



Beratungsdialog im WWW: Ein konversationales Modell und seine Implementierung

Hagen Engelmann¹, Marcello L'Abbate², Ulrich Thiel²

Fraunhofer Institut
Integrierte Publikations- und Informationssysteme
(IPSI)

Dolivostrasse 15
D-64293 Darmstadt

¹hagen.engelmann@epost.de

²{labbate,thiel}@ipsi.fhg.de

Zusammenfassung

Diese Arbeit beschäftigt sich mit der Entwicklung von Beratungsagenten im WWW. Aus Erfahrungen im EU-Projekt VIP-Advisor wurde ein allgemeines Dialogmodell für Beratungen entwickelt, welches als Grundlage für die Gestaltung eines Beratungsagenten dient. Dessen Dialogfähigkeit wurde durch Methoden des Case-Based-Reasoning (CBR) erweitert, sodass das System nicht nur den vorgesehenen Plan einer Beratung ausführt, sondern auf spezifische Probleme während des Dialogs flexibel reagieren kann.

Abstract

The paper deals with the development of advice-giving agents on the WWW. Based on work in the EU project VIP-Advisor, a general dialogue model was developed for advice sessions which formed the basis of the realisation of an advice-giving agent. Its dialogue capability was extended using case-based reasoning (CBR) methods such that the system was not only able to execute the intended dialogue plan of an advice session, but also able to react flexibly to specific problems arising during the dialogue.

1 Einführung

Beratung ist ein Bestandteil des Lebens, der in privaten, geschäftlichen und beruflichen Bereichen eine Rolle spielt. Berater sind das Ergebnis unserer



arbeitsteiligen Gesellschaft, deren Kennzeichen die Spezialisierung ihrer Mitglieder auf verschiedene Arbeitsfelder ist. Der Berater ist eine Person, die einer anderen Person, dem Klienten, bei der Bewältigung eines Problems behilflich ist. Er liefert ihm dazu Hilfestellungen für die Entscheidungsfindung, häufig mit einer kommerziellen Zielsetzung [Jon00]. Bei komplexen Produkten oder Dienstleistungen ist dies oft eine entscheidende Voraussetzung für die erfolgreiche Kaufentscheidung. Häufig wird dabei im Interesse einer nachhaltigen Kundenbindung die Beratung unabhängig vom Produktangebot durchgeführt. Im Zuge der Verlagerung von Kundenkontakten vom persönlichen Gespräch zu elektronisch vermittelten Informationen im WWW stellt sich die Frage, ob und wie weit es möglich ist, die Beratungsdienstleistung so zu automatisieren, dass sie den Kunden z.B. auf der Webseite eines Unternehmens angeboten werden kann. Beratungsdialoge sind wesentlich weniger strukturiert als z.B. Kaufinteraktionen, die inzwischen standardmäßig mit graphischen Schnittstellen realisiert werden. Um eine stärkere Flexibilität zu erreichen, bietet sich daher eine Interaktion in natürlicher Sprache an.

Die computerlinguistischen Systeme für natürlichsprachliche Dialoge zwischen Mensch und Maschine sind oft sehr komplex (siehe z.B. [Hoe99]). Lösungen wie z.B. in VERMOBIL¹ betreiben einen großen Aufwand, um Benutzereingaben linguistisch zu analysieren. Für einen Beratungsagenten, der viele Benutzer gleichzeitig bedienen und möglichst auf normalen Webserver laufen soll, ist solch eine aufwendige Analyse kaum realisierbar. Hier bieten sich Konversationale Agenten (oder auch „Chatterbots“) an. Diese wurden bisher hauptsächlich zum Zweck der Unterhaltung und Einführung auf Web-Seiten eingesetzt, hatten aber keinen Nutzen in ernsthaften Dialogszenarien, bei denen ein intelligentes und angepasstes Antwortverhalten notwendig ist². Ein menschlicher Experte ist in der Lage, gezielt auf Klärungsfragen einzugehen und sachgemäß auf verschiedenartige Anfragen zu reagieren. Auch mit den zuvor erwähnten komplexen Technologien ist es nicht möglich, das menschliche Expertenverhalten in einem virtuellen Berater komplett abzubilden. In diesem Beitrag erläutern wir das Potential der verfügbaren, unter ökonomischen Bedingungen einsetzbaren Verfahren am Beispiel einer experimentellen Implementierung eines Beratungsagenten, der potentielle Kunden einer Versicherungsgesellschaft bei einer Risikoanalyse berät. Dies wird zum einen durch ein ausgefeiltes Risikofaktorenmodell ermöglicht, dessen Parameter im Dialog mit dem

¹ <http://verbmobil.dfki.de/www.html>

² Kommerzielle Ansätze Chatterbots im E-Commerce zu nutzen gibt es z.B. durch Artificial-Life (www.artificial-life.com) oder Botizen (www.botizen.com).

Kunden zu ermitteln sind, zum anderen durch ein komplexes Modell des potentiellen Verlaufs von Beratungsgesprächen, das dem System eine kooperative Planung des Dialoges erlaubt. Zusätzlich erweitern wir die Fähigkeiten des Chatterbots durch Methoden des CBR.

Im Folgenden präzisieren wir zunächst den der Arbeit zugrunde liegenden Beratungsbegriff und skizzieren das Projekt, in dessen Kontext das danach vorgestellte Dialogmodell entwickelt wurde. Zum Abschluss zeigen wird an einem Beispiel, wie das Modell und das CBR-Verfahren bei der Gestaltung des Beratungsagenten Einsatz finden.

2 Verschiedene Beratungsformen

Anhand des Klientenproblems und der vom Berater genutzten Lösungsstrategie lassen sich vier Beratungsformen unterscheiden [Wasch98]:

Ist der Berater Informationslieferant zu einem Sachverhalt [Wal95], spricht man von einer **Gutachterberatung**. Hierbei soll ein Informationsdefizit des Klienten behoben werden, um ihm eine selbstständige Entscheidung zu ermöglichen. Der Berater besitzt Wissen über und Zugang zu spezifischen Informationsquellen. Dazu nutzt er entweder Methodenkenntnisse und disziplinäres Wissen, um spezifische Informationen zu beschaffen [Hill90 & Jung99] oder er sammelt und verkauft Informationen bzw. vermittelt Kontakte (z.B. Makler). Da der Klient meist die Informationen auch selbst beschaffen könnte, minimiert er nur seinen Zeit- und Suchaufwand [Sti61]. Diese Form der Beratung lässt sich mit dem Begriff des Information Agent³ in Verbindung bringen. Diese handeln ebenfalls im Auftrag eines Anwenders oder anderer Agenten. Sie sammeln, vermitteln und verwalten gewünschte Informationen, indem sie Zugriff auf ein oder mehrere, verteilte und heterogene Datenquellen nehmen.

Bei einer **Expertenberatung** werden Berater zur fachlichen bzw. inhaltlichen Hilfe herangezogen [Wal95]. Sie sind Problemlöser, die dem Klienten konkrete Vorschläge liefern. Der Berater qualifiziert sich durch einen Wissensvorsprung bzw. sein Expertenwissen. Er hat eine systematische Herangehensweise an die Problemanalyse, ein breites Methodenwissen und umfangreiche Erfahrungen mit ähnlichen Problemstellungen. Außerdem kann er oft eine andere Sichtweise auf die Problemstellung liefern und seinem Klienten eine erweiterte Informationsbasis zur Verfügung stellen, indem er

³ Siehe dazu www.dbgroup.unimo.it/IIA/briefintroduction.html.

systematisch externe Quellen erfasst und analysiert [Kime79]. Ist das Wissen formal repräsentierbar, lässt sich diese Beratungsform zum Teil mit Expertensystemen nachstellen [Fei92].

Die zwei bisher genannten klassischen Beratungsformen sind am häufigsten anzutreffen, während andere, wie die Prozess- und systemische Beratung nur 10,1 % ausmachen [Wasch98]. Bei der **Prozessberatung** unterstützt der Berater den Klienten in seiner Vorgehensweise beim Lösungsprozess, um dessen Problemlösungsfähigkeit zu verbessern [Wal95]. Durch die **systemischen Beratung** soll dem Klienten beim Erarbeiten einer angemessenen Problemsicht geholfen werden [Kvo93]. Der in diesem Beitrag erläuterte Agent soll als Informationslieferant dienen und eine fachliche Beratung der Kunden durchführen. Bei seiner Gestaltung kann daher auf Erfahrungen mit den oben genannten Techniken zurückgegriffen werden.

3 Das VIP-Advisor Projekt

Ziel des EU-geförderten Projektes VIP-Advisor (IST-2001-32440)⁴ ist die Entwicklung eines lebensecht erscheinenden, virtuellen Agenten als neuartiges Interface für Online-Beratungssysteme von Versicherungen und Banken. Der hier beschriebene Prototyp agiert im Bereich des Risikomanagements für kleine und mittelgroße Unternehmen (SMEs). Die Beratungsfunktionalität kann aber auch für die Abwicklung generischer Dienstleistungsgeschäfte mit Privatpersonen verwendet werden. In die Benutzerschnittstelle wurden verschiedene Technologien integriert. Durch natürlichsprachliche Dialoge, Spracheingabe und -ausgabe, maschinelle Übersetzung und Visualisierung mit einem 3D-Avatar, kombiniert mit intelligentem Information Retrieval wird eine hohe Benutzerfreundlichkeit angestrebt. Der Dialog mit dem Anwender soll eine zwischenmenschliche Beratungssituation nachstellen, um die Interaktion mit dem Agenten möglichst erwartungskonform zu gestalten. Ziel der Online-Beratung ist es, neue Kunden zu akquirieren, auf Produkte des Unternehmens aufmerksam zu machen und das Vertrauen des bestehenden Kundenstamms zu festigen.

Der im Rahmen von VIP-Advisor realisierte virtuelle Berater begleitet den Benutzer in Dialogform bei der Analyse seiner Risikosituation. Er bietet eine ausführliche Einleitung zu jedem Interaktionsschritt an und reagiert auf

⁴ Projektpartner waren: Winterthur (Schweiz), Pixelpark (Schweiz) AG, Mellon Techn. (Griechenland), Techn. University Madrid (Spanien), Avatar-Me Ltd. (London, UK), linguattec GmbH (Deutschland)

Benutzer-Fragen und -Probleme, indem er personalisierte Antworten bzw. Lösungen ermittelt. Dazu wurde eine Variante des CBR-Verfahrens gewählt [Lea96], bei dem für die Lösung aktueller Probleme vorhandene Lösungsansätze (die Cases) dem gegenwärtigen Dialogszenario angepasst werden. Die Anpassung erfolgt durch Hinzunahme kontextueller Informationen die sich aus einem Benutzerprofil und dem Dialogverlauf ergeben [L'Ab03]. Zu diesem Zweck wurde, ausgehend von verschiedenen Informationsquellen, eine Fallbasis (Case-Base) erstellt [L'Ab04], die etwa 500 Cases umfasst, wobei eine ständige Erweiterung, insbesondere durch die Analyse von Gesprächsprotokollen, möglich ist.

Für die Synchronisation der einzelnen Systemkomponenten sowie die Überwachung und Steuerung des Dialogs mit dem Benutzer ist in VIP-Advisor der „Interaction Manager“ zuständig. Er verwaltet die Interaktionsschritte, indem er Benutzereingaben interpretiert und Komponenten für die audiovisuellen Reaktionen anspricht. Bei einer Problemsituation wird eine interne Repräsentation des Problems erstellt, die als Input für das Case-Based-Reasoning-Modul dient. Eine ermittelte Lösung wird an die gegenwärtige Dialogsituation angepasst und dargestellt. Sie kann aus einer einfachen Systemantwort bestehen, aber auch Subdialoge enthalten, die der genaueren Identifikation des Problems zur Ermittlung einer individuellen Lösung dienen. Nach der erfolgreichen Problemlösung wird die Interaktion an der Stelle der Problemstellung fortgesetzt.

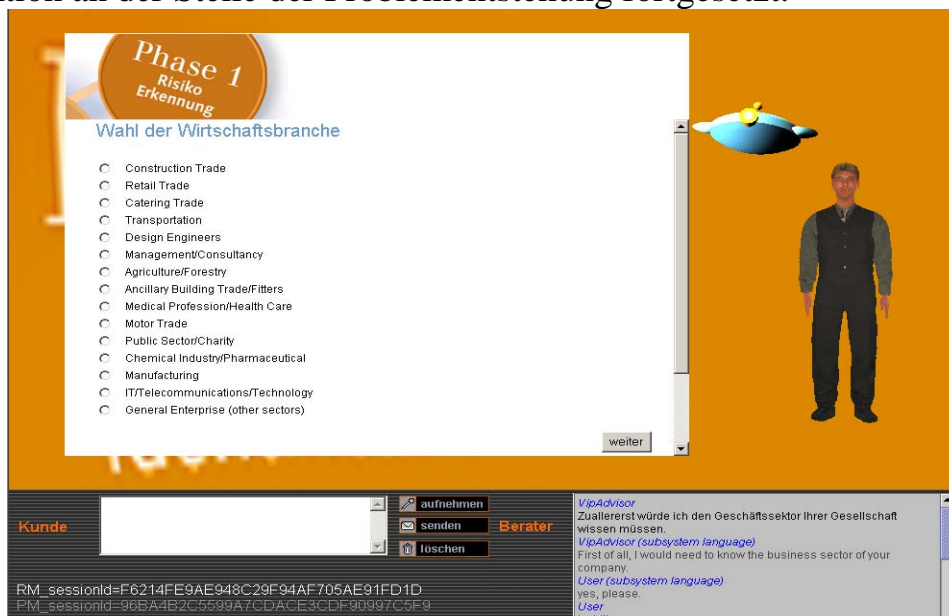


Abbildung 1: Screenshot des VIP-Advisor Prototypen

Im Rahmen der Projektaktivitäten wurden unterschiedliche Systemtests durchgeführt. Die Systemevaluierung erfolgte durch Benutzer mit unterschiedlichen Kompetenzniveaus, wobei anhand vordefinierter Szenarien das Verhalten und die Systemreaktionen protokolliert und analysiert wurden. Insgesamt ergab die Auswertung der Test-Interaktionen positive Ergebnisse. Die allgemeine Problemlösungsfähigkeit von VIP-Advisor wurde als zufrieden stellend eingestuft. Die erweiterten Interaktionsmechanismen, wie z.B. die Spracherkennung und maschinelle Übersetzung, bedürfen allerdings noch einer Verbesserung in zukünftigen Forschungsprojekten.

Die Erfahrungen im VIP-Advisor Projekt haben das Potential einer rechnergestützte Beratung aufgezeigt. In naher Zukunft sind dabei Lösungen interessant, die in der Praxis bereits eingeführte Verfahren zur Dialogführung wie z.B. „Chatterbots“ Systeme nutzen. Unter Verzicht auf Module zur Sprachanalyse und Generierung wird im Nachfolgenden ein Weg dorthin aufgezeigt. Um die Gestaltung von Beratungsdialogen zu erleichtern, wurde zunächst ein normatives Modell entwickelt, welches bei der Dialoggestaltung für Beratungsagenten zugrunde gelegt werden kann. Dieses wird im nachfolgenden Kapitel beschrieben.

4 Dialogmodell einer Beratung

Für unterschiedliche Formen der Beratung existieren jeweils Ansätze zur Gliederung des Beratungsverlaufs. Es stellt sich die Frage, welcher für den hier vorgestellten Fall bzw. seine Dialoge geeignet ist. Ziel des Agenten ist die Risikoberatung kleinerer und mittlerer Unternehmen im persönlichen Gespräch, weshalb die persönliche wie auch die Unternehmensberatung in Betracht kommen. Untersucht man die Charakteristika der beiden Beratungsformen, lässt sich der Ablauf des hier darzustellenden Dialogs schwerpunktmäßig mit einer persönlichen Beratung vergleichen.

Da das Dialogmodell von einer konkreten Realisierung unabhängig sein soll, finden ausschließlich Dialogbeiträge, die zur Beratung gehören, Beachtung. Befehle zur Systemsteuerung (z.B. zum Speichern von Daten oder dem Beenden des Agenten) sowie Probleme, die mit der Bedienung des Beratungsagenten oder mit dem Verständnis von Benutzereingaben auftreten können, bleiben unbeachtet. Bei der Gestaltung des Modells waren zwei Zielsetzungen zu vereinbaren:

1. Um eine Lösung für das Problem des Klienten zu finden, muss eine strukturierte Erfassung und Auswertung von Fakten erfolgen.
2. Der Dialog mit dem Klienten soll sich an einer Beratung zwischen Menschen orientieren. Dies beinhaltet einen möglichst freien Dialogverlauf und eine individuelle Ansprache durch den Berater.

Mit aktuellen Dialogsystemen ist diese Kombination kaum realisierbar. Ein möglichst freier Dialogverlauf, der jederzeit ein Abschweifen zu abweichenden Themen ermöglicht einerseits und die korrekte Erkennung von Benutzereingaben, die für die exakte Bestimmung der Problemsituation entscheidend ist andererseits, stellen meist widersprüchliche Ziele dar. Daher wird die pragmatische Zielsetzung verfolgt, eine Beratung zwischen Menschen bis zu einem bestimmten Grad nachzubilden. Der Dialogverlauf wird so gelenkt, dass die Auswahl an sinnvollen Dialogoptionen möglichst gering ist, um so die Analyse der Texteingabe zu erleichtern. Dies widerspricht aber nicht unbedingt einem natürlichen Beratungsverlauf. Gerade von Beratern wird häufig die Führung des Gesprächs erwartet, da sie gewöhnlich einen Informationsvorsprung gegenüber ihren Klienten besitzen.

4.1 COR als Grundlage des Dialogmodells

Die Darstellung des Dialogmodells basiert auf COR (Conversational Roles Modell) [Ste99], welches für die Beschreibung kooperative Information Retrieval Dialoge in einem multimodalen Benutzerinterface entworfen wurde und sich auf die Sprechakttheorie stützt. Auf dessen Darstellung aufbauend, wurde das Dialogmodell einer Beratung entwickelt. Bei diesem stellt COR eine adäquate Modellierung der Informationssuche von Beratern bzw. Klienten dar und findet in einer leicht abgewandelt Version bei entsprechenden Subdialogen Anwendung.

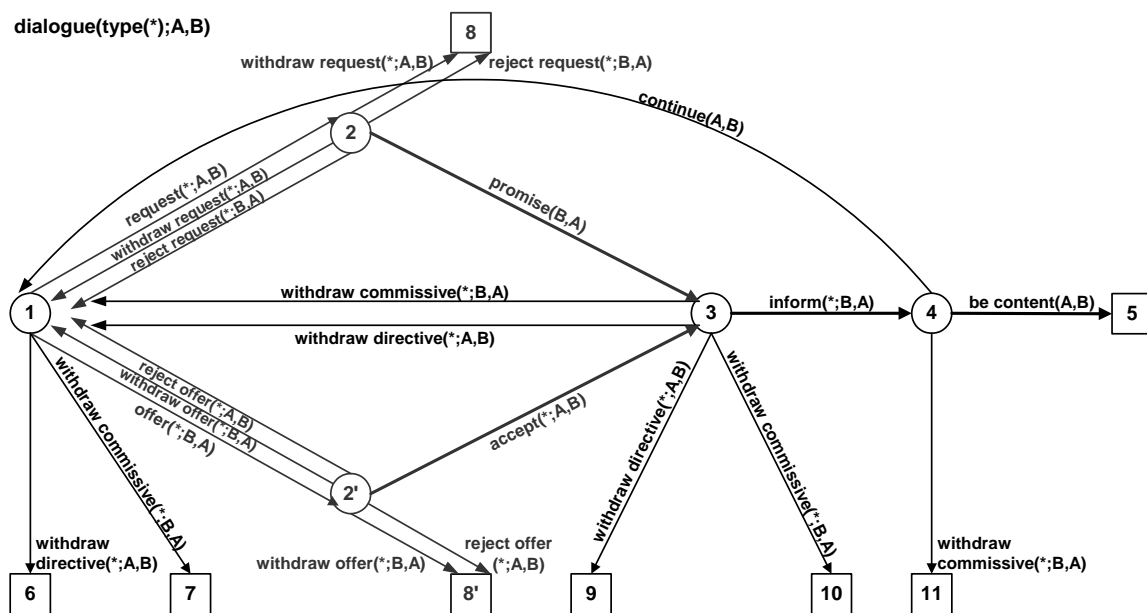


Abbildung 2: Zur Beratung abgewandelte Form des COR-Modells

Die Darstellung des Modells erfolgt als Netz aus Zuständen (dialog states) mit zugehörigen Zustandsübergängen (dialog moves). Ein Zustandsübergang wird durch Dialogbeiträge initiiert, die aus einem Dialogakt bzw. Subdialog bestehen, während die Zustände lediglich Verbindungsknoten zwischen zwei Übergängen (Kreise) bzw. die Endzustände (Quadrate) eines Dialogs darstellen. Abb. 2 zeigt das Schema des für die Beratung abgewandelten COR Dialogs zwischen den Dialogpartnern A (Informationssuchender) und B (Informationslieferant). Zustandsübergänge stellen meist Einstiegspunkte für durch Subnetzen modelliert Subdialoge dar. Dabei werden für verschiedenen Phase der Beratung spezifische Formen z.B. der request oder inform Subdialoge festgelegt. Die Semantik und der Verlauf eines Dialogs können vorgeben werden. Zum Beispiel würde `dialogue(request(personal client information);A,C)` festlegen, dass der „request“-Ast des Schemas verfolgt wird und der Dialoginhalt „personal client information“ ist, was einen Dialog zur Erfassung persönlicher Klienteninformationen beschreibt.

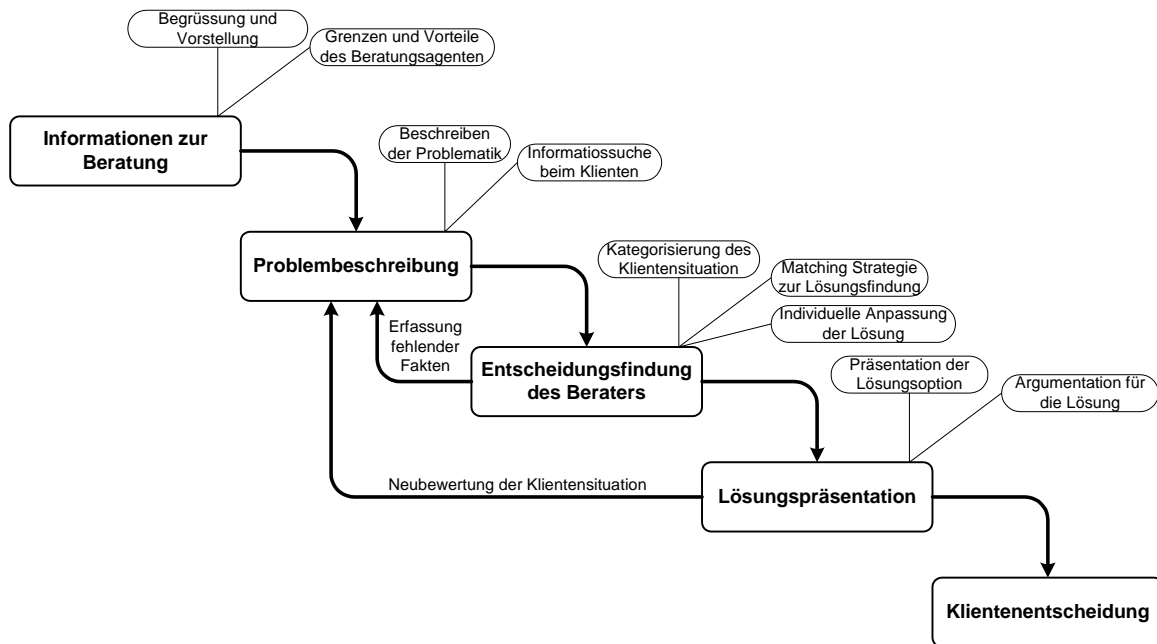


Abbildung 3: Abgewandelte Phasen des „Advice Giving and Taking“ Ansatzes

Eine Beratung lässt sich als Interaktionsprozess darstellen, der die Bearbeitung und Lösung von Problemen des Klienten zum Thema hat [Wasch98, S.18]. Jungermann charakterisiert den Beratungsprozess in seinem „Advice Giving und Taking“ Ansatz durch vier Phasen [Jung99], die auch bei der Gliederung unseres Dialogmodells Verwendung finden. Da sie allerdings den Ablauf einer Beratung und nicht deren Dialogverlauf darstellen, wurde die Phase „Informationen zur Beratung“ ergänzt. Aus Platzgründen ist es nicht möglich, hier das gesamte Dialogmodell zu erläutern. Stattdessen wird der Inhalt der einzelnen Phasen erläutert und exemplarisch auf das Dialogschema während der Problembeschreibungsphase eingegangen.

4.2 Modellierung der Phasen eines Beratungsdialogs

Die Beratung beginnt mit der Phase **Informationen zur Beratung**, in der ein Klient begrüßt und in den Dialog eingeführt wird. Da eine Beratung durch den Computer meist eine ungewohnte Situation ist, sind schon im Voraus einige wichtige Informationen zu geben und Fragen zu klären, um Befürchtungen zu zerstreuen und falsche Erwartungen zu vermeiden. Sollen sich die Dialogbeiträge des Agenten individuell an den Klienten anpassen, ist es außerdem sinnvoll, schon während der ersten Phase Daten dafür zu erfassen (z.B. Name oder Qualifikation). Neben einem Überblick über Verlauf und Ziele der Beratung müssen die Grenzen und Möglichkeiten des Systems aufgezeigt werden. Vorteile sind z.B. die Anonymität und die Verfügbarkeit rund um die Uhr. Möglicherweise sind Klienten aber wegen

des Datenschutzes besorgt, daher sollte über den Umgang mit personenbezogenen Daten informiert werden. Grenzen des Beratungsagenten ergeben sich außerdem aus dem relativ unflexiblen Dialogverlauf und Problemen beim Verständnis von Klientenäußerungen.

In der **Problembeschreibungsphase** wird der Klient über seine Problemstellung und die diesbezüglichen Bedürfnisse und Ziele befragt. Ihm sollten möglichst alle für eine Problemlösung relevanten Fakten entlockt werden. Dies ist oft ein erster Schritt, dem Klienten zu helfen, da er durch den Befragungsprozess Denkanstöße und neue Ideen erhält. Der Einsatz von Fachbegriffen sollte dabei vermieden werden, um den Klienten nicht mit Fakten zu überrollen oder gar zu verwirren. Eine möglichst vollständige Problembeschreibung ist für die Qualität einer Beratung entscheidend, entsprechend umfangreich und komplex ist das Dialogschema, da es dem wechselseitigen Informationsaustausch zwischen Klient und Berater Rechnung tragen muss. Der eine benötigt Informationen, um die Problemstellung analysieren zu können und der andere, um sein Problem bzw. die zugehörigen Fragen richtig zu verstehen.

Problem Description

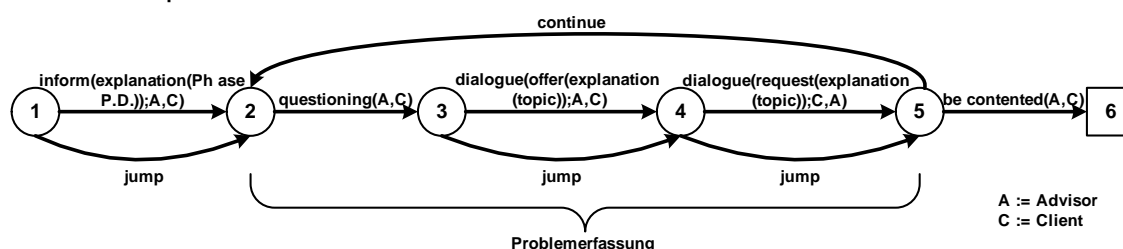


Abbildung 4: Dialogschema der Problembeschreibungsphase

Auf der obersten Ebenen des Dialogschemas für diese Phase (Abb. 4) wird der Klient mit `inform(explanation(...));A,C` zunächst über die neue Phase und ihre Ziele informiert. Danach beginnt die eigentliche Problemerkfassung. Dieser aus drei Subdialogen bestehende Abschnitt hat immer eine Fragestellung des Beraters zum Inhalt und wird so lange wiederholt, bis der Agent glaubt, genügend Informationen für eine Problemlösung gesammelt zu haben (`be contented(A,C)`).

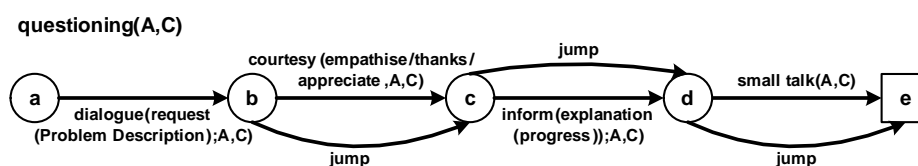


Abbildung 5: Dialogschema des Subdialogs questioning(A,C)

Im Subdialog `questioning(A,C)` (Abb. 5) erfolgt die Problemerkfassung, wobei die Befragung in `dialogue(request(Problem Description));A,B`) (vgl. Abb. 2)

stattfindet. Nachdem eine Frage vom Klienten beantwortet worden ist, kann dieser auf verschiedene Arten für den weiteren Verlauf der Beratung motiviert werden. Neben höflich formulierten Äußerungen des Agenten sind im Modell dazu spezifische Dialogelemente vorgesehen:

- Die Mitarbeit des Klienten sollte mit ermunternden bzw. dankbaren Äußerungen quittiert werden. Dazu existieren Sprechakte für Höflichkeitsfloskeln, die courtesy acts. Sie haben eine rein expressive Funktion, sollen das Gespräch auflockern und einen „natürlicheren“ Dialogablauf suggerieren. Beispiele dafür sind Gesten des Dankes („Danke, für ihre Hilfe.“) oder der Anerkennung („Gut gemacht!“).
- Damit die Entwicklung des Beratungsprozesses nachvollziehbar bleibt, sollte der Klient regelmäßig über den Fortschritt der Beratung auf dem Laufenden gehalten werden (inform(explanation(progress));A,B)).
- Zu Gesprächen zwischen Menschen gehört small talk, Subdialoge mit Themen, die nichts zur Beratung beitragen. Abhängig vom Klientel und dem Thema kann diese Fähigkeit zur Akzeptanz beitragen (z.B. Verkaufsberater für CD's), aber auch unerwünscht sein (professionelle Beratungen wie z.B. Risk-Management). Damit der Agent die Kontrolle über solche Dialoge hat, sollte er sie selbst initiieren (small talk(A,C)).

request(Problem Description;A,B)

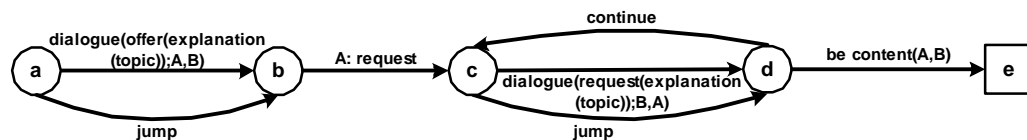


Abbildung 6: Dialogmodell des Subdialogs request(problem description;A,B)

Die Fragestellung in `dialogue(request(Problem Description);A,B)` erfolgt im Subdialog `request(Problem Description;A,B)`. Manchmal ist es dabei nötig, dem Klienten im Voraus Informationen zu geben, z.B. um ihn auf eine Frage einzustimmen und/oder damit er sie überhaupt richtig beantworten kann. Dies ist in `dialogue(offer(explanation(topic));A,B)` möglich. Sind diese Vorinformationen vermittelt, kann die eigentliche Fragestellung in `A: request` erfolgen. Sollte der Klient nun keine weiteren Nachfragen haben, kann der Subdialog verlassen werden (`jump`). Hat er aber Probleme z.B. mit unverständlichen Fachbegriffen oder der Fragestellung selbst, kann er sie in `dialogue(request(explanation(topic)))` klären. Dieser Schritt wird so lange wiederholen (`continue`), bis keine Fragen mehr offen sind (`be content(A,B)`).

Danach wird der Subdialog `dialogue(request(Problem Description);A,B)` mit der Antwort des Klienten (`inform(Problem Description;A,B)`) abgeschlossen.

Nach dem Abschluss des `questioning(A,C)` Subdialogs werden Möglichkeiten gegeben, allgemeine Themen zu diskutieren (Dialogschritte 3-5 in Abb. 4). Hier kann der Beratungsagent mit `dialogue(offer(explanation(topic));A,C)` pro-aktive Erläuterungen anbieten, z.B. um weiterführende Informationen zur Verfügung zu stellen oder aber der Klient wird eigenständig aktiv, indem er in `dialogue(request(explanation(topic));C,A)` Fragen zum Themenkomplex der Beratung stellt. Dies ahmt die Fähigkeit eines menschlichen Beraters nach, dem Klienten durch Fachinformationen neue Perspektiven in der Betrachtung seines Problems zu liefern.

Bei der **Entscheidungsfindung** nutzt der Berater die zuvor erhaltenen Informationen, um Empfehlungen für Problemlösungen zu entwickeln. Dafür können abhängig von der Art der Beratung unterschiedlichste Verfahren genutzt werden, z.B. die zu Beginn erwähnten Information Agents oder auch Expertensysteme. Teilweise existieren bereits Programme, die z.B. zur Unterstützung von Beratern entwickelt wurden. Diese nutzen häufig Kategorisierungs- und Matching-Strategien, bei denen der Klient und sein Problem in Kategorien eingeordnet werden, die als Basis für die Ermittlung der am besten passenden Lösungsmöglichkeiten dienen. Eine solche Klassifizierung geschieht teilweise schon während der Problembeschreibung (vgl. VIP-Advisor). In vielen Beratungssituationen ist nicht auf Anhieb eine optimale Lösung zu erwarten, da sich Problem und Klient nicht immer eindeutig einer Kategorie zuordnen lassen oder spezielle Anforderungen existieren, die es nötig machen, zwischen verschiedenen Lösungen abzuwägen oder vorhandene abzuwandeln. Reichen die gesammelten Informationen nicht aus, kann sogar ein Rücksprung in die Problembeschreibungsphase notwendig sein.

Die erarbeiteten Problemlösungen stellt der Berater dem Klienten während der **Lösungspräsentation** vor. Dabei ist es meist sinnvoll, nur das Ergebnis, nicht aber den detailliert Weg der Entscheidungsfindung, zu erläutern. Bei der Präsentation hat der Berater drei Möglichkeiten. *Eine Lösung vermarkten*, indem er sie empfiehlt und dies mit Argumenten untermauert (z.B. Verkaufsberatung). *Mehrere Lösungen vorstellen und eine empfehlen*, wobei die Gründe für diese Entscheidung dargelegt werden sollten (z.B. medizinische Beratungen). Der Berater kann auch *mehrere Lösungen ohne spezielle Empfehlung* mit ihren Vor- und Nachteilen präsentieren. Indem einer Empfehlung erst gegeben wird, wenn der Klient nachfragt, kann jeglicher

Eindruck einer Beeinflussung durch den Berater vermieden werden. Welche Präsentationsform gewählt wird, ist von der Erwartung des Klienten, den Zielen des Beraters und den verfügbaren Lösungen abhängig. Aus Dialogsicht ist zu beachten, dass ein Rollentausch stattfindet. War der Klienten bisher der Befragte, ist es nun an ihm, die Lösungsvorschläge durch Fragen an den Berater zu erkunden. Dabei können Einwände und Fragen Hinweise darauf geben, dass Fakten falsch eingeschätzt worden sind und eine Neubewertung notwendig ist. Allerdings ist es schwierig, diese Hinweise so zu interpretieren, dass sie als Fakten in eine neue Entscheidungsfindung einfließen können.

Die **Klientenentscheidung** stellt den Abschluss einer Beratung dar. Bei der Zustimmung bzw. Ablehnung der vorgestellten Lösungsmöglichkeiten spielen unterschiedlichste Kriterien eine Rolle. Von der Dialogseite gesehen besteht die Entscheidung lediglich aus einem zustimmenden bzw. ablehnenden Sprechakt. Bei einer Ablehnung wird der Agent nur selten den Grund bestimmen können, trotzdem muss das nicht unbedingt das Ende des Beratungsdialoges bedeuten. Zum einen können noch andere, vom Agenten als schlechter bewertete Lösungen, vorgestellt werden. Zum anderen lassen sich aus der Kritik des Klienten teilweise Fakten bestimmen, die in einen weiteren Beratungsdurchlauf einfließen können.

5 Realisierung

Im Gegensatz zur aufwendigen computerlinguistischen Software, die bei VIP-Advisor eingesetzt wird, um Probleme der Spracherkennung und Übersetzung zu lösen, beschränkt sich die hier vorgestellte Implementierung auf das textuelle Ein- und Ausgabeformat. Als Beschreibungssprache für die Dialogsteuerung des virtuellen Beraters kommt das von der ALICE FOUNDATION entwickelte AIML (Artificial Intelligence Markup Language) zum Einsatz, eine auf XML basierende Sprache, die einen verbreiteten Standard für „Chatterbot“-Systeme darstellt. Durch AIML wird festgelegt, wie auf Benutzereingaben reagiert werden soll. Auf Grundlage dieser Wissensbasis übernimmt ein Programm (hier „Program D“) die Ausführung des Chatterbots. Vorteile von AIML:

- Es lässt sich einfach erlernen, bietet aber trotzdem die Möglichkeit, komplizierte Konstrukte aufzubauen.
- Da es auf XML basiert, existieren Tools zum Editieren, Parsen und Verwalten. Außerdem kann der Sprachumfang einer XML basierenden Beschreibungssprache leicht erweitert werden.

- AIML Interpreter wurden in verschiedenen Programmiersprachen implementiert und sind meist frei verfügbar.
- Die verschiedenen Implementierungen von AIML bieten unterschiedliche Möglichkeiten Scriptsprachen einzubetten (z.B. Javascript oder Perl).

In der Arbeit von Neves und Barros [Neba03] werden Probleme im Dialogfluss von AIML basierten Chatterbots aufgezeigt. Anstatt wie dort eine vollkommen neue Sprache zu entwickeln, wurde hier versucht eine flüssigeren Dialogablauf durch Nutzung des zuvor vorgestellten Dialogmodells zu erreichen und die Problemlösungsfähigkeit durch ein CBR-Verfahren zu verbessern⁵. Das Dialogmodell dient als Schablone beim erstellen spezifischer Dialoginstanzen für konkrete Beratungssituationen, aus denen sich der Dialogplan entwickelt. Damit dieser realisiert werden kann, ist entscheidend, dass der Beratungsagent immer weiß, in welchem Zustand er sich gerade befindet. Dies kann in AIML wie folgt realisiert werden.

Die Einheit, in der Wissen in AIML gespeichert wird, ist eine Kategorie (AIML tag <category>). Sie besteht aus einem Muster (AIML tag <pattern>), welches festlegt, für welche Benutzereingaben die Kategorie gilt und einem Template, welches die Handlung (z.B. Textausgaben oder Aufruf eines SOAP-Service) bestimmt. Ergänzend zu einem Muster kann man that-Statements (AIML tag <that>) verwenden, die festlegen, dass eine bestimmte Ausgabe des Chatterbots vorangegangen sein muss, damit die Kategorie gültig ist. Dies ist ein nützliches Element zur Dialogsteuerung, da ein lokaler Kontext vorgegeben werden kann. Für die Realisierung des Dialogplans war allerdings wichtiger, dass eine Kategorie einem Thema (AIML tag <topic>) zugeordnet sein kann. Sollte ein Thema gesetzt sein, wird ein passendes Muster zunächst in Kategorien mit diesem Thema, dann in den themenfreien Kategorien gesucht. Jedem Zustand im Dialogplan kann ein eigenes Thema zugeordnet werden, wodurch ein Kontext für die Analyse einer Benutzereingabe vorgegeben wird, indem der Chatterbot primär auf die im Plan festgelegten Zustandsübergänge reagiert. Die themenfreien Kategorien dienen dann lediglich zur Behandlung unerwarteter oder falsch erkannter Benutzereingaben. Um den Wechsel zwischen Themen zu erleichtern und Kategorien gruppieren zu können, wurden Variablen eingeführt, in denen das letzte relevante und das als nächstes gewünschte Thema abgelegt sind. Damit ist es möglich, eine Form von Unterprogrammstruktur in AIML zu realisieren,

⁵ Im Agentenframework AINI wird ebenfalls eine AIML Wissensbasis um CBR-Verfahren ergänzt [Gofu03], allerdings kein detaillierter Dialogplan genutzt.

durch deren geschickte Ausnutzung sich mit einer relativ kleinen Wissensbasis eine leistungsfähige Dialogsteuerung realisieren lässt. Zusätzlich wird der Code dadurch übersichtlich und leichter erweiterbar.

Der in der Implementierung abgebildete Risikoanalyseprozess ist in mehrere Abschnitte untergliedert. Im ersten wird in mehreren Interaktionsschritten ein Profil des Klienten erstellt, dessen Analyse der Zusammenstellung eines Risiko-Portfolios dient. Unter anderem soll dabei die Wirtschaftsbranche des Benutzers ermittelt werden, indem ihm eine Liste mit verschiedenen Auswahlmöglichkeiten präsentiert wird. Dieser Interaktionsschritt soll beispielhaft die Arbeitsweise des Systems erläutern und darstellen, wie das Modell aus Kapitel 4.2 bei der Dialoggestaltung des Agenten Einsatz findet.

```

<1>Problem Description; inform(explanation(Business_sector);B,K); B:inform
B: „Sie stehen am Anfang des Risikoanalyseprozesses. Zunächst möchte ich ein wenig
mehr über Ihr Unternehmen erfahren. Dazu ist es notwendig, dass Sie einige allgemeine
Angaben machen. Auf Basis dieser Informationen wird dann die Analyse erstellt. Je
genauer Ihre Angaben sind, desto besser kann ich Ihr unternehmerisches Risiko
einschätzen.“
<2>questioning(B,K); dialogue(request(Problem Description;B,K)); B:request
      B: „Geben Sie zunächst an, welchem Wirtschaftszweig Ihr Unternehmen angehört.“
<3>request(ProblemDescription;B,K);
      dialogue(request(explanation(Motorfahrzeuggewerbe);K,B)); K:request
      K: „Was ist mit der Branche Motorfahrzeuggewerbe gemeint?“
<4> dialogue(request(explanation(Motorfahrzeuggewerbe);K,B)); B:inform
      B: „Die Wirtschaftsbranche Motorfahrzeuggewerbe beinhaltet die komplette
Fahrzeugindustrie“
<5>questioning(B,K); dialogue(request(Problem Description;B,K)); K:inform
      K: „Ich bin in der Brache Motorfahrzeuggewerbe tätig.“

B := Beratungsagent ; K := Klient
    
```

Abbildung 7: Verlauf eines Beispieldialogs mit dem Beratungsagenten

Der Berater bietet zunächst eine Einführung an, bei der die Ziele des Interaktionsschritte erläutert werden (Abb. 7, <1>), was dem ersten Zustandübergang im Problem Description Dialog (vgl. Abb. 4) entspricht. Daraufhin fordert der Berater den Benutzer auf, eine Auswahl aus einer Liste zu tätigen (Abb. 7, <2>), wobei die Möglichkeit einer Erläuterung im questioning(B,K) (vgl. Abb. 5) Subdialog nicht genutzt wird. Nehmen wir an, der Benutzer hat Schwierigkeiten beim Verständnis eines Begriffs in der Auswahlliste der Wirtschaftsbranchen und stellt daher eine Frage. Dies initiiert im request(Problem Description;B,K) Subdialog einen weiteren Subdialog (Abb. 7, <3>) zur Klärung des Problems.

Eine umfangreiche Frage- und Antwortliste in der Wissensbasis des Chatterbots zu verwalten wird schnell sehr aufwendig. Hier bietet es sich an, auf andere Möglichkeiten zurückzugreifen. Im konkreten Beispiel wird Analog zu VIP-Advisor CBR genutzt. Dazu wurde eine Schnittstelle geschaffen, um die Möglichkeiten der Dialogsteuerung durch externe Prozesse zu erweitern. Dies war relativ leicht zu realisieren, da die JAVA Implementierung von AIML im Quellcode vorliegt. Um die Schnittstelle möglichst universell zu halten, wurde der Sprachumfang von AIML um das Tag `<SOAPaccess>` erweitert, welches den Aufruf eines beliebigen SOAP-Service⁶ möglich macht. Der Aufruf des Case-Based-Reasoner soll dies beispielhaft darstellen.

```
<category>
  <pattern>*/</pattern>      <!--Fraglicher Begriff wird in "*" übergeben-->
  <template>
    <SOAPaccess> <!--Neues AIML-Tag für SOAP-Aufrufe-->
      <service>urn:case-based-reasoner</service> <!--URI des SOAP-Service-->
      <function>getCase</function>      <!--Funktion, die Aufgerufen wird-->
      <!-------Funktionsparameter----->
      <subject>
        <get name="retrieval_subject"/> <!--Themengebiet der Anfrage-->
      </subject>
      <matter>
        <get name="retrieval_matter"/> <!--Art der Anfrage-->
      </matter>
      <text>
        <get name="retrieval_text"/> <!--Ursprünglicher Text der Anfrage-->
      </text>
      <content>
        <star index="1"/> <!--Schlüsselwort der Anfrage-->
      </content>
    </SOAPaccess>
  </template>
</category>
```

Abbildung 8: Aufruf des SOAP-Service Case-Based-Reasoner

Der Case-Based-Reasoner ermittelt eine Lösung zu einem Benutzerproblem, welches nicht in den AIML Kategorien berücksichtigt ist. Dabei wird eine Kombination aus Text-Retrieval und Retrieval mit Attribut-Werte-Paaren benutzt. Die Cases bestehen aus der Problemdefinition und dem Lösungsteil. Der Definitionsteil enthält eine Metabeschreibung des Problems, kodiert durch Attribute in einer festen Reihenfolge sowie einer textuellen Beschreibung der Anfrage, die stellvertretend für alle Formulierungen desselben Problems ermittelt worden ist.

⁶ <http://www.w3.org/TR/SOAP>

Der Eingabe-Text, der das Dialogproblem ausgelöst hat, wird inhaltlich analysiert und auf eine Grundform reduziert. Dabei kommen Stoppworteliminierung und Wortstammreduktion zum Einsatz. Die daraus resultierenden Schlüsselwörter werden als Suchanfrage für die indexierte Case-Basis verwendet. Zur Steigerung der Retrieval-Präzision wird ein Problem durch mehrere Attribute beschrieben. Anhand der verwendeten Fragestellung lässt sich ein Wert für das Attribut „matter“ bestimmen. Wird eine Definition gewünscht (erkennbar durch allgemeine Satzstellungen wie z.B. „Was ist mit * gemeint?“, „Was bedeutet *?“), ergibt sich „definition“ als Attributwert, ist eine Erklärung oder Beschreibung einer Aktivität oder eines Prozesses erwünscht (z.B. erkennbar durch „Wie kann ich *?“, „Wie geht das *“), so wird der Wert „explanation“ gesetzt. Das „content“-Attribut übermittelt den gefragten Begriff. Üblicherweise ist dies die *-Komponente der obigen Beispiele. Der ursprüngliche Fragetext bleibt in der Variable „retrieval_text“ erhalten. Das Attribut „subject“ beschreibt eine mögliche Zuordnung des Problems zu einem Themengebiet, dabei wird das aktuelle AIML Thema als Wert benutzt. Damit wird ein Kontext für die Problemstellung vorgegeben, der durch weitere Attribute verfeinert werden kann. Zum Beispiel lassen sich Cases nach der Komplexität der Lösungsbeschreibung einstufen. So würden Lösungen mit hoher Komplexität unerfahrenen Benutzern nicht angeboten, wogegen Experten keine detaillierte Grundlagenbeschreibung erhalten. Ein anderes Beispiel betrifft das Herkunftsland des Klienten. Problemlösungen mit juristischem Anteil sollten nur Personen angeboten werden, für die das in der Lösung verwendete Rechtssystem relevant ist.

Die vom Case-Based-Reasoner gelieferte Antwort wird dem Benutzer präsentiert (Abb. 7, <4>), womit der Subdialog `dialogue(request(explanation(Motorfahrzeuggewerbe);K,B))` beendet ist. Da der Klient keine weiteren Fragen hat, beantwortet er die Frage des Beratungsagenten, entweder durch eine Texteingabe oder die Auswahl eines Radiobuttons in einer Liste mit Wirtschaftszweigen (Abb. 7, <5>). Dies beendet die Subdialoge `dialogue(request(Problem Description;B,K))` und `questioning(B,K)`. Die Interaktion wird danach mit der nächsten anstehenden Fragestellung des Beraters fortgesetzt.

6 Zusammenfassung und Ausblick

In diesem Beitrag wird ein Dialogmodell der Beratung und seine Anwendung bei einem virtuellen Beratungsassistenten im WWW vorgestellt. Ausgehend von den Erfahrungen aus dem EU-Projekt VIP-Advisor, wurde ein Modell

konzipiert, welches zwar vorsieht, dass Dialogführung beim Berater liegt, im Dialogverlauf aber genug Freiheiten zulässt, sodass der Eindruck einer Beratung durch einen Menschen vermittelt werden kann. Die Realisierung des Dialogmodells in AIML wird dabei durch ein CBR-Modul ergänzt, dessen Antworten durch Kontextinformationen individuell ermittelt werden. Es zeigt sich, dass die recht einfache Chatterbot Technik durch die Kombination mit anderen Verfahren leistungsfähige Ergebnisse auch in einem komplexen Dialogszenario liefert. Gerade für die Bereiche der Experten- und Gutachterberatung, wo bereits rechnergestützte Lösungsansätze existieren, bietet es sich an, eine natürlichsprachliche Dialogkomponente zu nutzen.

Das hier vorgestellte Dialogmodell stellt einen ersten Ansatz dazu dar, lässt sich aber aufgrund von Erfahrungen sicher noch verfeinern. Eine Verbesserung der Beratungsfunktion verspricht eine Erweiterung des CBR-Moduls um weitere kontextbezogene Attribute. Außerdem wäre es nützlich auf Basis des Dialogmodells eine graphische Oberfläche zur Generierung der Wissensbasis eines Beratungsagenten zu entwickeln. Ein weiterer Ansatzpunkt für Erweiterungen wäre eine Verbesserung im pro-aktiven Verhalten des Agenten. Unter Berücksichtigung des Dialogverlaufs und kontextueller Informationen, wie z.B. Benutzereigenschaften und – Einstellungen, können Lösungsansätze präsentiert werden, bevor der Anwender sich eines Problems bewusst ist. Zu diesem Zweck muss das System um eine Komponente erweitert werden, die den Dialog kontinuierlich verfolgt, um Situationen zu erkennen, in denen im Dialogverlauf eine solche Problemlösung angeboten werden soll.

7 Literaturangaben

- [Ek87] Elfgen, R. & Klaile, B. (1987): Unternehmensberatung : Angebot, Nachfrage. Poeschel, Stuttgart.
- [Fei92] Feigenbaum, E. A. (1992): A personal view of expert systems: Looking back and looking ahead. *Expert Systems with Applications*, 5, 1992, 193-201
- [Gofu03] Ong Sing Goh¹ & Chun Che Fung² (2003): Intelligent Agent Technology in E-commerce. In *Intelligent Data Engineering and Automated Learning*, 4th International Conference, IDEAL 2003, Hong Kong, China, Lecture Notes in Computer Science 2690 Springer 2003, 10-17.
- [Hill90] Hill, W. (1990): Der Stellenwert der Unternehmensberatung für die Unternehmensführung. In: *Die Betriebswirtschaft*, 50. Jg. (1990), Nr. 2., S.171-180.
- [Hoe99] Hoepfner, W. (1999): Der Mensch-Maschine Dialog, In: G.Antos, K.Brinker, W.Heidemann, S.F.Sager (Hrsg.): "Text und Gesprächslinguistik", 2. Halbband: "Gesprächslinguistik", Berlin: de Gruyter 1999.
- [Jon00] Jonas, E. (2000): Beraten und entscheiden: Experimentelle Untersuchungen zur Informationssuche in Beratungssituationen. Neuried bei München : Ars Una, S. 6 ff.

- [Jung99] Jungermann, H. (1999): Advice giving and taking. Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS-32), Maui, Hawaii: Institute of Electrical and Electronics Engineers, Inc. (IEEE).
- [Kime79] Kienbaum, G.. & Meissner, D. (1979): Zur Problematik des Effizienznachweises von Beratung. In: Betriebswirtschaftliche Forschung und Praxis (31), 109-116.
- [Kvo93] König, E. & Volmer, G. (1993): Systemische Organisationsberatung. Grundlagen und Methoden, Deutscher Studien Verlag, Weinheim 1993, 4. Aufl. 1996.
- [L'Ab04] L'Abbate, M.; Kamps, T. & Thiel, U. (2004): Content Engineering for Conversational Advisory Systems in the Finance and Insurance Domain. Proceedings of eChallenges 2004, Vienna, Austria. To appear
- [L'Ab03] L'Abbate, M. & Thiel, U. (2003): Use of Contextual Information in a Proactivity Model for Conversational Agents. In P. Blackburn, C. Ghidini, R. M. Turner, F. Giunchiglia (Eds.): Modeling and using Context, Proceedings of the 4th International and Interdisciplinary Conference, Stanford, CA, USA, June 2003, Springer LNAI.
- [Lea96] D. Leake (1996): CBR in context: The present and future. In: Leake, D. (ed), 1996, Case-based Reasoning: Experiences, Lessons and Future Directions. AAAI Press/MIT Press.
- [Neba03] Neves, A. M. M. & Barros, F. A. (2003): XbotML: A Markup Language for Human Computer Interaction via Chatterbots. In Web Engineering, International Conference, ICWE 2003, Oviedo, Spain, Lecture Notes in Computer Science 2722 Springer 2003, 171-181.
- [Ste99] Stein, A.; Gulla, J. A. & Thiel, U. (1999): User-Tailored Planning of Mixed Initiative Information-Seeking Dialogues. User Modeling and User-Adapted Interaction, 1999, 9(1-2): S. 133-166.
- [Sti61] Stigler, G.J. (1961): The Economics of Information. Journal of Political Economy 69, 213-225.
- [Wal95] Walger, G. (1995): Idealtypen der Unternehmensberatung. In: Walger, Gerd (Hrsg.): Systemische Unternehmensberatung, Organisationsentwicklung, Expertenberatung und gutachterliche Beratungstätigkeit in Theorie und Praxis. Köln 1995.
- [Wasch98] Walger, G. & Scheller, C. (1998): Das Angebot der Unternehmensberatungen in Deutschland, Österreich und der Schweiz. Eine empirische Analyse. In: QUEM-Report (Arbeitsgemeinschaft Qualifikations-Entwicklungs-Management). Schriften zur beruflichen Weiterbildung, Heft 54, Berlin.
- [Wei96] Weiershäuser, S. (1996): Mitarbeiterverhalten im Beratungsprozeß: eine ökonomische Betrachtung. Wiesbaden : Dt. Univ.-Verl. [u.a.], Gabler Edition Wissenschaft.

