Institut für theoretische Informatik
Fakultät für Informatik
Universität Karlsruhe

Interactive Systems Labs

Natürlichsprachliches Reservierungssystem für einen multimodalen Raum

Studienarbeit von
Philipp Sorg

betreut von
M.A. Petra Gieselmann
Prof. Alex Waibel

November 2006

Zusammenfassung

Diese Studienarbeit befasst sich mit einem natürlichsprachlichen Dialogsystem. Die Aufgabe dieses Systems ist die Verwaltung der Reservierungen eines multimodalen Raumes.

Auf das Reservierungssystem kann direkt im Raum über eine multimodale Benutzerschnittstelle zugegriffen werden, was eine einfache und intuitive Benutzung ermöglicht. Benutzereingaben erfolgen dabei über gesprochene natürliche Sprache und eine grafische Oberfläche. Zusätzlich ist auch ein Zugriff über das Internet möglich. Dabei erfolgt die natürliche Spracheingabe textbasiert.

Als Dialogmanager wird *Tapas* verwendet. Hierfür wird eine Wissensbasis, die auf diese Anwendung bezogen ist, definiert. Die grafische Oberfläche ist als Webseite realisiert, die über eine Webapplikation mit dem Dialogmanager verbunden ist. Diese Webapplikation ist auch für die Reservierungsverwaltung zuständig und verknüpft die verschiedenen Eingabemodalitäten.

Das System wird mit Hilfe einer Benutzerevaluation getestet. Das Ergebnis zeigt, dass die Reservierungsverwaltung für eine kleine Anzahl von Benutzern einsatzfähig ist, für viele Benutzer aber erweitert werden muss.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Die Motivation dieser Studienarbeit entstand aus der Idee, für einen multimodalen Raum ein wiederum multimodales Reservierungssystem zu erstellen. Konkret handelt es sich dabei um den multimodalen Raum des ISL an der Fakultät für Informatik der Universität Karlsruhe. Für dies Aufgabe wird ein System benötigt, das einfach und intuitiv zu bedienen ist und an verschiedenen Standorten zur Verfügung steht.

Die intuitive Bedingung wird durch die Multimodalität des Reservierungssystems erreicht, die gesprochene natürliche Spracheingabe mit einer einfachen grafischen Oberfläche kombiniert. Die Einlernphase für den Benutzer ist dadurch sehr kurz, da er ausformulierte Sätze verwenden kann, die er auch im Gespräch zur Reservierungsbuchung mit einer anderen Person benutzen würde. Alle Funktionen der grafischen Oberfläche werden direkt dargestellt und die Bedienung ist damit auch leicht erlernbar. Durch den Verzicht von Eingabegeräten wie Tastatur und Maus ist ein flexibler Standort möglich. Denkbar ist zum Beispiel direkt die Tür des Raumes. Mit Hilfe von Touchscreen, Mikrofon und Lautsprecher können dort dann direkt Reservierungen vorgenommen werden.

Die Bedienung von verschiedenen Standorten aus wird dadurch erreicht, dass die grafische Oberfläche durch eine Weboberfläche realisiert wird. Damit kann das System auch über das Internet bedient werden und steht somit jederzeit den Benutzern zur Verfügung.

1.2 Problematik und Vorgehensweise

Bei der Entwicklung des Systems müssen folgende Aspekte betrachtet werden:

Der Zugriff soll von verschiedenen Standorten aus möglich sein. Dies fordert eine vernetzte Struktur des Systems. Dabei muss entschieden werden, ob ein zentrales oder ein dezentrales System entwickelt wird. Bei einem zentralen System wird die Hauptanwendung auf einem zentralen Rechner ausgeführt, auf dessen Oberfläche von verschiedenen Orten aus zugegriffen werden kann. Bei einem dezentralen System wird die Anwendung auf verschiedenen Rechnern ausgeführt und nur die Reservierungsdaten werden synchronisiert. Bei beiden Ansätzen muss das Problem des gleichzeitigen Zugriffs mehrerer Benutzer gelöst werden.

Da es sich um ein multimodales System handeln soll, muss das Zusammenführen der verschiedenen Eingabe- und Ausgabemodalitäten gelöst werden. Der Benutzer soll die Möglichkeit haben, natürlichsprachliche Eingaben und Eingaben über die grafische Benutzeroberfläche beliebig zu kombinieren und zwischen diesen Modalitäten zu wechseln. Dazu wird eine einheitliche interne Darstellung der verschiedenen Eingaben benötigt. Bei den Ausgaben des Systems muss beachtet werden, dass die natürlichsprachliche Ausgabe immer mit der grafischen Ausgabe übereinstimmt.

Schließlich soll das System benutzerfreundlich und intuitiv bedienbar sein. Dazu muss eine grafische Oberfläche entwickelt werden, die die Reservierungsinformationen klar und verständlich darstellt und dem Benutzer genügend Hilfe bietet. Diese Oberfläche soll auch beim Zugriff von verschiedenen Standorten aus immer gleich aussehen. Dies erleichtert dem Benutzer das Einlernen in die Bedienung des Systems.

Aus dieser Problematik ergibt sich folgende Vorgehensweise:

Da es sich um ein Dialogsystem handelt, müssen die Dialogziele identifiziert werden. Dazu müssen die einzelnen Funktionen, die zum Bedienen eines Reservierungssystems benötigt werden, definiert werden. Diese werden dann zu den einzelnen Dialogzielen zusammengefasst.

Für die natürlichsprachliche Eingabe wird zuerst ein Spracherkenner benötigt. Danach muss das System den erkannten Text in eine semantische Darstellung umwandeln, um damit die Eingaben zu verstehen. Dazu muss ein semantisches Sprachmodell der verschiedenen Aufgaben und Objekte in der Domäne der Raumreservierung erstellt werden.

1.3 Überblick

Im Kapitel 2 werden Dialogsysteme vorgestellt, die ein ähnliches Anwendungsgebiet besitzen oder auf ähnlichen Mechanismen basieren.

Im Kapitel 3 wird auf die Grundlagen eines natürlichsprachlichen Dialogsystems eingegangen und der Dialogmanager *Tapas*, auf dem diese Studienarbeit basiert, vorgestellt.

Im Kapitel 4 werden die einzelnen Komponenten des *Reservation Managers* beschrieben. Dazu gehören die Wissensbasis für den Dialogmanager, die grafische Oberfläche und die Verbindung zum Dialogmanager.

Das Kapitel 5 beschreibt die Benutzerevaluation. Ergebnisse und Schlussfolgerungen dieser Evaluation werden präsentiert.

Im Kapitel 6 werden die Stärken und Schwächen des Systems diskutiert. Dabei wird vor allem darauf eingegangen, welche Erweiterungen bei einer hohen Anzahl von Benutzern notwendig werden.

Kapitel 2

Verwandte Arbeiten

Ein grundsätzlicher Ansatz eines multimodalen Dialogsystems ist in [QGS+01] beschrieben. Das System soll gesprochene natürlichsprachliche Befehle in einer Haushaltsumgebung verarbeiten. Als Beispiel wird das An- und Ausschalten von Lichtern beschrieben. Dabei wird eine agentenbasierte Architektur gewählt. So gibt es zum Beispiel einen Dialogmanager, der für das Sprachverstehen und die semantische Interpretation von Benutzereingaben verantwortlich ist, oder einen Aktionenmanager, der die Funktionalität in der tatsächlichen Umgebung des Systems zur Verfügung stellt. Der Vorteil dieser agentenbasierten Architektur ist die Möglichkeit, einzelne Komponenten je nach Anforderung des Systems auszutauschen, und die Abkapselung der verschiedenen Ressourcen, die dadurch einfacher definiert werden können. Das System dieser Studienarbeit verfolgt einen ähnlichen Ansatz. Der Dialogmanager, die Sprachausgabe und die Kalenderfunktionalitäten sind in unterschiedlichen Bereichen untergebracht und tauschen Daten über verschiedene Kommunikationskanäle aus, was in Kapitel 4 beschrieben wird. Auch die Vervollständigung von Dialogzielen wird von beiden Systemen ähnlich gelöst. Für das System dieser Studienarbeit wird dies in in Abschnitt 4.2.2 erläutert.

In [Wah03] wird die multimodale Dialogplattform *SmartKom* vorgestellt. Das System soll ein persönlicher Assistent des Benutzers sein, und diesem bei bestimmten Aufgaben helfen. Als Beispiel wird das Einscannen eines Objektes und das Verschicken des erstellten Bildes per Email beschrieben. Ein weiteres Beispiel ist die Anwendung als Routenplaner, der den Benutzer an eine bestimmte Adresse führt. Die Ein- und Ausgabemodalitäten sind bei diesem System symmetrisch. Die Benutzereingabe erfolgt über natürliche Sprache und einer grafischen Oberfläche, die mit einem Touchscreen bedient werden kann. Außerdem verfolgt das System Gesten und Gesichtsausdrücke des Benutzers per Kamera. Da es sich um ein symmetrisches multimodales

System handelt, befindet sich ein virtueller Gesprächspartner auf der grafischen Oberfläche, durch den auch wiederum Gesten und Gesichtsausdrücke dargestellt werden. Das hier in dieser Studienarbeit vorgestellte System beschränkt sich auf natürlichsprachliche Ein- und Ausgabe sowie auf eine grafische Oberfläche, die per Maus oder Touchscreen bedient werden kann. Wie in *SmartKom* müssen dabei die verschiedenen Eingaben parallel verarbeitet und kombiniert werden. Dieser Vorgang wird im Abschnitt 4.3 beschrieben. Bei *SmartKom* ist die Synchronisation der Ausgaben wichtig. Sprachausgabe, Gesten des virtuellen Gesprächspartners und grafische Darstellungen müssen den gleichen Systemstatus wiedergeben. Dies lässt sich direkt auf diese Studienarbeit übertragen. Bei diesem System müssen Sprachausgabe und grafische Oberfläche aufeinander abgestimmt werden.

In [McT02] werden mehrere natürlichsprachlichen Dialogsysteme beschrieben. Im Unterschied zum System, das in dieser Studienarbeit vorgestellt wird, sind diese Systeme für einen telefonischen Dialog vorgesehen. Die Ein- und Ausgabe erfolgt daher nur über gesprochene natürliche Sprache. Damit sind die Systeme nicht multimodal. Eines der vorgestellten Systeme, das *Nuance Automatic Banking System*, hat eine ähnliche Aufgabenstellung wie das System dieser Studienarbeit. Es ermöglicht dem Benutzer, eine Rechnung zu zahlen. Dabei werden alle relevanten Daten für diese Aufgabe abgefragt, also Empfänger, Betrag und Datum. Danach führt das System die Überweisung aus. Dies ist ähnlich zu einem Reservierungsvorgang. Dafür werden auch zuerst die Daten abgefragt, in dem Fall Name, Zeitpunkt und Dauer, und danach wird die Reservierung eingetragen. Eine Schwäche des *Nuance Automatic Banking System* ist die fehlende Flexibilität auf komplexere Benutzereingaben, was dieser Dialogauszug aus [McT02] zeigt:

⋮

System: *How much would you like to pay?*
User: *One hundred pounds next Monday.*
System: *What date would you like the payment to be made on?*

⋮

Das System erwartet eine bestimmt Eingabe und erkennt deshalb auch nur diese. Die zusätzliche Datumsangabe in der Antwort wird nicht erkannt. Dies führt zu einer Nachfrage des Systems, was für den Benutzer verwirrend sein kann, da die Antwort schon gegeben wurde. Das hier vorgestellte System kann Antworten dieser Art flexibler handhaben. Zusätzliche Angaben werden erkannt und im Dialog verwendet. Dadurch können Nachfragen dieser Art weitgehend vermieden werden.

Kapitel 3

Grundlagen

Der Reservierungsmanager lässt sich in verschiedene Komponenten unterteilen: Weboberfläche, Reservierungsverwaltung und Dialogmanager.

Das Generieren der Weboberfläche und die Reservierungsverwaltung werden von einer Webapplikation mit dahinterliegender Datenbank übernommen. Dies wird in Kapitel 4 ausführlich beschrieben.

Der Kern des Dialogsystems ist der Dialogmanager. Hierfür wird das in [Hol05] beschriebene System *Tapas* verwendet. Dieses ist eine Weiterentwicklung des Systems in [Den02]. *Tapas* stellt ein domänen- und sprachunabhängiges Grundgerüst eines Dialogmanagers zur Verfügung. Für eine bestimmte Anwendung müssen die domänenabhängigen Ressourcen definiert werden.

Abbildung 3.1 zeigt die Verarbeitungsschritte des *Tapas*-Systems bei Benutzereingaben. Die dünnen, durchgezogenen Pfeile stehen für den Informationsfluss zwischen den verschiedenen Komponenten. Natürliche Spracheingabe wird vom Spracherkenner in Text umgewandelt und an die Sprachversteherkomponente weitergeleitet. Eingaben über die grafische Oberfläche werden teilweise auch in Text umgewandelt, diese Vorgehensweise wird in Abschnitt 4.3 beschrieben. Die Sprachversteherkomponente wandelt den Text in eine semantische Darstellung um und leitet die ermittelten Konzepte an die Dialogmanagerkomponente weiter. Die dicken, gestrichelten Pfeile stehen für Zugriffe der Komponenten auf die domänenabhängigen Ressourcen. Diese Ressourcen, die Grammatiken, das Domänenmodell und das Aufgabenmodell, werden im Folgenden näher beschrieben. Dabei wird auch die Funktionsweise der einzelnen *Tapas*-Komponenten, die auf diese Ressourcen zugreifen, näher erläutert.

In den Grammatiken werden die möglichen natürlichsprachlichen Benutzereingaben definiert und diese dann auch mit den Konzepten aus dem Domänenmodell verknüpft. Die Spracherkennung akzeptiert nur Eingabetexte, die durch die Grammatik definiert sind und muss deshalb auf die Gram-

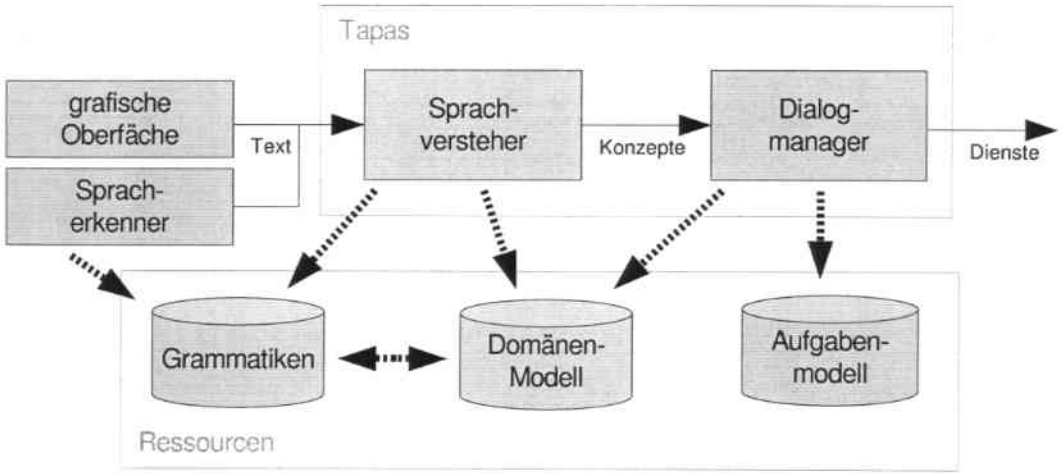

Abbildung 3.1: Darstellung der verschiedenen Ressourcen und Zugriffe auf diese Ressourcen in einem *Tapas*-System.

matikressource zugreifen. Die Sprachversteherkomponente wandelt mit Hilfe der Grammatiken den Eingabetext in eine semantische Repräsentation um, die auf dem Domänenmodell aufbaut. Die spezielle Grammatik für diese Arbeit wird in Kapitel 4.2.3 näher beschrieben.

Im Domänenmodell werden die Konzepte definiert, die für eine bestimmte Dialoganwendung notwendig sind. Durch Methoden wie Vererbung oder Verschachtelung können diese Konzepte in eine semantische Struktur gebracht werden. Die Sprachversteherkomponente benutzt diese Konzepte und deren Struktur für die semantische Darstellung des Eingabetextes. Die Dialogmanagerkomponente braucht das Wissen des Domänenmodells zur Prüfung von Vorbedingungen der Dialogziele und beim Einbinden von Diensten beim Erreichen eines Dialogziels. Eine Beschreibung des Domänenmodells dieser Anwendung befindet sich in Abschnitt 4.2.1.

Die Aufgabenmodellressource enthält Dialogziele und Vorgehensweisen, wie diese zu erreichen sind.

Zu den einzelnen Dialogzielen werden Vorbedingungen und Dienstaufrufe definiert. Die Vorbedingungen bestehen aus Konzeptstrukturen, die im Laufe des Dialogs vorgekommen sein müssen. Wenn dies für ein Dialogziel erfüllt ist, werden die entsprechenden Dienstaufrufe des Dialogziels ausgeführt.

Die Aufgabe der Dialogmanagerkomponente ist es, das vom Benutzer beabsichtigte Dialogziel auszuwählen. Dabei wird nach jeder Eingabe untersucht, welche Dialogziele in Frage kommen. Bei diesem Vorgang werden auch

Informationen von vorhergehenden Eingaben verwendet. So können zum Beispiel erkannte Konzepte aus mehreren aufeinander folgenden Eingaben als Vorbedingung von Dialogzielen verwendet werden. Ziel des Dialogmanagers ist dabei, ein einzelnes Dialogziel auszuwählen und alle anderen ausschließen zu können. Wenn dies der Fall ist, versucht der Dialogmanager, alle Vorbedingungen dieses Dialogziels zu erfüllen. Die Vorgehensweise dazu muss im Aufgabenmodell definiert werden. Eine Möglichkeit ist zum Beispiel die Definition von Klärungsfragen. Diese beschreiben, was der Benutzer gefragt werden soll, wenn bestimmte Konzepte aus der Vorbedingung des ausgewählten Dialogziels fehlen. Dabei kann auch definiert werden, wie die Antworten auf diese Klärungsfragen eventuell umgewandelt werden müssen. Die Dialogziele des Reservierungssystems dieser Studienarbeit werden in Abschnitt 4.2.2 beschrieben.

Kapitel 4

Umsetzung

Das System soll dem Benutzer ermöglichen, eine Reservierung an einem bestimmten Datum zu erstellen, sich Reservierungen anzeigen zu lassen oder Reservierungen zu löschen. Abbildung 4.1 zeigt ein Bild der grafischen Oberfläche. In den folgenden Abschnitten wird zuerst die grafische Oberfläche beschrieben. Danach werden die Definitionen der domänenabhängigen Ressourcen des *Tapas*-Systems erläutert. Schließlich wird der Reservierungsmanager vorgestellt, der die grafische Oberfläche und das *Tapas*-System verbindet und die Reservierungen verwaltet.

4.1 Grafische Oberfläche

4.1.1 Aufbau der grafischen Oberfläche

Die grafische Oberfläche besteht aus drei Bereichen (siehe Abbildung 4.1).

Im Dialogbereich oben links wird der Dialogverlauf mit Benutzereingaben und Antworten des Reservierungssystems angezeigt. Der Benutzer hat dort auch die Möglichkeit, Eingaben per Tastatur an das System zu schicken.

Im Bereich oben rechts wird dem Benutzer direkt Hilfe angeboten. Er kann einzelne Dialogziele direkt durch einen Mausklick auswählen. Dabei wird eine Standardeingabe zum Auslösen dieses Ziels übertragen und auch im Dialogverlauf angezeigt. Der Benutzer sieht diese Eingabe und kann daraus lernen, wie er das System mit natürlichsprachlicher Eingabe steuern kann.

In der Mitte wird ein Kalender dargestellt. Dabei werden Reservierungen innerhalb einer Woche von Montag bis Sonntag angezeigt. Der Benutzer kann per Mausklick zur nächsten oder vorherigen Woche navigieren oder ein bestimmtes Datum und eine bestimmte Uhrzeit auswählen. Diese werden dann direkt in den laufenden Dialog übernommen und an den Dialogmanager wei-

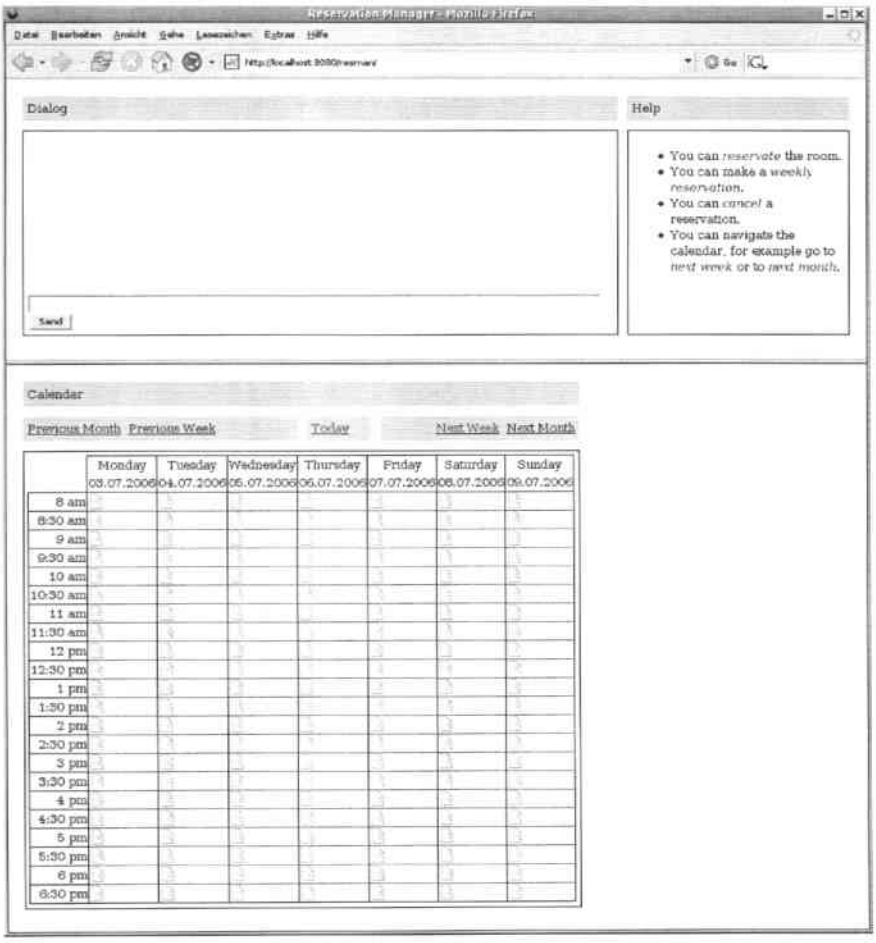

Abbildung 4.1: Die grafische Oberfläche des Reservierungssystems.

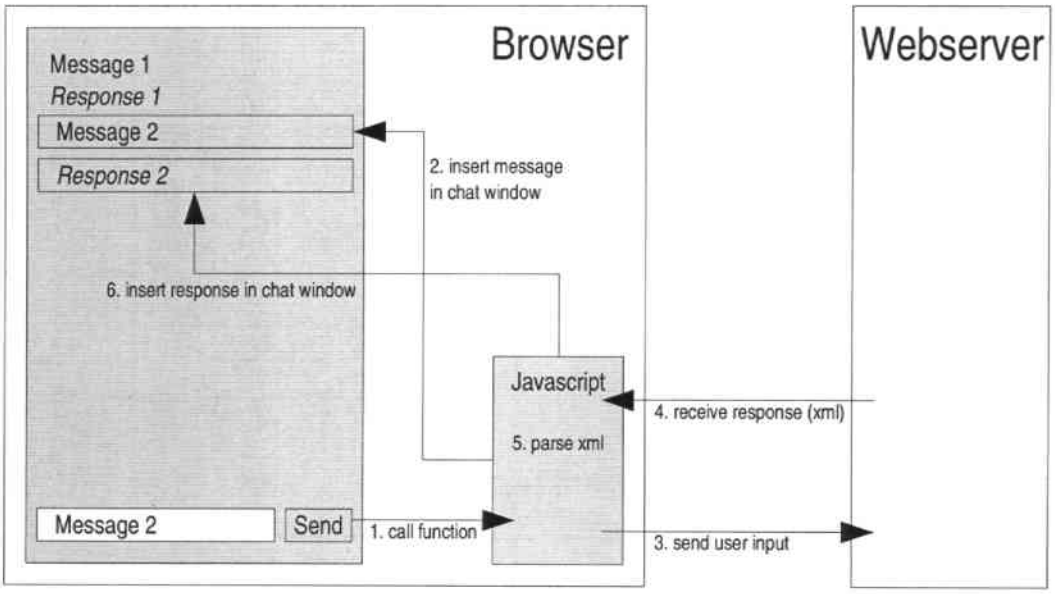

Abbildung 4.2: Verarbeitung der Benutzereingabe und der Antwort des Reservierungssystems im Browser.

tergeschickt.

4.1.2 Technische Umsetzung der grafischen Oberfläche

Die grafische Oberfläche ist als Weboberfläche implementiert. Dies hat den Vorteil, dass sie über das Internet von vielen Standorten aus erreicht werden kann. Außerdem kann sie auf einem Standardbrowser angezeigt werden und ist damit unabhängig vom benutzten Computersystem. Allerdings wird bei der Benutzung im Browser nur die Texteingabe über das Dialogfeld in der grafischen Oberfläche unterstützt. Natürlichsprachliche Eingabe ist nur an einem speziell eingerichteten Computer mit Mikrofon und Spracherkennungssoftware möglich.

Um dem Benutzer ein möglichst schnelles Feedback nach einer Eingabe im Dialogfeld zu geben, wird die Benutzereingabe direkt durch den Browser in den Dialogverlauf eingetragen. Im Hintergrund wird die Antwort des Dialogsystems abgewartet, die dann wiederum auch zum Dialogverlauf hinzugefügt wird. Dieses Prinzip beruht auf der *Ajax*-Programmierung, die in [Gar05] beschrieben wird. Abbildung 4.2 zeigt einen schematischen Ablauf dieses Vorgangs.

4.2 Ressourcen für das *Tapas*-System

In diesem System, wird wie in Kapitel 3 beschrieben, das domänen- und sprachunabhängige Dialogsystem *Tapas* verwendet. Im Folgenden wird die Definition der spezifischen Ressourcen für die Reservierungsverwaltung näher erläutert. Dabei wird auf die verschiedenen Ressourcen Grammatik, Domänenmodell und Aufgabenmodell eingegangen (siehe Abbildung 3.1).

4.2.1 Domänenmodell

Im Domänenmodell werden Konzepte und Beziehungen zwischen Konzepten definiert, die dazu notwendig sind, Benutzereingaben zu verstehen und darauf entsprechend zu reagieren. Die Definition erfolgt in Form einer Ontologie.

Bei dieser Anwendung werden dementsprechend alle Konzepte zur Reservierungsverwaltung eines Raumes benötigt.

Da Reservierungen an bestimmte Zeitpunkte geknüpft sind, gibt es den Sprechakt **act_datetime**. Dieser besitzt die Attribute **DATE** und **TIME**, die ein Datum und eine Uhrzeit darstellen. Alle Konzepte zur Datums- und Zeitdarstellung sowie ihre Hierarchie sind in Abbildung 4.3 dargestellt.

Für das Erstellen und Löschen von Reservierungen sind Konzepte definiert, die auf die Dialogziele, die in Abschnitt 4.2.2 beschrieben werden, abgestimmt sind. Mit Hilfe dieser Konzepte werden alle Informationen zum Erstellen oder zum Löschen von Reservierungen gesammelt. Das Konzept **act_newreservation** erbt das Konzept **act_datetime** und hat zusätzlich die Attribute **USER** und **LENGTH**, die den Benutzer und die Länge der Reservierung beinhalten. Das Konzept **act_cancelreservation** erbt genauso vom Konzept **act_datetime**. Das Attribut **ID** von **act_cancelreservation** wird nur benötigt, wenn eine Reservierung direkt durch ihre Identifikationsnummer gelöscht wird. Dieser Vorgang wird in Abschnitt 4.3.2, in dem die Verknüpfung zwischen grafischer Oberfläche und *Tapas*-Dialogmanager beschrieben wird, erklärt. Eine grafische Darstellung dieser Konzepte befindet sich in Abbildung 4.4.

4.2.2 Dialogziele

Wie im Kapitel 3 beschrieben sind die Dialogziele ein Teil des Aufgabenmodells und damit Teil der Wissensbasis des *Tapas*-Dialogmanagers (siehe Abbildung 3.1). Für die Reservierungsverwaltung werden Dialogziele zum Erstellen von Reservierungen und zum Löschen von Reservierungen benötigt.

Das Erstellen von Reservierungen wird durch zwei Dialogziele definiert. Das Dialogziel **NewReservation** beschreibt das Erstellen einer normalen

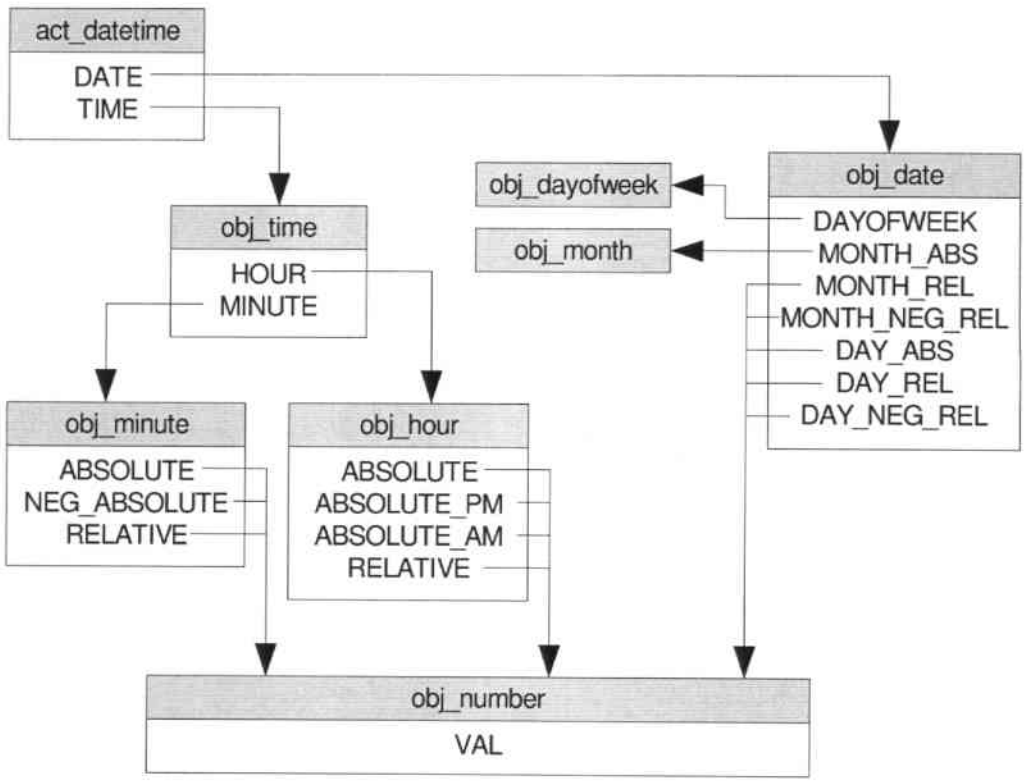

Abbildung 4.3: Konzepte zur Datums- und Zeitdarstellung und ihre Hierarchie.

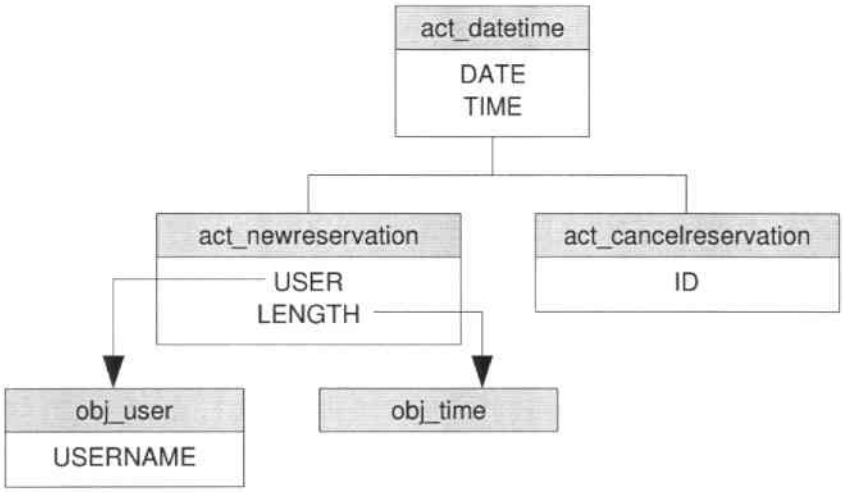

Abbildung 4.4: Konzepte zum Erstellen und Löschen von Reservierungen

Reservierung, das Dialogziel **NewWeeklyReservation** das Erstellen von Reservierungen, die sich wöchentlich wiederholen. Die beiden Dialogziele sind sehr ähnlich, da sie beide die gleichen Informationen im Diskurs benötigen. Sie unterscheiden sich lediglich durch den unterschiedlichen Dienstaufruf bei Erreichen des Dialogziels. Beide Dialogziele führen vor dem jeweiligen Dienstaufruf ein Skript zur Datums- und Zeitumformung aus. Die verschiedenen absoluten oder relativen Datums- oder Zeitangaben werden dabei in absolute Zeitpunkte umgewandelt. Beispiele dazu befinden sich in Abschnitt 4.2.3. Das Ergebnis dieser Umwandlung wird dann im Dienstaufruf verwendet.

Das Löschen von Reservierungen wird ebenfalls durch zwei Dialogziele definiert. Das Dialogziel **CancelReservationByDate** dient zum Löschen einer Reservierung an einem bestimmten Zeitpunkt. Wie bei den oben beschriebenen Dialogzielen **NewReservation** und **NewWeeklyReservation** wird bei Erreichen dieses Zieles und vor dem Dienstaufruf eine Datumsumwandlung durchgeführt. Das absolute Datum wird ermittelt und damit wird der Dienst aufgerufen, der die Reservierung löscht. Das zweite Dialogziel zum Löschen von Reservierungen, **CancelReservationById**, erwartet die Identifikationsnummer einer Reservierung im Dialogkontext. Dies wird für die Verknüpfung mit der grafischen Oberfläche benötigt und wird in Abschnitt 4.3.2 näher beschrieben.

Die weiteren Dialogziele **SayHello**, **SayGoodBye** und **Help** dienen zur

Begrüßung und Verabschiedung des Benutzers und zur Bereitstellung von Hilfe bei der Benutzung des Systems.

Die grafische Oberfläche stellt einen Kalender dar (siehe Abbildung 4.1). Um auch über natürlichsprachliche Eingabe in diesem Kalender navigieren zu können, ist das Dialogziel **Navigate** definiert. Da dieses Dialogziel auch ein Datum als Argument hat, wird hier bei Erreichen des Dialogziels und vor dem Dienstaufruf eine Datums- und Zeitumformung durchgeführt. Mit dem ermittelten absoluten Datum wird dann der Dienst aufgerufen, der dieses Datum im Kalender anzeigt.

4.2.3 Grammatik für natürlichsprachliche Eingaben

Ein weiterer Teil der Wissensbasis des *Tapas*-Dialogmanagers ist die Grammatik. Diese definiert alle gültigen natürlichsprachlichen Benutzereingaben für diese Anwendung. Außerdem verknüpft sie die Eingaben mit den Konzepten aus dem Domänenmodell. Durch diese Verknüpfungen wird die Benutzereingabe in eine semantische Darstellung gebracht. Damit können Dialogziele ausgewählt werden oder vorkommende Konzepte in bereits ausgewählten Dialogzielen ergänzt werden.

Für das System ist eine englische Grammatik definiert. Diese ist in zwei Teile aufgeteilt. Im einen Teil werden die Benutzereingaben zur Erkennung der Dialogziele definiert (Datei *resman.engl.jgram*), im anderen die Benutzereingaben zur Erkennung von Datums- und Zeitangaben (Datei *datetime2.engl.jgram*). Der Vorteil an dieser Aufteilung liegt darin, dass die Grammatik zur Datums- und Zeiterkennung auch für andere Anwendungen genutzt werden kann. Sie ist nicht speziell auf die Anwendung in einem Reservierungssystem ausgelegt sondern erkennt allgemeine Datums- und Zeitformate. Damit kann sie in anderen Anwendungen, die ebenfalls Datums- oder Zeitangaben benötigen, direkt verwendet werden.

Grammatik zur Erkennung von Dialogzielen

Um einzelne Dialogziele auswählen zu können, müssen die Konzepte, die mit den Dialogzielen verknüpft sind, in der Benutzereingabe erkannt werden. So wird zum Beispiel die Benutzereingabe

```
hello
```

in die semantische Darstellung

```
act_hello
```

umgewandelt, durch die wiederum der Dialogmanager das Dialogziel **SayHello** auswählt.

Komplexere Eingaben mit mehr Informationen werden am Beispiel des Dialogziels **NewReservation** veranschaulicht, für andere Dialogziele wie **CancelReservation** oder **Navigate** ist die Grammatik analog definiert.

Einfache Benutzereingaben, um eine neue Reservierung zu erstellen, sind zum Beispiel

```
I want to reserve the room
please book the room for me
```

Diese werden in die semantische Darstellung

```
act_newreservation
```

umgewandelt. Wenn mehr Informationen in der Eingabe enthalten sind, werden die erkannten Konzepte aus dem Domänenmodell in die Struktur der semantischen Darstellung eingefügt. So wird die Benutzereingabe

```
my name is philipp, I want to reserve the room
```

in die semantische Darstellung

```
act_newreservation
  resman:USER [ resman:obj_user
    resman:USERNAME [ "philipp" ]
  ]
```

umgewandelt, in der das Objekt **USER** des Konzepts **act_newreservation** bereits mit dem richtigen Benutzer belegt ist. Bei zusätzlichen Datums- oder Zeitangaben werden auch die entsprechenden Datums- oder Zeitobjekte belegt. Ein Beispiel für eine Benutzereingabe mit Datums- und Zeitangaben ist

```
I want to reserve the room on Friday at 4pm
```

Die daraus entstehende semantische Darstellung wird später näher beschrieben.

Grammatik zur Erkennung von Datums- und Zeitangaben

Die Grammatik zur Erkennung von Datums- und Zeitangaben ist allgemein gehalten und damit nicht auf ein bestimmtes Dialogziel oder eine bestimmte

Anwendung beschränkt. In der Grammatik der oben beschriebenen Dialogziele wird daher bei allen Konzepten, die Datums- oder Zeitattribute besitzen, immer auf die gleichen Datums- und Zeitdefinitionen zurückgegriffen.

Um dem Benutzer möglichst viele Freiheiten zu gewähren, werden in der Grammatik absolute und relative Datums- und Zeitangaben definiert. Eine absolute Datumsangabe ist zum Beispiel "May fourteenth", eine relative Datumsangabe "in tow days". Dabei ist es notwendig, die Datums- oder Zeiteingaben in eine semantische Darstellung umzuwandeln, da auch ein Kombination von absoluten und relativen Angaben möglich sein soll.

Die folgenden Beispiele erklären anschaulich, wie Datums- und Zeitangaben in der Grammatik definiert sind. Dies geschieht wieder anhand des Dialogziels **NewReservation**. Wichtig ist dabei das Erkennen von mehreren Objekten in einer Eingabe, in diesem Fall das Dialogziel und die Datums- und Zeitangaben.

Beispiel 1 (absolutes Datum) *Eingabe:*

```
May fourteenth
```

Semantische Darstellung:

```
datetime2:obj_date
  datetime2:DAY_ABS [ datetime2:obj_ordinal_number
    datetime2:VAL [ "14" ]
  ]
  datetime2:MONTH_ABS [ datetime2:obj_month
    generic:NAME [ "may" ]
    datetime2:VAL [ "5" ]
  ]
```

Die absolute Datumsangabe wird in das Konzept ***obj_date*** *umgewandelt. Dabei werden die Objekte* ***DAY_ABS*** *und* ***MONTH_ABS*** *initialisiert, da die dafür notwendigen Informationen aus der Eingabe gewonnen werden können.*

Auf das Dialogziel ***NewReservation*** *bezogen wird zum Beispiel die kombinierte Eingabe*

```
i want to make a new reservation at May fourteenth
```

in die semantische Darstellung

```
act_newreservation
  datetime2:DATE [ datetime2:obj_date
    datetime2:DAY_ABS [ datetime2:obj_ordinal_number
```

```
    datetime2:VAL [ "14" ]
  ]
  datetime2:MONTH_ABS [ datetime2:obj_month
    generic:NAME [ "may" ]
    datetime2:VAL [ "5" ]
  ]
]
```

gebracht. Das Objekt ***DATE*** *wird dann mit dem erkannten absoluten Datum belegt. Beim Erreichen des Dialogziels kann dieses Datum dann direkt zum Reservieren des Raumes verwendet werden.*

■

Beispiel 2 (relatives Datum) *Eingabe:*

```
in two days
```

semantische Darstellung:

```
datetime2:obj_date
  datetime2:DAY_REL [ datetime2:obj_number
    datetime2:VAL [ "2" ]
  ]
```

Bei dieser relativen Datumsangabe kann nur das Attribute ***DAY_REL*** *des Konzepts* ***obj_date*** *initialisiert werden. Information über den Monat sind nicht vorhanden.*

In Kombination mit dem Dialogziel ***NewReservation*** *wird dann wie im vorherigen Beispiel 1 das Objekt* ***DATE*** *belegt. Allerdings muss dann beim Erreichen des Dialogziels das Datum umgewandelt werden. Die relative Datumsangabe wird dann mit Hilfe des aktuellen Datums in ein absolutes Datum umgerechnet.*

■

Beispiel 3 (absolute Uhrzeit) *Eingabe:*

```
five thirty pm
```

semantische Darstellung:

```
datetime2:obj_time
  datetime2:HOUR [ datetime2:obj_hour
    datetime2:ABSOLUTE_PM [ datetime2:obj_number
      datetime2:VAL [ "5" ]
    ]
  ]
  datetime2:MINUTE [ datetime2:obj_minute
    datetime2:ABSOLUTE [ datetime2:obj_number
      datetime2:VAL [ "30" ]
    ]
  ]
```

Analog zum absoluten Datum wird diese absolute Zeitangabe in das Konzept ***obj_time*** *umgewandelt. Die Objekte* ***HOUR*** *und* ***MINUTE*** *werden dabei dementsprechend belegt.*

Wenn das Erstellen einer neuen Reservierung direkt mit Uhrzeitangabe erfolgt, wie bei dieser Eingabe

```
i want to make a new reservation at five thirty pm
```

wird das Objekt ***TIME*** *von* ***act_newreservation*** *direkt mit der Uhrzeit initialisiert.*

■

Beispiel 4 (relative Uhrzeit) *Eingabe:*

```
in two hours and twenty minutes
```

semantische Darstellung:

```
datetime2:obj_time
  datetime2:HOUR [ datetime2:obj_rel_hour
    datetime2:RELATIVE [ datetime2:obj_number
      datetime2:VAL [ "2" ]
    ]
  ]
  datetime2:MINUTE [ datetime2:obj_rel_minute
    datetime2:RELATIVE [ datetime2:obj_number
      datetime2:VAL [ "20" ]
    ]
  ]
```

Die Umwandlung von einer relativen Zeitangabe in eine semantische Darstellung erfolgt analog zur Umwandlung einer absoluten Zeit.

Wenn eine relative Zeitangabe als Attribut des Konzepts ***act_newreservation*** *angegeben ist, wird bei Erreichen des Dialogziels* ***NewReservation*** *diese Zeitangabe in eine absolute Uhrzeit umgerechnet. Dabei wird die aktuelle Uhrzeit als Ausgangspunkt verwendet.*

■

4.3 Reservierungsmanager

Die Aufgaben des Hauptprogramms sind die Generierung der grafischen Oberfläche sowie die Kommunikationsverbindung zwischen Benutzereingaben und *Tapas*-Dialogmanager. Dabei wird Informationsfluss in beide Richtungen ermöglicht. Benutzereingaben über gesprochene natürliche Sprache, Text oder die grafische Oberfläche werden an den Dialogmanager weitergereicht. Die Ausgaben des Dialogmanagers werden verarbeitet und auf der grafischen Oberfläche angezeigt. Die Daten müssen dabei auf unterschiedliche Formate konvertiert werden, was in den Abschnitten 4.3.1 und 4.3.2 näher beschrieben wird.

Die Generierung der Weboberfläche ist mit Java Server Pages realisiert. Die Kommunikation mit dem Dialogmanager und Zugriffe auf die Datenbank zur Verwaltung der Reservierungen sind in Java Bibliotheken implementiert. Diese werden in die Java Server Pages eingebunden und von dort aus aufgerufen.

4.3.1 Natürlichsprachliche Benutzereingaben

Der Informationsfluss bei natürlichsprachlicher Benutzereingabe zwischen grafischer Oberfläche, Hauptprogramm und Dialogmanager ist in Abbildung 4.5 schematisch dargestellt.

Dabei bestehen die Daten nicht nur aus Texteingaben und Textausgaben, sondern auch aus Befehlen. Ein Beispiel dafür ist der Befehl zum Erstellen einer neuen Reservierung an einem bestimmten Datum, der vom Dialogmanager bei Erreichen des Dialogziels **NewReservation** an das Hauptprogramm gesendet wird. Eine ausführlichere Darstellung des Informationsflusses und der Datenumwandlung befindet sich in Beispiel 5.

Beispiel 5 (Informationsfluss bei Texteingabe) *Der Informationsfluss bei der Eingabe von natürlichsprachlichem Text wird hier anhand der Eingabe*

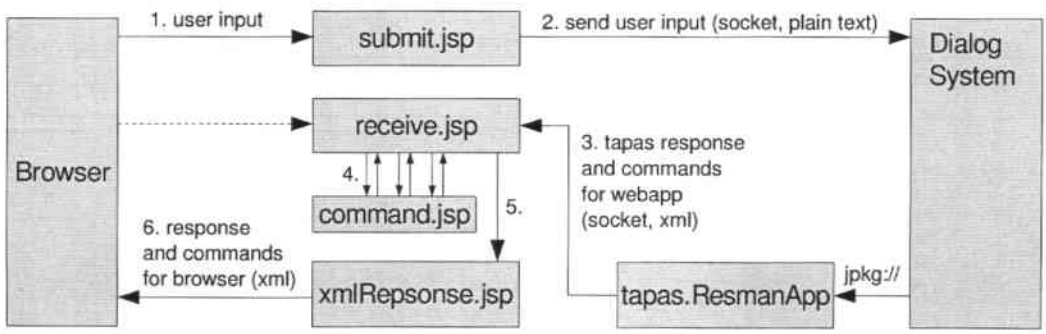

Abbildung 4.5: Informationsfluss bei Texteingabe des Benutzers.

```
Please show me next month.
```

anschaulich erklärt. Dabei wird jede Umwandlung der Daten beschrieben.

1. *Die Benutzereingabe im Browser wird als Argument an eine JSP-Seite überreicht, dabei wird der Eingabetext nicht umgewandelt.*

   ```
   Please show me next month.
   ```

2. *Der Eingabetext wird an den Dialogmanager weitergeleitet. Dabei werden Sonderzeichen wie zum Beispiel "." oder "?" entfernt.*

   ```
   Please show me next month
   ```

3. *Der Dialogmanager verarbeitet die Eingabe. In diesem Fall wird das Dialogziel* ***Navigate*** *ausgewählt. Da die Vorbedingungen erfüllt sind, werden die für dieses Dialogziel in den Ressourcen definierten Dienste aufgerufen. In diesem Fall wird eine Javamethode ausgeführt, die den Ausgabetext und das ausgewählte Datum als Argument erhält und eine XML-Ausgabe generiert, die sowohl die Textausgabe für die grafische Oberfläche als auch den Befehl zur Kalendernavigation enthält.*

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <resman>
     <message class="tapasText">Calendar navigation...</message>
     <command>NAVIGATE 2006;7;19</command>
   </resman>
   ```

4. *Die Webapplikation verarbeitet die Antwort des Dialogsystems. Dabei werden enthaltene Befehl ausgeführt. In diesem Fall wird die interne Repräsentation des Kalenders auf das neue Datum gesetzt. Die Anzeige im Browser wird dadurch noch nicht verändert.*

5. *Die XML-Antwort für den Browser wird generiert. Auch hier ist der Text und ein Befehl enthalten, der den Browser dazu veranlasst, den Kalender neu zu laden. Dies ist notwendig, um auch die Kalenderdarstellung im Browser auf das neue Datum zu aktualisieren.*

```
<?xml version="1.0" encoding="UTF-8"?>
<resman>
  <message class="tapasText">Calendar navigation...</message>
  <command>calendar_reload</command>
</resman>
```

6. *Der Text wird angezeigt und der Kalender neu geladen.*

■

4.3.2 Benutzereingaben über die grafische Oberfläche

Die grafische Oberfläche ist durch eine Weboberfläche innerhalb eines Browsers realisiert. Deshalb sind die Benutzereingaben über die grafische Oberfläche Klicks auf bestimmte Bereiche. Dies kann über ein Zeigegerät wie eine Maus oder über ein Touchscreen erfolgen. Es gibt dabei zwei verschiedene Reaktion auf solche Benutzereingaben.

In einem Fall wird durch den Klick ein Befehl an das Hauptprogramm gesendet. Die Verarbeitung ist dann je nach Befehl unterschiedlich. Beispiel 6, 7 und 8 zeigen verschiedene Verarbeitungsabläufe.

Im anderen Fall löst ein Klick auf eine bestimmte Schaltfläche eine normale Texteingabe aus. Der Text ist dabei vorher festgelegt. Der Informationsfluss ist damit identisch zu dem Informationsfluss, der in Abschnitt 4.3.1 beschrieben wird. Dies wird zum Beispiel für Hilfetexte verwendet. Der Benutzer kann auf den Hilfetext klicken, der erklärt, wie man eine neue Reservierung erstellt. Dabei wird ein Standardsatz übertragen und angezeigt, der dieses Dialogziel im Dialogmanager auslöst. Da der Benutzer den Satz im Dialog angezeigt bekommt, ist ein Lerneffekt gegeben, so dass bei der nächsten Reservierung die Hilfe nicht mehr benötigt wird.

Beispiel 6 (Kalendernavigation) *Ein Klick mit der Maus auf einen Navigationslink schickt den jeweiligen Befehl an das Hauptprogramm. Wenn*

zum Beispiel der Link Next Month *ausgewählt wird, dann wird der Befehl* ***next_month*** *übertragen. Das Hauptprogramm ändert daraufhin das Startdatum des Kalenders und generiert diesen neu.*

■

Beispiel 7 (Neue Reservierung an Datum) *Wenn das Symbol zum Erstellen einer neuen Reservierung in einem bestimmten Datumsfeld ausgewählt wird, wird dieses Datum zusammen mit der Information, dass eine neue Reservierung erstellt werden soll, an das Hauptprogramm gesendet. Dort wird das Datum direkt in die semantische Repräsentation des Dialogmanagers gebracht und an dessen semantischen Eingang weitergeschickt. Der weitere Informationsfluss erfolgt dann wie in Abschnitt 4.3.1.*

Wird zum Beispiel an dem Kalenderfeld mit dem Zeitpunkt

```
21. Juni 2006, 10 Uhr
```

das Symbol zum Erstellen einer neuen Reservierung angeklickt, dann schickt das Hauptprogramm die semantische Darstellung

```
resman:act_newreservation
  datetime2:DATE [ datetime2:obj_date
    datetime2:DAY_ABS [ datetime2:obj_number
      datetime2:VAL [ "21" ]
    ]
    datetime2:MONTH_ABS [ datetime2:obj_number
      datetime2:VAL [ "6" ]
    ]
  ]
  datetime2:TIME [ datetime2:obj_time
    datetime2:HOUR [ datetime2:obj_hour
      datetime2:ABSOLUTE [ datetime2:obj_number
        datetime2:VAL [ "10" ]
      ]
    ]
    datetime2:MINUTE [ datetime2:obj_minute
      datetime2:ABSOLUTE [ datetime2:obj_number
        datetime2:VAL [ "0" ]
      ]
    ]
  ]
```

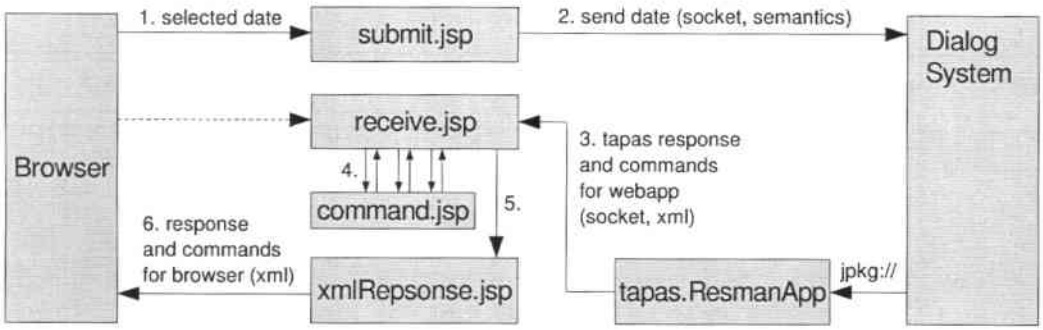

Abbildung 4.6: Informationsfluss bei semantischer Eingabe (z.B. Auswahl eines Datums).

an den semantischen Eingang des Dialogmanagers.

Eine schematische Darstellung dieses Informationsflusses befindet sich in Abbildung 4.6.

■

Beispiel 8 (Spezielle semantische Darstellungen) *Bei Benutzereingaben über die grafische Oberfläche werden teilweise auch semantische Informationen an den Dialogmanager übertragen, die durch natürlich sprachliche Eingabe nicht möglich sind, da dem Benutzer diese Informationen nie angezeigt werden.*

So wird beim Klicken auf das Löschen-Symbol neben einer Reservierung der Befehl zum Löschen sowie die Identifikationsnummer dieser Reservierung übertragen. Wenn diese zum Beispiel ***28*** *ist, dann wird die semantische Darstellung*

```
resman:act_cancelreservation
  resman:ID [ "28" ]
```

an den Dialogmanager geschickt. Diese Nummer ist für den Benutzer nie sichtbar und wird nur intern zur Verwaltung der Reservierungen benötigt.

■

Kapitel 5

Experimente und Evaluation

Die Benutzerexperimente wurden in zwei Phasen durchgeführt. In der ersten Phase wurde das System über das Internet getestet. Dabei war natürlichsprachliche Eingabe nur über das Textfeld möglich. In der zweiten Phase wurde das Experiment an einem speziellen Rechner durchgeführt, an dem ein Mikrofon mit Spracherkennung installiert war. Dabei waren Eingaben über gesprochene Sprache möglich. Die Ausgaben des Systems wurden mit einem Text-to-Speech-Programm als Sprachausgaben über Lautsprecher ausgegeben.

Im ersten Experimente wurde den Benutzern ein Experiment- und ein Auswertungsbogen vorgelegt. Diese befinden sich im Anhang A.1. Neben einer kleinen Einführung enthält der Experimentbogen einfache Aufgaben, die der Benutzer lösen soll. Diese Aufgaben sind zum Beispiel das Erstellen einer Reservierung an einem bestimmten Zeitpunkt oder die Navigation im Kalender zu einem bestimmten Datum. Auf dem Auswertungsbogen kann der Benutzer seine Meinung zu dem System äußern. Dabei gibt es eine Aufteilung der Bewertung in die Bereiche Dialog, grafische Oberfläche und Gesamteindruck. Bei der Beschreibung und Auswertung der einzelnen Experimente in den Abschnitten 5.1 und 5.2 wird näher auf diese Bewertungen eingegangen. Im zweiten Experiment wurden die Aufgaben von dem Versuchsleiter gestellt.

5.1 Benutzerexperiment mit Texteingabe über das Internet

An diesem Experiment haben 5 Testpersonen teilgenommen. Alle haben das Experiment bis zum Ende durchgeführt, auch wenn nicht alle Aufgaben gelöst werden konnten.

	Anzahl / Testperson					*Alle*
	P1	*P2*	*P3*	*P4*	*P5*	*Personen*
Eingaben über grafischer Oberfläche	5	-	18	8	8	14.0%
Korrekt erkannte Texteingaben	1	10	78	17	19	42.0%
Fehlerhaft erkannte Texteingaben	6	23	58	10	23	44.0%

Abbildung 5.1: Verteilung der Benutzereingaben.

Das Experiment wurde ohne Versuchsleiter durchgeführt. Die Teilnehmer erhielten den Experiment- und Auswertungsbogen zugeschickt und haben das System selbstständig getestet. Ein Vorteil bei einer Evaluation ohne Versuchsleiter ist, dass die Benutzer nicht beeinflusst werden. Sie müssen sich alleine mit dem System auseinandersetzten. Dadurch werden Schwächen des Systems besser erkannt. Ein Nachteil ist es, dass es zu Situationen kommen kann, in denen der Benutzer nicht mehr weiterkommt. Dies kann an Fehlern im System oder an einer fehlerhaften Benutzung liegen. Die Folge daraus ist eine Frustration der Testperson, die zu einer schlechten Bewertung des Systems oder sogar zum Abbruch des Experiments führen kann.

In Abschnitt 5.1.1 werden die Benutzereingaben und Reaktionen des Systems auf diese Eingaben analysiert. Im Abschnitt 5.1.2 wird auf die Bewertung der Teilnehmer eingegangen.

5.1.1 Auswertung des Dialogs

Benutzereingaben bei diesem Experiment waren in Textform oder über die grafische Oberfläche möglich. Abbildung 5.1 zeigt die Verteilung der Eingaben. Dabei wird bei den Texteingaben noch zwischen korrekt erkannten und fehlerhaft erkannten Eingaben unterschieden.

Diese Verteilung zeigt, dass die Texteingabe bei allen Benutzern mehr verwendet wurde als die Eingabe über die grafische Oberfläche. Dies liegt zum Teil daran, dass einige Eingaben nur über die Texteingabe möglich sind, zum Beispiel die Eingabe des Namens des Benutzers oder der Dauer der Reservierung. Dies alleine erklärt aber noch nicht den geringen Anteil der Eingaben über die grafische Oberfläche. Die Benutzer haben bei vielen Eingaben, die über beide Modalitäten möglich sind, die Texteingabe gewählt. Dies wird bei der Betrachtung der Bewertungsbögen in Abschnitt 5.1.2 noch einmal aufgegriffen. In welcher Situation welche Modalität verwendet wurde war zwischen den Benutzern sehr unterschiedlich. So hat zum Beispiel Benutzer 1 immer zuerst die grafische Eingabe verwendet. Nur bei den Fragen, die

	Anzahl / Testperson					*Alle*
	P1	*P2*	*P3*	*P4*	*P5*	*Personen*
fehlerhaftes Datumsformat	-	5	35	7	5	42.0%
Intention nicht erkannt	1	4	31	4	2	33.0%
Text nicht erkannt	-	-	6	4	12	18.0%
Schreibfehler	-	1	6	2	-	7.0%

Abbildung 5.2: Verteilung der fehlerhaft erkannten Texteingaben auf Fehlerkategorien.

zwingend die Texteingabe benötigen, wurde diese dann auch verwendet. Benutzer 3 hat versucht, alle Aufgaben über die Texteingabe zu lösen. Erst bei mehreren hintereinanderfolgenden Erkennungsfehlern wurde auf die grafische Oberfläche zur Eingabe von Datum und Uhrzeit zurückgegriffen.

Der Anteil der fehlerhaft erkannten Texteingaben ist (siehe Abbildung 5.1) relativ hoch. In Abbildung 5.2 werden diese Texteingaben in Fehlerkategorien aufgeschlüsselt. Dabei werden die Hauptursachen für fehlerhaftes Erkennen deutlich.

Den größten Anteil haben Eingaben mit falschem Datumsformat. Dieses Problem beruht hauptsächlich darauf, dass die Testpersonen Englisch als Fremdsprache gelernt haben. Es wurden oft Datumsformate verwendet, die im Englischen nicht korrekt sind. Ein Beispiel für solch eine fehlerhafte Eingabe ist "fourteen September". Ein weiterer großer Teil der fehlerhaft erkannten Datumseingaben enthält Abkürzungen oder nicht ausformulierte Zeitangaben wie zum Beispiel "14.09.2006" oder "1h". Diese werden vom natürlichsprachlichen Dialogsystem nicht erkannt.

Eingaben, die als Text erkannt wurden, aber bei denen die Intention des Benutzers nicht ermittelt wurde, haben den zweitgrößten Anteil an den fehlerhaft erkannten Eingaben. Ursachen dafür sind oft aus dem Kontext gerissene Benutzereingaben. Das System hat intern kein Dialogziel ausgewählt und muss eine Texteingabe verarbeiten, die einem bestimmten Dialogziel zugeordnet ist und nur damit Sinn macht. In diesem Fall kann die Intention nicht erkannt werden. Ein Beispiel, wie es zu dieser Situation kommen kann, ist die wiederholte Eingabe eines nicht erkannten Datums bei der Erstellung einer neuen Reservierung. In diesem Fall verwirft das System den Gesprächskontext. Wenn daraufhin ein korrektes Datum eingegeben wird, kann dieses nicht mehr dem Dialogziel zum Erstellen einer neuen Reservierung zugeordnet werden.

Beispiel 9 (aus dem Kontext gerissene Benutzereingabe)

User: *I want to make a weekly reservation.*
System: *What is your name?*
User: *In September.*

Das System erwartet einen Namen und kann mit der Datumsangabe nichts anfangen. Dadurch wird der Dialogkontext zurückgesetzt. Der Benutzer erhält die Rückmeldung, dass seine Eingabe nicht verstanden wurde.

User: *Fourteen September.*

Da der Dialogkontext zurückgesetzt wurde, wird dieses Datum nicht dem Dialogziel zur Erstellung einer neuen wöchentlichen Reservierung zugeordnet und der Benutzer erhält wieder eine Fehlermeldung.

■

Eine Verbesserung dieser hohen Fehlerquote bei der Erkennung von Texteingaben kann dadurch erreicht werden, dass die Benutzer auf das Problem des Datumsformats hingewiesen werden. Wenn nur noch ausformulierte und korrekte englische Daten eingegeben werden, wird die Erkennungrate deutlich verbessert. Dadurch werden auch die Fehler bei der Erkennung der Intention verringert, da diese oft dadurch entstehen, dass ein fehlerhaftes Datum eingegeben wird und deswegen dann das aktuell Dialogziel abgebrochen wird. Auf die Intention der nächsten Eingabe kann das System dann nicht mehr korrekt schließen. Eine weitere Möglichkeit der Verbesserung der Datumserkennung ist die Übernahme von falschen Datumsformaten in die Grammatik. Dies ist vor allem bei vorwiegender Nutzung von nicht muttersprachlichen Personen sinnvoll. Allerdings sind dieser Vorgehensweise Grenzen gesetzt. Je mehr unterschiedliche, nicht korrekte Datumsformate definiert werden, desto schwieriger wird die Abgrenzung dieser Formate zu gleichzeitig eingegebenen Uhrzeiten oder Zahlen. Ein paar dieser nicht korrekten Datumsformate sind in dem System dieser Studienarbeit implementiert. So wird zum Beispiel das Datum "the 5th may" erkannt, das vom deutschen Datumsformat abgeleitet ist.

Weitere Ursachen nicht erkannter Texte sind neben den falschen Datumsformaten auch Äußerungen der Benutzer, die nicht von der Grammatik abgedeckt sind oder außerhalb der Domäne liegen. Beispiele dazu sind die Benutzereingaben "I need a reservation" und "I want to know you".

Eine wichtige Information zur Auswertung des Dialogs ist die Anzahl der erkannten Dialogziele und ob diese vollständig erreicht, fehlerhaft ausgeführt oder abgebrochen wurden. Abbildung 5.3 enthält diese Verteilung.

Der Grund für die fehlerhafte Ausführung von Dialogzielen lag immer an falsch erkannten Datumsformaten. Ein Teil davon wurde komplett falsch erkannt, was in Beispiel 10 veranschaulicht wird. Bei komplexeren Datumseingaben, bei denen nur ein Teil als korrektes Datumsformat formuliert wurde,

	Anzahl / Testperson					*Alle Personen*
	P1	*P2*	*P3*	*P4*	*P5*	
vollständig erreicht	3	-	15	2	4	42.0%
abgebrochen	1	-	11	6	2	34.0%
fehlerhaft erreicht	1	5	1	3	4	24.0%

Abbildung 5.3: Status der im Dialogverlauf vom System ausgewählten Dialogziele.

wurde dieser Teil als Datum erkannt, der Rest wurde ignoriert. Dadurch wurde intern ein falsches Datum als Objekt eines Dialogziels gespeichert und damit das Dialogziel fehlerhaft ausgeführt. Der Ursprung dieses Fehlers liegt in der unvollständigen Erkennung der Eingabe. Anstatt den ganzen Eingabetext zurückzuweisen wird vom Parser nur ein Teil verwendet. Diese Fehlerquelle könnte durch das Einbinden eines anderen Parsers in das *Tapas*-System behoben werden.

Beispiel 10 (Fehlerhafte Ausführung) *Der Benutzer möchte eine neue Reservierung erstellen und hat bereits seinen Namen, Datum und Zeit eingegeben.*

⋮

System: *How long do you want to reserve the room?*
User: *One.*

⋮

Die Eingabe "one" wird als "one o'clock" interpretiert, und nicht als "one hour". Da es sich trotzdem um eine Zeitangabe handelt, wird das Dialogziel fertiggestellt. Die Länge der Reservierung wird dabei fälschlicherweise auf 0 gesetzt.

■

Der hohe Anteil an abgebrochenen Dialogzielen hängt auch direkt mit den Problemen bei der Datumseingabe zusammen. Nach mehreren nichterkannten Eingaben bricht der Dialogmanager das erkannte Dialogziel ab. Bei Verwendung von korrekten Datumsformaten kann der Anteil der abgebrochenen Dialogziele stark vermindert werden.

5.1.2 Auswertung der Benutzerbewertung

Alle Testpersonen haben den Bewertungsbogen ausgefüllt. Die Personen sind zwischen 26 und 29 Jahre alt. Wie in Abbildung 5.4 zu sehen, haben sie sehr

	Testpersonen				
	P1	*P2*	*P3*	*P4*	*P5*
Ich kenne mich allgemein mit Computern aus.	3	5	3	2	5
Ich kenne mich mit Dialogsystemen aus.	5	1	1	1	2
Das System hat die Eingaben verstanden.	2	4	2	2	3
Die Ausgaben des Systems waren verständlich.	1	5	3	2	4
Das System hat wie erwartet reagiert.	3	3	1	4	2
Die Dialogführung war nachvollziehbar.	2	4	1	3	3
Alle Aufgaben konnten erledigt werden.	2	1	5	1	5
Die grafische Oberfläche ist intuitiv bedienbar.	2	4	3	3	4
Alle wichtigen Informationen sind dargestellt.	2	5	4	3	4
Die Hilfebox ist nützlich.	1	5	2	3	5
Die grafische Steuerung liegt mir mehr als die natürlichsprachliche Eingabe.	2	4	4	4	4
Ich finde den Reservation Manager sinnvoll.	1	5	4	4	2
Die Kombination aus grafischer Benutzeroberfläche und Dialog gefällt mir besonders.	3	4	4	3	4

Abbildung 5.4: Antworten der Testpersonen im Bewertungsbogen.
(1: trifft nicht zu, 3: trifft teilweise zu, 5: trifft sehr gut zu)

unterschiedliche allgemeine Computerkenntnisse, von eher wenigen bis zu sehr guten Kenntnissen. Vorkenntnisse über Dialogsysteme sind außer bei der Testperson 1 nur sehr gering vorhanden.

Die Angaben im Bewertungsbogen variieren sehr stark. Dies deutet darauf hin, dass die Bewertung des Systems eng mit persönlichen Vorlieben verknüpft ist. Ein Beispiel dafür ist die Antwort auf die Frage im Bewertungsbogen, ob alle Aufgaben erledigt werden konnten. Wenn man diese Angaben in Bezug auf den Anteil der vollständig erreichten Dialogziele setzt (siehe Abbildung 5.5), ist erkennbar, dass die Bewertung mit höherem Anteil steigt. Aus der Reihe fällt dabei die Testperson mit dem höchsten Anteil an vollständig erreichten Dialogzielen, Testperson 1. Diese Person fand ein dialogbasiertes System nicht sehr sinnvoll und hat daraufhin auch eher schlechtere Bewertungen eingetragen. Im Gegensatz dazu hat Testperson 2 angegeben, dass die Aufgaben nicht erledigt werden konnten. Trotzdem ist seine Bewertung

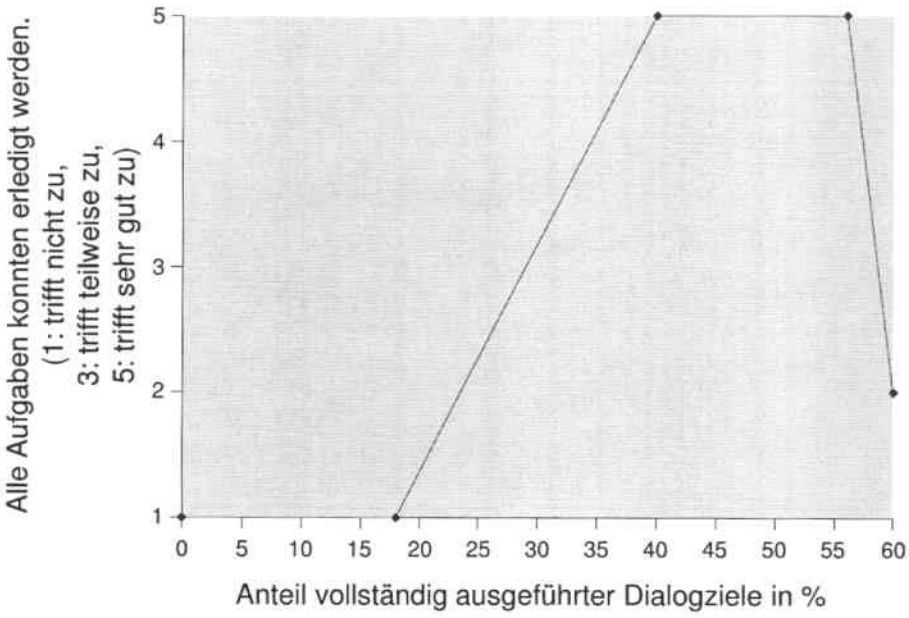

Abbildung 5.5: Antwort auf die Frage im Bewertungsbogen, ob alle Aufgaben erledigt werden konnten, in Bezug auf den Anteil der vollständig erreichten Dialogziele der jeweiligen Testperson.

für die Bedienbarkeit sehr gut.

Überraschenderweise haben fast alle Testpersonen angegeben, dass ihnen die grafische Steuerung mehr liegt als die natürlichsprachliche Eingabe. Dies entspricht nicht den Erwartungen, da die Testpersonen im Experiment die natürlichsprachliche Eingabe deutlich häufiger benutzt haben. Wie schon in Abschnitt 5.1.1 angesprochen könnte ein Grund dafür sein, dass die Benutzer die neue, eher unbekannte Modalität im Experiment testen wollten. Die bekannte und eingeübte grafische Steuerung wurde dann trotzdem als einfacher bewertet.

Die Kombination von grafischer Oberfläche und natürlichsprachlichem Dialog wurde positiv gewertet. Daraus ergibt sich, dass die grafische Steuerung zwar als einfacher empfunden wird, ein multimodales System aber auch als sinnvoll gesehen wird.

Bei der Frage, ob das System an sich sinnvoll ist, liegen die Bewertungen weit auseinander. Die Antworten hängen auch nicht mit dem Anteil der erfolgreich ausgeführten Dialogziele zusammen. Dies deutet wieder darauf hin,

dass die Bewertung aus persönlichen Vorlieben der einzelnen Testpersonen erfolgt ist.

5.2 Benutzerexperiment mit gesprochener natürlichsprachlicher Eingabe

Im Vorfeld dieses Experiments wurde das System mit zwei Benutzern ohne Auswertung getestet. Dies diente dazu herauszufinden, ob die gesprochene natürlichsprachliche Eingabe grundsätzlich funktioniert und ob noch Änderungen am System notwendig sind. Bei diesen Tests stellte sich heraus, dass die Fehlerrate beim Erkennen von Daten und Uhrzeiten sehr hoch war. Daraufhin wurde die Grammatik zur Erkennung von Daten und Uhrzeiten an den Stellen, an denen die Fehler auftraten, leicht angepasst. Ein Beispiel dafür ist das Streichen der Grammatikregel zur Erkennung von zwei hintereinander folgenden Zahlen als Uhrzeit, wie "ten fifty". Diese Regel führte oft zur fälschlichen Erkennung von Uhrzeiten, auch wenn im gesprochenen Text überhaupt keine Uhrzeit vorkam. Weiterhin erkannt werden Formulierungen wie "ten fifty am". Durch die zusätzliche Angabe "am" wird das Fehlerrisiko deutlich verringert.

Das Benutzerexperiment wurde mit der leicht modifizierten Grammatik durchgeführt. Dabei nahmen drei Personen teil. Alle diese Personen waren nicht an dem ersten Experiment über das Internet beteiligt. Das Experiment wurde mit Versuchsleiter durchgeführt. Die Segmentierung der Spracheingabe wurde von den Benutzern selbst durchgeführt, indem sie während der Eingabe einen Button gedrückt hielten. Zusätzlich zu den Ausgaben der grafischen Oberfläche wurde den Benutzern nach jeder Eingabe noch der erkannte Text des Spracherkenners angezeigt.

Die Aufgaben, die die Benutzer erledigen sollten, wurden während des Experiments vom Versuchsleiter gestellt. Ziel dabei war es, die Aktionen Erstellen von Reservierungen, Löschen von Reservierungen und das Navigieren im Kalender abzudecken. Um den Benutzern möglichst wenig Vorgaben zu geben wurde die Aufgabenstellung aus dem ersten Experiment nicht verwendet. Die abstrakten Aufgabenstellungen, wie "erstelle eine Reservierung", sollten die Benutzer dazu führen, eigene Lösungswege und Formulierungen zu finden.

5.2.1 Auswertung des Dialogs

Abbildung 5.6 listet die Anzahl der korrekt und der fehlerhaft erkannten Eingaben auf. Daraus ergibt sich eine durchschnittliche Turnfehlerrate von

	Anzahl / Testperson			*Alle Personen*
	P1	*P2*	*P3*	
korrekt erkannte Eingabe	18	17	18	70.0%
Text falsch erkannt	4	1	14	25.0%
Intention nicht erkannt	2	-	3	5.0%

Abbildung 5.6: Anzahl der korrekt und fehlerhaft erkannten natürlichsprachlichen Eingaben.

30%.

Für das falsche Erkennen des Textes gab es zwei verschiedene Ursachen.

Zum einen wurden Eingaben mit undeutlicher Sprache nicht erkannt. Dazu zählen auch Eingaben, die Zwischenwörter wie "äh" oder "hmm" enthalten. Das Auftreten dieses Fehlers war bei den verschiedenen Testpersonen sehr unterschiedlich. Bei Testperson 1 sind 75% der Eingaben, bei denen der Text nicht erkannt wurde, auf undeutliche Sprache zurückzuführen. Bei Testperson 2 ist dies gar nicht vorgekommen.

Eine andere Ursache für das nicht Erkennen des Textes sind Formulierungen in der Eingabe, die nicht in der Grammatik definiert sind. Der Spracherkenner benützt die Grammatik zum Erkennen und ordnet jeder Spracheingabe einen gültiger Satz aus der Grammatik zu. Falls der Eingabetext nicht in der Grammatik vorhanden ist, kann dies zu komplett falschen Ergebnissen führen. Dieser Fehler ist bei der Testperson 3 oft vorgekommen. Ein Beispiel ist die Eingabe "anyway anywhen next week", die als "January eighteenth ten four am" erkannt wurde.

Das Nichterkennen der Intention bedeutet, dass der Spracherkenner den korrekten Text erkannt hat, der Dialogmanager aber die Intention nicht verstand. Ursachen davon sind analog zu den Fehlern bei der Intentionserkennung im vorherigen Experiment über das Internet und werden im Abschnitt 5.1.1 besprochen.

In Abbildung 5.7 wird die Anzahl der gelösten Aufgaben pro Benutzer aufgelistet. Dazu ist die Anzahl der Turns angegeben, die die einzelnen Benutzer im Durschnitt zum Lösen der Aufgaben benötigt haben. Im Vergleich zum ersten Experiment mit der textbasierten natürlichsprachlichen Eingabe fällt diese Anzahl kleiner aus. Erwartungsgemäß müsste das Ergebnis umgekehrt sein, da bei der gesprochenen natürlichsprachlichen Eingabe noch eine weitere Fehlerquelle durch den Spracherkenner gegeben ist. Allerdings gab es beim zweiten Experiment deutlich weniger Fehler beim Erkennen der Benutzereingaben. Mögliche Gründe dafür werden in Abschnitt 5.2.2 aufgelistet.

	Anzahl / Testperson		
	P1	*P2*	*P3*
Anzahl gelöster Aufgaben	4	4	5
Turns pro Aufgabe	6,0	4,5	7,0

Abbildung 5.7: Anzahl gelöster Aufgaben pro Benutzer und durchschnittliche Anzahl der Benutzereingaben pro Aufgabe.

Durch die geringe Anzahl von Erkennungsfehlern mussten die Testpersonen ihre Eingaben nicht wiederholen, dadurch ergibt sich die geringere Anzahl von Turns pro Aufgabe.

5.2.2 Ergebnisse

Im Vergleich zur Texteingabe beim Experiment über das Internet gab es bei diesem Experiment mit Spracheingabe deutlich weniger Fehler beim Erkennen von Datumsformaten. Ein Grund dafür ist, dass bei der Spracheingabe die Wörter nicht abgekürzt werden können. Dies führte bei der Texteingabe oft zu Erkennungsfehlern. Ein weiterer Grund ist, dass die Versuchspersonen bei der Spracheingabe größtenteils korrekte englische Datumsformate verwendet haben. Da bei diesen Experimenten nur Personen teilnahmen, die Englisch als Fremdsprache gelernt haben, könnte eine Begründung dafür sein, dass es den Personen bei gesprochener Sprache leichter viel, die korrekten Formulierungen abzurufen. Eine mögliche Erklärung ist allerdings auch, dass die Testpersonen, die am zweiten Experiment teilgenommen haben, über die besseren Englischkenntnisse verfügen.

Ein großes Problem bei der Zeiterkennung ist die Unterscheidung zwischen "am" und "pm". Diese Angaben wurden vom Spracherkenner oft verwechselt. Ein Vermeiden dieses Problems ist nur durch andere Formulierungen wie "9 o'clock" oder "4 in the afternoon" möglich.

Auffallend beim Experiment mit Spracheingabe war, dass die Testpersonen die grafische Oberfläche fast gar nicht verwendet haben. Vor dem Experiment wurde den Personen erklärt, dass Eingaben über beide Modalitäten möglich sind. Alle Personen lösten die gestellten Aufgaben dann aber nur über Spracheingaben. Eine mögliche Erklärung ist, dass der Reiz, die Aufgabe mit Spracheingabe zu erledigen, höher war. Die Bedienung der grafischen Oberfläche war den Personen geläufiger, sie haben sich also bewusst für die neue Modalität entschieden.

Kapitel 6

Ausblick

Um das System in größerem Rahmen produktiv einsetzen zu können, muss vor allem das Problem des gleichzeitigen Zugriffs geklärt werden. Bei der derzeitigen Implementierung ist das System nur für einen Benutzer gleichzeitig verfügbar. Bei einem weiteren Zugriff wird eine Seite mit der Meldung angezeigt, dass das System zur Zeit nicht verfügbar ist. Die technische Umsetzung des gleichzeitigen Zugriffs auf die grafische Oberfläche ist durch die Architektur als Webapplikation von Grund aus gegeben. Die Schwierigkeit besteht in der Fehlerbehebung bei gleichzeitigem Zugriff auf eine bestimmte Uhrzeit oder einen bestimmten Termin. Dies muss sofort vom System bemerkt werden und eine entsprechende Meldung an die Benutzer ausgegeben werden. Außerdem muss der *Tapas*-Dialogmanager für jeden Benutzer separat gestartet werden, um Dialogverlauf und Dialogkontext einzeln zu speichern. Dafür ist der *Tapas*-Dialogmanager grundsätzlich nicht ausgelegt. Eine Lösung dieses Problems wäre zum Beispiel das Verwalten von einer bestimmten Anzahl von *Tapas*-Dialogmanager-Instanzen in einem Pool. Bei einem Zugriff auf das System wird dem Benutzer ein freier *Tapas*-Dialogmanager aus dem Pool zugewiesen. Nach dem Beenden des Dialogs wird dieser dann wieder zurück in den Pool gegeben. Diese Probleme wurden im Rahmen dieser Studienarbeit nicht angegangen. Das System ist somit auf gleichzeitigen Zugriff von einer Person beschränkt.

Eine weitere fehlende Funktion ist die Einteilung der Termine in Kategorien. Dabei muss geklärt werden, ob eine bestimmte Auswahl an Kategorien angeboten wird, oder ob beliebige Kategorien von den Benutzern eingetragen werden können. Letzteres ist mit dem *Tapas*-Dialogmanager allerdings nur schwer zu realisieren, da das System nicht darauf ausgelegt ist, neue permanente Grammatikregeln in die Konfiguration aufzunehmen. Allerdings wird dieses Lernen von neuen Formulierungen in verschiedenen Forschungsarbeiten untersucht.

Schließlich fehlt ein Programm zur Verwaltung der im System registrierten Benutzer. Diese sind ein einer Datenbank gespeichert. Vorstellbar wäre eine Webapplikation als Verwaltungssystem oder ein auf Email basierendes System. Der Zugriff auf beide Systeme ist von verschiedenen Standpunkten aus möglich und damit für die Benutzerverwaltung geeignet.

Wenn diese Probleme gelöst sind, kann das System mit vielen Benutzern eingesetzt werden. Einer geringe Anzahl von Benutzern aus einem eingeschränkten Personenkreis ist es möglich, das hier vorgestellte System ohne Veränderungen produktiv zu benutzen. Gleichzeitiger Zugriff kommt durch die geringe Anzahl der Benutzer selten vor und durch den eingeschränkten Personenkreis ist die Einteilung in Kategorien nicht entscheidend, da die Benutzer sich gegenseitig kennen. Schließlich ist die Verwaltung der Benutzer aufgrund der kleinen Anzahl nicht mit hohem Aufwand verbunden.

Literaturverzeichnis

[Den02] M. Denecke. Rapid prototyping for spoken dialogue systems. *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[Gar05] Jesse James Garrett. Ajax: A new approach to web applications. 2005.

[Hol05] H. Holzapfel. Towards development of multilingual spoken dialogue systems. *Proceedigs of the 2nd LTC*, 2005.

[McT02] Michael F. McTear. Spoken dialogue technology: enabling the conversational user interface. *ACM Comput. Surv.*, 34(1):90–169, 2002.

[QGS+01] J. F. Quesada, F. Garcia, E. Sena, J. A. Bernal, and G. Amores. Dialogue management in a home machine environment: Linguistic components over an agent architecture. *Procesamiento del Lenguaje Natural*, 2001.

[Wah03] Wolfgang Wahlster. Smartkom: Symmetric multimodality in an adaptive and reusable dialogue shell. 2003.

Anhang A

A.1 Experiment- und Auswertungsbogen der Benutzerevaluation

Testbogen für *Reservation Manager*

Statistische Angaben

Aktuelle Uhrzeit: (wird zur Auswertung des Dialogs benötigt)

Alter: Geschlecht:

(1: trifft nicht zu, 3: trifft teilweise zu, 5: trifft sehr gut zu)

	1	*2*	*3*	*4*	*5*
Ich kenne mich allgemein mit Computern aus.					
Ich kenne mich mit Dialogsystemen aus.					

Einführung

Am ISL der Universität Karlsruhe gibt es einen Multimediaraum, der von vielen verschiedenen Mitarbeitern genutzt wird. Der *Reservation Manager* soll die Reservierungen für diesen Raum verwalten. Es könnte zum Beispiel ein Computer mit Touchscreen direkt an der Tür des Raumes installiert werden, an dem dann mit natürlicher Sprache und grafischer Eingabe Reservierungen vorgenommen werden können.

Über den Link

http://y80.de:8080/resman/

kann man den *Reservation Manager* erreichen.

Aufgabe

Mache dich kurz mit dem System vertraut und versuche, das System zur Reservierung eines Raumes sinnvoll zu nutzen.

Spezielle Aufgaben

Versuche jetzt, diese speziellen Aufgaben zu lösen:

1. Erstelle eine Reservierung für morgen um 10 Uhr für 30 Minuten.
2. Erstelle eine Reservierung am 14. September um 16 Uhr für 1 Stunde.
3. Gehe zum 19. Juli im Kalender.
4. Erstelle eine sich wöchentlich wiederholende Reservierung an diesem Tag.
5. Lösche diese Reservierung wieder.

Bewertung

(1: trifft nicht zu, 3: trifft teilweise zu, 5: trifft sehr gut zu)

1. Dialog

	1	*2*	*3*	*4*	*5*
Das System hat die Eingaben verstanden.					
Die Ausgaben des Systems waren verständlich.					
Das System hat wie erwartet reagiert.					
Die Dialogführung war nachvollziehbar.					
Alle Aufgaben konnten erledigt werden.					

2. Grafische Oberfläche

	1	*2*	*3*	*4*	*5*
Die grafische Oberfläche ist intuitiv bedienbar.					
Alle wichtigen Informationen sind dargestellt.					
Die Hilfebox ist nützlich.					

3. Gesamteindruck

	1	*2*	*3*	*4*	*5*
Die grafische Steuerung liegt mir mehr als die natürlichsprachliche Eingabe.					
Ich finde den *Reservation Manager* sinnvoll.					
Die Kombination aus grafischer Benutzeroberfläche und Dialog gefällt mir besonders. *Warum:*					

4. Kommentare und weiterführende Vorschläge
