an den Benutzer gestellt werden, mit der Absicht, weitere Informationen im Diskurs zu erlangen, oder widersprüchliche oder falsche Information zu ersetzen.

3.6 Interaktionsmuster

Neben dem allgemeinen Steuerungsteil hat man auch über die Anwendungsbeschreibung die Möglichkeit, den Dialogfluss zu beeinflussen. Grundlage hierfür sind die Interaktionsmuster. Es gibt vier verschiedene Grundtypen, die von der Dialogstrategie gewählt werden können:

- Fragen-Interaktionsmuster (fügt Information zum Diskurs hinzu)
- Korrektur-Interaktionsmuster (löscht Information aus dem Diskurs)
- Anpassen-Interaktionsmuster (ersetzt Information im Diskurs mit neuer Information)
- Zustands-Interaktionsmuster (interne Zustandsübergänge im Dialogsystem)

Welche Interaktionsmuster von der Dialogstrategie gewählt werden, findet man in [Den02] beschrieben. Instantiierungen von Interaktionsmuster erfolgen in der Anwendungsbeschreibung. Jedes Interaktionsmuster hat eine Menge von Schablonen, die von der Dialoganwendung verwendet werden können. Eine Schablone definiert im Allgemeinen Vorbedingungen und Aktionen. Die Vorbedingungen werden verwendet, um zu überprüfen, ob ein Interaktionsmuster ausgeführt werden kann oder nicht. Die Aktionen definieren Methodenaufrufe an die Anwendung und Aufrufe weiterer Dienste, die das Dialogsystem anbietet. Als Beispiel für Schablonen sind im Anhang auf Seite 93 die Klärungsfragen des Fahrplan-Informationssystems, wie es in Kapitel 5.2 beschrieben wird angegeben.

3.7 Beschreibung der Wissensquellen und der Dialoganwendung

Die Konfiguration einer Dialoganwendung enthält die Information, wie die Eingabe in das System erfolgt, wie die Ausgabe über Spracherzeugung erfolgt und welche Wissensquellen eingebunden werden. Weiterhin enthält sie die Information, welche Module geladen werden, um Variablen, Facetten und Vorbedingungen zu definieren und zu berechnen. Die Wissensquellen enthalten die Ontologie, die Dialogzielbeschreibungen, Satzanalyse-Grammatiken, Datenbankbeschreibungen, Schablonentypen und Generierungsgrammatiken. Sie bilden die Beschreibung der Dialoganwendung.

3.7.1 Ontologie

Die Ontologie beschreibt die semantischen Rahmen und Objekte, die in der Domäne existieren. Folgendes Beispiel repräsentiert ein Musikstück als Objekt (Quelle: *ARIADNE* Dokumentation / Tutorial):

```
class obj_song inherits generic:object {
  base:string : FILENAME;
  base:string : ARTIST;
  base:string : STYLE;
  base:string : ALBUM;
};

class act_player  inherits generic:action;

class act_playsong inherits act_player {
```

```
    obj_song   : generic:ARG;
};
```

3.7.2 Dialogzielbeschreibungen

Die Dialogzielbeschreibung definiert, welche Information im Diskurs vorhanden sein muss, um eine Aufgabe ausführen zu können. Zusätzlich enthält es die Beschreibung, welche Aktion(en) ausgeführt werden sollen. (Quelle: *ARIADNE* Dokumentation / Tutorial):

```
goal PlaySong {
  precondition:
    [ act_playsong
      generic:ARG    [ obj_song
                       generic:NAME   [ base:string ]
                       FILENAME       [ base:string ]
                     ]
    ]
  ->
  bindings:
    jpkg://localhost:5454/playSong individual:
                          $objs.[generic:ARG|FILENAME] ,
                          $objs.[generic:ARG|generic:NAME];
};
```

3.7.3 Datenbankbeschreibungen

Datenbankinformationen werden entweder in einer Access-Datenbank abgelegt, oder über eine Applikation angebunden. Die Einbindung erfolgt wie im Quelltextsegment gezeigt. In dem gegebenen Beispiel wird eine Java-Applikation verwendet, die Informationen über verfügbare Musikstücke liefert. Wie die Applikation die Daten speichert bleibt hinter der Schnittstelle verborgen.

```
database Songs obj_song jpkg://localhost:5454/Songs?jpkg  {
  dbtable Songs obj_song {
    dbfield Name = [FILENAME];
  };
};
```

Kapitel 4

Entwurf und Architektur

Wir haben im letzten Kapitel das Dialogsystem *ARIADNE* und dessen Eigenschaften kennen gelernt. Dieses Kapitel beschreibt das vom Autor erstellte Rahmenwerk für emotionssensitive Dialogverarbeitung. Es werden die Verfahren zur Emotionsbehandlung der Eingabe (entsprechend der Sprachverarbeitung) des Dialogsystems, Verfahren für das Dialogmanagement und die Erweiterung durch neue Variablen vorgestellt.

4.1 Gesamtarchitektur

Der Dialogmanager ist in ein Gesamtsystem eingebettet, das auf gesprochene Sprache reagiert und mit gesprochener Sprache antwortet. Abbildung 4.1 zeigt die Gesamtarchitektur der beteiligten Komponenten. Basierend auf dieser Architektur werden in den folgenden Abschnitten Konzepte für Emotionen beschrieben.

- Der Benutzer: kommuniziert durch Sprache mit dem System. Das System empfängt vom Benutzer ein digitalisiertes Sprachsignal und sendet ein synthetisiertes Sprachsignal an den Benutzer.

- Der Spracherkenner: verwendet die digitalisierte Sprache und erzeugt einen Text als Hypothese.

- Der Emotionserkenner: verwendet ebenfalls die digitalisierte Sprache als Eingabe und zusätzlich das Ergebnis des Spracherkenners, um auch textbasiert Rückschlüsse über Emotionen des Benutzers machen zu können. Die Ausgabe von Emotionserkenner und Spracherkenner wird in ein spezielles Eingabeformat für den Dialogmanager fusioniert.

- Der Parser: bekommt die fusionierte Eingabe von Spracherkenner und Emotionserkenner und analysiert die Spracheingabe. Er erstellt anhand der Eingabegrammatik einen semantischen Zerteilungsbaum, welcher durch die Konvertierungsregeln in eine multidimensionale Merkmalsstruktur (TFS) überführt wird.

- Der Dialogmanager: erhält als semantische Eingabe eine multidimensional Merkmalsstruktur. Er verwendet die Informationsquellen, wie in Kapitel 3 beschrieben, sendet eine Antwort an die Sprachsynthese und führt definierte Aktionen aus.

- Die Sprachsynthese (Text-to-Speech): erhält die Antwort des Dialogsystems in Textform und erzeugt ein Sprachsignal, das über die Soundkarte an den Benutzer ausgegeben wird.

- Die Applikation: empfängt vom Dialogmanager Befehle. Die Applikation kann entsprechend ihrem Einsatz unterschiedliche Aufgaben erfüllen und z.B. eine Befehlssteuerung für einen Roboter implementieren.

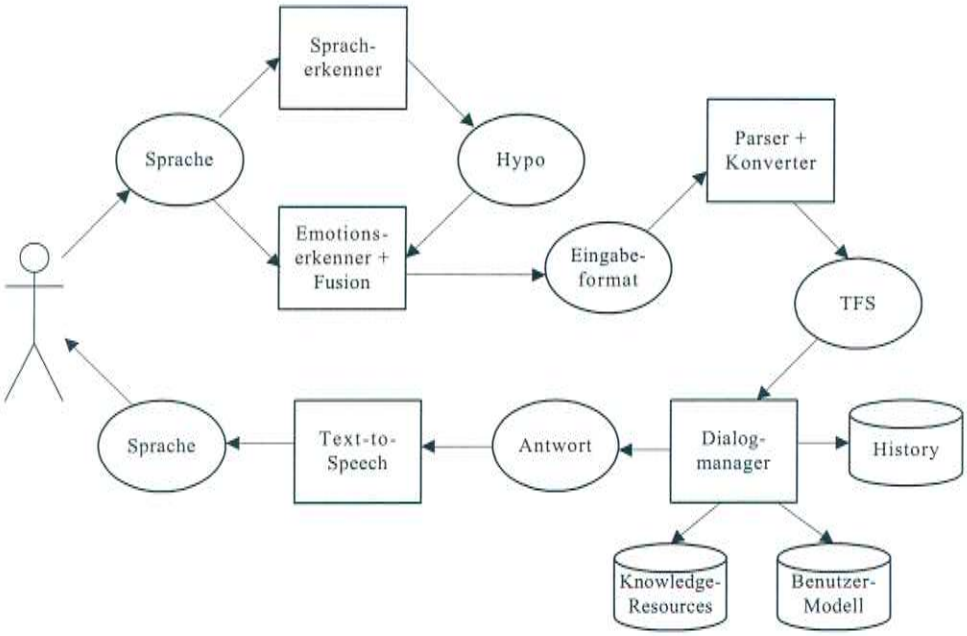

Abbildung 4.1: Gesamtarchitektur

4.2 Integration von Emotionen in der Eingabe

Das Verstehen von Emotionen soll dem Dialogsystem dabei helfen, zusätzliche Information vom Benutzer zu erlangen. Was Emotionen sind und durch welche Modelle sie erfasst werden können, wurde bereits in Kapitel 2 ausführlich behandelt. Dieser Abschnitt beschreibt, welche Aspekte von Emotionen berücksichtigt wurden und wie diese strukturell in das Dialogsystem integriert sind.

4.2.1 Emotionen als Modalität der Spracheingabe

Wenn man ein Sprachsignal als Eingabe verarbeitet, ist hierbei nicht nur interessant, *was* der Benutzer sagt, sondern auch *wie* der Benutzer etwas sagt. Zum einen können Emotionen Auswirkungen auf die Sprach- und Wortwahl des Benutzers haben, was z.B. durch eine *Emotionale Semantische Grammatik* (siehe Abschnitt 4.2.4), oder durch ein anderes Sprachmodell modelliert werden kann (hierbei wird das *was* betrachtet). Zum anderen haben Emotionen des Sprechers auch besonders Auswirkungen auf die Prosodie und andere akustische Merkmale (hierbei wird das *wie* betrachtet). Dieser Fall wird durch einen eigenen Emotionserkenner behandelt, wie bereits in Abbildung 4.1 integriert dargestellt.

Neben den Auswirkungen auf das Sprachsignal, hat der emotionale Zustand des Benutzers auch Einfluss auf andere Faktoren, wie z.B. den Gesichtsausdruck. Genauso wie die die akustischen Merkmale zur Emotionsklassifikation herangezogen werden, lässt sich auch ein Emotionserkenner verwenden, der die Emotion aufgrund des Gesichtsausdrucks oder einer Kombination mehrerer Faktoren erkennt. Dieses Klassifikationsergebnis wird als emotionaler Zustand des Benutzers zum Zeitpunkt seiner Aussage interpretiert und daher als Modalität der Spracheingabe verarbeitet.

4.2.2 Emotionen als selbständige Modalität

Allgemein ist emotionaler Ausdruck nicht unbedingt an eine Sprachnachricht gekoppelt. Zum Beispiel, für die Modellierung von Rückmeldungen, kann visuelle Information verwendet werden, ohne dass der Benutzer etwas sagt. Auch für Emotionen in Sprache kann die Modellierung über eine selbständige Modalität Sinn machen. Zum Beispiel werden Ausrufe des Erstaunens 'Oh', Interesse bekundendes 'aha', bewunderndes 'mhm' oder zustimmendes 'aha' unter Umständen von einem Spracherkenner nicht als korrekte Textsegmente,

sondern als Störgeräusche erkannt. Dann liefert der Spracherkenner keine Ausgabe, aber ein auf Sprache basierender Emotionserkenner könnte eine Emotion klassifizieren. Eine eigene Modellierung der emotionalen Eingabe ist also nötig.

Um Emotionen, die nicht an eine Spracheingabe gekoppelt sind, zu erfassen, muss man die emotionale Eingabe als selbständige Modalität beschreiben. Die Emotionsparameter sind in diesem Fall nicht als Facetten eines semantischen Eintrags zu verstehen, sondern werden eigenständig berechnet und durch eigene semantische Einträge erfasst. Diese Modellierung erfolgt analog zur Fusion weiterer Modalitäten, wie z.B. Gestik. Wie semantische Eingaben verschiedener Modalitäten, besonders im Hinblick auf zeitliche Zusammengehörigkeit, fusioniert werden können, wird zur Zeit an unserem Lehrstuhl erforscht und wird nicht in dieser Arbeit behandelt. Im Gegensatz dazu können linguistische Merkmale als eigene semantische Werte integriert werden. Die Modellierung ist in Abschnitt 4.2.4 beschrieben.

4.2.3 Facetten

Wir haben *Facetten* bereits in Kapitel 3.3 als Elemente der Semantik kennen gelernt. Facetten bieten die Möglichkeit, weitere Merkmale der Sprache in das selbe semantische Element mit aufzunehmen. Sie eignen sich damit hervorragend, um Emotionen als Modalitäten der Spracheingabe zu modellieren.

Emotionsfacette

Das Ziel der Implementierung ist schließlich, den emotionalen Zustand des Benutzers so zu modellieren, dass er den Zustand des Benutzers zum Zeitpunkt seines Turns am besten beschreibt. Daher wurde in dieser Arbeit die Annahme gemacht, dass der Emotionswert für die gesamte Spracheingabe konstant bleibt. Eine experimentelle Bestätigung hierfür ist in Kapitel 6 zu finden. Deshalb wird auch nur ein Emotionswert für die gesamte Hypothese berechnet. Um der Definition aus Kapitel 3.3 aber gerecht zu werden, wird in jede Spalte der Eingabematrix der selbe Emotionswert geschrieben. Sei also O.B.d.A. $\vec{v_2}$ eine Emotionsfacette eines Merkmalsknotens, der durch die Eingabe erzeugt wird. Dann gilt $M_{2,1} = M_{2,i}\ \forall 1 \leq i \leq n$.

Der Wert der Emotionsfacette entstammt einem Emotionsmodell. Somit ist die Definition und Berechnung der Facettenwerte erstens von diesem Modell und zweitens auch von dem verwendeten Emotionserkenner abhängig. Allerdings soll die Berechnung der Facette zumindest eine Abstraktion über

verschiedene Emotionserkenner bilden, um als Schnittstelle zu den Variablen nur die Varianz durch verschiedene Emotionsmodelle zu haben.

Ein neuer Emotionserkenner lässt sich relativ leicht in das System integrieren, wenn sich die Ausgabe des Erkenners durch Konvertierung auf das Modell eines existierenden Erkenners abbilden lässt. Besonders wenn der Emotionserkenner ein kontinuierliches Modell verwendet, wie z.B. Arousal-Valence, sind zumindest kleinen Varianzen zu erwarten. Es muss daher für jeden Erkenner eine spezielle Diskretisierung, wie in Abbildung 4.2 gezeigt, vorgenommen werden. Die Emotionsfacette verwendet als Ausgabewerte nur die diskretisierten Werte.

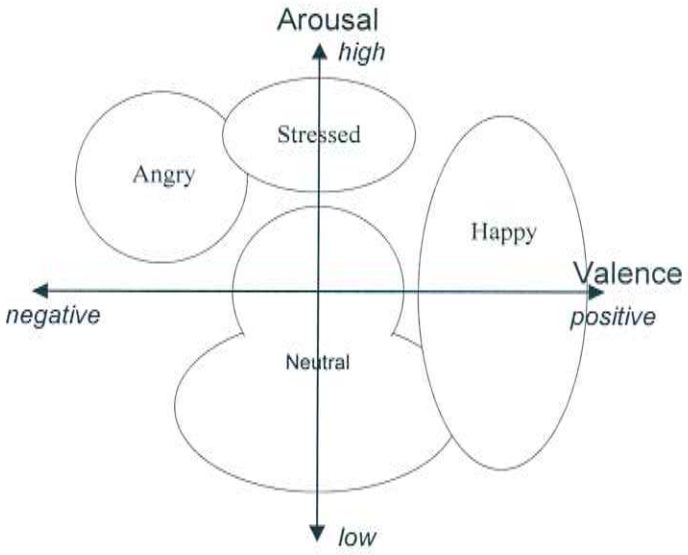

Abbildung 4.2: Diskretisierung der Arousal-Valence Ebene

4.2.4 Emotionale Semantische Grammatiken

Abschnitt 4.2.2 beschreibt die Existenz von Emotionen als selbständige Modalität. *Emotionale Semantische Grammatiken* sind semantische Grammatiken, mit zusätzlichen Konvertierungsregeln für emotionale semantische Werte. Sie können z.B. dazu verwendet werden, Auswirkungen von Emotionen auf die Satzstruktur normaler gesprochener Sätze zu erfassen. Die Satzstruktur, die z.B Schimpfwörter enthält, hat direkte Auswirkungen auf den Zerteilungsbaum (siehe Abbildung 4.3). Entsprechende Konvertierungsregeln der

semantischen Grammatik überführen diesen Zerteilungsbaum in eine semantische Repräsentation mit Emotionswerten. Die Modellierung von semantischer Bedeutung des Satzes und der Modellierung von Emotionen wird dabei vermischt. Dieses Konzept ist orthogonal zur Verwendung von Variablen. Mit emotionalen semantische Grammatiken kann z.B. auch Affekt bezüglich Objekten ausgedrückt werden. Zum Beispiel "Würden Sie mir freundlicherweise sagen, warum diese doofe Maschine nicht funktioniert?" enthält ein negatives Attribut für die Maschine. Dieses wird auch als solches in der semantischen Repräsentation dargestellt. Hingegen muss der emotionale Zustand des Benutzers, der durch die Emotionsvariablen modelliert wird, davon nicht betroffen sein.

```
<obj_thing,NP,_> =  <hlp_det>* <modif,A,_> 'thing';
<hlp_det> = 'the'
          : 'a';
<modif,A,_> =
    'stupid' { EMO "negative" }
  : 'beautiful' { EMO "positive" };
```

Abbildung 4.3: Grammatikfragment mit emotionalen Zusätzen

Die Grammatiken können auch dazu verwendet werden, Rückmeldungen aufzunehmen. Abbildung 4.4 zeigt eine Regel der semantischen Grammatik, die Rückmeldungen mit den zugehörigen semantischen Werten modelliert. Die semantische Repräsentation der Rückmeldungspartikel enthält alleine nur eine schwache Auskunft über Emotionen. Diese können zusätzlich über einen externen Emotionserkenner geliefert und über Facetten und Variablen integriert werden, wie schon in den letzten Abschnitten beschrieben. Die Anwendungsbeschreibung kann dann die semantische Information und die Werte der Variablen kombinieren.

```
public <backch,_,_> =
    'mhm'  { BCKCH_ID "mhm" }
  : 'ohh'  { BCKCH_ID "ohh" }
  : 'well' { BCKCH_ID "well" };
```

Abbildung 4.4: Grammatikalische Regel für Rückmeldungspartikel mit semantischen Werten

4.3 Emotionsparameter im Dialogmanagement

4.3.1 Dialogziele

Anwendungsfälle

Im letzten Kapitel wurde die Funktionsweise von Dialogzielen beschrieben. Wenn man die Dialogziele abhängig vom emotionalen Zustand des Benutzers definiert, ergeben sich z.B. folgende Möglichkeiten, zwischen unterschiedlichen Dialogzielen auszuwählen:

- je nachdem ob der Benutzer während des Dialoges eher fröhlich oder wütend war, kann man nach Erreichen des Zieles eine Entschuldigung aussprechen, nach Verbesserungswünschen oder positivem Feedback fragen.

- wenn der Benutzer eines Beratungs- oder Verkaufssystem über die Firma oder das System sehr verärgert war, kann dem Benutzer nach Beendigung des Gesprächs ein kleiner Rabatt gewährt werden. Ähnliches wird bei Fluggesellschaften schon eingesetzt, wenn der Benutzer eine bestimmte Zeitdauer in der Warteschlange am Telefon verbringen muss.

- abhängig davon, welche Informationen der Benutzer mit überwiegender Selbstsicherheit eingegeben hat oder bei welchen Informationen er sich unsicher war, kann man die gesammelte Information unterschiedlich bewerten. Das ist interessant für Tutor-Systeme, um herauszufinden, welche Fragen zwar richtig beantwortet wurden, aber durch weitere Lernaufgaben weiter gefestigt werden müssen.

Implementierung

Dialogziele haben Vorbedingungen und semantische Rahmen, die gesetzt sein müssen, damit das Dialogziel ausgeführt werden kann. Die Vorbedingungen können ebenfalls über Werte von Variablen definiert werden.

Weiterhin sind Dialogziele abhängig von Vorbedingungen, welche über Variablen des *Abstrakten Dialogzustandes* definiert werden können. Die Verwendung von Emotions-Variablen wird im nächsten Abschnitt beschrieben. Mit diesen Emotions-Variablen lassen sich also auch auf einfache Weise neue Dialogziele, in Abhängigkeit vom emotionalen Zustand des Benutzers, definieren und oben genannte Wünsche erfüllen. Beispiel siehe Abbildung 4.5.

```
goal Stop3 {
    precondition:
        [ act_no ]
    variable:(UserEmotion = stressed)
    ->
    bindings:
        jpkg://localhost:5454/rbStop;
};
```

Abbildung 4.5: Dialogziel mit einer Emotions-Variablen als Vorbedingung

4.3.2 Dialogstrategie

Anwendungsfälle

Die zweite Möglichkeit, Emotionen zu integrieren, zielt nicht auf das Endergebnis (das Dialogziel) ab, sondern auf den Weg, wie dieses erreicht wird. Hierbei wird der Dialogablauf von den Emotionen des Benutzers beeinflusst. Dies ist eine sehr mächtige Fähigkeit, da hiermit wesentlich spezifischere Aktionen ausgewählt werden können, als dies bei unterschiedlichen Dialogzielen der Fall ist. Ähnliches wird auch in [WLKA98] beschrieben, wobei der Schwerpunkt auf der Optimierung der Dialogstrategie liegt. [TW01] hingegen legt mehr Wert auf eine größere Benutzer-Zufriedenheit.

In beiden Fällen wird eine Optimierung des Dialoges vorgenommen. Eine Optimierung des Dialoges kann z.B. bedeuten:

- schneller ans Ziel zu kommen: wobei sowohl die absolute Zeit als auch die Anzahl der Turns gemessen werden kann.
- überhaupt ein Ziel zu erreichen: das Ziel wird z.B. auch verfehlt, wenn der Benutzer frustriert ist und den Dialog abbricht.
- die Benutzerzufriedenheit zu erhöhen: was einerseits teilweise schon implizit durch einen schnellen Ablauf des Dialogs erreicht werden kann, andererseits z.B. dadurch erreicht werden kann, dass man den Benutzer positiv stimmt, ohne dabei eine besonders gute Zeit-Effektivität zu erzielen.

Implementierung

Die Dialogstrategie ist auf dem *Abstrakten Dialogzustand* und dem Kontext der Anwendung definiert. In dieser Arbeit werden verschiedene neue Varia-

blen vorgeschlagen, die den *Abstrakten Dialogzustand* sinnvoll erweitern und damit der Dialogstrategie neue Möglichkeiten gewähren. Die Variablen werden in Abschnitt 4.4 definiert.

Die Variablen des *Abstrakten Dialogzustandes* werden sowohl von der grundlegenden Steuerungsstrategie des Dialogmanagers, als auch von den instantiierten Interaktionsmustern genutzt. In der Dialoganwendung werden Instantiierungen dieser Interaktionsmuster durch Schablonen definiert. Diese Schablonen haben ebenso wie die Dialogziele eine Reihe von Vorbedingungen, die definieren, ob ein solcher Schablonentyp ausgeführt werden darf. Auch hier können die Vorbedingungen über die Variablen des *Abstrakten Dialogzustandes* definiert werden, allerdings wird hiermit nicht in der Auswahl unterschiedlicher Dialogziele unterschieden, sondern Einfluss auf die Dialogstrategie genommen.

Mit Vorbedingungen in Schablonentypen kann also auf jede einzelne Eingabe des Benutzers, in Abhängigkeit des Emotions-Zustandes, eine entsprechende Antwort gegeben werden (siehe Abbildung 4.6) oder der Ablauf der Dialogstrategie geändert werden.

```
move NoParse on variable Quality changed to noParse {
    variable:(UserEmotion = angry)
    ->
    bindings:
      internal://dialogue/say #NoParseAngry;
      internal://dialogue/abort;
};
```

Abbildung 4.6: Schablonentyp mit einer Emotionsvariablen als Vorbedingung

4.4 Variablen

Variablen sind Teile des Abstrakten Dialogzustandes, wie im Grundlagenkapitel 3.4 beschrieben. Die existierenden Variablen des Dialogmanagers beschreiben Aspekte des Dialogverlaufs. Damit existiert keine explizite Modellierung des Benutzers. Die Emotionsvariablen sind der erste Ansatz für dieses Dialogsystem, die den Kontext "Benutzer" als Parameter des Dialogs mit in Betracht ziehen.

4.4.1 Benutzer-Emotion

Die Variable *Benutzer-Emotion* (implementiert als *UserEmotion*) modelliert den aktuellen emotionalen Zustand des Benutzers. Ihre Berechnung erfolgt auf Grundlage der Emotionsfacette, wobei die aktuelle Implementierung den Wert der Facette direkt übernimmt. Da schon die Facette eine Abstraktion über verschiedene Modelle von Emotionserkennern bildet, kann man die Variable als unabhängig vom jeweiligen Erkenner bezeichnen. Die Facette bildet auch eine Abstraktion über unterschiedliche Diskretisierungen von kontinuierlichen Modellen. Allerdings haben unterschiedliche Emotionsmodelle auch unterschiedliche Werte bzw. Emotionsnamen. Daher müsste eigentlich für jedes neue Emotionsmodell eine neue Variable implementiert werden. Um dem entgegen zu wirken, werden hier zwei Modelle für die Variable Benutzer-Emotion beschrieben: eine problemspezifische Definition für die Anwendungen, die in dieser Arbeit beschrieben werden und eine generische Möglichkeit, deren Werte in der Anwendungsbeschreibung definiert werden können.

Standardmodell

Das Standardmodell verwendet Emotionen der Arousal-Valence Ebene und der darauf basierenden Diskretisierung, wie in Abbildung 4.2 gezeigt. Die vier verschiedene Emotionszustände sind in Tabelle 4.1 abgebildet. Das Standardmodell enthält typische Emotionswerte, die für die Anwendungen, die in dieser Arbeit beschrieben werden geeignet sind.

Zustand	Beschreibung
neutral	Standardwert
happy	stellvertretend für alle positive Werte
angry	Wut oder Ärger, stark negative Einfluss
stressed	Stress oder Hektik

Tabelle 4.1: Die Werte der Variable *UserEmotion*, entsprechend der Arousal-Valence Diskretisierung

Generisches Modell

Das generisches Modell erlaubt es, die Emotionswerte in der Anwendungsbeschreibung zu definieren. Intern werden die Emotionen nur als Zahlen repräsentiert. Der Dialogentwickler kann den Emotionen selber Namen zuord-

nen. Dadurch wird ermöglicht, dass verschiedene Emotionsmodelle, mit einem entsprechenden Emotionserkenner, einfach integriert werden können. Dabei ist zu beachten, dass andere Variablen, die auf der Definition der Benutzer-Emotion basieren, wie z.B. die Emotionstendenz oder die System-Emotion, ohne Modifikation nicht verwendet werden können.

4.4.2 Emotionstendenz

Die Emotionstendenz $E_t(t)$ (implementiert als *EmotionTrend*) des Benutzers wird als Interpolation über die Historie der Variable Benutzer-Emotion berechnet. Die Variable kann die in Tabelle 4.2 dargestellten Zustände annehmen. Die fünf Werte repräsentieren die Einstellung des Benutzers gegenüber dem System und sind als *sehr negativ*, *negativ*, *neutral*, *positiv* und *sehr positiv* kategorisiert. Die Zeitpunkte sind diskret und bezeichnen einzelne Gesprächsbeiträge des Benutzers, wobei der Zeitpunkt $t + 1$ den Gesprächsbeitrag bezeichnet, der direkt nach dem Gesprächsbeitrag, zum Zeitpunkt t, folgt. $E_u(t)$ ist der Emotionswert des Benutzers zum Zeitpunkt t entsprechend den Werten in Tabelle 4.2. Er wird aus der Variable Benutzer-Emotion über eine Zuordnungstabelle berechnet. Die Berechnung ist vom Emotionsmodell für die Variable Benutzer-Emotion abhängig. Für das Standardmodell der Benutzer-Emotion (Tabelle 4.1) ist die Zuordnung in Tabelle 4.3 dargestellt.

Zustand	Bedeutung
strnegative	sehr negativ Einstellung
negative	negative Einstellung
neutral	normale oder neutrale Einstellung
positive	positive Einstellung
strpositive	sehr positive Einstellung

Tabelle 4.2: Die Werte der Variable Emotionstendenz

Linear Gewichtete Summe

Die Berechnung erfolgt als linear gewichtete Summe. Dabei werden die letzten $N0$ Zustände der Variable Benutzer-Emotion (mit linear abfallender Gewichtung) in Betracht gezogen.

$UserEmotion$	E_u
neutral	neutral
happy	strpositive
angry	strnegative
stressed	negative

Tabelle 4.3: Zuordnung der Benutzer-Emotion auf Werte der Emotionstendenz

4.4.3 System-Emotion

Die *System-Emotion* (implementiert als *SystemEmotion*) ist im Unterschied zur Benutzer-Emotion eine Applikations-spezifische Implementierung. In der ersten prototypischen Implementierung wurde nur die Benutzer-Emotion verwendet und auf eine System-Emotion verzichtet [HFDW02]. Die Variable System-Emotion ist aber aus folgenden Gründen sinnvoll:

- Bei Verwendung eines Avatars, kann die System-Emotion durch den Avatar dargestellt werden, um dem Benutzer die aktuelle Strategie zu veranschaulichen.
- Bei Erweiterung des Systems mit einer eigenen Persönlichkeit und emotionaler Ausdrucksstärke, repräsentiert die Variable System-Emotion den emotionalen Zustand des Systems.
- Die Berechnung der Variable System-Emotion bildet eine Abstraktion über die Reaktionen des Dialogsystems. Die Berechnung entspricht damit der Strategie, wie das System in unterschiedlichen Situationen reagiert, während der Wert der Variablen die Strategie vor dem Dialogentwickler "versteckt".

Die Abstraktion der System-Emotion über Reaktionen des Dialogsystems liefert dem System keine neue Funktionalität, bietet allerdings eine Vereinfachung für den Entwickler einer Applikation. Die Vereinfachung liegt darin, dass der Entwickler die Sprachausgabe und Klärungsfragen entsprechend der Variablen entwirft. Er muss nicht zusätzlich eine Strategie entwickeln, wie die Reaktion des System auf Emotionen des Benutzers ausfallen soll und dabei auf Konsistenz achten. Die Trennung durch die System-Emotion bildet zudem eine Reduktion des Parameterraumes im Sinne eines Lernalgorithmus. Es muss statt vieler einzelner Reaktionen des Systems in Abhängigkeit der Benutzer-Emotion nur die Berechnung der System-Emotion (die Strategie) optimiert werden.

Modell

Die Variable System-Emotion soll in der Lage sein, ein komplizierteres Modell zu ermöglichen als eine einfache Abbildung der Benutzer-Emotion auf Werte der System-Emotion. Dadurch können zu einem späteren Zeitpunkt komplexere Modelle implementiert werden, die weitere Faktoren berücksichtigen. Schließlich soll die Berechnung vor dem Dialogentwickler versteckt sein. Daher muss die Berechnung der Variable System-Emotion im Dialogsystem durchgeführt werden und kann nicht als offene Schnittstelle an die Dialoganwendung weitergegeben werden. Also muss der System-Emotion ein eindeutiges Modell der Benutzer-Emotion zugrunde gelegt werden. In unserer Implementierung haben wir uns für das Arousal-Valence Modell, mit der Diskretisierung wie oben beschrieben, entschieden. Dieses Modell bietet die in unseren Anwendungen grundlegenden Emotionstypen. Die Berechnung der System-Emotion ist als Abbildung in Tabelle 4.4 angegeben.

Die zugrunde liegende Strategie sieht vor, dass das System keinen negativen Zustand annimmt. Die folgende Beschreibung soll die Intention der Implementierung erläutern, die Verwendung in anderen Strategien kann davon natürlich abweichen. Der Wert *stressed* (als Reaktion auf Ärger des Benutzers) bezeichnet einen Zustand, wobei das System eigene Fehler als Ursache für den Ärger des Benutzers vermutet und eine entsprechende Vorgehensweise wählen soll, um die Fehler zu beheben. Als Reaktion auf Stress des Benutzers verhält sich das System sehr sachlich, aufgabenbezogen und soll Fehler zu vermeiden, indem häufiger Rückfragen gestellt werden. Die Emotionen *neutral* und *happy* werden eins zu eins von der Benutzer-Emotion auf die System-Emotion abgebildet.

Eine wesentlich größere Menge an Emotionswerten macht nicht unbedingt Sinn. Um ein aussagekräftige Evaluation durchführen zu können, sind wenige Parameter der Strategie besser. Auch ist es bei zu vielen Werten für den Benutzer nicht mehr möglich, zwischen den unterschiedlichen Strategien auch entsprechend zu unterscheiden. Soll das System auch seine Emotionen visualisieren, wie ein Avatar (z.B. in Abschnitt 5.3), gelten auch Beschränkungen in den Darstellungsmöglichkeiten des Avatars.

4.4.4 Erweiterte Intention Variable

Um Unsicherheiten im Dialog besser gerecht zu werden, wurde eine neue Variable *IntentionConfirm* eingeführt, die von der Variable "Intention" (siehe Abschnitt 3.4) abgeleitet ist. Diese Variable wird nicht direkt für Emotionsstrategien benutzt, sondern ist ein zusätzliches Hilfsmittel. Die ursprüngliche Variable "Intention" nimmt den Zustand *Finalized* an, sobald die Informa-

Benutzer-Emotion (Kategorie)	System-Emotion
NEUTRAL	neutral
POSITIVE	happy
ANGRY	stressed
STRESSED	clinical

Tabelle 4.4: Abbildung von Kategorien der Variablen Benutzer-Emotion auf Werte der Variablen System-Emotion

tion im Diskurs ausreichend genau spezifiziert ist. Das lässt allerdings keine Bestätigung von Seiten des Benutzers mehr zu, was in manchen Situationen allerdings erwünscht ist, um die eingegebene Information zu überprüfen. Man denke dabei z.B. an einen per Sprache gesteuerten Fahrkarten-Automaten, der eine falsche Fahrkarte verkauft, weil der Spracherkenner das Ziel falsch verstanden hat. Die Zustände der erweiterten Intention-Variable sind mit den erlaubten Übergängen in Abbildung 4.7 dargestellt.

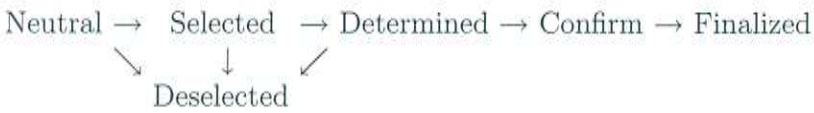

Abbildung 4.7: Zustände und Zustandsübergänge der erweiterten Intention-Variable bei monoton wachsender Spezifizität

4.5 Dialog-Historie

Die abgearbeiteten Dialogziele, mit ihren semantischen Repräsentationen werden vom Dialogmanager auch nach Abarbeitung eines Dialogziels gespeichert. Siehe hierzu [Den02]. Für die Aktien-Anwendung (Abschnitt 5.3) und die Roboter-Anwendung (Abschnitt 5.4) wurde ein Zugriff auf Informationen aus der Historie (der Diskurs abgearbeiteter Dialogziele) benötigt, siehe Abschnitt 5.3. Durch den Aufruf

```
internal://dialogue/assign "past", "sem";
```

in einem Schablonentyp werden in der internen Variablen "past" die Werte des aktuellen Diskurses gespeichert.

Abbildung 4.8 zeigt eine Klärungsfrage aus der Roboter-Applikation. Sie wird verwendet, um die letzte Aktion zu überprüfen und gegebenenfalls rückgängig zu machen.

```
move RevertDlActionPut {
  variable:(Intention = selected),
  path:($past.[PLACE|P_NAME]      is defined),
  path:($sem.[]   is undefined) ->
  bindings:
    internal://dialogue/say "You have selected to put the",
         $past.[OBJECT|generic:NAME],
         " on the ",$past.[PLACE|P_NAME], ". Is this correct?";
    internal://dialogue/addSymbolToSubGrammar
                        "speech", [], <generic:yes,V,_>;
    internal://dialogue/addSymbolToSubGrammar
                        "speech", [], <generic:no,V,_>;
    internal://dialogue/createSubGrammar "speech", "answer.xml";
};
```

Abbildung 4.8: Klärungsfragen mit Zugriff auf Diskurs des abgearbeiteten Dialogziels

Kapitel 5

Umsetzung in Dialoganwendungen

Um zu überprüfen, wie sich die theoretischen Überlegungen in der Praxis umsetzen lassen, wurden verschiedene prototypische Anwendungen implementiert. Hierbei wurde darauf geachtet, ob das Konzept der Emotionsvariablen allgemein genug ist bzw. genügend abstrahiert und sinnvoll in unterschiedlichen Anwendungen eingesetzt werden kann.

Die Applikationen sind in ihrer Komplexität von Abschnitt 5.1 bis Abschnitt 5.4 steigend. Die jeweils beschriebenen Konzepte werden in komplexeren Applikationen wieder verwendet.

5.1 Emotionen-Spiegel

Die erste Anwendung hat das einfache Ziel, die Emotion des Benutzers und die Emotion des Systems zu visualisieren. Die hier beschriebene Implementierung ist orthogonal zu einer ARIADNE Dialoganwendung ohne Emotionen. Zusätzlich wurde eine einfache Java-Applikation entwickelt, die in 2 Fenstern jeweils die Emotionswerte des Benutzers und des Systems mit *Smileys* bzw. *Emoticons* visualisiert. Für die Dialoganwendung werden die Variablen *Benutzer-Emotion* und *System-Emotion* verwendet. Die Erweiterung der Dialoganwendung geschieht durch einfache Schablonentypen, die bei Änderung einer Variablen ausgeführt werden (siehe Abbildung 5.1).

Die Schablonentypen und Variablen können in jede bestehende Anwendung integriert werden, ohne dass sie Auswirkungen auf die Teile der Anwendung haben, die nicht abhängig von Emotionsparametern sind.

```
move SE1 on variable SystemEmotion changed to happy {
->
bindings:
  jpkg://localhost:5454/expression  happy;
};

move SE2 on variable SystemEmotion changed to neutral {
->
bindings:
  jpkg://localhost:5454/expression okay;
};

...

move UE1 on variable UserEmotion changed to neutral {
->
bindings:
  jpkg://localhost:5454/userMirror "neutral";
};

move UE2 on variable UserEmotion changed to happy {
->
bindings:
  jpkg://localhost:5454/userMirror "happy";
};

...
```

Abbildung 5.1: Schablonentypen für den Emotionen-Spiegel

5.2 Bahn-Anwendung

Die Grundlage der Bahn-Anwendung ist ein Fahrplan-Informationssystem. Hierfür wurde ein Basissystem ohne Emotionsmodellierung erstellt und danach erweitert. Die Dialogstrategie ohne Emotionen erfragt nach und nach Informationen über Abfahrtsbahnhof, Zielbahnhof und Tageszeit, bis die Anfrage genügend spezifiziert ist. Das System hat nur prototypischen Charakter, um den Einsatz von Emotionsvariablen zu testen. Aus diesem Grund wurde auch darauf verzichtet, umfangreiche Fahrplandaten zu integrieren.

Basissystem und Dialogziele

Die Bahn-Anwendung hat nur ein einziges Dialogziel (siehe Abbildung 5.2). Das Dialogziel bedingt die folgenden Informationen: Zielbahnhof, Abfahrtsbahnhof, Uhrzeit und eine Bestätigung, ob das Ticket gekauft werden soll. Zur Wiederholung: Das Dialogziel kann ausgeführt werden, sobald alle Informationen im Diskurs vorhanden sind. Solange noch nicht alle Informationen vorhanden sind, wird durch Klärungsfragen versucht, weitere Informationen vom Benutzer zu erlangen. Die Klärungsfragen für diese Anwendung sind im Anhang auf Seite 93 abgebildet. Entsprechend den Klärungsfragen wird jede im Dialogziel verlangte Information einzeln abgefragt.

Emotionen

Die Emotionsstrategie betrachtet emotionale Reaktionen des Benutzers, die auf Kommunikationsfehler hindeuten. Diese Erweiterung kann in einem öffentlichen Informationssystem benutzt werden, um

1. den Dialog abzubrechen, um den Benutzer an einen menschlichen Berater weiter zu leiten,

2. existierende Information im Diskurs zu löschen, um die Anfrage erneut zu starten und Fehlerkorrekturmechanismen zu unterstützen.

Punkt 1 ist sinnvoll, wenn die Kommunikation zwischen Benutzer und System nicht funktioniert und auch nicht wieder hergestellt werden kann. Punkt 2 ist sinnvoll wenn die Information im Diskurs widersprüchlich ist, oder die Kommunikation zwischen Benutzer und System nicht mehr funktioniert, aber dennoch versucht wird, wieder einen gültigen Zustand im Diskurs herzustellen. Das System benutzt Wut und Ärger des Benutzers als Hinweise. Die Variable Benutzer-Emotion reicht alleine nicht als Indikator aus. Erstens können einzelne Klassifikationsfehler des Emotionserkenners nicht ausgeschlossen werden. Zweitens muss auch eine ärgerliche Reaktion des Benutzers nicht unbedingt einen schlechten Dialogverlauf als Ursache haben, sondern kann z.B. auch wegen einer falschen Hypothese des Spracherkenners auftreten. Allerdings kann diese Eigenschaft durch die Variable Emotionstendenz modelliert werden, da sie nicht nur den aktuellen emotionalen Zustand berechnet, sondern über mehrere Gesprächsschritte interpoliert. Abbildung 5.3 zeigt einen einfachen Schablonentyp, der die aktuelle Information im Diskurs löscht (Punkt 2 von oben), wenn der Wert der Emotionstendenz stark negativ wird. Der Schablonentyp kann in jede Anwendung integriert werden, da er unabhängig von der restlichen Strategie definiert ist.

```
goal SellTicket {
  precondition:
  [ act_sellTicket
    DESTINATION_TOWN  [ town_dest
                        TOWN_NAME [ base:string ]
                      ]
    DEPARTURE_TOWN    [ town_dep
                        TOWN_NAME [ base:string ]
                      ]
    TIME              [ obj_time
                        HOUR   [ base:string ]
                        MINUTE [ base:string ]
                      ]
    CONFIRM           [ meta_confirm
                        CONF [ base:boolean ]
                      ]
  ]
  ->
  bindings:
    internal://dialogue/say
               "You have chosen a ticket from",
               $objs.[DEPARTURE_TOWN|TOWN_NAME],
               "to", $objs.[DESTINATION_TOWN|TOWN_NAME],
               ". You want to leave around ",
               $objs.[TIME|HOUR], $objs.[TIME|MINUTE],
               ". You will receive your ticket instantaneously.";

};
```

Abbildung 5.2: Einziges Dialogziel der Bahn-Anwendung

5.3 Aktien-Assistent

Der Aktien-Assisten liefert Informationen über den aktuellen Kurs verschiedener Aktien. Es gibt ein aufgabenorientiertes Dialogziel, das die Informationen definiert, die zur Abfrage der Aktien-Information nötig sind. Das System soll auf verschiedene Reaktionen der Benutzer, z.B. Rückmeldungen, die Erstaunen, Freude oder Frustration ausdrücken, reagieren. Die vorgestellte

```
on variable EmotionTrend changed to strnegative {
->
bindings:
  internal://dialogue/abort;
};
```

Abbildung 5.3: Schablonentyp zum löschen des aktuellen Diskurses

Realisierung ist nicht auf diese Anwendung beschränkt, sondern kann ebenso mit anderen Konzepten kombiniert werden und in andere Anwendungen integriert werden.

Die Reaktion des Systems auf die Rückmeldungen des Benutzers soll nicht nur die emotionalen Parameter des Benutzers berücksichtigen, sondern auch die letzte Aktion und Aussage des Systems. Die Strategie des Dialogsystems sieht vor, die Information über den Aktienkurs bei Abarbeitung des Dialogziels zu geben, dann den aktuellen Diskurs in die Historie zu schreiben und darauf den aktuellen Diskurs zu leeren, um neue Aufgaben zu beginnen. Daher musste eine der folgenden zwei Möglichkeiten geschaffen werden, um auf die abgearbeiteten Informationen zuzugreifen:

- Das Dialogziel definiert zusätzlich eine Reaktion des Benutzers, sodass das Dialogziel noch nicht abgearbeitet ist, wenn die Information gegeben wird.

- Es wird ein Zugriff auf die Historie ermöglicht, um z.B. auf den Kontext des letzten Dialogziels wieder zugreifen zu können.

Variante eins ist nicht möglich, da Rückmeldungen nicht zwingend im Dialogverlauf vorkommen müssen, daher dürfen Reaktionen des Benutzers nicht als obligatorische Parameter im Dialogziel vorhanden sein. Daher wurde Variante zwei gewählt. Die Implementierung, zum Zugriff auf die Historie, wurde bereits in Kapitel 4.5 beschrieben. Die Rückmeldungen werden durch einzelne Dialogziele beschrieben. Abbildung 5.4 zeigt ein Dialogziel, das nach einer positiven Reaktion des Benutzers ausgeführt wird. Abbildung 5.5 zeigt eine entsprechende Implementierung für eine negative Reaktion des Benutzers. Die emotionale Information wird über die Variable Benutzer-Emotion integriert.

```
goal BackchPos1 {
  precondition: [ spk_backch ]
  path:($past.[INFO_MSG]  is unique),
  path:($past.[STOXX]     is unique),
  variable:(UserEmotion = happy) ->
  bindings:
    internal://dialogue/say "The news about ",
              $past.[STOXX], "seem to be good!";
};
```

Abbildung 5.4: Reaktionen des Assistenten auf positive Rückmeldungen des Benutzers

```
goal BackchNeg1 {
  precondition: [ spk_backch ]
  path:($past.[INFO_MSG]  is unique),
  path:($past.[STOXX]     is unique),
  variable:(UserEmotion = angry) ->
  bindings:
    internal://dialogue/say
              "I'm sorry for that," ,
              "perhaps other news are better?";
};
```

Abbildung 5.5: Reaktionen des Assistenten auf negative Rückmeldungen des Benutzers

5.4 Kommunikation mit Robotern

Roboter, die sich in einer realen Umgebung gemeinsam mit Menschen bewegen, oder als humanoide Roboter Arbeiten in einer Küche tätigen, stellen potentiell eine Gefahr für Menschen und Inventar dar, wenn keine besonderen Vorkehrungen und Sicherheitsmaßnahmen getroffen werden. Daher benötigt ein Roboter einen Mechanismus, über den er Gefahrensituationen erkennen kann. Eine solche Erkennung kann durch Sprache, oder durch Emotionen erfolgen. Diese Implementierung stellt eine Möglichkeit zur Verfügung, bei gleicher Semantik aber unterschiedlichen Emotionen die Eingabe unterschiedlich zu interpretieren. Diese Implementierung wurde auch in der Roboter-Anwendung (siehe Anhang A.3) eingesetzt, die für die Datensammlung (Kapitel 6) verwendet wurde.

Stopp-Strategien und Dialogkontext

Entsprechend den Emotionen des Benutzers werden Rückschlüsse über das Gefahrenpotential der aktuellen Situation und der aktuell ausgeführten Aktion gezogen. Die direkte folgerichtige Entscheidung ist, dass der Roboter seine aktuelle Aktion in einer gefährlichen Situation unterbricht. Dies ist keinesfalls eine ausreichende Behandlung der Situation und einzige Reaktionsmöglichkeit. Der Schwerpunkt liegt hierbei darauf, zu zeigen, was mit dem Rahmenwerk möglich ist. Ähnlich wie schon im letzten Abschnitt, benötigt das Dialogsystem einen Zugriff auf die Historie, um zu überprüfen, welche Aktionen ausgeführt werden oder eine Modellierung des Kontextes der Anwendung. Der Kontext der Anwendung liefert darüber hinaus auch die Information, ob eine Aktion gerade ausgeführt wird oder schon beendet wurde. Alternativ kann aber auch in erster Instanz ein Stopp-Befehl an die Anwendung geschickt werden ohne den Kontext zu berücksichtigen. In dem Fall hat die Anwendung selber die Aufgabe, heraus zu finden, ob aktuell eine Aktion ausgeführt wird.

Nachdem der Stopp-Befehl an den Roboter geschickt wurde, kann eine Fehleranalyse oder ein Korrekturdialog durchlaufen werden, um herauszufinden, was der Fehler war und wie ein normaler Zustand wieder hergestellt werden kann. Hierfür wird der Diskurs der letzten Aktion benötigt, um den Benutzer nach dem Fehler fragen zu können und eventuelle Korrekturen vorzunehmen oder bei einem falschen Alarm die Aktion weiter auszuführen.

Disambiguierung der Spracheingabe

Abbildung 5.6 zeigt verschiedene Dialogziele der Anwendung ohne eine explizite Kontextmodellierung. Das Ziel "Stop" wird allein durch den semantischen Inhalt einer Eingabe (z.B. "stopp", "halt") erreicht. Die Ziele "'Stop2" und "Stop3" bedingen die semantische Eingabe "no". Da aber "no" ebenso eine Antwort auf eine Frage sein kann, werden diese Dialogziele nur dann erreicht, wenn der Benutzer gleichzeitig Aufregung zeigt, wobei Aufregung oder Ärger hier als Indikator für eine Stress erzeugende Situation verwendet wird.

Um die Stopp-Ziele jederzeit abarbeiten zu können, werden die Dialogziele im Dialogmanager als bevorzugte Ziele behandelt. Dies ist nötig, da die Ziele auch dann erreicht werden sollen, wenn schon Information im Diskurs vorhanden ist. In diesem Fall würde die Unifikation der Semantik im allgemeinen zu einem inkonsistenten Zustand des Diskurses führen, siehe Abschnitt 3.4.

```
goal Stop {
    precondition:
        [ act_stop ]
    ->
    bindings:
        jpkg://localhost:5454/rbStop;
};
goal Stop2 {
    precondition:
        [ act_no ]
    variable:(UserEmotion = angry)
    ->
    bindings:
        jpkg://localhost:5454/rbStop;
};
goal Stop3 {
    precondition:
        [ act_no ]
    variable:(UserEmotion = stressed)
    ->
    bindings:
        jpkg://localhost:5454/rbStop;
};
```

Abbildung 5.6: Dialogziele mit unterschiedlichen Emotionswerten

5.5 Fazit

Dieses Kapitel zeigt verschiedene Anwendungsmöglichkeiten des Rahmenwerks auf. Jeder der verschiedenen Abschnitte betrachtet einen anderen Aspekt der emotionssensitiven Dialogverarbeitung. Implementierungen der einzelnen Abschnitte lassen sich allerdings kombinieren. Außerdem sind sie so definiert, dass sie eine normale Dialoganwendung, durch Hinzunahme der Schablonentypen und Dialogziele, erweitern können.

Kapitel 6

Benutzerstudien

Nachdem in Kapitel 5 gezeigt wurde, wie sich das Konzept der Emotionsvariablen in verschiedenen Dialoganwendungen einsetzen lässt, geht dieses Kapitel auf Versuche mit Benutzern ein. Die Versuche wurden als einfache Studien durchgeführt. Dabei wurde auf emotionale Reaktionen der Benutzer geachtet und teilweise schon einfache Emotionsstrategien implementiert.

Für die Versuche wurde eine Roboter-Applikation verwendet. Die Applikation setzt sich aus Dialogapplikation, Robotersteuerung mit Anbindung an die Dialogapplikation und Simulation zusammen. Die Robotersteuerung und die Simulation sind im Anhang in Kapitel A.3 genauer beschrieben. Die Simulation dient dem Benutzer als graphische Oberfläche, über die er den Kontext der Applikation einfach aufnehmen, Objekte mit eigenen Worten beschreiben, und Befehle an den Roboter selber formulieren kann.

6.1 Evaluierungsmethoden

Nach der Durchführung der Vorstudie wird eine Zwischenbilanz gezogen und Erweiterungen für das zweite Experiment beschrieben. Nach dem zweiten Experiment wird eine Evaluation für beide Teile erstellt. Da sich das gesamte Dialogsystem aus Einzelkomponenten zusammensetzt, lassen sich sowohl einzelne Komponenten, als auch das Gesamtsystem evaluieren. In Betracht gezogen werden sowohl subjektive Maße, durch Feedback der Benutzer, als auch objektive Maße, wie Zeitmessungen und Fehlerraten.

Evaluation der Einzelkomponenten

Für den Spracherkenner wurde die Wortfehlerrate und die *Satz-Fehlerrate* gemessen. Für das Dialogsystem wurde die Antwortzeit und die Satz-Fehlerrate

der Grammatik (bei korrekter Spracherkennerausgabe) gemessen. Die Satz-Fehlerrate bezieht sich auf die Anzahl der eingegebenen Äußerungen, die nicht korrekt in die semantische Repräsentation transformiert werden. Damit lassen sich objektive Werte angeben, z.B. über die Qualität der textuellen Eingabe, und Vergleiche anstellen, wie groß der Zeitverlust bei Verwendung der Wizard-of-OZ Schnittstelle ist.

Evaluation des Gesamtsystems

Für das Gesamtsystem wurde die Gesamtzeit gemessen, die zwischen Eingabe der Sprache und Ausgabe von Sprache bzw. Ausführung einer Aktion vergeht. Zusätzlich wurde die *Turn-Fehlerrate* gemessen, was Fehler von Spracherkenner, Sprachverarbeitung und Dialogmanager einschließt. Die Turn-Fehlerrate bezieht sich auf die Anzahl der Eingaben, auf die das System keine korrekte Antwort liefert. Um ein subjektives Feedback von den Benutzern zu erhalten, das unter anderem ein Maß für Erfolg und Misserfolg bildet, wurde ein Fragebogen (im Anhang auf Seite 103) vorbereitet. Die Fragen wurden mündlich gestellt und auch mündlich beantwortet. Die Antworten wurden jeweils durch den Fragesteller schriftlich festgehalten.

6.2 Vorstudie

Aufgabe (Task) der Anwendung ist es, durch Anweisungen den Roboter den Tisch decken zu lassen. Die Interaktion erfolgt kommandoorientiert, wobei die Attribute eines Kommandos auch über verschiedene Gesprächsschritte im Dialog erlangt werden können. Der Detaillierungsgrad für eine Aktion beschränkt sich auf "hole Objekt A von Position B" oder "stelle Objekt A auf den Tisch". Die Information, wie die Objekte auf dem Tisch angeordnet werden, also Regeln wie "das Messer liegt rechts vom Teller" und "die Gabel liegt links vom Teller", wird in der Anwendung gehalten und muss vom Benutzer nicht explizit angegeben werden. Außerdem existiert die Einschränkung, dass der Simulations-Roboter nur ein Objekt auf einmal tragen kann. Zusätzlich kennt der Roboter weitere Befehle, wie "gehe zum Position A", die er ausführen kann, aber nicht nötig sind um das Ziel der Aufgabe zu erreichen.

Ziel der Studie

Die Vorstudie hatte als Ziel herauszufinden, wie Benutzer ihre Anfragen formulieren, dementsprechend die Grammatik und Konvertierungsregeln zu entwickeln und evtl. zusätzliche Dialogziele aufzunehmen. Weiterhin sollte her-

ausgefunden werden, welche Emotionen bei den Benutzern auftreten und wie sinnvolle Reaktionen darauf aussehen können.

6.2.1 Aufbau und Durchführung

Wizard-of-OZ

Vor Beginn der Benutzerstudie wurde ein prototypisches System mit Grammatiken, Konvertierungsregeln, Task-Modell, Dialogzielen und Klärungsfragen implementiert. Da aber zu erwarten war, dass Benutzer mit anderen Formulierungen agieren und ebenfalls Informationen erfragen, die nicht im Task-Modell vorgesehen waren, wurde zu Beginn der Vorstudie eine reine Wizard-of-OZ Variante verwendet. Das Wizard-of-OZ System und dessen Einbindung in das übrige System ist im Anhang in Kapitel A.2 beschrieben.

Instruktion der Benutzer und Vorwissen

Die Benutzer wurden nur instruiert, dass sie die Aufgabe haben, den Roboter durch Anweisungen den Tisch decken zu lassen. Es wurde keine Kommandostruktur vorgegeben. Dadurch erreichten wir eine ungebundene und freie Sprechweise der Benutzer, was ein möglichst realistisches Szenario simuliert.

Es waren zwei Personen zur Versuchsbetreuung involviert: der Operator, der den Benutzer instruierte und jederzeit für Fragen bereit stand und der "Wizard" der über die Wizard-of-OZ Schnittstelle die Fragen des Benutzers beantwortete und die entsprechenden Befehle an die Roboter-Simulation sandte.

Durchführung

Die Benutzerstudie wurde komplett in englischer Sprache durchgeführt. Es wurde eine Vorstudie mit acht Probanten und eine Hauptstudie mit zehn Probanten durchgeführt. Die meisten der Probanten waren männlich. Nach der Vorstudie wurden Verbesserungen und Anpassungen am System vorgenommen, und darauf die Hauptstudie, als umfassenderes Experiment, durchgeführt. Die Aufnahmen wurden mit Nahbesprechungsmikrophonen und push-to-talk durchgeführt. Push-to-talk bedeutet, dass der Benutzer einen Schalter gedrückt hält, während er spricht. Die meisten Benutzer willigten ein, während der Studie zusätzlich durch eine Videokamera mit Bild und Ton aufgezeichnet zu werden.

Begünstigungen und Beeinträchtigungen bei der Durchführung

Um unterschiedliche emotionale Reaktionen zu testen, wurden manche Sprecher durch den Wizard begünstigt und manche beeinträchtigt. Bei einer Begünstigung wurde auf (fast) jede Eingabe eine passende Antwort gegeben. Wenn die Eingabe dem Task-Modell entsprach, wurde auch die passende Aktion ausgeführt. Eine Beeinträchtigung entstand dann, wenn der Benutzer Eingaben machte, die dem Task-Modell eigentlich entsprachen, aber entweder durch die Grammatik nicht abgedeckt oder durch den Spracherkenner nicht korrekt erkannt wurden, und ohne Korrektur an den Dialogmanager geschickt wurden.

6.2.2 Auswertung der Vorstudie

Auswertung der Simulation

Die Benutzer konnten durch die Simulation alle Objekte eindeutig identifizieren und waren damit in der Lage, mit ihren eigenen Formulierungen den Roboter zu instruieren. Zwei Sprechern war der englische Begriff für das Wort Theke nicht bekannt, wobei aber der Operator weiterhelfen konnte. Die Simulation hat sich damit als sinnvoll erwiesen, da die Benutzer so ihre eigenen und ungebundenen Formulierungen bilden konnten. Die Simulation war detailliert genug, so dass jeder Benutzer die aktuelle Situation erkennen, deuten und begreifen konnte.

Verhalten der Benutzer

Alle Benutzer haben sich im Laufe des Versuchs an das System adaptiert. Zu Beginn ihrer Aufgabe wusste keiner der Benutzer, welche Sätze und Befehle das System versteht. Die Benutzer mussten also zuerst herausfinden, was das System versteht und versuchten Sätze, wie z.B. "set the table". Abbildung 6.1 zeigt von den Benutzern eingegebene Sätze, die das System in entsprechenden Situationen nicht in die gewünschten Aktionen umwandeln kann. Nach ein paar Fehlversuchen adaptierten sich alle Benutzer allerdings sehr schnell an das System und formulierten ihre eigenen Eingaben mit den Worten des Dialogsystems und wenn möglich im Stil bereits erfolgreicher Eingaben.

Aufgabenabweichende Äußerungen

Außer den Eingaben, die direkt zur Erfüllung der Aufgabe benötigt wurden (aufgabenorientierte Eingaben), wurden auch aufgabenabweichende Einga-

- "set the table": Ist zu ungenau. Der Roboter antwortet, dass er Schritt für Schritt instruiert werden muss.
- "put the plate on the table": Ist kein gültiger Befehl, wenn der Roboter aktuell keinen Teller in den Händen hält. Er muss zuerst instruiert werden, einen Teller von der Theke aufzunehmen.
- "get the plate and the knife": Ist kein gültiger Befehl, da der Roboter nur ein Element gleichzeitig nehmen kann.

Abbildung 6.1: Benutzereingaben, die nicht durch das Dialogsystem abgedeckt werden

ben gemacht. Einige typische Eingaben sind "hi robbi", "thanks", "great job".

Emotionen

Um die Emotionen der Benutzer zu schätzen, wurden die Benutzer nach einer Selbsteinschätzung gefragt und beobachtet. Dei Befragung erfolgte nach Beendigung des Versuchs, mit den anderen Fragen des Fragebogens. In den meisten Fällen stimmten diese Aussagen mit den Beobachtungen überein, die a.) während des Versuchs und b.) nach Ansicht des Videos gemacht werden konnten.

Unterschiede der emotionalen Reaktionen:

Manche Benutzer hatten gar keine Emotionen, andere haben viel gelacht und sich über die Aktionen des Roboters gefreut, und wieder andere haben sich über das System geärgert, wenn es nichts oder zu wenig verstand. Die Emotionen waren stark von der Leistung des Systems abhängig. Die meisten Sprecher, die beeinträchtigt wurden, zeigten Ärger, während die begünstigten Sprecher keine oder nur flüchtige und manchmal positive Emotionen zeigten.

Die beobachteten Emotionen lassen sich grob unterteilen in

- **kurzzeitige Emotionen**: Als Reaktion auf Aussagen oder Aktionen des Roboters, wie z.B. stutzig, erstaunt und enttäuscht. Sie treten auf, wenn der Roboter etwas falsches oder unerwartetes macht. Typisch im Experiment war, dass die emotionale Reaktion nur kurz nach außen

zu erkennen war, aber sich nicht mehr im Sprachsignal wiederspiegelte, wenn der Benutzer wieder mit dem System interagierte.

- **anhaltende Emotionen**: Spiegelt sich auch im Sprachsignal wieder, wenn Benutzer den nächsten Befehl an das System gibt. Ein Beispiel hierfür ist Ärger darüber, dass das System die Eingabe des Benutzers nicht verstanden hat. Der Zustand bleibt mindestens für den Gesprächsbeitrag erhalten. Im Experiment waren Freude und Ärger am häufigsten vertreten.

- **wiederkehrende Emotionen**: Durch größeren zeitlichen Rahmen beeinflusst als auf nur einem Gesprächsbeitrag basierend. Beispiele sind Freude über lustige Aktionen des Roboters. Das Emotionspotential sinkt wieder auf neutral ab, hebt sich aber wieder stark an nach der nächsten "lustigen" Aktion des Roboters. Negative Aktionen des Roboters, die bei anderen Versuchen Ärger verursachten, wirkten sich in diesem Zustand kaum negativ aus. Ähnlich wie Freude, war auch Ärger (bzw. Wut) über schlechte Leistung des Systems vorhanden.

In verschiedenen Versuchen gab es Personen, die nach außen keine negativen Emotionen zeigten, um eine schlechte Spracherkennerleistung zu vermeiden. Dies waren vor allem Personen, die in der Spracherkenner-Forschung tätig sind und daher wissen, dass sich jede Abweichung von der Normalität (wie z.B. Überbetonung) schlecht auf die Erkennungsleistung auswirkt.

Feedback der Benutzer und vorläufige Auswertung des Fragebogens

Von der Simulation waren die meisten Benutzer sehr angetan und fanden die Darstellung und den Roboter "süß". Ähnlich wie auch bei den Emotionen, war das Feedback über die Leistung des Systems starken Schwankungen unterworfen. Je nachdem ob die Benutzer begünstigt oder beeinträchtigt wurden, war auch das Feedback positiv oder negativ.

Interessant erscheint eine vorläufige Tendenz, dass Sprecher, die während dem Dialog häufiger Ärger zeigen, auch beim Fragebogen eher negatives Feedback über das System geben. Dieses negative Feedback bezieht sich dabei nicht nur auf die Komponenten, über die sich der Benutzer geärgert hatte (in dem Fall meistens Spracherkenner oder Grammatik), sondern auch auf andere Teile des Systems, die von anderen Sprechern als positiv gewertet wurden. Beeinträchtigte Sprecher empfanden beispielsweise erstens als negativ, dass der Roboter nur ein Objekt gleichzeitig tragen kann, zweitens die Aufgabe zu

schwer, drittens den Roboter meist nicht so "süß", viertens den Detaillierungsgrad, zuerst einen Teller holen zu müssen und ihn dann erst auf den Tisch stellen zu können, als zu fein und würden das lieber durch ein Kommando erledigen.

6.3 Weiterentwicklung am System

Dialogsystem

Basierend auf den Daten der Vorstudie wurden die Grammatik und die Konvertierungsregeln erweitert. Entsprechend der Schnittstelle der Anwendung, standen die grundlegenden aufgabenorientierten Dialogziele schon vor der Datensammlung fest (s.o.). Zusätzlich wurden weitere Dialogziele hinzugenommen, um auf die typischen aufgabenabweichenden Äußerungen, wie Begrüßungen oder Kommentare, reagieren zu können. Außerdem wurde noch eine weitere, grüne Tasse in das Modell mit aufgenommen. Dies sollte eine Erschwernis für den Benutzer bilden, um komplexere Fehlerdialoge und Stoppstratgien testen zu können. Der Benutzer sollte vermeiden, dass der Roboter die Tasse aufnimmt und auf den Tisch stellt. Als weitere Erschwernis nimmt der Roboter automatisch die grüne Tasse, wenn der Benutzer sich auf eine Tasse bezieht, aber nicht explizit das Attribut "weiß" hinzufügt. Das führt dazu, dass der Roboter automatisch die grüne Tasse aufnimmt. Um die Tasse wieder abzustellen, muss der Benutzer einen Korrektur-Dialog durchlaufen oder die letzte Aktion rückgängig machen.

Simulation

Auch in der Simulation wurde die grüne Tasse eingefügt. Entsprechend dem Weltmodell im Dialogsystem, wurden auch in der Datenbank und dem Kontextmodell der Anwendung neue Einträge für die weitere Tasse erstellt. Zweitens wurde der Anwendung die Funktionalität hinzugefügt, Aktionen rückgängig machen zu können. Hierzu wurden alle Aktionen, die als Kommandos an die Roboter-Simulation gesendet wurden, in einer Stack-Struktur mit den entsprechenden Umkehrfunktionen gespeichert.

Emotionsvariablen und Emotionsstrategien

Die kurzzeitigen Emotionen sind für unsere Anwendung nicht so interessant, da sie durch unsere Aufnahmemechanismen nicht erfasst werden können, und auch nicht den tatsächlichen Zustand des Benutzers wiederspiegeln. Siehe

hierzu Abschnitt 2.2.4. Interessant sind aber die anhaltenden Emotionen und die wiederkehrenden Emotionen.

Im Folgenden ist eine Liste der Fälle angegeben, die mit den Emotionsvariablen modelliert wurden und für die Dialogverarbeitung interessant sind.

- Eine häufige Reaktion war Ärger, (siehe oben). Im Fragebogen haben die betroffenen Versuchspersonen angegeben, dass sie darauf keine besondere Reaktion des Systems erwarten, sondern sich mehr eine bessere Leistung des Spracherkenners wünschen. Allerdings ließ die Analyse der ersten Daten vermuten, dass das Feedback für das Gesamtsystem besser ausfallen kann, wenn der Benutzer sich verstanden fühlt. Dafür wurden Antworten eingebaut, dass sich das System entschuldigt, wenn sich der Benutzer ärgert.

  Ärger konnte sehr gut mit der Variable Benutzer-Emotion modelliert werden und die Strategie des Systems (wie entschuldigendes Verhalten) durch die Variable System-Emotion.

- Eine weitere Reaktion war Freude. Auch Freude konnte sehr gut durch die Variable System-Emotion modelliert werden, allerdings war nicht zu erkennen, welche andere Strategie das System sinnvoller Weise anwenden sollte. In diesem Fall konnte man vermutlich die Freude des Benutzers als eine Bestätigung der richtigen Strategie deuten.

- Die wiederkehrenden Emotionen konnten am besten mit der Variablen Emotionstendenz modelliert werden, da diese über mehrere Turns hinweg interpoliert. Gute Werte konnten für die linear gewichtete Summe mit N0=4 gefunden werden.

6.4 Hauptstudie

6.4.1 Aufbau und Durchführung

Das zweite Experiment basiert auf der Weiterentwicklung (Abschnitt 6.3), und verwendet ansonsten den gleichen Aufbau wie die Vorstudie. Die Sprecher wurden ebenfalls auf Video aufgezeichnet. Auch in diesem Experiment wurde die gleiche Wizard-of-OZ Schnittstelle verwendet, um Eingriffe vornehmen zu können, damit z.B. in Fehlersituationen, die der Dialogmanager nicht alleine auflösen kann, eingegriffen werden kann. Allerdings wurden grundsätzlich alle Eingaben des Benutzers automatisch an das Dialogsystem gesendet. Ebenso wurden die Ausgaben des Dialogsystems automatisch an

die Applikation geschickt, wenn der Datenfluss an der Wizard-of-OZ Schnittstelle nicht explizit unterbrochen wurde. Die Testpersonen der Hauptstudie waren nicht mit den Testpersonen der Vorstudie identisch.

bei allen Sprechern Wortfehlerrate	33%
minimale Anzahl Turns	10
mittlere Anzahl Turns	26
Turn-Fehlerrate	53%

Tabelle 6.1: Evaluationsergebnisse der Benutzerstudie

6.4.2 Auswertung der Hauptstudie

Auswirkung von negativen Emotionen

Auch im zweiten Experiment wurden einige Benutzer beeinträchtigt. Zwei Benutzer wurden nicht instruiert, die Adaptionssätze zu lesen, was eine wesentlich höhere Wortfehlerrate zur Folge hatte. Auch diese Benutzer ärgerten sich, ebenso wie die Benutzer der Vorstudie, über die schlechte Erkennungsleistung und gaben ebenfalls ein negatives Feedback, wie schon in Abschnitt 6.2.2 beschrieben.

Verhalten der Benutzer

Während den Dialogen kam es auch vor, dass das System mit einer falschen Spracherkennerhypothese auch eine falsche semantische Repräsentation arbeiten musste, was zu Fehlersituationen führte. Zum Beispiel verstand das System bei "get the spoon from the board" den Satz "'get the knife", worauf das System zurück fragt "Do you want me to get the knife from the board?". Die einfachste Antwort, um den Löffel aufzunehmen, wäre den vorherigen Satz noch einmal zu wiederholen, was die vorherige semantische Eingabe ersetzt. Da diese technische Sicht den Benutzern nicht bekannt ist, widersprechen die Benutzer dem System eher, wie z.B. "No I don't want you to get the knife" oder "no don't get the knife, get the spoon". Dies bestätigt auch Ergebnisse in [SNG+02], mit der Aussage, dass Benutzer nicht wiederholt sprechen ("users don't respeak"). Besonders zu Beginn ihres Tests dachten Benutzer selten daran, dass der Grund, weshalb das System nicht korrekt antwortet, ein Spracherkennungsfehler ist, sondern wählten im folgenden bevorzugt andere Formulierungen. Allerdings adaptierten sich alle

Benutzer sehr schnell an das System, formulierten ihre eigenen Eingaben mit den Worten des Dialogsystems und verwendeten die Struktur bereits erfolgreicher Eingaben. Diese Adaption der Benutzer führt dazu, dass die Benutzer erfolgreicher mit dem System kommunizieren können. Allerdings sollte sich bevorzugt ein System an den Benutzer anpassen können und nicht umgekehrt.

Antworten des Systems bei leerer Eingabe

In manchen Situationen war die Ausgabe des Spracherkenners leer, oder die Grammatik konnte keinen Parse auf der Eingabe finden. Zu Beginn der Datensammlung antwortete das System darauf gar nicht, was den Benutzer sehr verunsicherte. Die nächste Version konnte zumindest den Satz "I didn't understand you" sprechen, worüber sich die Benutzer aber auch ärgerten, weil das System immer das gleiche sagte. Die dritte, emotionsabhängige Variante konnte unterschiedliche Sätze sprechen, je nachdem, ob der Benutzer neutral, gestresst oder verärgert war. Die letzte Variante wurde von den Benutzern bedeutend besser bewertet, da sie zumindest gegenüber der ersten Variante, dem Benutzer Informationen über den aktuellen Stand gibt, und anders als die zweite Variante, durch unterschiedliche Formulierungen weniger Monotonie bietet.

Stoppdialoge und Reparaturdialoge

Einigen Benutzern wurde die grüne Tasse als ein Gefahrenpunkt vorgestellt. Ihnen wurde gesagt, dass diese ihre Lieblingstasse sei und der Roboter sie kaputt machen würde, wenn er sie anfasst. Da der Roboter automatisch beginnt die grüne Tasse zu holen, wenn der Benutzer ihn instruiert eine Tasse zu holen, musste er vom Benutzer unterbrochen werden. Hierbei gab es Unterschiede bei den Reaktionen zwischen verschiedenen Benutzern. Manche, besonders die frühen Benutzer der Testreihe, nahmen die Situation nicht ernst genug um den Roboter zu unterbrechen ("er holt jetzt die grüne Tasse, und?"). Sie zeigten allerdings teilweise Erstaunen, dass der Roboter die falsche Tasse nahm. Weiteren Benutzern wurde noch deutlicher dargestellt, dass der Roboter auf keinen Fall die grüne Tasse nehmen sollte, und dass sie den Roboter auf jeden Fall unterbrechen sollen, wenn er die grüne Tasse nehmen will. Diese Benutzer nahmen das Experiment auch ernster und versuchten den Roboter zu unterbrechen.

Die Kommandos um zu unterbrechen waren "stopp, stopp, stopp!", "wait!", "no!". "stopp, stopp, stopp!" und "wait!" waren durch die Grammatik abgedeckt und konnten in das semantische Symbol *act_stop* konvertiert werden,

sodass der Roboter die entsprechende Stopp-Strategie starten konnte. "no!" war vor dem Experiment nicht vorgesehen, sondern in der Grammatik nur als Gegensatz von "yes" zum Bestätigen oder Verneinen von Fragen vorgesehen. Dieser Fall konnte aber nachträglich durch eine Kombination mit Emotionsparametern modelliert werden, wie im folgenden Code-Segment gezeigt und auch schon in Kapitel 5.4 beschrieben.

```
goal Stop3 {
    precondition:
        [ act_no ]
    variable:(UserEmotion = stressed)
    ->
    bindings:
        jpkg://localhost:5454/rbStop;
};
```

6.5 Diskussion der Ergebnisse

Die hier vorgestellten Ergebnisse können, aufgrund der relativ kleinen Anzahl von 20 Personen, nur Tendenzen beschreiben. Während der Studie mussten verschiedene Parameter geändert werden, um mit verschiedenen Einflüssen z.B. bei den Benutzern Emotionen auszulösen. Dies führt dazu, dass die einzelnen Tests nur teilweise miteinander vergleichbar sind. Unter diesem Aspekt müssen auch die Zahlen in Tabelle 6.1 betrachtet werden. Sie bilden einen Mittelwert über normale Dialoge, begünstigte Dialoge und beeinträchtigte Dialoge. Auf verschiedene Ansätzen, Emotionen beim Benutzer auszulösen, wird in Abschnitt 7.2.2 noch einmal eingegangen. Dies stellt eine schwierige Aufgabe dar, und darf bei der Evaluierung emotionssensitiver Dialogstrategien nicht vernachlässigt werden.

Allerdings zeigt die Studie auf, dass Benutzer emotional im Dialog reagieren. Diese Erkenntnis stimmt mit anderen Studien, siehe z.B. Picard [Pic97], Reeves und Nass [RN99], überein. Es ist daher zu vermuten, dass Benutzer auch in anderen Experimenten emotional reagieren.

In diesem Kapitel wurden verschiedene Strategien des Systems erprobt. Dabei existieren jeweils unterschiedliche Schwerpunkte für Evaluationsmaße. Verschiedene Antwortstrategien, die darauf abzielen, den Benutzer aufzuheitern, oder zu beruhigen, können nur mit subjektiven Maßen ermittelt werden. Um zu zeigen, dass die Strategie eine Verbesserung bringt, müssen Dialoge ohne emotionssensitive Erweiterungen mit emotionssensitiven Dialogen verglichen werden. Dabei ist zu berücksichtigen, dass auch diese Maße wiederum nur

qualitativer Art sind und, wie die meisten subjektiven Maße, Schwankungen unterliegen. In wieweit die hier vorgestellten Erweiterung tatsächlich eine statistisch relevante Verbesserung bringen, muss also in weiteren Experimenten gezeigt werden. Die Disambiguierung der Eingabe für Stopp-Strategien des Roboters (siehe oben) soll die Effizienz der Interaktion erhöhen und die Satzfehlerrate verbessern. Die wenigen vorhandenen Beispiele zeigen, dass die vorgeschlagene Strategie zur Disambiguierung funktioniert. In weiteren Experimenten mit einem automatischen Emotionserkenner muss gezeigt werden, wie robust diese Verfahren z.B. bei Fehlklassifikation sind.

Kapitel 7

Erkenntnisse, Zusammenfassung und Ausblick

7.1 Zusammenfassung

In dieser Arbeit wurde eine Integration von Emotionsparametern in das bestehende Dialogsystem *ARIADNE* vorgestellt. Es wurden verschiedene Verfahren aufgezeigt, Emotionen im Dialog einzusetzen und entsprechende Implementierungen beschrieben. Dazu wurden verschiedene Emotionsmodelle und am Beispiel von *ARIADNE*, auch deren Einsatz in sprachbasierten Dialogsystemen untersucht. Die Integration in *ARIADNE* ist als Erweiterung des bestehenden Rahmenwerks zu verstehen, die es ermöglicht, auf einfache Art und Weise emotionssensitive Anwendungen zu schreiben.

Die Übertragbarkeit auf verschiedene Anwendungen wurde in Kapitel 5 durch die Implementierung in unterschiedlichen Anwendungen gezeigt. Für eine Benutzerstudie wurde eine Roboter-Anwendung, eine Roboter-Steuerung und eine Simulation entwickelt. Mit der Studie konnten Daten für eine Roboter-Steuerung (für die Aufgabe einen Tisch zu decken) gesammelt werden. Dabei wurden, unter anderem durch Beeinträchtigungen und Begünstigungen, Emotionen der Benutzer ausgelöst. Die Ergebnisse der Studie lassen vermuten, dass eine Verwendung von Emotionsparametern, so wie in der vorliegenden Arbeit beschrieben, sinnvoll ist.

In dieser Arbeit wurde weiterhin ein einfaches Klassifikationsschema zur Emotionserkennung beschrieben. Der Klassifikator wurde darauf trainiert, zwischen hoher Intensität (*high arousal*) und niedriger Intensität (*low arousal*) zu unterscheiden. Das Modell reicht aber nicht aus, um z.B. zwischen den Emotionen *happy* und *angry* zu unterscheiden, welche sich bei gleicher

Intensität nur in ihrer Valenz unterscheiden.

Zur Modellierung von Emotionen und deren Auswirkungen auf den linguistischen Eigenschaften der Sprache, wurden Emotionale Semantische Grammatiken eingeführt, welche aus der Spracheingabe direkt semantische Knoten in der Merkmalsstruktur erzeugen. Für Emotionen, die als Eigenschaften der Eingabe oder als emotionaler Zustand des Sprechers zu verstehen sind, wurden Facetten benutzt. Die Facetten erfassen Emotionsparameter, die zusätzlich zur Spracherkennerausgabe durch einen Emotionserkenner geliefert werden. Die Facetten bilden Elemente in multidimensionalen Merkmalsstrukturen. Es wurden verschiedene Emotionsvariablen eingeführt, die mit Hilfe der Facetten berechnet werden. Weiterhin wurden verschiedene Techniken vorgestellt und implementiert, um unter Verwendung der Emotionsvariablen verschiedene Dialogziele erreichen zu können, die Dialogstrategie durch Schablonentypen zu verändern und verschiedene Klärungsfragen zu stellen, verschiedene Antworten zu geben und eine Disambiguierung der semantischen Eingabe zu ermöglichen.

7.2 Erkenntnisse

Aus der Datensammlung für die Roboter-Steuerung konnten verschiedene Erkenntnisse gezogen werden. Grundlegend wurde veranschaulicht, dass Emotionen im Dialog auftreten und dass es wichtig ist, diese entsprechend zu modellieren. Weiterhin haben sich einige Erkenntnisse für den Entwurf von Dialoganwendungen ergeben (Abschnitt 7.2.1). Abschnitt 7.2.2 beschreibt, wie sich die Simulationsumgebung auf Emotionen des Benutzers auswirkt und welche Versuche unternommen wurden, um auf Emotionen des Benutzers Einfluss zu nehmen.

7.2.1 Entwurf der Dialoganwendung

Drei Hauptpunkte waren besonders relevant für den Entwurf der Dialoganwendung.

1. Nicht nur aufgabenorientierte Kommunikation ist relevant, sondern auch soziale Kommunikation, siehe Auswertung der Studie 6.2.2.

2. Eine Visualisierung ist wichtig. Die Simulation hat sich als sehr wertvoll herausgestellt, da die Benutzer den Zustand schnell erfassen können, frei sprechen können und eigene Formulierungen finden.

3. Sowohl negative als auch positive Emotionen wirken sich auf die Einstellung des Benutzers gegenüber dem gesamten System aus. Es ist auch aus diesem Grund wichtig, Ärger des Benutzers zu vermeiden.

7.2.2 Hervorrufen von Emotionen

Um ein System mit Emotionen zu evaluieren, benötigt man "echte" Emotionen der Benutzer. Allerdings ist es keine leichte Aufgabe, eine Situation herzustellen, die bei den Benutzern die gewünschten Emotionen hervorruft. [WP99] beschreibt, dass in ihren Versuchen mehrere Benutzer (meist dafür bezahlte Benutzer) unaufmerksam und desinteressiert waren. Weiterhin wird in [TW00] und [RR00] auch ein großer Einfluss der Umgebung und der anwesenden Personen auf das emotionale Verhalten der Benutzer beschrieben. Die Auswirkungen führen so weit, dass die Benutzer ihre Emotionen komplett unterdrücken.

In der Studie wurden verschiedene Wege getestet, bei den Benutzern Emotionen hervorzurufen. Neben den bereits beschriebenen Ansätzen, haben einige (ursprünglich vielversprechende) Ansätze keine im Dialog nutzbaren

Emotionen hervorgebracht. In verschiedenen Dialogen wurde versucht, die Benutzer unter Zeitdruck und damit unter Stress zu setzen. Dazu wurde ein Sekundenzähler gestartet. Der Sekundenzähler befand sich deutlich sichtbar auf der rechte Seite der Simulation (siehe Abbildung im Anhang auf Seite 87). Zusätzlich wurden die Benutzer auf einen Wettstreit, mit Belohnung für den schnellsten Dialog hingewiesen. Die Auswirkung auf die Sprechgeschwindigkeit war deutlich zu erkennen und die Benutzer kontrollierten oft, zu Beginn des Dialogs, die ablaufende Zeit. Dieser Effekt lies allerdings schon nach wenigen Interaktionen nach, sodass der Zeitdruck keine messbare Auswirkung auf den Dialog hatte.

Eine Auswirkung auf das Verhalten der Benutzer hat vermutlich auch, ob die Sprecher Muttersprachler sind oder nicht. Es fiel auf, dass nicht-Muttersprachler im Dialog konzentrierter waren und mehr Aufmerksamkeit darauf verbrauchten, den gesprochenen Satz korrekt zu formulieren.

Die größte Auswirkung war zu bemerken, wenn das System den Benutzer nicht verstehen konnte. Gründe dafür waren entweder Fehler des Spracherkenners, unzureichende Abdeckung der Grammatik oder fehlende Aktionen im Dialog. Dies führte sehr häufig zu Ärger seitens der Benutzer (siehe auch Kapitel 6.2.2). Daher scheint es wichtig, beim Entwurf eines Systems diese Situationen zu vermeiden, indem passende Antworten gegeben werden, um die Fehler aufzufangen und den Benutzer zumindest über das, was das System erfasst hat und plant, zu informieren.

7.3 Ausblick

Diese Arbeit stellt ein Rahmenwerk zur Verfügung, mit dem auf einfache Art und Weise Emotionen des Benutzers und eigene Emotionen des Systems im Dialog verwendet werden können. Neben den vorgestellten Implementierungen, bietet das Gebiet noch Platz für weitere Forschung. Zusätzlich kann auch das Rahmenwerk als Grundlage für weitere Arbeiten dienen. Die folgenden drei Abschnitte beschreiben kurz weiterführende Themen, die sich aus dieser Arbeit ergeben oder daran angelehnt sind.

7.3.1 Emotionsmodelle

Kurzzeitige Emotionen

Wir haben außer Emotionen, die gleichzeitig mit dem Sprachsignal gemessen werden, auch andere Emotionstypen feststellen können. Interessant sind hier z.B. kurzzeitige Emotionen, wobei die Eigenschaften über eine sehr kurze Zeitdauer betrachtet werden. Allerdings ist noch nicht klar, wie diese Merkmale integriert werden können. Auch für die Erkennung, die über Bildverarbeitung erfolgt, stehen uns noch keine Verfahren zur Verfügung. Weiterhin müsste das Dialogsystem dafür in der Lage sein, die Schwankungen und Eigenschaften in kleineren Einheiten, als nur Gesprächsschritten, zu erfassen. Dieses Problem lässt sich allgemein als Fusion verschiedener Modalitäten beschreiben, die nicht a-priori aneinander gekoppelt sind.

Auswirkung von Aktionen

Unser Rahmenwerk misst den emotionalen Zustand des Benutzers, um daraufhin entsprechende Aktionen zu tätigen. Interessant könnte aber auch eine Modellierung sein, die im voraus schon abschätzen kann, welche Auswirkungen verschiedene Aussagen oder Aktionen des Systems auf den emotionalen Zustand des Benutzers haben. Dies erfordert allerdings vermutlich entweder eine komplexe Modellierung oder eine große Menge an statistischen Daten, da hierbei viele Einflussfaktoren zu berücksichtigen sind.

Allgemeinere Modellierungen

In der Literatur werden viele verschiedene Emotionsmodelle vorgeschlagen. Nicht alle folgen dabei der klassischen Vorstellung von Emotionen. Zum Beispiel geben die *Ephemeral Emotions* (Abschnitt 2.2.4) eher Informationen über den Dialogverlauf, als dem klassischen Verständnis von Emotionen zu

folgen. Daher kann man sich überlegen, wie weit man von dem Begriff Emotionen abstrahiert und Eigenschaften der Spracheingabe allgemein betrachtet. Wieso soll man sich auf Emotionen beschränken, wenn es auch andere sinnvolle Informationsquellen gibt. Die Integration in das vorgestellte Rahmenwerk erfolgt sogar analog zu Emotionen, wobei nur neue Namen für die jeweiligen Zustände bzw. Werte vergeben werden müssen.

7.3.2 Reaktionen der Benutzer

Wir haben gesehen, dass Benutzer positiver reagieren, wenn das System zwischen unterschiedlichen Antworten wechselt und nicht immer dieselbe Antwort gibt. Ebenso wurde festgestellt, dass nicht nur aufgabenorientierte Aussagen wichtig sind, sondern dass Benutzer auch auf einer sozialen Ebene, im Sinne von Reeves und Nass [RN99], mit dem System interagieren. Beides muss also beim Entwurf von Dialoganwendungen und -systemen berücksichtigt werden, wenn man dem Benutzer eine möglichst natürliche Schnittstelle bieten möchte. Dies sollte auch bei Weiterentwicklungen von ARIADNE berücksichtigt werden, so dass eine soziale Komponente besser unterstützt wird.

7.3.3 Evaluation

Bisher konnte in Ermangelung eines passenden Emotionserkenners keine Evaluation mit einem Gesamtsystem (mit Emotionsklassifikation) gemacht werden. Für den Einsatz im Dialogsystem gelten für den Emotionserkenner besondere Bedingungen, was die Risikomaße der Klassifikation betrifft. Diese sind von der jeweiligen Strategie abhängig. In den meisten Fällen ist es wichtiger, dass der Erkenner nicht *wütend* klassifiziert, wenn der Benutzer fröhlich ist, als wenn er im gleichen Fall *neutral* klassifiziert. Beides kann zu einer komplett anderen Dialogstrategie führen.

Die Benutzerstudie, die in dieser Arbeit beschrieben wird, reicht nicht aus, um zu zeigen, dass die vorgeschlagenen Strategien eine tatsächliche Verbesserung bringen. Eine umfassende Evaluation sprengt allerdings den Rahmen dieser Arbeit. Es werden dazu nicht nur mehr Personen benötigt, die an der Evaluation teilnehmen, sondern insbesondere auch mehr Daten, in denen "echte" Emotionen auftreten. "Echt" ist hier besonders herausgehoben, da die meisten Arbeiten zur Emotionserkennung auf simulierten Daten von Schauspielern basieren. Um aber zu messen, welche Verbesserungen die emotionssensitive Dialogstrategie bringt, müssen die Benutzer diese Emotionen auch tatsächlich erleben.

7.3.4 Konfidenzmaße

Bei Verwendung eines Emotionserkenners muss man grundsätzlich auch mit Fehlklassifikationen rechnen. Entsprechend dem Modell des Emotionserkenners wäre es demnach sinnvoll, Konfidenzmaße für die Klassifikation zu verwenden. Die Integration der Konfidenzen erfolgt über Facetten und kann zusätzlich auch durch eine eigene Variable modelliert werden. Die technische Modellierung erfolgt analog zur Integration von Konfidenzen des Spracherkenners, die von Denecke und Yang [DY00] beschrieben wird.

7.3.5 Facetten

Die aktuelle Version des Dialogmanagers kann die Werte der verschiedenen Facetten nicht direkt nutzen, um die Dialogstrategie zu beeinflussen. Erst die Variablen, die aus den Emotionsfacetten berechnet werden, können von den Schablonentypen genutzt werden. Allerdings sind die Facetten als Einträge der multidimensionalen Merkmalsstrukturen im Diskurs vorhanden. Daher wäre es sinnvoll, die semantischen Rahmen der Dialogziele nicht nur auf den semantischen Werten der lexikalischen Eingabe, sondern auch auf den Werten der Facetten definieren zu können.

7.3.6 Domänenunabhängige emotionssensitive Dialogstrategie

Neben einer Anpassung der Dialogstrategie durch Schablonen, wie in Kapitel 4.3 beschrieben, kann man auch Veränderungen an der grundlegenden Dialogsteuerung vornehmen. Diese grundlegende Strategie ist domänenunabhängig definiert und steuert z.B. die Berechnung des Abstrakten Dialogzustandes, Datenbankabfragen und Ausführung der Aktionen, siehe Abschnitt 3.5. In diesen Teil lassen sich auch grundlegende Emotionsstrategien integrieren. Wenn z.B. negative Emotionen des Benutzers als Unzufriedenheit mit dem System interpretiert werden sollen, kann man z.B. Einfluss auf die Berechnung der Variable *TurnQuality* nehmen. Allerdings wird hierdurch die grundlegende Dialogsteuerung strategieabhängig und in ihrem Einsatz beschränkt. Aufgabe einer weiterführenden Arbeit wäre es z.B. herauszufinden, ob sich so auch domänenunabhängige Konzepte finden lassen.

7.3.7 Lernen

Eine weitere interessante Funktion, die Emotionen im Dialog übernehmen können, sind Hinweise für Lernalgorithmen. Verschiedene Arbeiten haben

gezeigt, wie motivationsgetriebene Systeme sowohl zielbasiert, als auch emotionsbasiert bzw. "affektbasiert" lernen können [Blu96], [Vel98b], [GH98]. Emotionen könnten z.B. für Algorithmen des Reinforcement Learning verwendet werden, indem sie als Feedback darüber gewertet werden, ob die letzte Aktion gut oder schlecht war.

Literatur

[AKG+99] ANDRE, E., M. KLESEN, P. GEBHARD, S. ALLEN und T. RIST: *Integrating Models of Personality and Emotions into Lifelike Characters.* In Proceedings of the Workshop on Affect in Interactions, (Siena, Italy):136–149, 1999. 8

[AKG+00] ANDRE, E., M. KLESEN, P. GEBHARD, S. ALLEN und T. RIST: *Exploiting Models of Personality and Emotions to Control the Behavior of Animated Interactive Agents.* In: *Agents2000 Workshop*, 2000. 12

[BA02] BREAZEAL, CYNTHIA und LIJIN ARYANANDA: *Recognition of Affective Communicative Intent in Robot-Directed Speech.* Autonomous Robots, 12(1):83–104, 2002. 11

[Bat97] BATES, J.: *The Role of Emotion in Believable Agents.* Communications of the ACM, 37(7):122–125, 1997. 12

[BB99] BALL, G. und J. BREESE: *Modeling the Emotional State of Computer Users.* In: *Workshop on Attitude, Personality and Emotions in User-Adapted Interaction, UM '99*, 1999. 97

[BGS90] BILANGE, E., M. GUYOMARD und J. SIROUX: *Separating dialogue knowledge and task knowledge from oral dialogue management.* In: *COGNITIVA'90*, Madrid, 1990. 20

[Bil91] BILANGE, E.: *A task independent oral dialogue model.* In: *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics.* European Chapter of the Association for Computational Linguistics, Berlin, 1991. 20

[Blu96] BLUMBERG, B.: *Old tricks, new dogs: Ethology and interactive creatures.* Doktorarbeit, MIT, 1996. 74

[Bre98] BREAZEAL, CYNTHIA (FERRELL): *A Motivational System for Regulation Human-Robot Interaction.* In: *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, Seiten 54–62. American Association for Artificial Intelligence, 1998. 14

[BS99] BREAZEAL, C. und B. SCASSELLATI: *A context-dependent attention system for a social robot.* In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999. 13

[BTM96] BLUMBERG, B., P. TODD und P. MAES: *No Bad Dogs: Ethological Lessons for Learning.* In: *From Animals to Animats 4, Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB)*, Seite 295. Bradford, Oktober 1996. 14

[Car92] CARPENTER, B.: *The Logic of Typed Feature Structures.* Cambridge University Press, 1992. 22

[CDCR99] COWIE, R., E. DOUGLAS-COWIE und A. ROMANO: *Changing emotional tone in dialog and its prosodic correlates.* In: *ESCA International Workshop on Dialog and Prosody*, Seiten 41–46, 1999. 6

[CDCT+01] COWIE, R., E. DOUGLAS-COWIE, N. TSAPATSOULIS, G. VOTSIS, S. KOLLAIS, W. FELLENZ und J. G. TAYLOR: *Emotion recognition in human-computer interaction.* In: *IEEE Signal Processing Magazine*, Band 18, Seiten 32–80, 2001. 6

[Dam95] DAMASIO, A.: *Descartes Error: Emotion, Reason, and the Human Brain.* Perennial (HarperCollins) Taschenbuch, November 1995. 13

[Den00a] DENECKE, M.: *Informational Characterization of Dialogue States.* In: *Proceedings of the International Conference on Speech and Language Processing (ICSLP-00)*, Beijing, China, 2000. 22

[Den00b] DENECKE, M.: *Object-oriented Techniques in Grammar and Ontology Specification.* In: *Proceedings of the Workshop on Multilingual Speech Communication*, 2000. 22

[Den02] DENECKE, M.: *Generische Interaktionsmuster für aufgabenorientierte Dialogsysteme.* Doktorarbeit, Universität Karlsruhe, 2002. v, vii, 19, 22, 23, 25, 26, 27, 29, 44

[DPW96] DELLAERT, F.,T. POLZIN und A. WAIBEL:*R ecognizing Emotions in Speech.* In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Band 3, Seiten 1970–1973, Philadelphia, PA, 1996. 16

[dRG99] ROSIS, FIORELLA DE und FLORIANA GRASSO: *Affective Natural Language Generation.* In: *IWAI*, Seiten 204–218, 1999. 16

[DW97] DENECKE, M. und A.H. WAIBEL: *Dialogue Strategies Guiding Users to their Communicative Goals.* In: *Proceedings of Eurospeech '97*, Nummer Rhodos, Greece, 1997. 28

[DY00] DENECKE, M. und J. YANG: *Partial Information in Multimodal Dialogue.* In: *Proceedings of the International Conference on Multimodal Interfaces (ICMI-00)*, 2000. 73

[ED94] EKMAN, P. und R. DAVIDSON: *The Nature of Emotion: Fundamental Questions.* Series in Affective Science. 1994. 5

[Ekm92] EKMAN, P.: *An Argument for Basic Emotions.* In: *Cognition and Emotion*, Band 6, Seiten 169–200, 1992. 7, 8

[Ekm99] EKMAN, P.: *Basic Emotions.* In: DALGLEISH, T. und T. POWER (Herausgeber): *The Handbook of Cognition and Emotion*, Seiten 45–60. Sussex, U.K.: John Wiley & Sons, Ltd, 1999. 8

[Ell92]E LLIOT, C.: *The Affective Reasoner: A Process Model of Emotions in a Multi-agent System.* Doktorarbeit, Northwestern University, Institute for the Learning Sciences, 1992. 8

[Ell97]E LLIOT, C.: *Hunting for the Holy Grail with "emotionally intelligent" virtual actors.* Draft HTML paper for ACM's Intelligence97, 1997. http://condor.depaul.edu/~elliott. 13

[ELR97] ELLIOT, C., J.C. LESTER und J. RICKEL: *Integrating affective computing into animated tutoring agents.* In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-97) Workshop on Animated Interface Agents*, Seiten 113–121, Nagoya, Japan, 1997. 13

[Gav00] GAVALDA, M.: *SOUP: A Parser for Real-world Spontaneous Speech.* In: *Proceedings of the 6th International Workshop on Parsing Technologies (IWPT-2000)*, Trento, Italy, February 2000. 20, 25

[Geb01] GEBHARD, PATRICK: *Enhancing Embodied Intelligent Agents with Affective User Modelling.* Lecture Notes in Computer Science, 2109, 2001. 13

[GH98] GADANHO, S. und J. HALLAM: *Exploring the role of emotions in autonomous robot learning.* In: CANAMERO, D. (Herausgeber): *AAAI Fall Symposium — Emotional and Intelligent: The tangled knot of cognition*, 1998. 74

[GS86] GROSZ, BARBARA J. und CANDACE L. SIDNER: *Attention, Intentions, and the Structure of Discourse.* Computational Linguistics, 12(3):175–204, 1986. 20

[Hei00] HEINSTRÖM, J.: *The impact of personality and approaches to learning on information behaviour.* Information Research, 5(3), 2000. Verfügbar unter http://informationr.net/ir/5-3/paper78.html. 12

[HFDW02] HOLZAPFEL, H., C. FUEGEN, M. DENECKE und A. WAIBEL: *Integrating Emotional Cues into a Framework for Dialogue Management.* In: *Proceedings of the International Conference for Multimodal Interaction.* IEEE, October 2002. 42

[HMY92] HEISTERKAMP, P., S. MCGLASHAN und N. YOUD: *Dialogue semantics for an oral dialogue system.* In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP-92)*, 1992. 20

[HR83] HAYES, PHILIP und RAJ REDDY: *Steps toward Graceful Interaction in Spoken and Written Man-Machine Communication.* Int. J. Man-Machines Studies, 19:231–284, 1983. 19

[KM01] KAMPS, JAAP und MAARTEN MARX: *Words with Attitude.* CCSOM Working Paper 01-194, 2001. 16

[LBC90] LANG, P. J., M. M. BRADLEY und B. N. CUTHBERT: *Emotion, Attention, and the Startle Reflex.* Psychological Review, 97(3):377–395, 1990. 7

[LKF80] LAZARUS, R. S., A. D. KANNER und S. FOLKMAN: *Emotions: A cognitive-phenomenological analysis.* In: PLUTCHIK, R. und H. KELLERMAN (Herausgeber): *Emotion Theory, Research, and Experience*, Band 1 der Reihe *Theories of Emotion.* Academic Press, 1980. 5

[MA93] MURRAY, I.R. und J.L. ARNOTT: *Toward the Simulation of Emotion in Synthetic Speech: A Review of the Literature on Human Vocal Emotion.* Journal of the Acoustical Society of America, 93(2):1097–1108, Februar 1993. 6

[MA95] MURRAY, I.R. und J.L. ARNOTT: *Implementation and testing of a system for producing emotion-by-rule in synthetic speech.* Speech Communication, Seiten 369–390, 1995. 15

[MA96] MURRAY, I. R. und J. L. ARNOTT: *Synthesizing Emotions in Speech: Is it Time to Get Excited?* In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Band 3, Seiten 1816–1819, Philadelphia, PA, 1996. 15

[NMF+95] NASS, C.,Y . MOON,B.J. FOGG,B. REEVES und C. DRYER: *Can Computer Personalities be Human Personalities.* In: *CHI*, May 1995. 12

[OCC88] ORTONY, A., G. L. CLORE und A. COLLINS: *The Cognitive Structure of Emotions.* Cambridge University Press, Cambridge, MA, 1988. vii, 8, 9, 12, 14

[PC95] PADGETT, C. und G. COTTRELL: *Identifying emotion in static face images.* In: *Proceedings of the 2nd Joint Symposium on Neural Computation*, Band 5, Seiten 91–101. University of California, San Diego, 1995. 16

[Pec91] PECKHAM, J.: *Speech understanding and dialogue over the telephone: An overview of the ESPRIT SUNDIAL project.* In: *Proceedings of the Workshop on Speech and Natural Language*, Seiten 14–27. Pacific Grove, CA, 1991. 20

[Pic97] PICARD, ROSALIND: *Affective Computing.* The MIT Press, 1997. 3, 6, 7, 8, 14, 65

[Plu94] PLUTCHIK, R.: *The psychology and biology of emotion.* Harper Collins, New York, 1994. 7

[Pol99] POLZIN, T.: *Detecting verbal and non-verbal cues in the communication of emotion.* Doktorarbeit, Carnegie Mellon University, November 1999. 16

[PW98a] POLZIN, T. und A. WAIBEL: *Detecting Emotions In Speech.* In: *Proceedings of the CMC*, 1998. 16

[PW98b] POLZIN, T. und A. WAIBEL: *Pronunciation Variations In Emotional Speech.* In: *Proceedings of the Workshop Modeling Pronunciation Variation for Automatic Speech Recognition*, Rolduc, Netherlands, May 1998. 15

[RB92] REILLY, W. SCOTT und JOSEPH BATES: *Building Emotional Agents.* Technischer Bericht CMU-CS-92-143, Pittsburgh, PA, USA, 1992. 14

[Rei96] REILLY, W. SCOTT NEAL: *Believable Social and Emotional Agents.* Doktorarbeit, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, Mai 1996. Technical Report CMU-CS-96-138. 12

[RJ97] RICKEL, JEFF und W. LEWIS JOHNSON:*Inte grating Pedagogical Capabilities in a Virtual Environment Agent.* In: JOHNSON, W. LEWIS und BARBARA HAYES-ROTH (Herausgeber): *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, Seiten 30–38, New York, 5–8, 1997. ACM Press. 13

[RN99] REEVES, BYRON und CLIFFORD NASS: *The Media Equation: How People Treat Computers, Television, and New Media like Real People and Places.* CSLI Publications, Juni 1999. reprint edition. 65, 72

[RR00] RICKENBERG, RAOUL und BYRON REEVES: *The effects of animated characters on anxiety, task performance, and evaluations of user interfaces.* In: *Proceedings of the CHI 2000 conference on Human factors in computing systems*, Seiten 49–56. ACM Press, 2000. 69

[SB96] SCHERER, K. R. und R. BANSE: *Acoustic profiles in vocal emotion expression.* Journal of Personality and Social Psychology, 70:614–636, 1996. 6, 15

[SBP97] SADEK, M. D., P. BRETIER und E. PANAGET: *ARTIMIS: Natural Language Meets Rational Agency.* In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Seiten 1030–1035, 1997. 19

[Sch86] SCHERER, K. R.: *Vocal Affect Expression: A Review and a Model for Future Research.* Psychological Bulletin, 99(2):143–165, 1986. 16

[SLS84] SCHERER, K. R., D. R. LADD und K. E. A. SILVERMAN: *Vocal cues to speaker affect: Testing two models.* Journal of the Acoustic Society of America, 76:1346–1356, 1984. 15

[SNG+02] SHIN, J., S. NARAYANAN, L. GERBER, A. KAZEMZADEH und D. BYRD: *Analysis of User Behavior under Error Conditions in Spoken Dialogs.* In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Seiten 2069–2072, 2002. 63

[Sno72] SNOW, C. E.: *Mother's speech to children learning language.* Child Development, 43:549–565, 1972. 11

[TW00] TSUKAHARA, WATARU und NIGEL WARD:*Eval uating Responsiveness in Spoken Dialog Systems.* In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP-00)*, 2000. 69

[TW01] TSUKAHARA, WATARU und NIGEL WARD: *Responding to subtle, fleeting changes in the user's internal state.* In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, Seiten 77–84. ACM Press, 2001. vii, 2, 10, 38

[Vel97] VELASQUEZ, J.: *Modeling Emotions and Other Motivations in Synthetic Agents.* In: *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97).* Providence, RI: MIT/AAAI Press, 1997. 14

[Vel98a] VELASQUEZ, J.: *Modeling emotion-based decision-making.* In: *Proceedings of the 1998 AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition.* Orlando, FL: AAAI Press, 1998. (Technical Report FS-98-03). 14

[Vel98b] VELASQUEZ, J.: *When Robots Weep: Emotional Memories and Decision-Making.* In: *Proceedings of the Fifteenth National*

Conference on Artificial Intelligence (AAAI-98), Seiten 70–75. AAAI, Madison, WI: MIT/AAAI Press, 1998. 74

[Vel99] VELASQUEZ, J.: *From Affect Programs to Higher Cognitive Emotions: An Emotion-Based Control Approach.* In: *Proceedings of the Workshop on Emotion-Based Agent Architectures (EBAA)*, 1999. 14

[War94] WARD, W.: *Extracting Information From Spontaneous Speech.* In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP-94)*, September 1994. 20

[War00] WARD, N.: *The challenge of non-lexical speech sounds.* In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP-00)*, Seiten II: 571–574, 2000. 10

[Wei66] WEIZENBAUM, JOSEPH: *ELIZA – a computer program for the study of natural language communication between man and machine.* Communications of the ACM, 9(1):36–45, 1966. 19

[WK99] WARD, NIGEL und TAKESHI KURODA: *Requirements for a Socially Aware Free-standing Agent.* In: *Proceedings of the Second International Symposium on Humanoid Robots*, Seiten 108–114, 1999. 10

[WLKA98] WALKER, MARILYN A., DIANE LITMAN, CANDACE A. KAMM und ALICIA ABELLA: *Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies.* In Computer Speech and Language, 12(3), 1998. 38

[WP99] WARD, W. und B. PELLOM: *The CU communicator System.* IEEE ASRU, Seiten 341–344, 1999. 69

[Wri96] WRIGHT, I.: *Reinforcement Learning and Animat Emotions.* In: *From Animals to Animats 4, Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB)*, Seite 272. Bradford, Oktober 1996. 14

Anhang A

System-Aufbau der Datensammlung

A.1 Komponenten der Datensammlung

Für die Datensammlung wurden folgende Komponenten verwendet

- auf dem Client-Rechner
  - Eingabe:
    - (i) Aufnahme-Tool für Spracheingabe
    - (ii) Spracherkenner
  - Ausgabe:
    - (iii) Roboter-Simulation mit Visualisierung
    - (iv) Text-to-Speech System
- auf dem Server-Rechner
  - Wizard-of-OZ Eingabekontrolle
  - Dialogsystem
  - Roboter-Applikationsschnittstelle mit Datenbank und Kontextmodell
  - Wizard-of-OZ Ausgabekontrolle

A.2 Wizard-of-OZ System

Die Architektur der Wizard-of-OZ Komponente ist in Abbildung A.1 dargestellt. Der Name Wizard-of-OZ bezeichnet ein System, das für den Benutzer den Anschein macht, eine reine Maschine zu sein, aber hinter der Fassade von einem Menschen betrieben wird. Das in dieser Arbeit erstellte Wizard-of-OZ System bietet die Möglichkeit, das System automatisch laufen zu lassen und nur bei Bedarf einzugreifen. Es bietet die Möglichkeit, die Spracheingabe des Benutzers zu verändern, einen Emotionswert zu setzen und die modifizierte Eingabe weiter an den Dialogmanager zu senden. Außerdem kann auch die Ausgabe des Dialogmanagers verändert oder komplett simuliert werden. Dies beinhaltet eine komplette Steuerung der Roboter-Simulation und den Text, der an die Sprachsynthese gesendet wird.

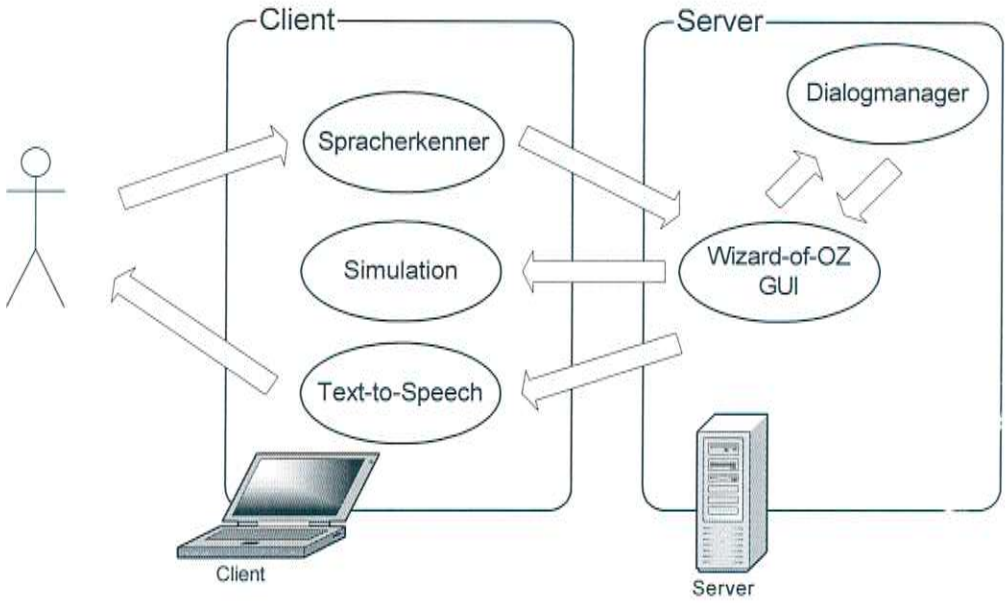

Abbildung A.1: Client-Server-Architektur für das Wizard-of-OZ Experiment

Bei den ersten Versuchen fiel auf, dass es zu lange dauert, den Text einzutippen, während der Benutzer auf eine Antwort des Systems wartet. Während die normale Antwortzeit inklusive Spracherkenner unter einer Sekunde lag, lag sie mit tippen bei mehreren Sekunden. Als Verbesserung wurden alle Befehle durch Auswahlboxen über die grafische Oberfläche der Wizard-of-OZ Schnittstelle zugänglich gemacht. Dies beinhaltet alle Befehle, die das System versteht und alle Kommandos, die an die Simulation geschickt werden

können. Zusätzlich ist für die Sprachausgabe eine Liste aller vom System gegebenen Antworten verfügbar, aus der die passende Antwort ausgewählt werden kann. Diese Erweiterung erleichterte den Umgang mit dem Wizard-of-OZ System und beschleunigte erheblich die Antwortzeit des Gesamtsystems auf knapp zwei Sekunden. Um dem Leser einen besseren Eindruck zu verschaffen, zeigt Abbildung A.2 die grafischen Oberfläche des Wizard-of-OZ Systems.

A.3 Roboter-Simulation

Simulation

Um realistische Experimente durchführen zu können, wurde eine Roboter-Simulation entwickelt. Als Szenario wurde die Aufgabe gewählt, den Roboter zum Tischdecken zu instruieren. Die Simulation verwendet eine zweidimensionale Darstellung des Raumes mit einem Tisch und einer Theke. Der Roboter kann sich frei im Raum bewegen. Zusätzlich gibt es eine Reihe von einzelnen Objekten, wie Tassen, Messer, Gabeln, die an verschiedenen Stellen positioniert werden können. Der rechte Rand des Fensters enthält eine Statusanzeige und Schalter zum Starten und Stoppen der Applikation. Die Statusanzeige bietet Informationen darüber, welches Objekt der Roboter gerade in der Hand hält und zeigt einen Sekunden-Zähler an, der die abgelaufene Zeit der Anwendung misst. Ein Abbild des Simulationsfensters ist in Abbildung A.3 auf Seite 87 gezeigt.

Die Simulation bietet die Möglichkeit über Schnittstellen niedriger Stufe den Roboter an eine beliebige Position zu bewegen oder Objekte an einer beliebigen Stelle mit beliebiger Ausrichtung zu positionieren. Die Positionen werden als X-Y-Koordinaten angegeben und beziehen sich auf zweidimensionale Positionen in einer Matrix. Die Standardeinstellung hat eine Matrix der Größe 10x10, wobei jedes Feld 40 Pixel breit und hoch ist. Die Größe der Matrix kann beim Starten angegeben werden, die Pixelgröße der Felder kann über einen Menüpunkt der Simulation geändert werden.

Über Schnittstellen hoher Stufe, bietet die Simulation die Möglichkeit den Roboter objektgebunden zu bewegen (z.B. die Anweisung zum Tisch zu fahren). Weiterhin werden Methoden angeboten, um den Tisch für bis zu vier Personen zu decken und dabei einzelne Objekte auf dem Tisch zu positionieren und zu entfernen. Die Logik, in welcher Richtung und auf welcher exakten Position die Objekte auf dem Tisch positioniert werden, ist in die Simulation integriert.

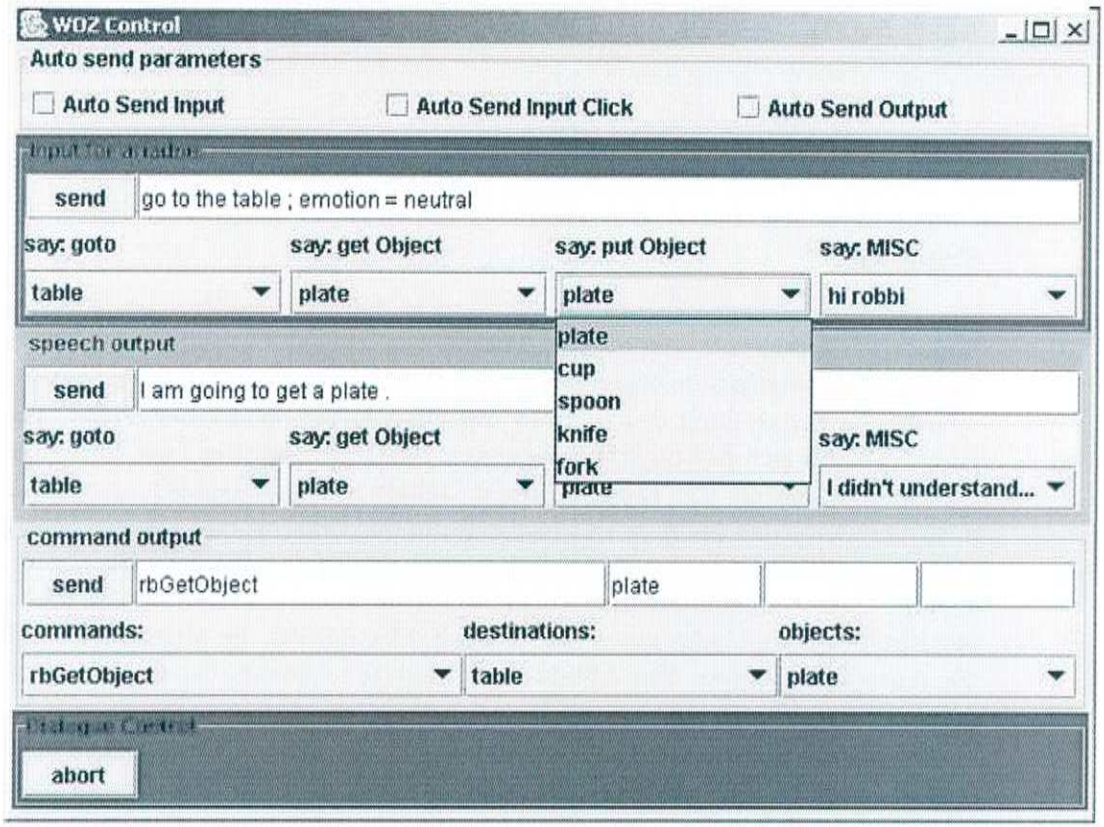

Abbildung A.2: Grafische Oberfläche der Wizard-of-OZ Schnittstelle

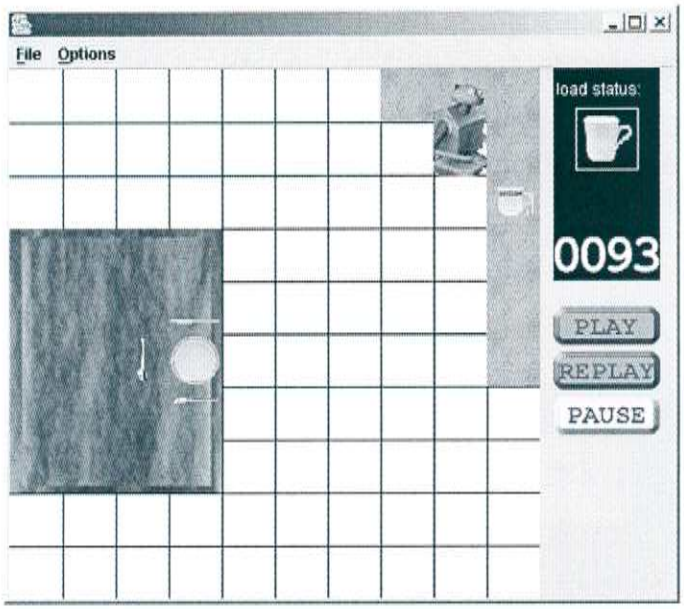

Abbildung A.3: Roboter-Simulation: Roboter beim Tischdecken

A.4 Aktien-Assistent

Zur Realisierung der grafischen Darstellung und der Sprachausgabe wurde die MSAgent Programmier-Schnittstelle (MSAgent API) benutzt. Genaue Informationen zur Verwendung der Schnittstelle sind auf der Webseite des Microsoft Development Networks[1] zu finden. Unterschiedliche Charaktere kann man ebenfalls über die Seite von Micosoft oder im MSAgent-Ring[2] finden.

Für die Darstellung des Aktien-Assistenten wurde die Figur James gewählt. Abbildung A.4 zeigt James mit Sprechblase, die während der Sprachausgabe angezeigt wird.

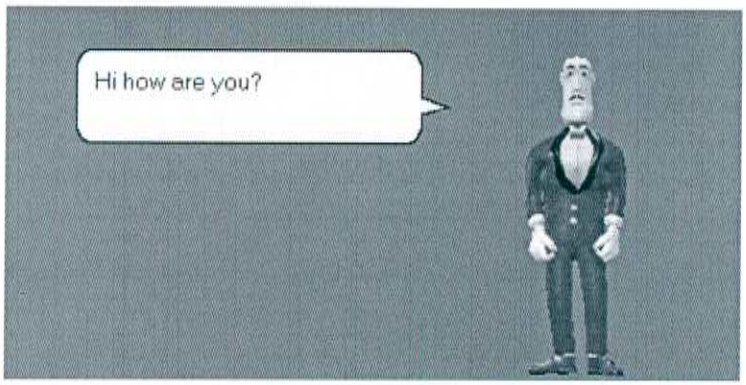

Abbildung A.4: "James" mit Sprechblase

[1] http://msdn.microsoft.com/msagent
[2] http://www.msagentring.org/

Anhang B

Kurzdokumentation Dialogziele, Grammatik und Schablonentypen

Dieses Kapitel liefert eine kurze Beschreibung zur Syntax der Dialogziele, Grammatikfragmente und Schablonentypen, die in dieser Arbeit verwendet werden.

B.1 Dialogziele und Schablonentypen

Allgemeiner Aufbau

Der Aufbau von Dialogzielen und Schablonentypen ähnelt sich stark und enthält jeweils eine Kopfzeile, Vorbedingungen und Aktionen.

Aufbau Dialogziel:

```
goal <Name> {
  <Vorbedingungen>
  ->
  bindings:
  <Aktionen>
}
```

Aufbau Schablonentyp:

```
move <Name> (optional: <Ereignis>) {
  <Vorbedingungen>
  ->
```

```
  bindings:
  <Aktionen>
}
```

Die Kopfzeile enthält eines der Schlüsselwörter goal oder move, welches das Konstrukt als Dialogziel oder als Schablonentyp kennzeichnet und den eindeutigen Namen des Konstrukts. Ein Schablonentyp kann zusätzlich ein Ereignis definieren, infolge dessen der Schablonentyp ausgeführt wird.

Die Vorbedingungen können die Merkmalsstruktur des Diskurses, Variablenabfragen und Abfragen einzelner Pfade im Diskurs enthalten.

Aktionen

Die Aktionen deklarieren Methodenaufrufe über eine Schnittstelle, die das Dialogsystem zur Verfügung stellt. Eine Aktion wird definiert durch

```
<Protokoll>://<Modul>/<Methode> <Parameterliste>;
```

Aktuell stehen zwei Protokolle zur Verfügung.

- Das Protokoll *internal* bietet einen Zugriff auf eine Reihe von Objekten, die ARIADNE zur Verfügung stellt. Der Platzhalter Modul bezeichnet das jeweilige Objekt, auf dem die spezifizierte Methode mit den entsprechenden Parametern aufgerufen wird.

- Das Protokoll *jpkg* ermöglicht die Anbindung einer externen Java-Applikation. Der Platzhalter Modul beschreibt eine TCP-Adresse[1], auf der ein Dienst zur Anbindung der Applikation läuft. Über diesen Dienst kann die angegebene Methode mit den entsprechenden Parametern ausgeführt werden.

B.2 Grammatik

Eine Regel der Grammatik wird in der Form
$linkeSeite = ABLEITUNG\_1 : ABLEITUNG\_2 : ... ;$
geschrieben, wobei mindestens eine Ableitung angegeben sein muss. Die linke Seite besteht aus dem optionalen Schlüsselwort *public* und einem Nichtterminalsymbol. Das Wort *public* ist optional und markiert ein Startsymbol der Grammatik. Ein Nichtterminalsymbol wird durch einen Vektor mit drei

[1] TCP-Adressen werden in der Form *Rechnername* : *Portnummer* angegeben

Einträgen in der Form $< sem, syn, sub >$ dargestellt, wobei *sem* die semantische Klasse, *syn* die syntaktische Kategorie und *sub* die syntaktische Unterkategorie ist. Eine Ableitung besteht aus Nichtterminalsymbolen, Terminalsymbolen, und Konvertierungsattributen. Die Konvertierungsattribute sind semantische Werte und können in geschweiften Klammern hinter die Elementen stehen. Sie werden zur Konvertierung in die Merkmalsstruktur benutzt. Durch einen Stern (*) hinter einem Element wird dieses als optionales Element markiert.

Beispiel Grammatikregel (siehe Abbildung 4.4):

```
public <backch,_,_> =
    'mhm'  { BCKCH_ID "mhm" }
  : 'ohh'  { BCKCH_ID "ohh" }
  : 'well' { BCKCH_ID "well" };
```

Beispiel Grammatikregeln (siehe Abbildung 4.3):

```
<obj_thing,NP,_> =
    <hlp_det>* <modif,A,_> 'thing';

<hlp_det> =
    'the'
  : 'a';

<modif,A,_> =
    'stupid' { EMO "negative" }
  : 'beautiful' { EMO "positive" };
```

Anhang C

Dialoganwendungsbeschreibung

C.1 Bahn-Anwendung Klärungsfragen

```
/* -------
 * ask departure town
 * -------*/

generate infoqst {
  variable:(Intention = determined),
  goal:(SellTicket = determined),
  path:($sem.[DEPARTURE_TOWN] is undefined),
  path:($sem.[DESTINATION_TOWN]   is unique)
  ->
  bindings:
    internal://dialogue/say
                        "You want a ticket to ",
                        $objs.first.[DESTINATION_TOWN|TOWN_NAME],
                        ". At which town do you want to start?";
    internal://dialogue/addSymbolToSubGrammar
                        "speech", [DEPARTURE_TOWN], <town_dep,NP,_> ;
    internal://dialogue/createSubGrammar
                        "answer_departure.soup";
};

generate infoqst {
  variable:(Intention = determined),
  goal:(SellTicket    = determined),
  path:($sem.[DEPARTURE_TOWN] is undefined) ->
```

```
  bindings:
    internal://dialogue/say
                        "At which town do you want to start?";
    internal://dialogue/addSymbolToSubGrammar
                        "speech" , [DEPARTURE_TOWN], <town_dep,NP,_> ;
    internal://dialogue/createSubGrammar
                        "answer_departure.xml";
};

/* -------
 * ask destination town
 * -------*/
generate infoqst {
  variable:(Intention = determined),
  goal:(SellTicket = determined),
  path:($sem.[DESTINATION_TOWN]   is undefined),
  path:($sem.[DEPARTURE_TOWN] is unique)
  ->
  bindings:
    internal://dialogue/say
                       "You want to leave from ",
                       $objs.first.[DEPARTURE_TOWN|TOWN_NAME],
                       ". What is your destination town?";
    internal://dialogue/addSymbolToSubGrammar
                        "speech" , [DESTINATION_TOWN], <town_dest,NP,_> ;
    internal://dialogue/createSubGrammar
                        "answer_destination.soup";
};

generate infoqst {
  variable:(Intention = determined),
  goal:(SellTicket = determined),
  path:($sem.[DESTINATION_TOWN]   is undefined) ->
  bindings:
    internal://dialogue/say
                        "What is your destination town?";
    internal://dialogue/addSymbolToSubGrammar
                        "speech" , [DESTINATION_TOWN], <town_dest,NP,_> ;
    internal://dialogue/createSubGrammar
                        "answer_destination.xml";
};
```

```
/* ================================
 * all information available
 * ask for confirmation
 * ================================*/

generate infoqst {
  variable:(Intention = determined),
  goal:(SellTicket = determined),
  path:($sem.[DEPARTURE_TOWN] is unique),
  path:($sem.[DESTINATION_TOWN]   is unique)
  ->
  bindings:
    internal://dialogue/say
                        "You want a ticket from",
                        $objs.first.[DEPARTURE_TOWN|TOWN_NAME], "to",
                        $objs.first.[DESTINATION_TOWN|TOWN_NAME],
                        ". Is this correct?";
    internal://dialogue/addSymbolToSubGrammar
                        "speech" , [CONFIRM], <meta_confirm,VP,_> ;
    internal://dialogue/createSubGrammar
                        "explicit_confirm.soup";
};


/* -------
 * ask departure time
 * -------*/
generate infoqst {
  variable:(Intention = determined),
  goal:(SellTicket = determined),
  path:($sem.[TIME]   is undefined),
  path:($sem.[DEPARTURE_TOWN] is unique)
  ->
  bindings:
    internal://dialogue/say
                        "At which time do you want to leave from ",
                        $objs.first.[DEPARTURE_TOWN|TOWN_NAME], "?";
    internal://dialogue/addSymbolToSubGrammar
                        "speech" , [DESTINATION_TOWN], <obj_time> ;
    internal://dialogue/createSubGrammar
```

```
                        "answer_dep_time.soup";
};

generate infoqst {
  variable:(Intention = determined),
  goal:(SellTicket = determined),
  path:($sem.[TIME]   is undefined) ->
  bindings:
    internal://dialogue/say
                        "At which time do you want to leave?";
    internal://dialogue/addSymbolToSubGrammar
                        "speech" , [TIME], <obj_time> ;
    internal://dialogue/createSubGrammar
                        "answer_dep_time.soup";
};
```

Anhang D

Emotionserkennung

Dieses Kapitel beschreibt ein einfaches Experiment zur Klassifikation von Emotionen. Da für diese Arbeit kein funktionsfähiger Emotionserkenner zur Verfügung stand, sollte dieses Experiment zeigen, ob mit einfachen Merkmalen zwischen den Emotionen wütend und neutral unterschieden werden kann. In dem Experiment wurde versucht, zwischen hoher und niedriger emotionaler Intensität (Arousal) zu trennen. Der Vorteil dabei ist, dass Arousal relativ gut bestimmt werden kann, während Valenz wesentlich komplizierter ist [BB99]. Der Nachteil ist, dass dabei nicht zwischen positiven und negativen Emotionen, mit gleichen Arousal-Werten, getrennt werden kann. Die Merkmale wurden entsprechend der Beschreibung von Ball und Breese [BB99] übernommen. Sie beschreiben, dass Arousal stark mit der mittleren Grundfrequenz und Energie im Sprachsignal korreliert ist. Mit diesen Merkmalen wurde ein mehrschichtiges Perzeptron mit einer versteckten Schicht sprecherunabhängig trainiert.

D.1 Korpus

Als Grundlage stand ein Emotionskorpus des LDC[1] zur Verfügung. Das Korpus enthält 16 Emotionsklassen. Für eine Aufnahme wurde nur ein Wort mit jeweils unterschiedlichen Emotionen gesprochen. Die Aufnahmen wurden von 20 ausgebildeten Schauspieler (jeweils zehn weibliche und zehn männliche) aufgenommen. Für die Klassifikation wurden jede Emotionsklasse entweder hoher Intensität (*high arousal*), niedriger Intensität (*low arousal*) oder keiner Gruppe zugeordnet. Die Zuordnung erfolgte entsprechend den Definitionen der Grundlagenliteratur über die Emotionsnamen und Hörproben zufolge.

[1] Linguistic Data Consortium *http://www.ldc.upenn.edu/*

Emotionskategorie	Zuordnung
anxiety	keine
boredom	low
coldAnger	high
contempt	keine
despair	keine
disgust	high
elation	keine
happy	high
hotAnger	high
interest	keine
neutral	low
panic	high
pride	low
sadness	low
shame	low

Tabelle D.1: Emotionskategorien des LDC-Korpus und deren Zuordnung zu hoher Intensität (*high*) bzw. niedriger Intensität (*low*)

Zweifelhafte Fälle wurden aus der Testmenge ausgeschlossen. Tabelle D.1 zeigt die Emotionskategorien mit ihren Zuordnungen.

D.2 Merkmale

Die verwendeten Merkmale Grundfrequenz und Energie (logarithmierte Werte) wurden je als ein Merkmal, durch Mittelwertbildung, für eine komplette Äußerung (jeweils ein Wort) berechnet. Die Grundfrequenz wurde mit dem Werkzeug Snack[2] bestimmt. Zusätzlich wurde eine sprecherspezifische Normalisierung durchgeführt. Die nicht normalisierten Merkmale weisen besonders bei der Grundfrequenz eine hohe Streuung auf, die teilweise größer ausfällt als sprecherspezifische Unterschiede verschiedener Emotionsklassen. Die Normalisierung berechnet für jeden Wert die Differenz zum sprecherspezifischen Mittelwert eines Merkmals. Diese Berechnung erfolgt sowohl für die Energie, als auch für die Grundfrequenz. Abbildung D.1 zeigt die enge Ballung aller Emotionswerte der Klasse "Neutral".

Das Energie-Merkmal gibt weder in der nicht normalisierten Version noch

[2]Snack Sound Toolkit *http://www.speech.kth.se/snack/*

in der normalisierten Version gute Hinweise für eine Klasse. Der Grund dafür ist, dass verschiedene Aufnahmen mit unterschiedlichem Pegel aufgenommen wurden, was nicht trivial in eine sinnvolles Energie-Merkmal umgewandelt werden konnte. Dennoch scheint Energie im Signal generell ein sinnvolles Maß zu sein, wenn entsprechende Voraussetzungen für die Aufnahmen eingehalten werden können. Zum Beispiel muss immer gleiche Distanz zwischen Mund und Mikrophon eingehalten werden, um zu hohe und zu niedrige Pegel zu vermeiden. Zusätzlich muss natürlich eine Normalisierung durchgeführt werden, um Merkmale im gleichen Wertebereich, wie in der Trainingsmenge zu erhalten.

Abbildung D.2 zeigt die normalisierten Werte der kompletten Klasse für *low arousal* und Abbildung D.3 zeigt die normalisierten Werte der kompletten Klasse für *high arousal*.

D.3 Evaluation

Das Training des einfachen Neuronalen Netzes ergab eine Erkennungsgenauigkeit von 63% wobei beide Klassen als gleich wahrscheinlich und der Klassifikationsfehler ebenso gleich gewichtet wurde. Weshalb keine größere Genauigkeit erreicht werden konnte, sieht man aus der Übereinanderlegung beider Emotionsklassen in Abbildung D.4.

D.4 Einsetzbarkeit

Die Daten scheinen zwar ein relativ gutes Ergebnis zu bescheinigen, allerdings liegt das auch in der Art der Daten begründet. Die *hotAnger* Daten (Abbildung D.5) sind von der Klasse *low arousal* deutlich trennbar, allerdings sind diese Aufnahmen sehr deutlich als Wut, durch eine hohe Stimme und lautes Schreien erkennbar. Die *coldAnger* Daten (Abbildung D.6), die zwar erkennbar Wut zeigen, aber nicht so überzogen sind wie die *hotAnger* Daten, lassen sich hingegen nur ansatzweise von der Klasse *low arousal* trennen.

Aus diesen Gründen ist der entstandene Emotionserkenner nicht genügend geeignet, dass man ihn in der Benutzerstudie, wie in Kapitel 6.2 beschrieben, einsetzen kann. Die dort verwendete Dialog-Applikation setzt voraus, dass der Emotionserkenner Wut nur mit großer Präzision klassifiziert. Allerdings kann man sich als Sprecher so auf den Emotionserkenner adaptieren, dass er in einer Demonstration eingesetzt werden könnte.

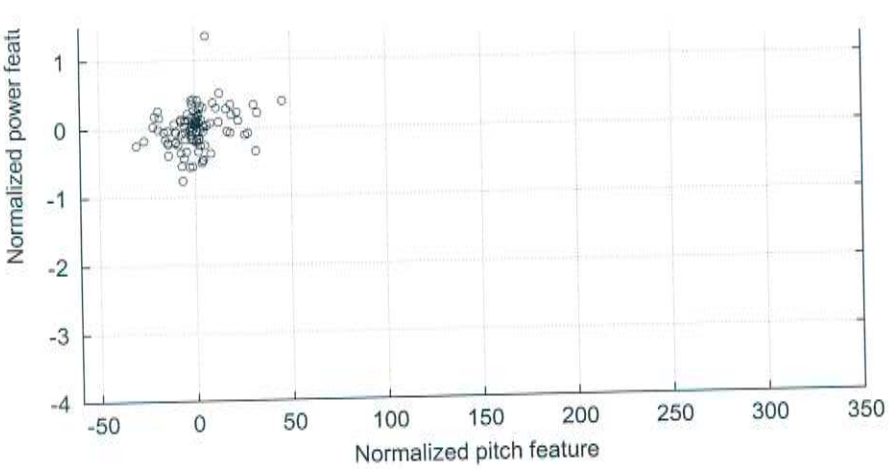

Abbildung D.1: Emotion "Neutral" mit normalisierten Werten von männlichen und weiblichen Sprechern

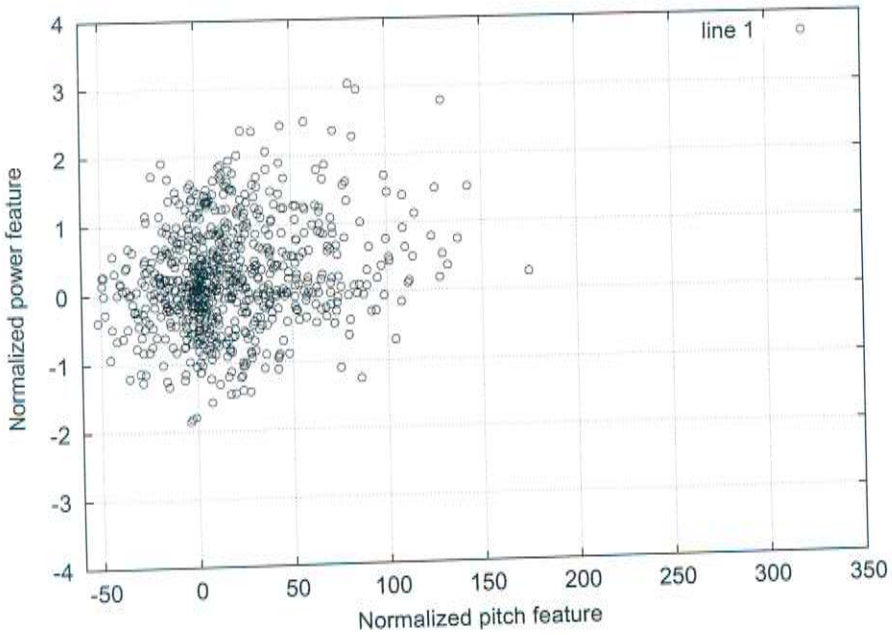

Abbildung D.2: LOW Class

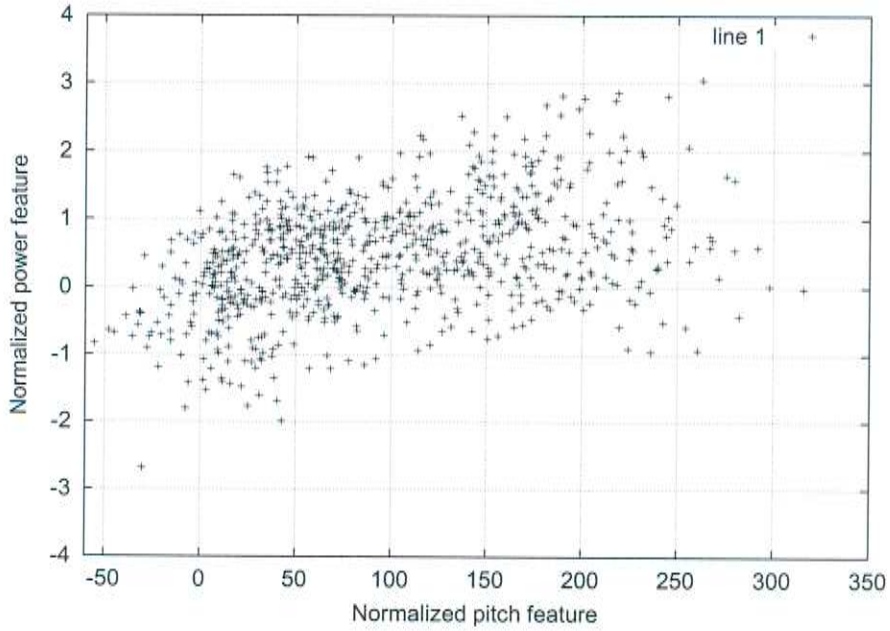

Abbildung D.3: HIGH Class

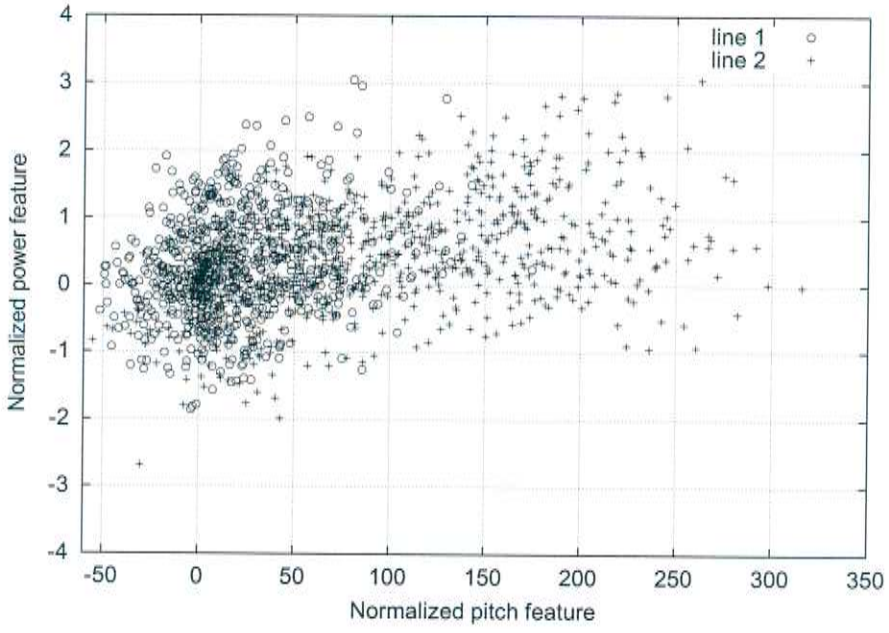

Abbildung D.4: BOTH Classes

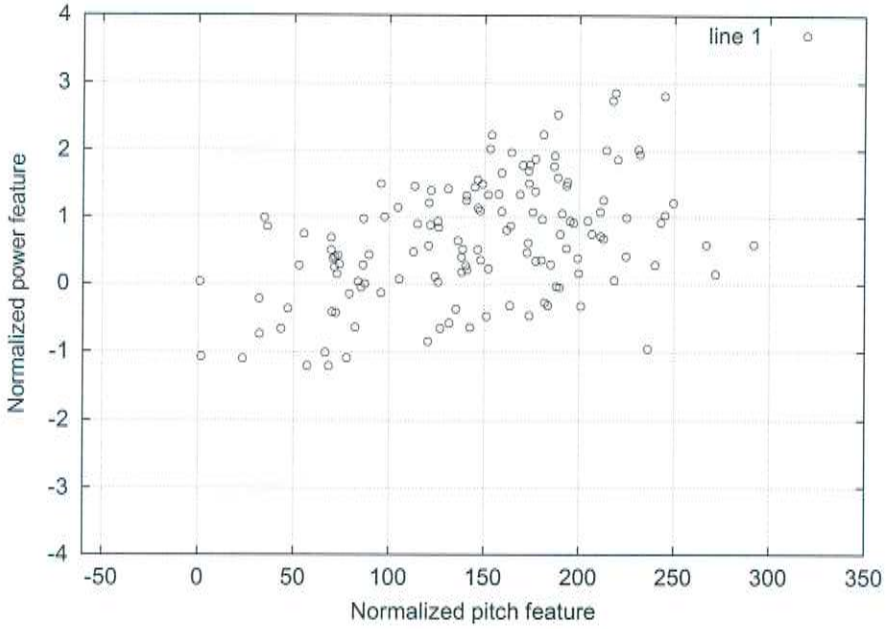

Abbildung D.5: Die hotAnger Daten

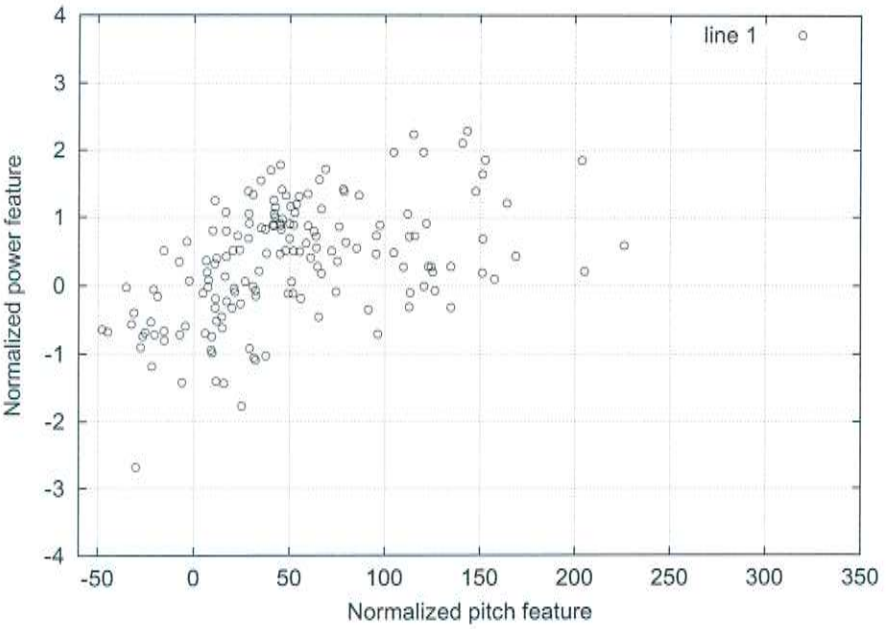

Abbildung D.6: Die coldAnger Daten

Anhang E

Fragebogen

- Zufriedenheit mit dem System allgemein
- Zufriedenheit mit der Leistung des Systems
- Zufriedenheit mit der eigenen Leistung
- Emotionen während dem Dialog, z.B. gegenüber dem System
- sollte das System besonders auf die Emotionen reagieren?
- Verbesserungsvorschläge, Lob und Kritik