

Zu Komplexen Diensten mit einfachen natürlichsprachlichen Interaktionen.

Volker Coors¹, Christian Kray², Robert Porzel³

Keywords: Personal memory; multimodale Dialoggestaltung

1. Einführung

Im Rahmen den interdisziplinären Forschungsbereiches „Personal Memory“ ist es Ziel des Deep Map Projektes neue Interaktionsparadigmen und –modalitäten am Beispiel eines web-basierten und eines mobilen elektronischen Touristenführers zu entwickeln. Die grundlegende Idee eines „personal memories“ oder eines persönlichen Assistenten in digitaler Form beinhaltet zunächst Speicherung und Verknüpfung aller persönlicher Daten [Bush45].

Persönliche Daten	/stunde	/Leben
Text lesen	100 KB	25 GB
Sprache hören	40 MB	10 TB
Fernsehen	2 GB	8 PB

Nachdem die Speicherung solcher Datenmengen gerade machbar wird stellt sich heraus, dass die Nutzbarkeit damit noch nicht gegeben ist. Selbst wenn aus den Daten durch eine gegebene Semantik Wissen wird, also bekannt ist, was in einem Eintrag steht, bzw. welche Eigenschaft eines Objektes er repräsentiert, ist noch kein Nutzen gegeben. Aus den gespeicherten Daten bzw. des gespeicherten Wissens wird erst Information, wenn das Wissen in einer konkreten Situation genutzt werden kann. Nach Kuhlén [Kuh90] ist Information „... die Teilmenge von Wissen, die von jemandem in einer konkreten Situation zur Lösung von Problemen benötigt wird.“ Das heißt, dass Wissen dadurch zu Information wird, dass es zu einem bestimmten Zeitpunkt in einer bestimmten Situation benötigt wird und zur Verfügung steht.

Eine Wissensbasis bringt also nur Nutzen, wenn Informationen in einer konkreten Situation abgefragt werden kann. Häufig ist der Benutzer einer Wissensdatenbank jedoch gar nicht in der Lage, seine Anfragen so präzise zu stellen, das sie genau den Informationsbedarf wiedergibt und die gewünschte Teilmenge von Wissen aus dem System extrahiert wird. Vielmehr werden vage Anfragen gestellt, die dadurch gekennzeichnet sind, dass die Antwort a priori nicht eindeutig definiert ist. Hierzu zählen neben Fragen mit unscharfen Kriterien insbesondere auch solche, die nur im Dialog iterativ durch Reformulierung (in Abhängigkeit

¹ Fraunhofer Institut für Graphische Datenverarbeitung, Rundeturmstr.6, D - 64283 Darmstadt

² Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D - 66123 Saarbrücken

³ European Media Laboratory – EML, Villa Bosch, Schloss-Wolfsbrunnengasse 33, D - 69118 Heidelberg

von den bisherigen Systemantworten) beantwortet werden können; häufig müssen zudem mehrere Datenbasen zur Beantwortung einer einzelnen Anfrage durchsucht werden.

Solche Dialoge werden überwiegend in künstlichen Anfragesprachen formuliert, die vom Informationssystem eindeutig interpretierbar sind. Die Formulierung der Anfragen wird durch eine graphische Benutzerschnittstelle unterstützt und lehnt sich dabei an der etablierten Desktop-Metapher und dem Point-und-Click-Paradigma an. Eine wesentlich intuitivere Form der User-Interface Gestaltung bieten natürlichsprachliche Dialoge. Diese neuartige Art der Interaktion und Dialoggestaltung läßt sich ebenso auf viele Bereiche des Digital Storytelling übertragen.

Fortschritte aus Gebieten der künstlichen Intelligenz, der Computerlinguistik und der Informatik ermöglichen Erfüllung zweier notwendigen Bedingungen. Zum einen Interfaces und Verarbeitungstechniken, welche nicht nur auf Beispielszenarien und kleinen Datensätzen operieren. Zum anderen die Verknüpfung unterschiedlichster Datenquellen, welche sowohl heterogen als auch verteilt sein können.

Im folgenden stellen wir, nach einem Überblick über bisherige Systeme, anhand des Prototypen eines elektronischen Touristenführers verschiedene dieser Techniken vor, um zu zeigen wie informationstechnisch ungeschulten Benutzern über ein multimodales Interface komplexe Funktionalitäten unterschiedlicher Dienste und Datenquellen zur Verfügung gestellt werden kann. Ein weiterer Aspekt des Deep Map Systems ist die Unterstützung des Benutzer in verschiedenen Umgebungen: sowohl am stationären PC als auch unterwegs mit einem wearable PC.

2. State of the Art

2.1 Natürlichsprachliche Mensch-Maschinen Interaktion

Der Bereich der Sprachverarbeitungssysteme kann in drei Kategorien unterteilt werden:

- Unterstützung der Mensch-Mensch Kommunikation, z.B. automatische Übersetzungssysteme wie VERBMOBIL [Wahl97] und C-STAR [Waib96] im Forschungsbereich oder der *Personal Translator* von LinguaTec im Produktbereich.
- Unterstützung des Menschen beim Umgang mit Sprache, z.B: Diktiersysteme (FreeSpeech (Philips), ViaVoice (IBM), Naturally Speaking (L&H), etc.) oder orthographische und textuelle Tools bei der Textverarbeitung.
- Dialogische Mensch-Maschinen Systeme, z.B. EVAR [Gall98], das Philips Train Timetable System [Aust95], das SRI Autoroute System [Alsh92] oder TRAINS System zur Routenplanung [Alle96].

Für den Bereich des Digital Storytellings sind vor allem Dialogsysteme von übergeordneter Bedeutung. Die in diesem Bereich implementierten Systeme zeichnen sich entweder durch eine starke Fokussierung auf das Sprachverstehen oder auf die Sprachproduktion aus. Eine weitere Charakteristik der oben genannten Dialogsysteme sind ihre Gebundenheit an eine einzelne Domäne und die Verwendung a priori definierter Skripte zur Dialogablaufsteuerung. Die im Deep Map Projekt implementierte Mensch-Maschinen Schnittstelle umfasst mehrere Domänen (z.B. touristische, raumbezogene oder historisch-geographische Informationen) und ist auf Erweiterung der Domänen ausgelegt. Ebenso ist durch das unten Beschriebene *Query and Answer Translator*- System (QUATRA) die Dialogablaufsteuerung durch ein

verteiltes System von Agenten realisiert, welches weder zentrale Steuerungskomponenten noch statische Ablaufpläne aufweist.

Darüber hinaus wurde im Deep Map Projekt gleichermaßen auf die Ausarbeitung des Sprachverstehen und auf die Qualität der Sprachproduktion geachtet. Andere gleichmäßig elaborierte *end-to-end* Systeme befinden sich noch im Forschungs- und Entwicklungsstadium. In diesem Bereich sind vor allem die Verbundprojekte EMBASSI (www.embassi.de) und SmartKom (www.smartkom.com) zu nennen, welche nicht nur gleichermaßen tiefe Analyse und Generierung zum Ziel haben, sondern ebenfalls mit externen Diensten und heterogenen Applikationen interagieren. Ziel beider Verbundprojekte ist das Verstehen und multi-modalen Inputs und das Generieren multi-modalen Outputs, wobei im SmartKom Projekt die Integration mit Office, PDA und Informationsdiensten, im Gegensatz zur Steuerung von Unterhaltungs- und Haushaltselektronik bei EMBASSI, im Vordergrund steht.

3. Deep Map

Der Deep Map Prototyp umfasst eine Reihe interner Dienste, welche die multimodale Kernfunktionalität des Systems garantieren. Des weiteren werden wir anhand des Beispiels von Hotelreservierungsdiensten die Einbindung externer Dienste und heterogener Datenquellen durch Deep Map beschreiben. Das Gesamtsystem besteht aus mehreren autonomen Komponenten, die über eine Kommunikationsinfrastruktur miteinander verbunden sind. Diese in Abbildung 1 dargestellten Komponenten werden im folgenden kurz vorgestellt.

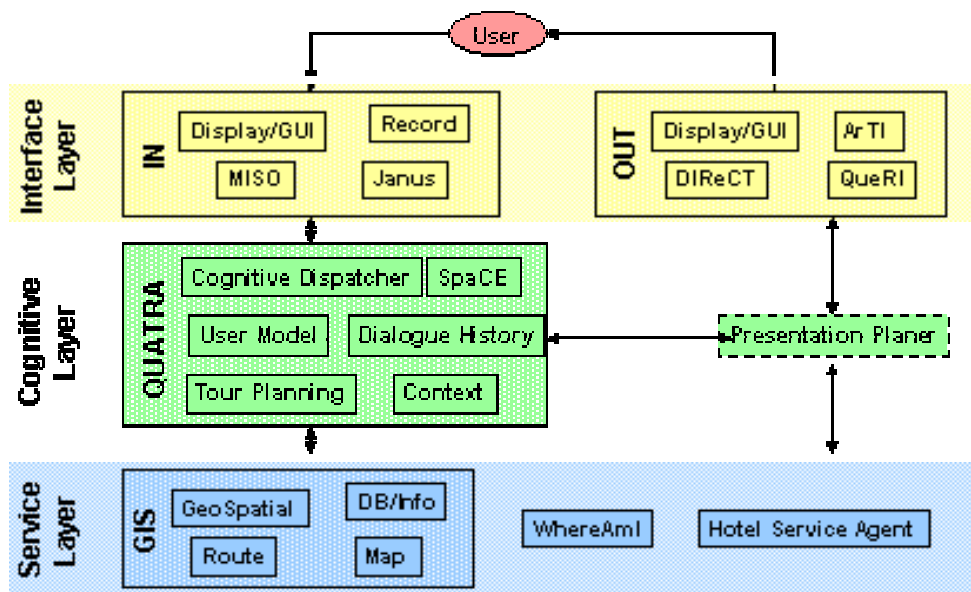


Abbildung 1: Aufbau des Deep Map Systems

Das Deep Map System besteht aus einer Gemeinschaft von Agenten, die zusammenarbeiten, um dem Benutzer intuitiven Zugriff auf die bereitgestellten Dienste zu bieten. Die Agenten verteilen sich auf drei Schichten:

- den *Interface Layer*, wo die direkt mit dem Benutzer interagierenden Agenten angesiedelt sind,

- den *Cognitive Layer*, auf dem die Agenten residieren, die für die Interpretation der Eingaben des Benutzers verantwortlich sind, und die entsprechende Antworten generieren, und
- den *Service Layer*, wo die Agenten zusammengefaßt sind, die das auf dem Cognitive Layer benötigte Wissen und die Services, auf die der Benutzer zugreifen will, zur Verfügung stellen.

Um multimodal mit dem Benutzer interagieren zu können, arbeiten auf dem Interface Layer verschiedene Agenten zusammen. Auf der Eingabeseite (IN) sind dies zum einen ein *Display/GUI-Agent*, der dem Benutzer eine graphische Oberfläche zur Verfügung stellt. Die Agenten *Record*, *Janus* und *MISO* dienen der Spracheingabe und werden im Abschnitt 3.1 näher erläutert. Auf der Ausgabeseite (OUT) kooperieren ebenfalls vier Agenten, um dem Benutzer die Erwidern des Systems multimodal und situationsgerecht präsentieren zu können. Hier sind zunächst *DIRECT* und *QueRI*, die für die Realisierung von Äußerungen bzw. Fragen in einer konkreten Sprache verantwortlich sind. Dabei überführen sie die interne (sprachunabhängige) Beschreibung der Systemausgabe anhand verschiedener Regeln in natürlichsprachliche Sätze. Diese können dann entweder akustisch realisiert werden – was das Aufgabengebiet von *ArTI* ist – oder aber textuell angezeigt werden. Letzteres fällt in den Zuständigkeitsbereich des auch auf der Eingabeseite tätigen *Display/GUI-Agenten*, der daneben auch graphische Ausgaben wie beispielsweise Kartendarstellungen verwaltet.

Die Agenten des Cognitive Layer lassen sich ebenfalls in zwei große Gruppen unterteilen, nämlich solche, die für das Verstehen der Eingabe und die Erzeugung der Ausgabe zuständig sind, und solche, die planen, wie die Ergebnisse dem Benutzer vermittelt werden. Unter die erste Gruppe (QUATRA) fällt der *Cognitive Dispatcher*, der die (auf dem Interface Layer erzeugte) Eingaberepräsentation analysiert, in Teilaufgaben zerlegt und diese an die entsprechenden Agenten weiterleitet. Empfänger solcher (Teil-)Aufgaben können beispielsweise der *SpaCE*- oder der *Tour Planning-Agent* sein. Ersterer ist für die Auflösung räumlicher Bezüge zuständig, letzterer für die Planung an den Benutzer angepaßter Touren. Daneben enthält QUATRA noch Agenten, die Inferenz- und Informationsdienste anbieten. Dazu gehören der *User Model-Agent*, der Eigenschaften und Interessen des Benutzers verwaltet, der *Dialogue History-Agent*, der den Verlauf des Dialogs protokolliert, sowie der *Context-Agent*, der kontextuelle Informationen wie beispielsweise Wetter, Systemressourcen und die aktuelle Zielsprache verwaltet.

Um dem Benutzer die vom System generierten Antworten adäquat zu präsentieren, muss die Präsentation nicht nur auf der Realisierungsebene (Interface Layer) sondern auch auf einer höheren Ebene geplant werden. Der *Presentation Planer-Agent* übernimmt diese Aufgabe auf dem Cognitive Layer. Er trifft Entscheidungen wie z.B. welche Information in welchem Medium/Modus übermittelt werden soll, und stellt sicher, dass die Bezüge zwischen den einzelnen Teilen der Präsentation erhalten bleiben, beispielsweise bei multimodalen Referenzen.

Auf dem Knowledge Layer werden die Informationen und Dienste zur Verfügung gestellt, auf die der Benutzer zugreifen möchte und die auch von anderen Agenten benötigt werden, um die Anfragen des Nutzers bearbeiten zu können. Den größten Anteil stellen (domänenbedingt) die Agenten des Geoinformationssystems (GIS). Darunter fallen der *GeoSpatial-Agent*, der ein geometrisches Modell der Welt vorhält, der *DB/Info-Agent*, der nicht-geometrische Information mit dem geometrischen Weltmodell verknüpft, dem *Route-Agenten*, der Wege berechnet, und dem *Map-Agenten*, der für die dynamische Generierung von Karten zuständig ist. Zur Verwendung in einem mobilen Szenario gibt es den *WhereAmI-Agenten*, der anhand von GPS-Daten die aktuelle Position des Benutzers bestimmt. Der

HotelService-Agent schließlich ist für die Bereitstellung von Hoteldaten sowie das Finden und Buchen von Hotels zuständig.

3.1 Interne Dienste

Im folgenden Kapitel werden die Komponenten des Deep Map Systems näher erläutert, die für die Beispielanwendung Hotelreservierung eine besondere Aufgabe übernehmen. Dabei handelt es sich zum einen um Dienste zur multimodalen Dialoggestaltung (Spracheingabe, Sprachverstehen, Dialoggestaltung), zum anderen um Dienste, die eine Wissensbasis zur Beantwortung der Benutzeranfragen bereitstellen (Raumkognition, *HotelService*, GIS).

Kommunikation:

Kommunikationsschicht von Deep Map wird durch Benutzung des am EML entwickelten *Java Agent Framework* als *Middleware* realisiert. Dieses ermöglicht die Kommunikation zwischen *HotelService*, GIS und User Interface (Query-Engine und Visualisierung) durch die Weiterleitung von Objekten. Dazu wird entweder ein Broadcast- oder ein direkter Adressierungsmechanismus verwendet, welche beide auf dem Grundprinzip des Austauschs serialisierter Java-Objekte basieren.

Spracheingabe:

Das natürlichsprachliche Interface von Deep Map umfasst spontansprachliche Spracherkennung, welche mit Hilfe des Janus Recognition Toolkits 3.24 ein sprecherunabhängiges Vokabular von ~5000 Wörtern und ein für touristische Anfragen entwickeltes Sprachmodell zur Verfügung stellt [Wosz97]. Die Resultate der Spracherkennung werden die einen syntaktischen und semantischen Parser [GaWa98] in sprachunabhängige typed-feature structures überführt, welche neben zusätzlichem Input durch Interaktion des Benutzers mit dem graphischen user interface den Input für die kognitiven und dialogische Module darstellen.

Sprachverstehen:

Das Verstehen der Benutzerintention erfordert die Extraktion der linguistischen Informationen einer Äußerung des Benutzers. Dadurch ist aber noch kein vollständiges Verstehen gewährleistet: Selbst eine scheinbar verständliche Äußerung wie „Jetzt will ich zur Peterskirche.“ braucht für eine sinnvolle Beantwortung zusätzliche Information, z.B. wo der Benutzer im Augenblick ist oder welches Objekt in der realen Welt mit „Peterskirche“ gemeint ist. Die Berechnung sinnvoller Entgegnungen erfolgt mit Hilfe von Deep Map Wissensbasen, z.B. der Datenbank TouristBase oder des geographischen Informationssystems DeepMap-GIS, und durch Anbindung externer Dienste (siehe folgendes Kapitel). Darüber hinaus sind hierbei auch Fragen der Dialoggestaltung und der Raumkognition von zentraler Bedeutung, und zwar sowohl beim Verstehen als auch bei der Beantwortung der Benutzeranfragen.

Raumkognition:

Selbst wenn in obigem Beispiel die Positionen des Nutzers und der Kirche bekannt sind und ein Weg zwischen beiden berechnet wurde, ist dies nur bedingt ausreichend: erfolgt die Kommunikation verbal, muß die interne Repräsentation in sprachliche Konzepte überführt

⁴ Die Erkennungs- und Parsing Komponenten wurden entwickelt und eingebunden in Kooperation mit dem ISL der Universität Karlsruhe und der CMU Pittsburgh.

werden. Dazu müssen räumliche Relationen [Gapp94] evaluiert werden, Referenzrahmen etabliert werden und die Gesamtstrecke in verbalisierbare Segmente zerlegt werden [KrBI99]. Ferner ist es notwendig geeignete Bezugsobjekte für die Beschreibung der Route zu identifizieren. Umgekehrt kann ein System eine Frage wie „Was ist das für ein Turm links von dem großen Haus?“ auch nach einer rein linguistischen Analyse nicht ohne weiteres beantworten. Zunächst muss das Bezugsobjekt identifiziert werden, danach ein Referenzrahmen bestimmt werden (zur Berechnung des links gelegenen Raums) und schließlich das Zielobjekt gefunden werden⁵. Danach erst kann die Frage mit einer Anfrage an die Datenbank beantwortet werden. Zur Lösung derartiger Aufgaben (Wegbeschreibung, Objektidentifikation, räumliches Schließen, Übersetzung zwischen Sprache und interner Repräsentation) verfügt Deep Map über einen Raumkognitionsagenten (SpaCE).

Dialoggestaltung:

Sinnvolle Interaktion mit dem Benutzer und komplexere Dienstleitungen werden erst durch dialogische Fähigkeiten eines Systems möglich. Hierbei dienen manche Agenten als semi-passive Wissensquellen, wie z.B. die Dialog-Historie, das Benutzer-Modell oder der Kontext-Manager, welche Informationen über den Diskursverlauf, Benutzerinteressen und –wissen und die außersprachliche Situation den interessierten Agenten mittels Ein- und Ausleseverfahren sowie auch über Subskriptionsmechanismen zur Verfügung stellen. Weitere Agenten benutzen diese Informationen zur Adressierung, Verarbeitung und Beantwortung der Anfragen des Benutzers oder anderer Agenten. Am Beispiel eines Hotelreservierungsdialoges werden wir Einzelheiten dieses Prozesses in Kapitel 3.3 beschreiben.

3.2. Externe Dienste

Hotelservice

Die Komponente Hotelservice integriert verschiedene externe Hotelbuchungssysteme und ermöglicht eine einheitliche Sicht auf die Hotelinformationen im Internet. Das Ziel dieser Komponente besteht im Wesentlichen darin, vorhandene Online-Buchungssysteme aus einer einheitlichen Sicht zu nutzen. Dies ermöglicht einen einheitlichen Zugriff auf verschiedene Hotelanbieter im Internet, ohne die Suchanfragen für jedes System explizit zu formulieren. Durch die Integration verschiedener Systeme ist neben einer vereinfachten Suche auch ein Preisvergleich unterschiedlicher Anbieter einfach möglich. Die Komponente Hotelservice besteht aus drei Teilen:

- Zur Beschreibung des Inhalts eines Buchungssystems dienen *Metadaten*. Da die Suchparameter nicht durch ein festes Formular eingeschränkt sind, sind diese Metadaten für die Entscheidung notwendig, ob bestimmte (freie) Suchparameter überhaupt im entsprechenden Buchungssystem enthalten sind.
- Der Abgleich, welche Attribute in welchen Buchungssystemen zur Verfügung stehen, kann durch einen semantischen Übersetzer verbessert werden. Dieser sorgt dafür, daß Attribute gleicher Bedeutung, die in unterschiedlichen Buchungssystemen verschieden benannt sind, als Suchparameter berücksichtigt werden. Der Einsatz eines *Thesaurus* ermöglicht ein 1:1 Mapping von unterschiedlichen Begriffen gleicher Semantik in verschiedenen Buchungssystemen.
- Eine Schnittstellenbeschreibung der unterschiedlichen Buchungssysteme wird im *Interface Repository* vorgehalten. Diese Schnittstellenbeschreibung stellt einen flexiblen Zugriff auf die integrierten Buchungssysteme sicher.

⁵ Die Identifikation findet jeweils in der internen Datenbank bzw. dem GIS statt.

Zur Kommunikation zwischen den einzelnen Komponenten wird ein gemeinsamer Satz von Objekten verwendet. Die Suchanfrage des Nutzers wird vom User Interface zum Hotelservice über das *HotelQuery* Objekt transportiert. Ein *HotelQuery* Objekt beinhaltet häufig verwendete Suchparameter wie Preis, An- und Abreisedatum und Komfort des Hotels als Attribute. Die Suche nach freien Parametern wird durch freie Attribute unterstützt. Der *Hotelservice*-Agent versucht durch Nutzung von Metadaten und einem semantischen Übersetzer entsprechende Attribute in den bekannten Buchungssystemen zu finden. Mit Hilfe der Metadaten kann ermittelt werden, ob bestimmte Informationen überhaupt in einem Hotelbuchungssystem vorhanden sind. Der semantische Übersetzer hilft bei dieser Prüfung, indem er zum einen zu suchenden Attribute Begriffe gleicher Bedeutung liefert.

Abbildung 2 zeigt beispielhaft den Zugriff auf das Hotelbuchungssystem HRS in Köln. Dieses System verfügt über eine offene Remote Procedure Call (RPC) Schnittstelle [Beut98].

Bei Nutzung dieser Schnittstelle können (insbesondere bei „unscharfer“ Suche) erhebliche Datenvolumina auftreten, weil bei jeder Abfrage das komplette Suchergebnis übertragen wird. Aus diesem Grund bietet die HRS-Schnittstelle zwei Hotelsichten an: eine minimale Hotelbeschreibung mit den wichtigsten Hotelparametern, (z.B. Preis, Adresse) und eine ausführliche Hotelbeschreibung, die alle vorhandenen Attribute enthält.

Zur Hotelsuche ist die RPC-Schnittstelle im Internet nur mit der minimalen Hotelbeschreibung einsetzbar. Damit kann aber nur ein eingeschränkter Satz an Kriterien zur Auswahl eines Hotels herangezogen werden. Um alle vorhandenen Attribute des Buchungssystems zur Suche nutzen zu können, ohne das zu übertragene Datenvolumen unnötig zu erhöhen, wurde eine CORBA-Wrapper für das HRS implementiert. Über diese Schnittstelle kann gezielt auf einzelne Attribute zugegriffen werden, d.h. es ist es möglich

gezielt auszuwählen, welche Daten als Ergebnis der Suchanfrage an den Client zurückgeliefert werden. Die notwendige CORBA/IDL Schnittstellendefinition wird zusätzlich im Interface Repository des Hotelservice verwendet.

Geoinformationssystem (GIS)

Das GIS unterstützt den Hotelservice bei der Auswertung räumlicher Suchanfragen. Die meisten Hotelbuchungssysteme enthalten nur wenig Angaben über die Lage des Hotels in Bezug auf andere Lokalitäten einer Stadt. Durch den Einsatz eines GIS können entsprechende raumbezogene Anfragen sehr viel flexibler gestellt und beantwortet werden. Weiterhin dient das GIS zur Georeferenzierung der einzelnen Hotels, soweit diese nicht schon im eigentlichen Buchungssystem durchgeführt wurde.

Zur Visualisierung der Suchergebnisse wird Kartenmaterial und ggfs. auch ein digitales Geländemodell vom GIS verwaltet und die Lage der gefundenen Hotels darin geographisch korrekt dargestellt.

3.3. Ein Beispiel zur Hotelreservierung

Die Aufgaben der einzelnen DeepMap Agenten werden im folgenden anhand eines kleinen Szenarios beispielhaft dargestellt. Herr Duck aus Entenhausen reist mit der Bahn nach Heidelberg. Am Bahnhof angekommen, benötigt er erst einmal ein Hotel. Reisen ist anstrengend, also soll das Hotel nicht zu weit entfernt sein. Da sich Herr Duck in Heidelberg leider überhaupt nicht auskennt, beauftragt er seine Deep Map mit der Suche nach einer geeigneten Unterkunft: „Ich suche ein Hotel in der Nähe“

Durch diese Anfrage werden verschiedene Agenten des Deep Map Systems aktiv. Abb 3 zeigt die an der Beantwortung dieser Suchanfrage beteiligten Agenten mit Hilfe eines UML-Sequenzdiagramms [FoSc97].

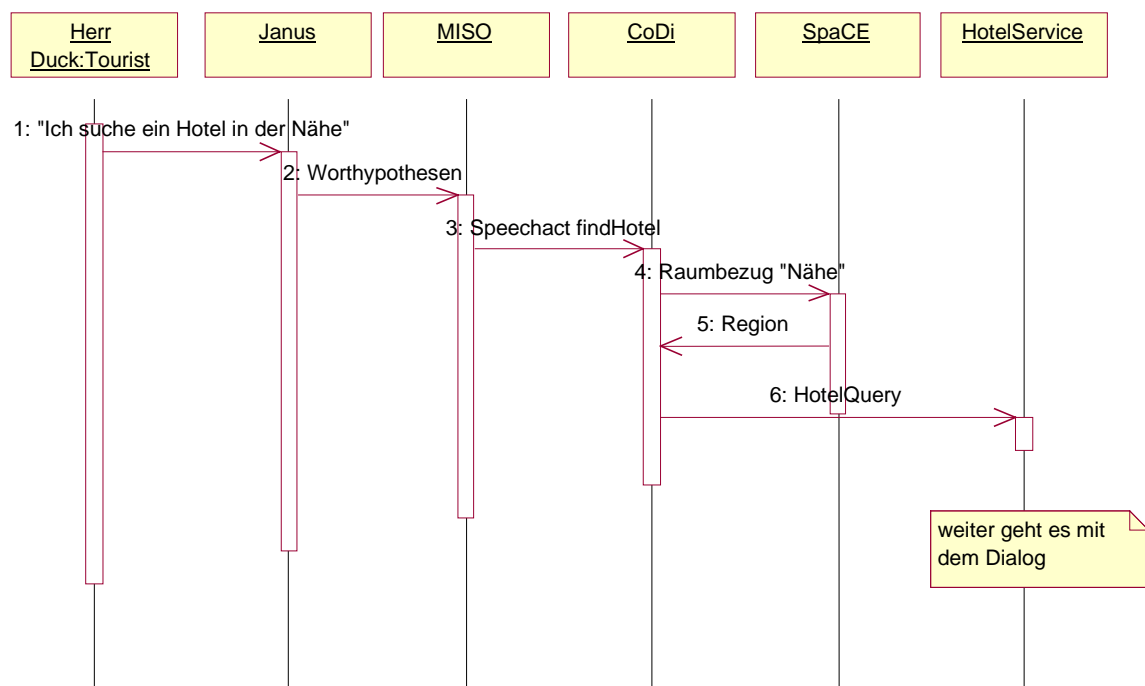


Abb. 3a: Beispieldialog zur Hotelbuchung, dargestellt als Sequenzdiagramm

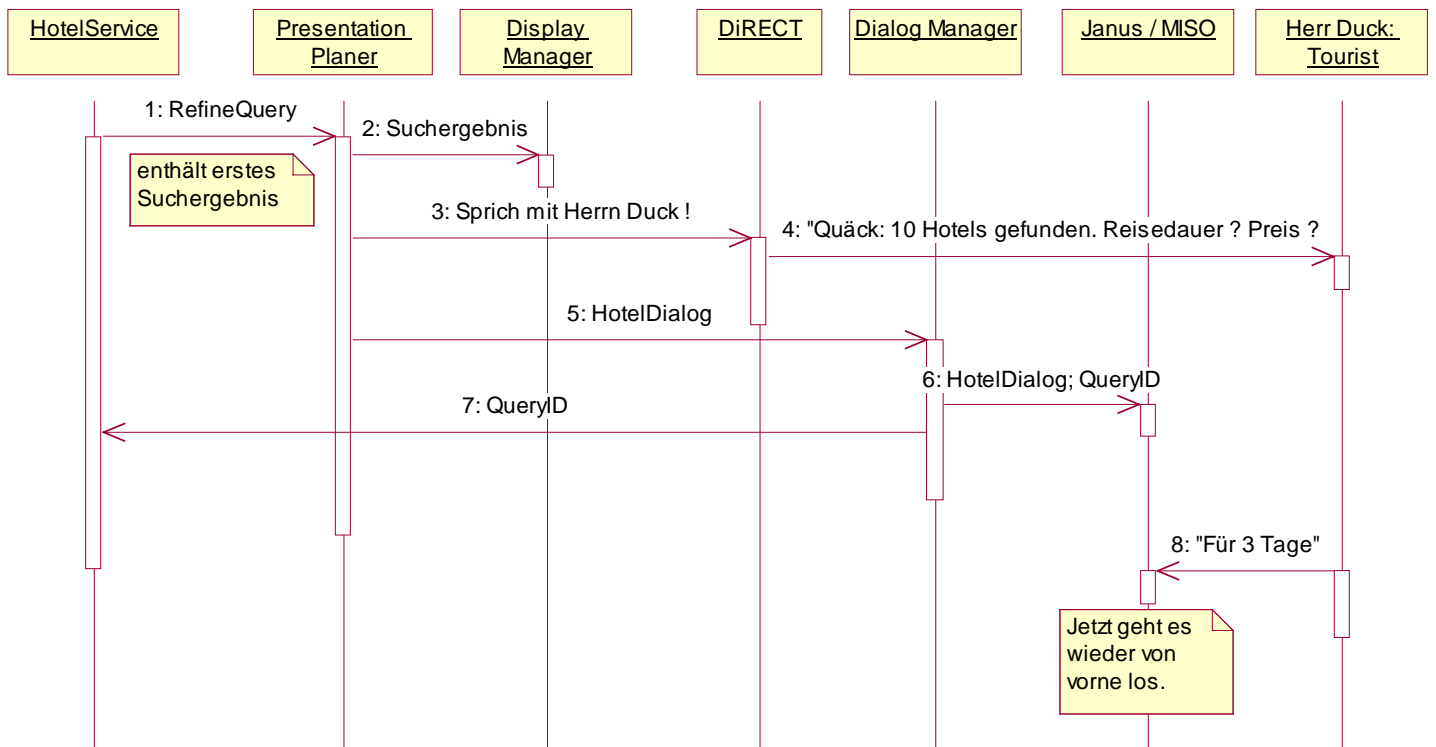


Abb. 3b: Beispieldialog zur Hotelbuchung, dargestellt als Sequenzdiagramm

Der Spracherkenner *Janus* setzt die akustischen Signale in eine für das System verständliche Form um. Dieser sog. Worthypothesengraph wird von *MISO* in eine strukturierte Form gebracht, die von anderen Agenten weiterverarbeitet werden kann. Im vorliegenden Beispiel wird aus der natürlichsprachlichen Eingabe „Ich suche ein Hotel in der Nähe“ der im System definierte *speechact findhotel*.

Dieser Sprachakt wird vom *Cognitive Dispatcher* (CoDi) als Hotelanfrage erkannt und analysiert. CoDi stellt fest, daß es sich bei der Formulierung „in der Nähe“ um einen nicht aufgelösten räumlichen Teilaspekt handelt. Der räumliche und persönliche Kontext spielen eine entscheidende Rolle bei der Auflösung des Begriffs „Nähe“. Ist der Reisende mit dem Auto unterwegs, beschreibt „Nähe“ beispielsweise eine wesentlich größere räumliche Region, als wenn er das Hotel zu Fuß erreichen will.

Der räumliche Teilaspekt wird vom SpaCE-Agenten mit Hilfe des Kontextes, des Benutzermodells und des GIS aufgelöst. Neben der Bedeutung von „Nähe“ muß auch ein Bezugsobjekt abgeleitet werden. Im vorliegenden Beispiel ist das Bezugsobjekt die aktuelle Position des Benutzers, die *WhereAmI* liefert. Aus dem Kontext konnte SpaCE entnehmen, daß der Reisende mit der Bahn unterwegs ist und liefert als Ergebnis entsprechend ein Polygon einer Region, die vom momentanen Standort zu Fuß gut zu erreichen ist.

Nachdem der Begriff „in der Nähe“ ausgelöst wurde, kann die Hotelanfrage an den *HotelService* weitergeleitet werden. Dort werden entsprechende Hotels gesucht. Auch der *HotelService* bedient sich dabei des Kontextes, da nicht näher spezifiziert ist, für welchen Zeitraum das Hotel gebucht werden soll. Aus dem Kontext wird ersichtlich, das Herr Duck

allein unterwegs ist und das Hotel sofort benötigt. Allerdings sind weder Informationen über das geplante Abreisedatum noch über das Reisebudget vorhanden. Der HotelService findet in unserem Beispiel eine Liste von 10 Hotels, die für diese Nacht ein freies Zimmer haben.

Diese Hotelliste wird mit dem Hinweis auf die fehlenden Angaben *Abreisedatum* und *Preisvorstellung* an den *Presentation Planer* übermittelt. Der *Presentation Planer* entscheidet aufgrund des Kontextes, die gefundenen Hotels zur Visualisierung an den *Display Manager* zu übergeben. Der *Display Manager* stellt die Hotels in einem dreidimensionalen Stadtplan dar. Dieses Modell zeigt die Topographie der Stadt und vermittelt einen Eindruck über die Lage des Hotels. Für nähere Informationen zur Visualisierung der Hoteldaten siehe [Coor99]. Gleichzeitig wird über *DIReCT* ein natürlichsprachlicher Dialog initialisiert, die nach den fehlenden Angaben Abreisedatum und Preis fragt, beispielsweise: „10 Hotels in Ihrer Nähe haben für heute ein freies Zimmer. Nähere Angaben zu den Hotels sehen Sie auf dem Display. Wie lange wollen Sie in Heidelberg bleiben ? Wieviel wollen Sie für Ihr Hotelzimmer ausgeben ?“

Um die Spracherkennung zu optimieren, wird den Erkennernmodulen *Janus* und *MISO* mitgeteilt, daß es sich bei der nächsten natürlichsprachlichen Eingabe um einen Hoteldialog handelt. Die Erkennernmodule können dadurch ihr Sprachmodell entsprechend anpassen. Der Dialog wird solange durchgeführt, bis der Reisende ein Hotel gebucht hat oder den Dialog abbricht, um einige Hotels, die das System vorschlägt, persönlich anzuschauen.



Abb. 4: mobile Variante des Deep Map Systems auf dem wearable PC MA IV der Fa. Xybernaut

4. Zusammenfassung

Um *Digital Storytelling* in der Kommunikation zwischen Menschen und Maschinen zu ermöglichen müssen die entsprechenden Systeme sowohl über den Zugang zu den benötigten Inhalten, welche im „digitalen Gespräch“ vermittelt werden sollen, wie auch über geeignete Kommunikationsschnittstellen verfügen. Wir haben mit dem elektronischen Touristenführer, Deep Map, ein System vorgestellt, welches eine linguistische und visuelle Schnittstelle liefert, die neben dem Verstehen und dem Management multi-modaler Interaktionen auch deren Verarbeitung und Beantwortung steuert und dabei auf die benötigten Wissensquellen zugreifen kann. Im Gegensatz zu kommerziellen Systemen ist die natürlichsprachliche Interaktion im Deep Map System sprecherunabhängig und muss daher nicht zeitintensiv trainiert werden. Dies ist eine notwendige Voraussetzung für die Akzeptanz von Mensch-Maschine-Kommunikation mit natürlicher Sprache. Die Sprecherunabhängigkeit wird erreicht, indem ein mittelgrosses Vokabular aus der Tourismusdomäne Verwendung findet.

Verschiedene Elemente des vorgestellten Systems sind über die Anwendung der elektronischen Touristenführers hinaus für verschiedene multi-modal interagierende und adaptive Systeme einsetzbar. Für Anwendung ausserhalb des Tourismus ist es jedoch unbedingt notwendig, einen geeigneten anwendungsspezifischen Wortschatz zu entwickeln, um eine Sprecherunabhängigkeit zu gewährleisten.

Ein grosses Hindernis beim Einsatz von sprecherunabhängigen Systemen ist der eingeschränkte Sprachumfang. Ein vielversprechender Ansatz, dies zu überwinden, ist eine dynamische Anpassung des Wortschatzes bei Kontextwechsel des Benutzers oder des Dialogs. Hierzu sind geeignete statistische Modelle und Lexika zu erheben, die eine Vorhersage über das zukünftig verwendete Vokabular erlauben. So ist beispielsweise bei einem Dialog zur Hotelbuchung zu erwarten, dass bei einem gerade erfolgten Angebot von 10 verschiedenen Hotels einer dieser Hotelnamen vom Benutzer erwähnt wird. Sämtliche Hotelnamen müssen also dynamisch in den Wortschatz aufgenommen werden.

Weiterführende interessante Forschungsarbeiten sind im Bereich der Präsentationsplanung gegeben. Hier drängt sich die Frage nach Entscheidungsregeln auf, welche Informationen visuell präsentiert werden sollen und welche in natürlicher Sprache. Dies hängt sehr stark vom Typ der Information ab. So kann zum Beispiel eine Karte visuell wesentlich schneller erfasst werden als über einen gesprochenen Text. Die Eingabe ist jedoch in vielen Fällen über Sprache wesentlich effektiver, da es der Mensch gewohnt ist, über Sprache zu kommunizieren.

5. Danksagung

Die Verwendung des Hotelreservierungssystems HRS erfolgte mit freundlicher Genehmigung und Unterstützung der Robert Ragge GmbH, Köln.

Die Autoren danken der Klaus-Tschira-Stiftung für die Förderung der Projekte. Teile der hier vorgestellten Arbeiten wurden im Rahmen des vom BMBF Leitprojektes SmartKom gefördert (Fördernummern 01IL904D2).

Literatur:

- [Alle96] Allen, J.F., B. Miller, E. Ringger and T. Sikorski [A Robust System for Natural Spoken Dialogue](#), Proc. 34th Meeting of the Assoc. for Computational Linguistics, 1996.
- [Alsh92] Alshawi, H., *The Core Language Engine*. MIT Press. 1992
- [Aust95] Aust, H., Oerder, M., Siede, F. and Stainbiss, V. *A spoken language enquiry system for automatic train timetable information*. Philips Journal of Research, 49(4), 399-418. 1995
- [Beut98] Beutler, G. *Dokumentation der HRS RPC Schnittstelle*, Hotel Reservation Service (HRS), Robert Rague GmbH, Köln, Stand 10.07.1998.
- [Bush45] Bush, V. *As We May Think* The Atlantic Monthly, July 1945
- [FoSc97] Fowler, M. and Scott, K. *UML Distilled. Applying the Standard Object Modeling Language*, Addison Wesley, 1997, ISBN 0-201-32563-2
- [Gall98] Gallwitz, F., Aretoulaki, M., Boros, M., Haas, J., Harbeck, S., Huber, R. and Niemann, H., *The Erlangen Spoken Dialogue System EVAR: A State-of-the-Art Information Retrieval System*. In Proceedings of the 1998 International Symposium on Spoken Dialogue (ISSD-98), Invited Talk, 1998, Sydney, Australia
- [Gapp94] Gapp, K.-P., *Basic Meanings of Spatial Relations: Computation and Evaluation in 3D space*, Proceedings of AAAI-94, Seattle, WA, USA 1994
- [GaWa98] Gavalda, M. und Waibel A. *Growing Semantic Grammars*. Proceedings of ACL/Coling. Montreal, Canada. 1998.
- [KrBl99] Kray, C. und Blocher, A., *Modeling the Basic Meanings of Path Relations*, in Kaufmann, M (Hrsg.) Proceedings of the 16th IJCAI, San Francisco, CA, USA, 1999
- [Kuhl90] Kuhlen, R. *Zum Stand pragmatischer Forschung in der Informationswissenschaft*. In J. Herget & R. Kuhlen (Eds.): *Pragmatische Aspekte beim Entwurf und Betrieb von Informationssystemen*. Proceedings des 1. Internationalen Symposiums für Informationswissenschaft, Konstanz. Universitätsverlag, 1990
- [Wahl97] Wahlster, W., *VERBMOBIL: Erkennung, Analyse, Transfer, Generierung und Synthese von Spontansprache*. DFKI GmbH. Juni 1997
- [Waib96] Waibel, A., *Interactive Translation of conversational Speech*. IEEE Computer, 29(7), July 1996
- [Wosz97] Woszczyzna, M., *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition Dissertation*, Universität Karlsruhe. 1997