

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

Klärungsfragen und Subdialoge durch  
Anomalieanalyse  
für natürlichsprachliche Dialogsysteme

Studienarbeit  
von  
cand. Inform. Ulf Krum

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

März 2005

# Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Einleitung</b>   | <b>1</b>  |
| 1.1      | Motivation . . . . .  | 1         |
| 1.2      | Gliederung . . . . .  | 1         |
| <b>2</b> | <b>Grundlagen</b>   | <b>3</b>  |
| 2.1      | Aufbau von Dialogsystemen . . . . .                                 | 3         |
| 2.2      | Strategien zur Dialogsteuerung . . . . .                            | 4         |
| 2.2.1    | Endliche Automaten . . . . .  | 5         |
| 2.2.2    | Rahmenbasierte Systeme . . . . .                                    | 6         |
| 2.2.3    | Agentenbasierte Systeme . . . . .                                   | 6         |
| 2.3      | Klärungsdialoge . . . . .   | 7         |
| 2.4      | Optimierung von Dialogen . . . . .                                  | 8         |
| 2.5      | Der Dialogmanager Tapas . . . . .                                   | 9         |
| 2.5.1    | Dialogziele und Diskurs . . . . .                                   | 9         |
| 2.5.2    | Abstrakter Dialogzustand . . . . .                                  | 10        |
| 2.5.3    | Dialogsteuerung . . . . .   | 11        |
| <b>3</b> | <b>Bestehende Arbeiten</b>  | <b>15</b> |
| 3.1      | Die Holdstrategie . . . . .   | 15        |
| 3.2      | Konfidenzen als Entscheidungsgrundlage für Klärungsfragen . . . . . | 16        |
| <b>4</b> | <b>Beschreibung der Arbeit und Theorie</b>                          | <b>19</b> |
| 4.1      | Einsatz von Klärungsdialogen in der Dialogsteuerung . . . . .       | 19        |
| 4.1.1    | Kriterien für Klärungsdialoge . . . . .                             | 19        |
| 4.1.2    | Endliche Automaten für Klärungsdialoge . . . . .                    | 20        |
| 4.2      | Trigger für Klärungsdialoge . . . . .                               | 20        |
| 4.2.1    | Die Datenfuß-Anomalieanalyse . . . . .                              | 21        |
| 4.2.2    | Anomalieanalyse der Zustandsfolgen in Tapas . . . . .               | 21        |
| 4.2.3    | Trigger als Klassifikatoren für Anomalien . . . . .                 | 23        |
| 4.2.4    | Entkoppelung von Klärungsdialogen und Dialogstrategie . . . . .     | 24        |
| 4.3      | Operationen und Relationen für Zustandsvariablen . . . . .          | 24        |

|          |  |           |
|----------|--|-----------|
| 4.4      | Katalog von Triggern und Klärungsdialogen . . . . .        | 24        |
| 4.4.1    | Missverständnis . . . . .                                  | 25        |
| 4.4.2    | Subdialog . . . . .  | 25        |
| 4.4.3    | Ruhezeit . . . . .   | 25        |
| 4.4.4    | Alternative . . . . .                                      | 26        |
| 4.5      | Triggerlose Klärungsdialoge . . . . .                      | 27        |
| 4.5.1    | Fortführung . . . . .                                      | 27        |
| <b>5</b> | <b>Experimente</b>   | <b>29</b> |
| 5.1      | Beschreibung . . . . .                                     | 29        |
| 5.2      | Systeme im Vergleich . . . . .                             | 29        |
| 5.3      | Testszenarien . . . . .                                    | 30        |
| 5.4      | Durchführung . . . . .                                     | 30        |
| 5.5      | Maße . . . . .   | 31        |
| 5.5.1    | Erfüllungsrate . . . . .                                   | 31        |
| 5.5.2    | Logische Erfüllungsrate . . . . .                          | 31        |
| 5.5.3    | Anzahl der Userturns . . . . .                             | 32        |
| 5.5.4    | Natürlichkeit des Dialoges . . . . .                       | 32        |
| 5.5.5    | Länge des Dialoges . . . . .                               | 32        |
| 5.5.6    | Behandlung von Erkennungsfehlern . . . . .                 | 32        |
| <b>6</b> | <b>Ergebnisse und Diskussion</b>                           | <b>33</b> |
| 6.1      | Ergebnisse der Experimente . . . . .                       | 33        |
| 6.2      | Analyse der Ergebnisse . . . . .                           | 35        |
| 6.2.1    | Logische Erfüllungsrate und Anzahl der Userturns . . . . . | 35        |
| 6.2.2    | Erfüllungsrate und abgebrochene Dialogziele . . . . .      | 35        |
| 6.2.3    | Länge der Dialoge . . . . .                                | 35        |
| 6.2.4    | Natürlichkeit der Dialoge . . . . .                        | 36        |
| 6.3      | Subdialoge und Erkennungsfehler . . . . .                  | 36        |
| <b>7</b> | <b>Zusammenfassung</b>                                     | <b>39</b> |
| 7.1      | Ausblick . . . . .   | 39        |
| <b>8</b> | <b>Danksagung</b>  | <b>41</b> |

# Abbildungsverzeichnis

|     |   |    |
|-----|---|----|
| 2.1 | Aufbau eines aufgaben orientierten Dialogsystems . . . . .  | 3  |
| 2.2 | Beispieldialog mit einem System basierend auf endlichen Automaten. Entnommen aus [McT02] . . . . .  | 5  |
| 2.3 | Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02] . . . . .  | 6  |
| 2.4 | Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02] . . . . .  | 6  |
| 2.5 | Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02] . . . . .  | 7  |
| 2.6 | Nachfrage, die auf einem akustischen Problem basiert. Entnommen aus [Sch04] . . . . .   | 7  |
| 2.7 | Nachfrage, die auf einer mehrdeutigen Äußerung basiert. Entnommen aus [Sch04] . . . . .   | 8  |
| 2.8 | Werte und Zustandsübergänge für INTENTION . . . . .   | 11 |
| 2.9 | Diskurs als Typisierte Merkmalstruktur dargestellt. . . . .   | 12 |
| 3.1 | Beispiel für Konfidenzen für "Withdraw fifteen hundred dollars from savings", in einem rahmen basierten Dialogsystem. Entnommen aus [Gab03] . . . . . | 17 |
| 4.1 | Einbettung von Klärungsdialogen in die Dialogsteuerung . . . . .  | 20 |
| 4.2 | Variable mit zugehöriger Zugriffssequenz über einen Programmpfad. Entnommen aus [Bal98]. . . . .  | 21 |
| 4.3 | Anomalie, die durch falsche Sprachverarbeitung entstand. . . . .  | 22 |
| 4.4 | Anomalie und drohender Kontextwechsel, durch falsche Sprachverarbeitung. . . . .  | 22 |
| 4.5 | Anomalie, die durch eine Änderung der Benutzerintention zustande kam. . . . .   | 23 |
| 4.6 | Missverständnis . . . . .   | 25 |
| 4.7 | Subdialog . . . . .   | 26 |
| 4.8 | Alternative . . . . .   | 27 |
| 4.9 | Fortführung . . . . .   | 28 |

# Tabellenverzeichnis

|     |  |    |
|-----|--|----|
| 6.1 | Durchschnittliche Erfüllungsrate von Dialogzielen . . . . .  | 33 |
| 6.2 | Durchschnittliche logische Erfüllungsrate von Dialogzielen . . .                                   | 33 |
| 6.3 | Durchschnittliche Dialoglänge in Benutzerturns pro logisch fi-<br>nalisiertem Dialogziel . . . . . | 33 |
| 6.4 | Durchschnittliche Bewertung der Natürlichkeit eines Dialoges .                                     | 34 |
| 6.5 | Durchschnittliche Bewertung der Länge eines Dialoges . . . . .                                     | 34 |
| 6.6 | Rahmendaten der Experimente . . . . .  | 34 |

# Kapitel 1

## Einleitung

### 1.1 Motivation

Die Verwendung von Rechnergestützten Systemen dringt heute in fast beliebige Domänen vor. Damit geht nicht nur eine wachsende Komplexität, sondern auch ein Anforderungswandel an die verwendeten Benutzerschnittstellen einher.

In einer Umgebung, in der rechnergestützte Systeme als reine Werkzeuge verwendet werden, reicht eine kommandozeilenorientierte oder grafische Schnittstelle aus. Heute kommen rechnergestützte Systeme in Umgebungen zum Einsatz, die eine natürliche und für den Menschen intuitive Kommunikation erfordern.

Wie die verschiedenen Paradigmen der Programmiersprachen, entwickeln sich Bedienungskonzepte weg von "Maschinennähe" und hin zu Konzepten, die dem Menschen näher sind, weil er sie von Geburt an gewohnt ist. So zum Beispiel Sprache und Gestik.

Ein Dialogsystem für die natürlich sprachliche Kommunikation zwischen Mensch und Maschine soll es dem Benutzer ermöglichen, einen künstlichen Dienstleister zu steuern. Dabei muß es den Benutzer soweit unterstützen, daß dieser keinerlei initiales Training benötigt.

### 1.2 Gliederung

In Kapitel 2 werden der allgemeine Aufbau von Dialogsystemen, gängige Strategien zur Dialogsteuerung, die Rolle und Bedeutung von Klärungsdialogen, sowie das Basissystem für die vorliegende Arbeit beschrieben. Kapitel 3 beschreibt eine Strategie zur impliziten Klärung von Unsicherheiten im Dialog. Außerdem wird eine mögliche Entscheidungsgrundlage für Klärungs-

fragen vorgestellt. Die, in dieser Arbeit entwickelten, Konzepte werden in Kapitel 4 erörtert. Die zugehörigen Experimente und Ergebnisse werden in den Kapiteln 5 und 6 vorgestellt. Eine Zusammenfassung wird in Kapitel 7 gegeben.

# Kapitel 2

## Grundlagen

### 2.1 Aufbau von Dialogsystemen

Der Aufbau eines Dialogsystems für die natürlich sprachliche Kommunikation zwischen Mensch und Maschine und seine Einbettung in ein Gesamtsystem, sind in Abbildung 2.1 dargestellt und soll im Folgenden kurz beschrieben werden.

**Spracherkenner:** Der Spracherkenner hat die Aufgabe, vom Benutzer gesprochene Eingaben in Text zu umzuwandeln. Dies kann mit Hilfe von statistischen N-Gram Modellen oder kontextfreien Grammatiken geschehen. Bei multilingualen Systemen kann die Detektion der Sprache des Benutzers hinzukommen.

Als Ausgabe liefert er in der Regel eine Hypothese, die mit Konfidenzen annotiert ist, oder eine n-besten Liste von Hypothesen und zugehörigen Konfidenzen.

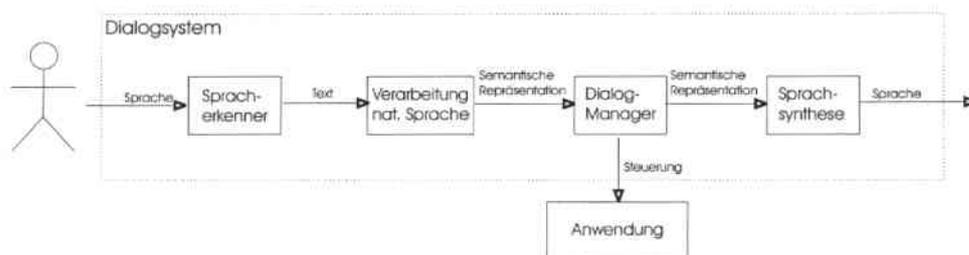


Abbildung 2.1: Aufbau eines aufgaben orientierten Dialogsystems

**Verarbeitung natürlicher Sprache:** Der Text des Spracherkenners wird hier in eine semantische Repräsentation überführt. Dabei wird überprüft, welchen Informationsgehalt der Text im Sinne einer natürlichen Sprache hat.

**Dialogmanager:** Ist die verbale Eingabe des Benutzers semantisch aufgearbeitet, muß der Dialogmanager die Information pragmatisch interpretieren und den weiteren Verlauf des Dialoges in geeigneter Weise steuern. Die Eingabe muß in einen, zum Nutzen der Anwendung passenden, Kontext gebracht werden. Objektdaten müssen evtl. aus einer Datenbank erfragt werden, sowie fehlende Information vom Benutzer akquiriert werden, sodaß Dienste der Anwendung ausgeführt werden können.

**Sprachsynthese:** Die Sprachsynthese erzeugt Sprache aus Text. Bei multilingualen Systemen muß zusätzlich der Text in der korrekten Sprache aus Vorlagen heraus generiert werden.

**Anwendung:** Die Anwendung, die gesteuert werden soll, liefert die eigentliche Daseinsberechtigung für ein aufgaben orientiertes Dialogsystem. Das Dialogsystem bildet in der Nutzungsumgebung der Anwendung eine bessere, oder sogar die einzig sinnvolle Alternative zu klassischen Ein- und Ausgabegeräten, wie Maus, Tastatur und Bildschirm.

Handelt es sich, wie in dieser Arbeit zugrunde gelegt, um ein autonomes, mobiles Robotersystem, welches Dienstleistungen im Haushalt anbietet, ist die Steuerung desselben mit Maus und Tastatur inakzeptabel [GFH<sup>+</sup>03].

Stattdessen ist es wünschenswert dem Roboter, auf einem natürlich sprachlichen Dialog basierend, Aufgaben erteilen zu können. So wie man sie auch beispielsweise einem menschlichen Bediensteten erteilt.

Weitere Beispiele sind Auskunfts- und Buchungssysteme, die per Telefon vom Benutzer erreichbar sein sollen, wie das Fahrplansystem von Philips [McT02], oder ein intelligenter Büro- bzw. Konferenzraum, wie er im CHIL Projekt entstehen [WSS04] soll.

## 2.2 Strategien zur Dialogsteuerung

Um alle, für einen Dienst der Anwendung, relevanten Daten zu erfassen, werden im wesentlichen drei Formen von informationsbasierten Strategien zur Dialogsteuerung verwendet [McT02]:

- Systeme, die auf endlichen Automaten basieren.
- Rahmenbasierte Systeme.

- Agentenbasierte Systeme.

Diese drei werden im folgenden kurz erläutert.

### 2.2.1 Endliche Automaten

Der Dialogablauf wird durch einen vordefinierten, endlichen Automaten bestimmt. Jede Benutzereingabe führt zu einem Zustandsübergang. Für jeden Zustand gibt es nur eine bestimmte Menge an Eingabeformulierungen, die in einen neuen Zustand führen. Der Benutzer hat also sehr wenig Spielraum für die Formulierung seiner Eingaben.

Ein oder mehrere Finalzustände repräsentieren die Dienste der zugrunde liegenden Anwendung. Zu jedem Finalzustand führen ein oder mehrere Pfade, entlang derer die benötigten Informationen vom Benutzer angesammelt werden.

Meist erlauben diese Systeme nur einzelne Worte oder Phrasen als Benutzereingabe, insbesondere nur eine Information innerhalb einer Eingabe. Diese Systeme sind also sehr restriktiv.

Die meisten kommerziell verfügbaren Dialogsysteme verwenden diese Form der Dialogkontrollstrategie.

Abbildung 2.2 zeigt einen Dialogablauf mit solch einem System, welches zudem die Eingabe des Benutzers in jedem Zustand verifiziert.

```
System: What is your Desitination?  
User: London.  
System: Was that London?  
User: Yes.  
System: What day do you want to travel?  
User: Friday.  
System: Was that Sunday?  
User: No.  
System: What day do you want to travel?
```

Abbildung 2.2: Beispieldialog mit einem System basierend auf endlichen Automaten. Entnommen aus [McT02]

System: What is your Desitination?  
User: London.  
System: Was that London?  
User: Yes.  
System: What day do you want to travel?  
User: Friday.

Abbildung 2.3: Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02]

System: What is your Desitination?  
User: London on Friday around 10 in the morning.  
System: I have the following connection... .

Abbildung 2.4: Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02]

## 2.2.2 Rahmenbasierte Systeme

Bei einem rahmen- oder vorlagenbasierten (framebased, templatebased) System wird der Benutzer so befragt, daß freie Felder mit Informationen gefüllt werden können. Der Dialogfluß ist in diesem Fall nicht vorherbestimmt, sondern hängt vom Informationsgehalt der jeweiligen Äußerung ab. In Abbildung 2.3 verhält sich das System wie in Abschnitt 2.2.1 beschrieben. Sind mehr Informationen in der Äußerung des Benutzers, als verlangt wurden, gehen diese nicht verloren. Und in den nächsten Fragen werden diese nicht explizit angefordert, wie in Abbildung 2.4 dargestellt.

## 2.2.3 Agentenbasierte Systeme

Agentenbasierte Dialogsysteme sind darauf ausgelegt, komplexe Kommunikation zwischen dem Benutzer und der zugrunde liegenden Anwendung zu ermöglichen, um ein Problem oder eine Aufgabe zu lösen. In Abbildung 2.5 ist ein beispielhafter Dialog gezeigt. Die Antwort des Systems ist zwar negativ, jedoch werden dem Benutzer Alternativen angeboten.

In [McT02] wird ein System vorgestellt, das dem Benutzer beim Beheben eines Fehlers in einem elektronischen Schaltkreis assistiert. Dabei geht das System auf die Fachkenntnis des Benutzers ein und gibt, falls nötig Hilfestellung.

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

Abbildung 2.5: Beispieldialog mit einem rahmenbasierten System. Entnommen aus [McT02]

A: Did you bring a 3-5 torx?  
B: A what?

Abbildung 2.6: Nachfrage, die auf einem akustischen Problem basiert. Entnommen aus [Sch04]

## 2.3 Klärungsdialoge

Klärungsdialoge können in einer Vielzahl von Variationen vorkommen und haben hauptsächlich zwei Ziele. Zum einen dienen sie dazu, akustische Missverständnisse zu eliminieren, wenn eine Äußerung des Kommunikationspartners nicht oder nur bruchstückhaft zu verstehen war. Zum anderen dienen sie dazu, Äußerungen in den vom Sprecher verstandenen Kontext zu setzen. Dies ist häufig bei Ellipsen notwendig.

Klärungsdialoge betreffen im allgemeinen nicht Objekte oder Aktionen, die zum gerade besprochenen Thema gehören, sondern markieren ein Verständnisproblem das durch vorhergehende Äußerungen entstand [Sch04].

In [Gab03] wird ein Klärungsdialog genau als Nachfrage eines Sprechers verstanden, der sich über die Äußerung des Vorgängers nicht vollkommen im Klaren ist:

We define the term *clarification question* or *clarification request* in a very narrow sense. Intuitively, a speaker asks a clarification question only when he did not (fully) understand or is uncertain about what the previous speaker said or meant with an utterance.

Abbildung 2.6 zeigt ein Beispiel für eine klärende Nachfrage, also den Beginn eines Klärungsdialoges, die aufgrund eines akustischen Problems gestellt werden sein kann.

Abbildung 2.7 zeigt ein Beispiel für eine Nachfrage, die durch die Mehrdeutigkeit einer Äußerung verursacht wurde.

- A: I saw a man with a telescope.  
B: What did you mean? Did you see  
a man who was holding a telescope,  
or did you use a telescope to watch  
him?

Abbildung 2.7: Nachfrage, die auf einer mehrdeutigen Äußerung basiert. Entnommen aus [Sch04]

Für weitere Aspekte, Formen und Ursachen klärender Fragen und Dialoge wird an hier auf [Sch04] und [Gab03] verwiesen.

Klärungsdialoge spielen eine signifikante Rolle, um in natürlich sprachlichen Mensch-Maschine Dialogen ein gegenseitiges Verstehen zu ermöglichen. Dies ist insbesondere in Anbetracht heutiger Spracherkennung notwendig, die zwar schon sehr gut, aber nicht perfekt funktionieren. Klärende Rückfragen müssen also wesentlich häufiger gestellt werden als in zwischenmenschlichen Dialogen [Gab03].

Um den Dialogablauf aber nicht mit Rückfragen zu "überfluten", ist eine geeignete Dialogstrategie nötig.

## 2.4 Optimierung von Dialogen

Natürlich sprachliche Dialoge zwischen Mensch und Maschine zu optimieren heißt, den Dialog möglichst nahe an einen zwischenmenschlichen Dialog zu bringen. Der Benutzer soll kaum bemerken, daß er einen künstlichen "Gesprächspartner" hat.

Bei der Optimierung werden die folgenden Kriterien herangezogen:

**Kriterium 1** : Korrektheit, Robustheit des Dialoges

**Kriterium 2** : Länge, natürlicher Fluß des Dialoges

Korrektheit, Robustheit und Natürlichkeit des Dialogflusses sollten maximiert, die Länge des Dialoges muß aber minimiert werden, um Dialogablauf zu optimieren.

Beide Kriterien haben gegensätzliche Auswirkungen aufeinander. Das heißt, eine möglichst robuste Dialogstrategie geht zu Lasten des natürlichen Flusses sowie der Länge des Dialoges und umgekehrt.

## 2.5 Tapas - Ein domänen- und sprachunabhängiger Dialogmanager

Als Basis für die in dieser Arbeit entwickelten Strategien zu Klärungsdialogen, wurde der domänen- und sprachunabhängige Dialogmanager *Tapas* verwendet. *Tapas* basiert auf den in der Dissertation von Matthias Dencke [Den02] beschriebenen Dialogmodellen, und ist das Nachfolgesystem des ebenfalls darauf basierenden Systems *ARIADNE*.

Ein wesentlicher Vorteil, den die in [Den02] beschriebenen Modelle mit sich bringen, ist die Tatsache, daß der Benutzer seine Eingabe auf verschiedene Weisen formulieren kann. Für den Umgang mit einem auf *Tapas* basierenden System müssen also nicht erst die richtigen Kommandos erlernt werden. Es ist im weitesten Sinne möglich, mit dem System zu kommunizieren, als hätte man einen menschlichen Gegenüber [GFH<sup>+</sup>03].

Weiterhin ist *Tapas* für ein Rapid Prototyping von natürlich sprachlichen Dialogsystemen konzipiert. Dies wird erreicht, indem nur die domänen- und sprachabhängigen Komponenten spezifiziert werden müssen.

Die Dialogsteuerungsstrategie in *Tapas* ist zielbasiert und taskorientiert. Dialogziele sind durch die, in [Den02] erläuterten, typisierten Merkmalstrukturen beschrieben.

### 2.5.1 Dialogziele und Diskurs

Den Diensten der zu steuernden Anwendung werden sog. Dialogziele zugeordnet [GFH<sup>+</sup>03]. In einer Dialogzielbeschreibung werden alle Informationen, die zur Ausführung des zugehörigen Dienstes benötigt werden, definiert. Durch die Eingabe des Benutzers werden passende Dialogziele ausgewählt. Sind mehrere Dialogziele gewählt oder nicht alle benötigten Informationen vorhanden, fordert der Dialogmanager weitere Eingaben, durch geeignete Fragen.

Ein Dialogziel muß nicht notwendigerweise mit einem oder mehreren Diensten der Anwendung verknüpft sein, obwohl dies für die meisten Fälle gilt. Es kann auch ein kommunikatives Ziel darstellen oder zur Steuerung der Dialogstrategie dienen (z.B. Zurücksetzen der Strategie).

Die Informationen in den Eingaben des Benutzers werden in einem Diskurs [GS86] gespeichert. Mit jeder neuen Eingabe wird versucht, die in ihr enthaltenen Informationen, passend mit den vorhandenen im Diskurs, zu vereinen, sodaß ein Zuwachs des Informationsgehaltes die Folge ist.

## 2.5.2 Abstrakter Dialogzustand

Zu einem bestimmten Zeitpunkt in einem Dialog befindet sich der Dialogmanager in einem dedizierten Zustand. Anhand dessen entscheidet er, was er dem Benutzer als nächstes antworten wird. Das kann eine Erfragung noch fehlender Informationen sein oder die Bestätigung der Ausführung einer bestimmten Aktion der Anwendung.

Ein Zustand  $d = \langle v_1, v_2, v_3, \dots, v_n \rangle$  wird durch die Belegung der Variablen  $v_1, v_2, v_3, \dots, v_n$ , die den sog. Abstrakten Dialogzustand beschreiben, definiert [Den02] und [HG04]. Die in dieser Arbeit berücksichtigten Zustandsvariablen sind die folgenden:

$v_1$ : **INTENTION** Gibt an, wie gut bzw. welche Dialogziele zum Diskurs passen.

$v_2$ : **SPEECHACT** Sprechakt

$v_3$ : **SELECTEDGOALS** Die Menge der selektierten Dialogziele.

$v_4$ : **FINALIZEDGOALS** Die Menge der finalisierten Dialogziele.

Weiterhin werden in *Tapas* die, in [Den02] beschriebenen Zustandsvariablen **INPUTCONFIDENCE** und **OVERALLQUALITY** berechnet. Sie repräsentieren verschiedene Konfidenzen, die in die aktuellen Dialogstrategien einfließen. Prinzipiell ist es möglich, beliebige weitere Variable hinzuzufügen, wie bspw. den **HOLDSTATE** aus [HG04], siehe Abschnitt 3.1.

### SELECTEDGOALS

Die Variable **SELECTEDGOALS** enthält eine Menge  $G_s$  mit all denjenigen Dialogzuständen  $g_i$ , die den Wert *selected* haben. Ein Dialogziel kann die Werte *selected*, *deselected* und *finalized* annehmen. Und zwar je nach dem, ob das Dialogziel zum aktuellen Diskurs passt oder nicht. Im Falle *selected* sind noch nicht alle, für dieses Ziel notwendigen, Informationen vorhanden.

### FINALIZEDGOALS

Die Variable **FINALIZEDGOALS** enthält die Menge  $G_f$  mit dem ausführbaren Dialogziel. Alle relevanten Informationen sind im Diskurs vorhanden und dem Ziel ist der Wert *finalized* zugeordnet.

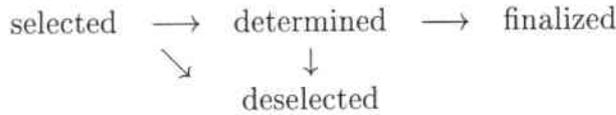


Abbildung 2.8: Werte und Zustandsübergänge für INTENTION

### INTENTION

Die Variable INTENTION kann die Werte *selected*, *determined*, *finalized* und *deselected* annehmen. INTENTION ist eine Abgeleitete Variable von SELECTEDGOALS und FINALIZEDGOALS. sie wird nach folgender Vorschrift belegt:

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

Abbildung 2.8 zeigt die möglichen Zustandsübergänge für INTENTION.

### SPEECHACT

Die Variable SPEECHACT kann Werte aus der zugrunde liegenden Grammatik annehmen (z.B. *act\_setTable*, *act\_switch*).

### 2.5.3 Dialogsteuerung

Die Dialogsteuerung kann grob in zwei Phasen geteilt werden, die nach der Sprachverarbeitung erfolgen:

1. Aktualisierung des Diskurses
2. Ausführen von Schritten der Dialogstrategie bzw. von Dialogzielen

### Typisierte Merkmalstrukturen

Die Informationen im Diskurs werden durch sog. Typisierte Merkmalstrukturen beschrieben. Sie ermöglichen es, ähnlich der Aggregation in der Objektmodellierung, hierarchisch aufgebaute Datenstrukturen zu beschreiben. Jedem Element ist ein bestimmter Typ zugeordnet [Car92]. Abbildung 2.9 zeigt ein Beispiel einer solchen Struktur.

$$\left[ \begin{array}{l} act\_switch \\ ONOFF \left[ \begin{array}{l} prp\_onoff \\ BOOL[true] \end{array} \right] \\ OBJ \left[ \begin{array}{l} obj\_lamp \\ SG[false] \\ ON[false] \\ ROBPOSEX[-1.00] \\ ROBPOSPHI[16.20] \\ ROBPOSY[1.62] \end{array} \right] \end{array} \right]$$

Abbildung 2.9: Diskurs als Typisierte Merkmalstruktur dargestellt.

### Aktualisierung des Diskurs

Die Informationen jeder Eingabe des Benutzers werden, ebenfalls wie die des Diskurses, in einer Typisierten Merkmalstruktur gespeichert. Diese muß nun in geeigneter Weise mit dem Diskurs in Einklang gebracht werden. Es ergeben sich drei Möglichkeiten, mit den Eingabedaten umzugehen:

**unifizieren:** Die alten Daten werden um die neuen ergänzt. Der Informationsgehalt nimmt, passend zum bisherigen Kontext, zu.

**abbrechen:** Die neuen Daten werden komplett übernommen und ersetzen die bisherigen des Diskurses. Das bisherige Dialogziel wird also abgebrochen.

**ignorieren:** Der alte Diskurs wird komplett beibehalten, die neuen Daten ignoriert.

Bspw. seien nach dem letzten Benutzerturn und einer Aktualisierung des Diskurses folgende Daten gespeichert:

$$\left[ \begin{array}{l} act\_switch \\ ONOFF \left[ \begin{array}{l} prp\_onoff \\ BOOL[true] \end{array} \right] \\ OBJ \left[ \begin{array}{l} obj\_lamp \\ SG[false] \\ ON[false] \end{array} \right] \end{array} \right]$$

Der Benutzer möchte also, daß eine Lampe eingeschaltet wird.

### Schritte und Dialogziele

Nun berechnet das System seine nächsten Schritte. Sind alle benötigten Informationen im Diskurs gespeichert (und damit ein dediziertes Dialogziel ausgewählt, also *finalisiert*), kann ein entsprechender Aufruf an die Anwendung erfolgen. Andernfalls müssen, mit einer entsprechenden Antwort an den Benutzer, weitere Informationen akquiriert werden.

Im Beispiel hat eine Abfrage in der Objektdatenbank eine große und eine kleine Lampe zum Ergebnis. Da das System aber nicht weiß welche gemeint ist, muß es dies vom Benutzer erfragen.

In der neuen Eingabe befinden sich die folgenden Daten:

$$\left[ \begin{array}{l} act\_switch \\ OBJ \left[ \begin{array}{l} obj\_lamp \\ NAME["thebiglamp"] \\ SG[true] \end{array} \right] \end{array} \right]$$

In der weiteren Aktualisierungsphase ist eine Unifikation möglich und der Diskurs beinhaltet danach die Merkmalstruktur aus Abbildung 2.9. Alle Daten sind vorhanden, das Dialogziel kann ausgeführt werden: die große Lampe wird angeschaltet.

# Kapitel 3

## Bestehende Arbeiten

### 3.1 Die Holdstrategie

Probleme in der Dialogsteuerung kann es geben, wenn die aktuelle Benutzereingabe inkonsistent zur Information ist, die schon im Diskurs gespeichert ist [HG04]. Es gibt zwei Fälle, die dies provozieren:

1. Es gab einen Fehler in der Spracherkennung. Der Benutzer hat also etwas anderes gesagt, als vom System verstanden wurde. Die Information kann deshalb nicht in den Diskurs aufgenommen werden.
2. Der Benutzer hat nicht das gesagt, was das System erwartet hat. Beispielsweise könnte das System eine bestimmte Information erfragen, der Benutzer möchte aber plötzlich ein ganz anderes Dialogziel verfolgen und evtl. das alte Dialogziel zu einem späteren Zeitpunkt wieder aufnehmen (Subdialog).

Der Dialogmanager kann nicht unterscheiden, welcher Fall vorliegt. Es ist also keine Lösung, einfach auf die erwartete Information zu beharren und die Frage zu wiederholen, denn es könnte Fall 2 eingetreten sein. Das Dialogziel zu wechseln ist ebenfalls nicht sinnvoll, da dies der Benutzer evtl. nicht mitbekommt, bzw. Fall 1 eintrat.

In [HG04] wurde die sog. Hold-Strategie entwickelt, um eine Form der impliziten Bestätigung durch den Benutzer zu realisieren. Dabei läßt die Strategie die Eingabe wiederholen, wenn anhand der Belegung der Variablen INTENTION, INPUTCONFIDENCE, OVERALLQUALITY, HOLDSTATE des abstrakten Dialogzustandes von einem Fehler ausgegangen werden kann. Wird die geforderte Information erneut nicht erbracht, der Benutzer beharrt also offensichtlich auf seiner Eingabe, ignoriert die Strategie die bisherige Information im Diskurs und ersetzt sie komplett durch die neue.

Die Variable `INTENTION` beschreibt, wie gut die Information im Diskurs die Intention des Benutzers trifft, siehe auch Abschnitt 2.5.2.

`INPUTCONFIDENCE` beinhaltet einen Konfidenzwert für die aktuelle Eingabe. Dieser Wert wird vom Spracherkenner geliefert.

Das Konfidenzmodell `OVERALLQUALITY` beschreibt, wie zuverlässig das System über die letzten Eingaben gearbeitet hat.

Speziell für die Holdstrategie wurde die Variable `HOLDSTATE` eingeführt. Sie beschreibt, ob die Holdstrategie angewendet werden kann, bzw. wie häufig sie angewendet wurde. Wie oft also der Benutzer zur Wiederholung seiner Eingabe aufgefordert wurde. Praktikabel ist eine Wiederholung.

Die Holdstrategie konnte gegenüber einer Standardstrategie die Anzahl der finalisierten Dialogziele und damit die Erfüllungsrate<sup>1</sup> beträchtlich erhöhen.

## 3.2 Konfidenzen als Entscheidungsgrundlage für Klärungsfragen

In [Gab03] werden zunächst verschiedene Formen von Klärungsfragen, mit unterschiedlichen Zielen, vorgestellt. Diese werden mit verschiedenen Ebenen der sog. Misskommunikation kombiniert und ergeben eine Klassifikation für Klärungsfragen.

Ein möglicher Lösungsansatz wird für das Problem diskutiert, wann und wie durch das Dialogsystem entschieden werden kann, welche Klärungsfrage generiert wird.

Für ein rahmen basiertes Dialogsystem wird vorgeschlagen, daß die Konfidenz über die gesamte Eingabe zur Entscheidung herangezogen wird. Die Klärungsfrage kann die Eingabe komplett abweisen, bestätigen lassen oder akzeptieren. Die Konfidenz wird vom Spracherkenner zusammen mit der Hypothese oder einer n-besten Liste von Hypothesen mitgeliefert.

Alternativ können die Konfidenzen der einzelnen Teilinformationen (Slots) eine Klärungsfrage veranlassen, die eben nur diese Information bestätigen läßt.

Abbildung 3.1 zeigt ein Beispiel für Konfidenzen in einem rahmen basierten Dialogsystem zur Eingabe *"Withdraw fifteen hundred dollars from savings"*.

---

<sup>1</sup>Die verwendeten Maße und ihre Bedeutung werden in Abschnitt 5.5 erläutert

### 3.2. KONFIDENZEN ALS ENTSCHEIDUNGSGRUNDLAGE FÜR KLÄRUNGSFRAGEN<sup>1</sup>

| Slot    | Value     | Confidence |
|---------|-----------|------------|
| action  | withdraw  | 49         |
| amount  | \$1500.00 | 60         |
| account | savings   | 61         |
| overall |           | 56         |

Abbildung 3.1: Beispiel für Konfidenzen für "Withdraw fifteen hundred dollars from savings", in einem rahmen basierten Dialogsystem. Entnommen aus [Gab03]

# Kapitel 4

## Beschreibung der Arbeit und Theorie

### 4.1 Einsatz von Klärungsdialogen in der Dialogsteuerung

Die Notwendigkeit für Klärungsdialoge und was man darunter versteht, wurde schon in Abschnitt 2.3 diskutiert.

Abbildung 4.1 gibt einen Überblick über die Einbettung von Klärungsdialogen in *Tapas*.

Der prinzipielle Ablauf der Dialogsteuerung in *Tapas* wurde in dieser Arbeit um die Möglichkeit erweitert, Klärungsdialoge zu starten. Nach der Aktualisierung des Diskurses und der Berechnung des Dialogzustandes, also der einzelnen Zustandsvariablen, wird überprüft, ob ein Klärungsdialog notwendig ist (Abschnitt 4.2). Je nach Antwort des Benutzers kann dann der Diskurs weiter vervollständigt werden, Teilinformationen gelöscht oder ersetzt, sowie der Diskurs auf einem Stack gespeichert werden. Letzteres ermöglicht die Eröffnung eines Subdialoges. Der Benutzer verfolgt also ein anderes Dialogziel und hat später die Möglichkeit, das vorherige wieder aufzunehmen.

Nach der Ausführung eines Klärungsdialoges werden die nächsten Schritte vom Dialogmanager ausgeführt, oder ein Funktionsaufruf an die Anwendung gesendet.

#### 4.1.1 Kriterien für Klärungsdialoge

Klärungsdialoge sollten kurz und robust gehalten werden. Kurz, um die Länge des Gesamtdialoges nicht unnötig zu vergrößern und sich damit vom eigentlichen Ziel, der Optimierung des Gesamtdialoges zu entfernen. Robust, um

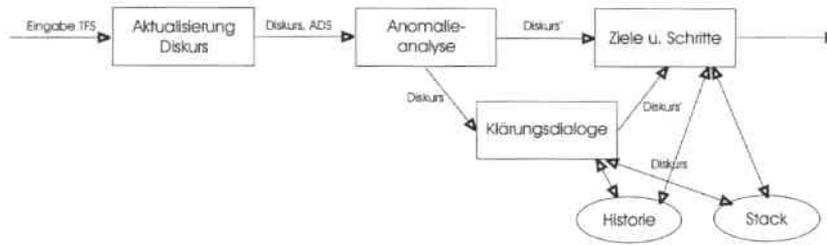


Abbildung 4.1: Einbettung von Klärungsdialogen in die Dialogsteuerung

weitere Unsicherheiten oder Missverständnisse in der Kommunikation möglichst zu unterbinden.

In dieser Arbeit, und wahrscheinlich in den meisten Fällen, beschränkt sich ein Klärungsdialog auf eine Nachfrage des Systems und einer entsprechenden Antwort des Benutzers, d.h. genau auf einem Benutzerturn.

Die Fragen sind meist geschlossene Fragen oder Alternativfragen, daher fallen die Antworten in einen klar umrissenen Rahmen.

#### 4.1.2 Endliche Automaten für Klärungsdialoge

Den Ausführungen im vorangehenden Abschnitt folgend, bieten sich endliche Automaten für die Modellierung von Klärungsdialogen an. Diese wurden bereits in Abschnitt 2.2 beschrieben.

Die Zustandsübergänge sind mit einer Semantik annotiert. Diese wird bei Eintritt in den Quellzustand mit einer Subgrammatik erzeugt.

Bei Erreichen eines Zustandes fordert die Dialogstrategie eine Information vom Benutzer, oder führt eine der oben genannten Manipulationen des Diskurses durch.

### 4.2 Trigger für Klärungsdialoge

Die Entscheidung darüber, wann die Dialogkontrollstrategie einen bestimmten Klärungsdialog einsetzt, wird in dieser Arbeit mit einer Anomalieanalyse über den Verlauf der abstrakten Dialogzustände gefällt. Die Idee dazu wurde durch die aus der Softwaretechnik bekannte Datenfluß-Anomalieanalyse [Bal98] inspiriert.

### 4.2.1 Die Datenfluß-Anomalieanalyse

Bei der Datenfluß-Anomalieanalyse werden anhand des Quellcodes statisch mögliche Programmpfade ermittelt. Entlang dieser Pfade werden Zugriffe für jede Variable annotiert. Es gibt drei verschiedene Arten von Variablenzugriffe:

- Definition (d) einer Variable; sie erhält einen Wert.
- Referenzierung (r) einer Variable; sie wird in einer Berechnung oder Bedingung verwendet.
- "Undefinierung" (u) einer Variable; der Wert wird zerstört (z.B. beim Verlassen des Scopes).

Für jede Variable wird die zugehörige Zugriffssequenz auf Anomalien untersucht.

In Abbildung 4.2 sind zwei Beispiele gegeben.

Für die Variable *max* tritt die Teilsequenz "dd" auf. Die Variable wurde zweimal in direkter Folge definiert. Der erste Wert geht also verloren.

In der Sequenz für die Variable *hilf* treten gleich zwei Anomalien auf: "ur" und "du". Bei der ersten wird *hilf* verwendet, obwohl die Variable undefiniert ist. Bei der zweiten hat die Definition keine Wirkung, weil die Variable nicht vor der Zerstörung des Wertes referenziert wird.

### 4.2.2 Anomalieanalyse der Zustandsfolgen in Tapas

Nach jeder Eingabe des Benutzers werden, wie in Abschnitt 2.5.3 beschrieben, die Daten aus der Eingabe des Benutzers in den Diskurs aufgenommen. Die Variablen des Abstrakten Dialogzustandes können neu berechnet werden. Die Eingabe hat evtl. einen Übergang in einen neuen Zustand zur Folge, falls sich der Wert einer der Variablen geändert hat.

| Variable | Sequenz |
|----------|---------|
| max      | udrddru |
| hilf     | urdu    |

Abbildung 4.2: Variable mit zugehöriger Zugriffssequenz über einen Programmpfad. Entnommen aus [Bal98].

---

|   |                 |  |
|---|-----------------|--|
| 1 | Benutzer:       | Switch on the light  |
|   | Verstanden:     | Switch on the lamps  |
|   | $d_1 = \langle$ | $determined, act\_switch, \{SwitchObjectSg\}, \{\}\rangle$ |
|   | System:         | Which lamp do you want me to switch on?                    |

---

|   |                 |  |
|---|-----------------|--|
| 2 | Benutzer:       | The big light                                    |
|   | Verstanden:     | Yes  |
|   | $d_2 = \langle$ | $deselected, yes, \{\}, \{\}\rangle$             |
|   | System:         | I misunderstood you. Please try this once again. |

---

Abbildung 4.3: Anomalie, die durch falsche Sprachverarbeitung entstand.

---

|   |                 |  |
|---|-----------------|--|
| 1 | Benutzer:       | Please set the table robbi                             |
|   | Verstanden:     | Please set the table please                            |
|   | $d_1 = \langle$ | $determined, act\_setTable, \{SetTable\}, \{\}\rangle$ |
|   | System:         | For how many Persons do you want me to set the table?  |

---

|   |                 |   |
|---|-----------------|---|
| 2 | Benutzer:       | For two persons please                    |
|   | Verstanden:     | Put two glasses please                    |
|   | $d_2 = \langle$ | $determined, act\_put, \{\}, \{\}\rangle$ |
|   | System:         | ...                                       |

---

Abbildung 4.4: Anomalie und drohender Kontextwechsel, durch falsche Sprachverarbeitung.

Über den Verlauf eines Dialoges gibt es also eine Folge von Zuständen, in denen sich die Dialogsteuerung befindet. Diese Folge ist in einer Historie (Abbildung 4.1) gespeichert und wird einer Anomalieanalyse unterzogen.

In Abbildung 4.3 führt die zweite Äußerung des Benutzers zu einer Anomalie: die Variable `INTENTION` wechselt von *determined* nach *deselected*. Die übrigen Übergänge sind dabei unerheblich. Der Grund dafür war die zweite Äußerung, "yes" kann in keinen sinnvollen Kontext gebracht werden. Das System kann nicht einmal entscheiden, ob der Benutzer ein anderes Ziel verfolgen will.

In diesem Fall liegt die Vermutung nahe, daß – wie auch geschehen – ein Fehler in der Verarbeitung der Eingabe aufgetreten ist. Dabei steht die Ursache nicht zur Debatte (z.B. Rauschen im Kanal, Nebengeräusche). Durch das Erkennen der Anomalie ist es möglich den Benutzer auf das Problem aufmerksam zu machen und zu versuchen es zu lösen. Es könnte ihn etwa auffordern, seine Äußerung zu wiederholen.

In Abbildungen 4.4 führt ebenfalls ein Fehler der Sprachverarbeitung zu ei-

---

|   |  |
|---|--|
| 1 | Benutzer: Set the table<br>$d_1 = \langle determined, act\_setTable, \{SetTable\}, \{\} \rangle$<br>System: For how many Persons do you want me to set the table?      |
| 2 | Benutzer: For one person<br>$d_1 = \langle determined, act\_setTable, \{SetTable\}, \{\} \rangle$<br>System: Which kind of glasses do you want me to put on the table? |
| 3 | Benutzer: Make me a cup of coffee<br>$d_3 = \langle determined, act\_make, \{MakeDrink\}, \{\} \rangle$<br>System: ...   |

---

Abbildung 4.5: Anomalie, die durch eine Änderung der Benutzerintention zustande kam.

ner Anomalie. Hier kann das System die zweite, falsch erkannte Äußerung in einen sinnvollen Kontext bringen, der allerdings nicht zum Diskurs passt: Nach der ersten Äußerung soll der Tisch gedeckt werden, die zweite verfolgt aber das Dialogziel etwas abzulegen. Das System kann nicht entscheiden, ob es sich tatsächlich um einen Fehler handelte, oder ob der Benutzer seine Intention geändert hat, wie es in Abbildung 4.5 der Fall ist. Ein Klärungsdialog löst das Problem und macht den Benutzer zudem darauf aufmerksam, daß Klärungsbedarf besteht.

### 4.2.3 Trigger als Klassifikatoren für Anomalien

Die Beispiele in den Abbildungen 4.4 und 4.5 zeigen zwei Anomalien, die sich in einer Klasse zusammenfassen lassen: Wenn die Variable INTENTION im vorherigen und aktuellen Zustand den Wert *determined* hat, die Mengen der Variablen SELECTEDGOALS aber disjunkt sind, kann ein beabsichtigter Kontextwechsel oder ein Fehler vorliegen. Es muß ein entsprechender Klärungsdialog gestartet werden.

Die Anomalien, die zum selben Problem führen, kann man also mit einem Satz Regeln beschreiben. Dieser Satz Regeln wird in dieser Arbeit als *Trigger* bezeichnet, da in der Folge ein passender Klärungsdialog gestartet wird.

Ein Trigger klassifiziert Anomalien und zugehörigen Klärungsdialoge.

Eine komplette und formale Beschreibung des hier gezeigten Triggers ist in Abschnitt 4.4.1 zu finden.

#### 4.2.4 Entkoppelung von Klärungsdialogen und Dialogstrategie

Die Verwendung von Triggern und das Herauslösen der Klärungsdialoge als endliche Automaten aus der kompletten Dialogstrategie, ermöglicht ein einfacheres Kombinieren verschiedener Strategien mit dem, an sich orthogonal dazu verlaufenden, Aspekt der Klärungsdialoge.

Der Ansatz passt damit gut in das Konzept von *ARIADNE* und *Tapas Rapid Prototyping* für natürlich sprachlich gesteuerte Anwendungen realisieren zu können.

### 4.3 Operationen und Relationen für Zustandsvariablen

Um die Trigger definieren zu können, müssen Werte der Variablen des abstrakten Dialogzustandes in Relation zueinander gebracht werden können. Handelt es sich bei der Wertemenge einer Variable um Skalare oder Mengen, können selbstverständlich alle darauf gültigen Operationen und Relationen verwendet werden.

Für die Variable *INTENTION* wird eine Ordnung definiert, um die üblichen Relationen anwenden zu können:

$$deselected < selected < determined < finalized$$

Die Relation entspricht dem anwachsenden Informationsgehalt des Diskurses. Der zu erstrebende Wert ist *finalized* und daher als größtes Element definiert.

### 4.4 Katalog von Triggern und Klärungsdialogen

In diesem Abschnitt werden einige Trigger und deren zugehörige Klärungsdialoge zusammengestellt. Die ersten zwei (Abschnitt 4.4.1 und 4.4.2) wurden in der Dialogstrategie dieser Arbeit implementiert und evaluiert.

Die Variablen des abstrakten Dialogzustandes sind, wenn nicht anders beschrieben, diejenigen aus Abschnitt 2.5.2.

Die Reihenfolge der Analyse welcher Trigger passt ist unerheblich, nicht aber die Ausführung des zugehörigen Klärungsdialoges. Dies ist leicht ersichtlich, wenn man sich ein System vorstellt, in dem die Trigger *Subdialog* (Abschnitt 4.4.2) und *Alternative* (Abschnitt 4.4.4) implementiert sind.

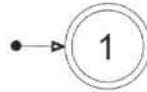
Die Trigger umfassen jeweils den aktuellen Dialogzustand  $d_k$  und den vorherigen Zustand  $d_{k-1}$ . Für die Zustände sind jeweils Belegungsregeln angegeben. Sogenannte "don't care" Belegungen sind mit "-" gekennzeichnet.

### 4.4.1 Missverständnis

Gelangt der Dialogmanager in den Zustand INTENTION = *deselected*, war eine sinnvolle Interpretation des Benutzerturns nicht möglich. Nachfolgende Belegungen lösen den Klärungsdialog in Abbildung 4.6 aus:

$$d_{k-1} = \langle -, -, -, - \rangle$$

$$d_k = \langle deselected, -, -, - \rangle$$



System: "I misunderstood you, please try this once again."

Abbildung 4.6: Missverständnis

### 4.4.2 Subdialog

Führt die letzte Eingabe des Benutzers zu einem Kontextwechsel (Wechsel des Dialogzieles) muß festgestellt werden, ob ein Fehler in der Sprachverarbeitung vorlag, oder ob der Benutzer tatsächlich das vorherige Dialogziel abbrechen möchte, bzw. nicht zu diesem Zeitpunkt weiter verfolgen möchte. Nachfolgende Belegungen lösen den Klärungsdialog in Abbildung 4.7 aus:

$$d_{k-1} = \langle \{selected|determined\}, -, G_{s_{k-1}}, \emptyset \rangle$$

$$d_k = \langle \{selected|determined|finalized\}, -, G_{s_k}, G_{f_k} \rangle$$

$$\text{mit } \forall g \in G_{s_k} : g \notin G_{s_{k-1}} \wedge g \notin G_{f_k}$$

### 4.4.3 Ruhezeit

Für den abstrakten Dialogzustand sei eine weitere Variable  $v_5$ , IDLETIME definiert. Sie beinhaltet die Zeit  $t_i$  in Sekunden, die seit der letzten Benutzereingabe verstrichen war.

$$d_{k-1} = \langle \{selected|determined\}, -, -, -, t_{k-1} \rangle$$

$$d_k = \langle -, -, -, -, t_k \rangle$$

$$\text{mit } t_k - t_{k-1} \geq t_m$$

Ist eine bestimmte Zeit  $t_m$  seit der letzten Eingabe verstrichen und das letzte Dialogziel konnte noch nicht ausgeführt werden, ist die Wahrscheinlichkeit hoch, daß der Benutzer kein Interesse mehr daran hat. Um sicher zu gehen, daß der Benutzer eine neue Intention verfolgt und um ihn an das ausstehende Dialogziel zu erinnern, startet der Dialogmanager einen Klärungsdialog. Dieser ist analog zum Dialog in Abbildung 4.9 und mit entsprechender Fragestellung an den Benutzer aufgebaut.

#### 4.4.4 Alternative

Falls die beste Hypothese in einer n-besten Liste nicht mit den Informationen im aktuellen Diskurs unifiziert werden kann, dafür jedoch eine andere, kann durch eine Nachfrage evtl. geklärt werden, ob der Benutzer falsch verstanden wurde und was er tatsächlich meinte.

Vorraussetzung dafür ist, daß für jede Hypothese  $h_i$  der potentielle Zustand  $d_{k,i}$  berechnet wird. Es ergibt sich die Menge  $D_k = \{d_{k,1}, d_{k,2}, d_{k,3}, \dots, d_{k,n}\}$ . Die  $h_i$  sind durch die Bewertung des Spracherkenners geordnet. Die beste Hypothese ist  $h_1$ . Die  $d_{k,i}$  sind ebenfalls geordnet und es gilt: für alle  $i \in \{1, 2, 3, \dots, n\}$  ist  $d_{k,i}$  abgeleitet von  $h_i$ .

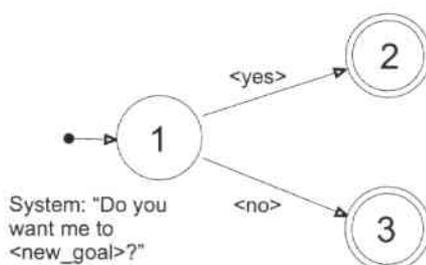


Abbildung 4.7: Subdialog

$$d_{k-1} = \langle \{selected|determined\}, -, G_{s_{k-1}}, \emptyset \rangle$$

$$D_k = \{d_{k,1}, d_{k,2}, d_{k,3}, \dots, d_{k,n}\}$$

$$\text{mit } \exists d \in D_k : d \neq d_{k,1} \wedge d = \langle \{selected|determined|finalized\}, -, G_s, G_f \rangle$$

$$\wedge \exists g \in G_{s_{k-1}} : g \in G_s \vee g \in G_f$$

Im zugehörigen Klärungsdialog lässt sich der Dialogmanager eine der Hypothesen vom Benutzer bestätigen. Der Klärungsdialog ist in Abbildung 4.8 gezeigt.

## 4.5 Triggerlose Klärungsdialoge

Die bisher besprochenen Klärungsdialoge werden alle direkt durch eine Äußerung des Benutzers – und in Abhängigkeit des Zustandes, in dem sich der Dialogmanager befindet – ausgelöst.

Diese Arbeit sieht auch die Möglichkeit vor, Klärungsdialoge direkt in Abhängigkeit des Systemzustandes zu jeder, für die Dialogstrategie notwendigen Zeit zu starten.

Für die korrekte Bearbeitung von Subdialogen wurde von dieser Möglichkeit Gebrauch gemacht.

### 4.5.1 Fortführung

Um nach einem Subdialog wieder zum vorherigen Dialogziel zurückkehren zu können, wird der Inhalt des Stacks überprüft. Befindet sich nach Finalisierung eines Dialogzieles ein Diskurs – der das alte Dialogziel und alle bisher

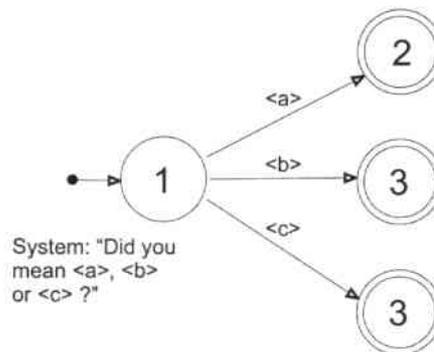


Abbildung 4.8: Alternative

dazu gesammelten Daten beinhaltet – auf dem Stack, wird er zum aktuellen Diskurs. Der Dialog kann an der Stelle fortgesetzt werden, an dem er unterbrochen wurde.

Um die Möglichkeit auszuschließen, daß der Benutzer eben dieses alte Dialogziel nicht einfach abbrechen wollte, wird ein entsprechender Klärungsdialog (Abbildung 4.9) gestartet, ohne auf eine Eingabe des Benutzers zu warten (Seine letzte Eingabe hatte dazu geführt, daß das Dialogziel des Subdialoges finalisiert werden konnte).

$$d_k = \langle \text{finalized}, -, -, - \rangle \wedge \neg \text{Stack.isEmpty}$$

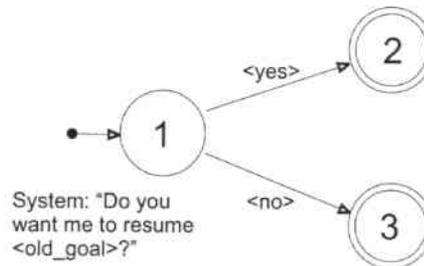


Abbildung 4.9: Fortführung

# Kapitel 5

## Experimente

### 5.1 Beschreibung

Ziel der Experimente war es zu zeigen, daß der Einsatz von Klärungsdialogen zu einer Optimierung des Gesamtdialoges führt und die Kriterien aus Abschnitt 2.4 besser erfüllt werden als ohne. Dazu wurde das im Rahmen dieser Arbeit entstandene System mit einem Referenzsystem ohne Klärungsdialoge in einer Benutzerevaluation getestet.

### 5.2 Systeme im Vergleich

Beide Systeme wurden auf einem Laptop mit 1 GHz *Pentium III (Coppermine)* Prozessor, 512 MB Speicher und *Linux* betrieben.

Als Spracherkenner kam *Janus 5* Patchlevel 12 und als Sprachausgabe *Festival* zum Einsatz.

Als Grundlage für beide Systeme diente *Tapas* Version 1.1 und sie wurden, wie *Tapas* selbst, in Java 1.4 implementiert. Als Laufzeitumgebung kam die entsprechende JVM<sup>1</sup> von Sun zum Einsatz.

Das Referenzsystem implementiert die Holdstrategie, die in Abschnitt 3.1 beschrieben wurde. Das in dieser Arbeit entstandene System implementiert eine Dialogstrategie mit Klärungsdialogen, wie sie in Kapitel 4 beschrieben wurde.

---

<sup>1</sup>Java Virtual Machine

### 5.3 Testszzenarien

Die Testszzenarien waren relativ stark reglementiert. Die Probanden hatten genaue Vorgaben, welche Aufgaben vom System ausgeführt werden sollten. Dies hatte den Zweck, das jeweils gewünschte Verhalten der Systeme zu provozieren. Damit wurden die Ergebnisse erst vergleichbar.

1. Zwei Dialogziele sollten vom System sequentiell ausgeführt werden. Die Vorgabe war: Licht einschalten, Tisch decken.
2. Ein Dialogziel sollte durch ein weiteres unterbrochen werden. Nach Finalisierung dieses Dialogzieles sollte das erste fortgeführt werden und ebenfalls finalisiert werden. D.h. es wurde ein Subdialog durchgeführt. Die Vorgabe war: Tisch decken, Kaffee kochen.
3. Ein verfolgtes Dialogziel sollte durch ein anderes abgebrochen werden. Die Vorgabe war: Kaffee kochen, Cola bringen lassen.

### 5.4 Durchführung

Insgesamt wurden die Systeme mit sieben Benutzern getestet. Jeder Benutzer hatte die in Abschnitt 5.3 beschriebenen Aufgaben mit beiden Systemen zu lösen. Welches System jeweils getestet wurde, war dem Benutzer nicht bekannt.

Um die Ergebnisse der Experimente vergleichen zu können wurden einerseits die vom System ausgegebenen Logdateien ausgewertet, andererseits eine Bewertung durch die Benutzer nach erfolgreichem Lösen jeder Aufgabe vorgenommen.

Somit zerfällt die Evaluation in einen objektiven und einen subjektiven Teil. Für die Bewertung wurden folgende Fragen gestellt:

1. Wie nahe kam der Dialog einem natürlichen Dialog?
2. War die Länge des Dialoges der Aufgabe entsprechend angemessen?
3. Hatten Sie das Gefühl falsch verstanden worden zu sein?
4. Wenn die letzte Frage mit "ja" beantwortet wurde:  
War die Behandlung von Erkennungsfehlern akzeptabel?

## 5.5 Maße

Die folgenden Basisgrößen dienen der Berechnung der Maße. Sie beziehen sich jeweils auf einen kompletten Dialog und wurden anhand der Ausgaben in den Logdateien der Systeme bestimmt.

$\#g_s$  **Anzahl der begonnenen Dialogziele.** Ein Dialogziel gilt als begonnen, sobald INTENTION den Wert *selected* angenommen hat.

$\#g_f$  **Anzahl der finalisierten Dialogziele.** Ein Dialogziel  $g$  gilt als finalisiert, wenn  $g \in \text{FINALIZEDGOALS}$  und damit INTENTION den Wert *finalized* angenommen hat.

$\overline{\#g_s}$  **Anzahl der begonnenen logischen Dialogziele.** Ein Dialogziel gilt als logisch begonnen, sobald der Benutzer in seiner Eingabe dieses als seine Intention erkennen läßt.

$\overline{\#g_f}$  **Anzahl der finalisierten logischen Dialogziele.** Analog zu oben gilt ein Dialogziel als logisch finalisiert, wenn es, vom Benutzer beabsichtigt, ausgeführt werden kann.

### 5.5.1 Erfüllungsrate

Die Erfüllungsrate  $cr$  berechnet sich durch

$$cr = \frac{\#g_f}{\#g_s} \quad (5.1)$$

und spiegelt die Robustheit des Dialogsystems gegenüber Fehlern in der Sprachverarbeitung wider. Je höher die Erfüllungsrate ausfällt, desto robuster war ein Dialog. Die Erfüllungsrate ist also ein Maß, das die Optimierung eines Dialoges nach Kriterium 1 aus Abschnitt 2.4 bewertet.

### 5.5.2 Logische Erfüllungsrate

Die logische Erfüllungsrate  $\overline{cr}$  berechnet sich durch

$$\overline{cr} = \frac{\overline{\#g_f}}{\overline{\#g_s}} \quad (5.2)$$

und spiegelt den Anteil der vom System ausgeführten Dialogziele wider, die in der tatsächlichen Intention des Benutzers lagen. Je höher die logische Erfüllungsrate ist, desto präziser und flüssiger konnte das System die Aufträge des Benutzers erfüllen.

$\overline{cr}$  bewertet die Optimierung eines Dialoges nach Kriterium 1 aus Abschnitt 2.4.

### 5.5.3 Anzahl der Userturns

Die Anzahl der Userturns  $turns(g)$  ist die Anzahl der Turns zu einem Dialogziel  $g$ , die seit dem logischen Beginn von  $g$  bis zu seiner logischen Finalisierung benötigt wurden.  $g$  gilt ab demjenigen Benutzerturn als logisch begonnen, ab dem  $g$  in der Intention des Benutzers erkennbar ist. Logisch finalisiert ist  $g$ , wenn es das System im Sinne des Benutzers ausgeführt hat.

$turns(g)$  bewertet die Optimierung eines Dialoges nach Kriterium 2 aus Abschnitt 2.4.

### 5.5.4 Natürlichkeit des Dialoges

Die Natürlichkeit  $nat$  wird durch den Benutzer festgelegt.

Die Frage nach der Natürlichkeit des Dialoges sollte mit einer Bewertung aus  $\{-2, -1, 0, 1, 2\}$  vorgenommen werden. Wobei eine 2 als "sehr natürlich" anzusehen ist.

$nat$  bewertet die Optimierung eines Dialoges nach Kriterium 2 aus Abschnitt 2.4.

### 5.5.5 Länge des Dialoges

Ebenso hatte der Benutzer darüber zu entscheiden, ob die Länge  $l$  des Dialoges angemessen war. Diese Frage war mit "angemessen" oder "zu lang" zu beantworten.

Es gab keinerlei Vorgaben an die Kriterien oder die Bewertungsgrundlage. Der Benutzer sollte quasi nach Gefühl entscheiden.

Erwartet wurde, daß ein Dialog als zu lang empfunden wurde, bei dem der Benutzer Eingaben wiederholen mußte, oder viele, evtl. als unnötig empfundene Klärungsfragen beantworten mußte.

$l(g)$  bewertet die Optimierung eines Dialoges nach Kriterium 2 aus Abschnitt 2.4.

### 5.5.6 Behandlung von Erkennungsfehlern

Schließlich sollte eine Bewertung für die Behandlung von Erkennungsfehlern  $bef$  aus  $\{-2, -1, 0, 1, 2\}$  durch den Benutzer vorgenommen werden, falls er das Gefühl hatte, daß das System ihn nicht verstanden hatte.

$bef$  bewertet die Optimierung eines Dialoges nach Kriterium 1 aus Abschnitt 2.4.

# Kapitel 6

## Ergebnisse und Diskussion

### 6.1 Ergebnisse der Experimente

Um die Verbesserung von Dialogen mit Klärungsfragen zu zeigen, wurden die in Kapitel 5 beschriebenen Experimente durchgeführt. Die Ergebnisse werden in diesem Abschnitt präsentiert. Einen Überblick geben die Tabellen 6.1 bis 6.5. Sie zeigen die erzielten Werte für die zwei getesteten Systeme, aufgeschlüsselt nach den drei Szenarien (Abschnitt 5.3) und dem Gesamtergebnis.

| System          | Sequentiell | Subdialog | Unterbrechung | Gesamt |
|-----------------|-------------|-----------|---------------|--------|
| Holdstrategie   | 79%         | 58%       | 56%           | 65%    |
| Klärungsdialoge | 91%         | 90%       | 60%           | 81%    |

Tabelle 6.1: Durchschnittliche Erfüllungsrate von Dialogzielen

| System          | Sequentiell | Subdialog | Unterbrechung | Gesamt |
|-----------------|-------------|-----------|---------------|--------|
| Holdstrategie   | 86%         | 95%       | 50%           | 83%    |
| Klärungsdialoge | 100%        | 95%       | 46%           | 84%    |

Tabelle 6.2: Durchschnittliche logische Erfüllungsrate von Dialogzielen

| System          | Sequentiell | Subdialog | Unterbrechung | Gesamt |
|-----------------|-------------|-----------|---------------|--------|
| Holdstrategie   | 4           | 6         | 1             | 5      |
| Klärungsdialoge | 3           | 4         | 1             | 3      |

Tabelle 6.3: Durchschnittliche Dialoglänge in Benutzerturns pro logisch finalisiertem Dialogziel

| System          | Sequentiell | Subdialog | Unterbrechung | Gesamt |
|-----------------|-------------|-----------|---------------|--------|
| Holdstrategie   | -0.29       | -0.71     | 0.6           | -0.29  |
| Klärungsdialoge | 0           | 0.29      | 0.2           | 0.21   |

Tabelle 6.4: Durchschnittliche Bewertung der Natürlichkeit eines Dialoges

| System          | Sequentiell | Subdialog | Unterbrechung | Gesamt |
|-----------------|-------------|-----------|---------------|--------|
| Holdstrategie   | 1           | -0.14     | 1             | 0.79   |
| Klärungsdialoge | 0.71        | 0.71      | 0.2           | 0.79   |

Tabelle 6.5: Durchschnittliche Bewertung der Länge eines Dialoges

Insgesamt haben die Experimente gezeigt, daß ein Dialogsystem mit expliziten Klärungsfragen natürlichere Dialoge erzeugen kann, als eines ohne. Es wurde eine Optimierung des Dialogablaufes nach den Kriterien aus Abschnitt 2.4 erzielt.

Dies zeigt eine deutliche Steigerung der Erfüllungsrate zum Baselinesystem, bei gleichzeitiger Verkürzung der Dialoge (Tabellen 6.1 und 6.3).

Diese rein objektiv, aus Logdateien gewonnenen Daten werden durch die Beurteilung der Natürlichkeit der Benutzer untermauert: die Benutzer empfanden Dialoge mit Klärungsstrategie natürlicher, wie Tabelle 6.4 zeigt.

Die Länge der Dialoge beider Systeme wurden im Allgemeinen als angemessen empfunden (Tabelle 6.5).

|                 |     |
|-----------------|-----|
| Anzahl Benutzer | 7   |
| Dialoge gesamt  | 38  |
| Turns gesamt    | 353 |

Tabelle 6.6: Rahmendaten der Experimente

Die Durchführung der Evaluation war nicht unproblematisch, da die Verwendung von automatisch generierten Eingabedaten bei Dialogsystemen nach aktuellem Stand der Technik nicht möglich ist. Schon die Berücksichtigung subjektiver Maße, wie die Natürlichkeit eines Dialoges, erfordert reale Benutzer. Jeder Proband benötigte ungefähr eine dreiviertel Stunde für die drei Testszenerien. Dies ermöglichte nur einen kleinen Datenkorpus.

Tabelle 6.6 zeigt eine Übersicht über die statistischen Rahmenbedingungen. Die Evaluation konnte mit sieben Benutzern durchgeführt werden. Dabei wurden insgesamt 38 Dialoge mit 353 Turns geführt.

## 6.2 Analyse der Ergebnisse

### 6.2.1 Logische Erfüllungsrate und Anzahl der Userturns

Die logische Erfüllungsrate in Tabelle 6.2 erscheint für beide Strategien ungewöhnlich hoch. Außerdem unterscheiden sich die beiden Strategien, bis auf Szenario Nr. 1, in den erzielten Werten kaum.

Die hohen Werte sind durch die Bemühungen der Benutzer zu erklären, die gestellten Aufgaben unter allen Umständen zu lösen. Daher ist die logische Erfüllungsrate zusammen mit der Anzahl der Turns pro logisch finalisiertem Dialogziel aussagekräftiger. Die hohe Konvergenz ist durch die guten Ergebnisse zu erklären, die die Holdstrategie schon liefert.

Dennoch konnte die Strategie mit Klärungsdialogen offensichtlich im Szenario Nr. 1 ein besseres Ergebnis liefern als die Holdstrategie. Dies läßt den Schluß zu, daß die Holdstrategie zumindest eher Situationen zuließ, in denen Verständnisprobleme auftraten, die nicht aufgelöst werden konnten.

### 6.2.2 Erfüllungsrate und abgebrochene Dialogziele

Bei Szenario Nr. 3 wäre eigentlich eine Erfüllungsrate von max. 50% zu erwarten. Denn es sollte ein Dialogziel durch den Beginn eines anderen abgebrochen und später nicht weiter verfolgt werden. Dennoch treten Werte über 50% auf, weil die Benutzer oft das System begrüßt haben ("Hello Robbi"). Dies führt sofort zu einem finalisierten Dialogziel.

### 6.2.3 Länge der Dialoge

Die Länge der Dialoge konnte insgesamt verkürzt werden (Tabelle 6.3). Beim dritten Testzenario ergab sich keine Verbesserung, die beiden Strategien liefern das gleiche Ergebnis. Das ist auf die Tatsache zurückzuführen, daß bei der Holdstrategie die letzte Frage vor der Eingabe, die den Kontextwechsel provoziert, wiederholt wird und die Strategie mit Klärungsdialogen explizit nachfragt, ob der Wechsel beabsichtigt war. Im ersten Fall findet eine implizite Klärung statt, im zweiten Fall eine explizite, die jeweils zu einer zusätzlichen Frage des Systems führen.

Die Bewertung der angemessenen Länge eines Dialoges durch die Benutzer hat im Ganzen keinen Unterschied gebracht. Allein im Szenario mit dem provozierten Subdialog konnte das implementierte System mit Klärungsdialogen eine deutliche Verbesserung erzielen. Dieses Ergebnis war aufgrund der Wahl des Szenarios zu erwarten. Tabelle 6.5 zeigt die Ergebnisse.

### 6.2.4 Natürlichkeit der Dialoge

Im Schnitt haben die Benutzer Dialoge mit Klärungsfragen als natürlicher angesehen. Im Falle des Zielabbruchs, im dritten Szenario, wurden Dialoge mit Holdstrategie als natürlicher angesehen. Dies lag an der Tatsache, daß am Ende des Dialoges nachgefragt wurde, ob das abgebrochene Dialogziel wieder aufgenommen werden soll. Das System kann hier nicht unterscheiden, ob der Kontextwechsel durch einen Subdialog oder einen Abbruch hervorgerufen wurde. Es war Klärungsbedarf vorhanden und daher diese Frage notwendig. Die Ergebnisse sind in Tabelle 6.4 zusammengefaßt.

## 6.3 Subdialoge und Erkennungsfehler

In einem Dialogzustand, in dem die letzte Eingabe des Benutzers nicht zum aktuellen Kontext passt, gibt es zwei extreme Varianten, wie die Dialogstrategie reagieren kann:

**Erkennungsfehler:** Die Strategie beharrt auf einem Erkennungsfehler der Sprachverarbeitung. Sie wird den Benutzer "zwingen", das aktuelle Dialogziel weiter zu verfolgen.

**Subdialog:** Die Strategie nimmt an, daß der Benutzer ein anderes Dialogziel verfolgen will und beginnt einen Subdialog. Der Benutzer kann seine Intention nicht weiter verfolgen.

Ziel jeder Dialogstrategie muß es sein, einen geeigneten Mittelweg zu gehen. Die in dieser Arbeit verglichenen Strategien verfolgen jeweils mit unterschiedlichen Ansätzen diesen Mittelweg.

Die Holdstrategie verwendet implizite Klärungsfragen: Passt die letzte Eingabe des Benutzers nicht zum Kontext, wird die letzte Frage des Systems wiederholt.

Die Strategie mit Anomalieanalyse<sup>1</sup> verwendet explizite Klärungsfragen, indem sie den Benutzer konkret bestätigen läßt, daß er einen Subdialog starten möchte.

Beide Strategien verlängern den Dialog bei Klärungsbedarf um einen Benutzerturn, indem sie die letzte Frage wiederholen, bzw. eine Klärungsfrage stellen. Dennoch hat die Evaluation gezeigt, daß eine Strategie mit expliziten Klärungsfragen den Dialog verkürzt (Tabelle 6.3).

Bei Verwendung expliziter Klärungsfragen wird der Benutzer konkret auf das Problem hingewiesen. Dies könnte sich positiv auf die Robustheit des

---

<sup>1</sup>Natürlich wäre es denkbar, nach Entdecken einer Anomalie ebenso vorzugehen wie die Holdstrategie.

Dialoges ausgewirkt haben. In jedem Fall haben sie die Bewertung der Natürlichkeit positiv beeinflusst. Insbesondere haben sie dies beim Szenario Nr. 2, in dem der Benutzer einen Subdialog beginnen sollte.

Der in Abbildung 6.1 dargestellte Dialog wurde mit dem Baselinesystem während eines Benutzertests geführt. Er beinhaltet beide Fälle: In Turn 3 soll ein Subdialog gestartet werden, in Turn 5 liegt ein Erkennungsfehler vor. Der Verlauf des Dialoges läßt den Benutzer darauf schließen, daß er das System unterstützen muß (Turn 6).

Eine explizite Klärungsfrage hätte in diesem Fall sicher ein Fehlverhalten des Systems verhindert.

|   |             |   |  |
|---|-------------|---|--|
| 1 | Benutzer:   | Set the table   |  |
|   | Verstanden: | Set the table   |  |
|   |             | $d_1 = \langle \text{determined}, \text{act\_setTable}, \{\text{SetTable}\}, \{\}\rangle$ |  |
|   | System:     | For how many Persons do you want me to set the table?                                     |  |
| 2 | Benutzer:   | For three persons please  |  |
|   | Verstanden: | For three persons   |  |
|   |             | $d_1 = \langle \text{determined}, \text{act\_setTable}, \{\text{SetTable}\}, \{\}\rangle$ |  |
|   | System:     | Which kind of glasses ... ?   |  |
| 3 | Benutzer:   | Make me a coffee  |  |
|   | Verstanden: | Make me a coffee  |  |
|   |             | $d_1 = \langle \text{determined}, \text{act\_setTable}, \{\text{SetTable}\}, \{\}\rangle$ |  |
|   | System:     | Which kind of glasses ... ?   |  |
| 4 | Benutzer:   | Make me a coffee  |  |
|   | Verstanden: | Make me a coffee  |  |
|   |             | $d_1 = \langle \text{determined}, \text{act\_make}, \{\text{MakeDrink}\}, \{\}\rangle$    |  |
|   | System:     | Going to make you a cup of coffee. Do you want milk?                                      |  |
| 5 | Benutzer:   | No milk please  |  |
|   | Verstanden: | Now move please   |  |
|   |             | $d_1 = \langle \text{determined}, \text{act\_go}, \{\text{GoSomewhere}\}, \{\}\rangle$    |  |
|   | System:     | Where should I go?  |  |
| 6 | Benutzer:   | Go to the kitchen and make me a cup of coffee   |  |
|   | Verstanden: | Go to the table   |  |
|   |             | $d_1 = \langle \text{finalized}, \text{act\_go}, \{\text{GoSomewhere}\}, \{\}\rangle$     |  |
|   | System:     | Going to the table.   |  |
| 7 | Benutzer:   | Ähm make me a coffee  |  |
|   | Verstanden: | Make me a coffee  |  |
|   |             | $d_1 = \langle \text{determined}, \text{act\_make}, \{\text{MakeDrink}\}, \{\}\rangle$    |  |
|   | System:     | Going to make you a cup of coffee. Do you want milk?                                      |  |

Abbildung 6.1: Dialog mit Baselinesystem (Holdstrategie): ein unbemerkter Kontextwechsel.

# Kapitel 7

## Zusammenfassung

In dieser Arbeit wurde eine Methode entwickelt, Situationen im Dialogablauf zu erkennen, in denen Klärungsbedarf besteht. Eine Anomalieanalyse prüft die einzelnen Zustände des Dialogablaufes und entscheidet mit Hilfe von Triggern, ob ein Klärungsdialog notwendig ist. In Form von endlichen Automaten wurden die Klärungsdialoge implementiert.

Die Verwendung von Triggern und endlichen Automaten ermöglicht eine Art "separation of concerns", wie sie auch in der aspektorientierten Programmierung vorgesehen ist. Klärungsdialoge können mit den Methoden dieser Arbeit orthogonal mit verschiedenen Dialogsteuerstrategien verwendet werden.

Eine Benutzerevaluation hat gezeigt, daß der Einsatz von Klärungsdialogen eine Optimierung von Dialogen nach den Kriterien Korrektheit, Robustheit, Natürlichkeit und natürlicher Fluß erzielen kann.

### 7.1 Ausblick

Um eine höhere statistische Relevanz zu erreichen, müssen weitere Benutzer-tests durchgeführt werden.

Dabei sollten auch in großer Zahl "freie" Dialoge geführt werden, bei denen die Probanden keine Vorgaben zu beachten haben.

Der vorgestellte Katalog mit Triggern ist keineswegs vollständig. Im Gegenteil ist zu erwarten, daß mit zunehmender Erfahrung im Umgang mit der Anomalieanalyse, in Zukunft weitere Trigger gefunden werden. Insbesondere die Einführung einer neuen Zustandsvariable kann weitere Anomalien und damit neue Trigger nach sich ziehen.

Der vorhandene Katalog ist also um weitere Trigger zu erweitern, die implementiert und getestet werden müssen.

Die Klärungsdialoge und Trigger sind derzeit noch in Klassen gekapselt, also hart kodiert. Um Systeme schneller entwickeln zu können, wäre ein grafisches Werkzeug zur Modellierung der Dialoge (Endliche Automaten) hilfreich. Die Tatsache, daß in Zukunft weitere Trigger hinzukommen werden, legt die Entwicklung eines Editors mit dem die Trigger leichter bearbeitet werden können nahe.

Klärungsdialoge werden in dieser Arbeit mit Endlichen Automaten realisiert. Das zieht automatisch eine Einschränkung der Äußerungsmöglichkeiten für den Benutzer nach sich. Integriert man die Behandlung der Äußerungen in aktiven Klärungsdialogen in die Gesamtstrategie, erreicht man mehr Flexibilität.

Eine zusätzliche Variable `CLARIFICATIONON` im abstrakten Dialogzustand mag dies ermöglichen.

# Kapitel 8

## Danksagung

Bedanken möchte ich mich bei meinem Betreuer Hartwig Holzapfel, der mich bei dieser Arbeit hervorragend unterstützt hat. Insbesondere hat er Änderungswünsche meinerseits an der Schnittstelle zu *Tapas* immer prompt umgesetzt.

Für Korrekturen in der Ausarbeitung geht mein Dank an Patrycja Holzapfel. Weiterhin möchte ich mich bei den Probanden bedanken, die mir mit ihrer Geduld die nötigen Tests und die Benutzerevaluation ermöglichten. Dies waren im einzelnen: Keni Bernardin, Annika Frommer, Petra Gieselmann, Hartwig Holzapfel, Patrycja Holzapfel, Tobias Kluge, Annika Krum, Gena Batalski, Ulrich Klee, Tobias Gehrig, Borislava Mimer, Susanne Peterson und Michael Voit

# Literaturverzeichnis

- [Bal98] Helmut Balzert. *Lehrbuch der Softwaretechnik - Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer verlag, 1998.
- [Car92] Bob Carpenter. *The Logic of Typed Feature Structures*. Cambridge University Press, 1992.
- [Den02] Matthias Denecke. *Generische Interaktionsmuster für Aufgabenorientierte Dialogsysteme*. Dissertation, University of Karlsruhe, 2002.
- [Gab03] Malte Gabsdil. Clarification in spoken dialogue systems, 2003.
- [GFH<sup>+</sup>03] P. Giesermann, C. Fügen, H. Holzapfel, T. Schaaf, and A. Waibel. Towards multimodal communication with a household robot, 2003.
- [GS86] Barbara J. Grosz and Candace L. Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3), 1986.
- [HG04] H. Holzapfel and P. Giesermann. A way out of dead end situations in dialogue systems for human-robot interaction. In *Humanoids 2004*, Los Angeles, 2004.
- [McT02] Micheal F. McTear. Spoken dialogue technology: Enabling the conversational user interface, 2002.
- [Sch04] David Schlangen. Causes and strategies for requesting clarification in dialogue, 2004.
- [WSS04] Alex Waibel, Hartwig Steusloff, and Rainer Stiefelhagen. CHIL - Computers in the Human Interaction Loop. In *NIST ICASSP Meeting Recognition Workshop*, Montreal, Canada, May 2004.