

Selbstreflexion in einem planbasierten Dialogsystem

Peter Reiß, Bernd Ludwig, Günther Görz

{reiss, ludwig, goerz}@informatik.uni-erlangen.de

Abstract: Wir stellen eine Erweiterung eines existierenden Dialogsystems um Funktionalitäten zur Selbstreflexion und Selbsterklärung vor. Ergebnis ist ein individualisiertes und personalisiertes Assistenzsystem, das den Benutzer mit Informationen darüber versorgen kann, welche Schritte er in einer konkreten Situation ausführen muss, um seine Ziele zu erreichen.

1 Motivation

Natürlichsprachliche Dialogsysteme sind auf dem Wege, sich in Praxisanwendungen zu etablieren. Vor allem im Automobilbereich und bei Callcenter-Anwendungen werden sie inzwischen dafür genutzt, dem Benutzer eine komfortable Ansteuerung der verschiedensten Funktionen zu ermöglichen. Ein weiterer sinnvoller Anwendungsbereich ist die Hilfestellung für behinderte Menschen, die mit Hilfe natürlichsprachlicher Anweisungen ihre Umwelt in einer Weise manipulieren können, die ihnen ansonsten nicht zugänglich wäre.

In Abbildung 1 ist eine typische Bediensituation dargestellt. Der Benutzer steuert mit Hilfe des Dialogsystems eine bestimmte Applikation. Je nach Typ der Applikation hat er eine bestimmte Vorstellung von ihrem Leistungsumfang; beispielsweise wird er erwarten, dass bei einem Kontoführungssystem der Kontostand abgefragt, eine Überweisung getätigt oder einen Dauerauftrag angelegt werden kann. Das Wissen darüber hat er sich durch die bisherige Verwendung seines Kontos erworben. Diese Beziehung ist durch die gestrichelte Linie zwischen Benutzer und Applikation angedeutet.

Bei der Bedienung über ein Dialogsystem gibt er die Kommandos indirekt: Er äußert einen Wunsch (dargestellt durch die gepunktete Linie), der vom Dialogsystem in eine entsprechende Anweisung an die Applikation übersetzt wird (durchgehende Linien). Auch hier hat ein menschlicher Benutzer eine bestimmte Vorstellung vom Ablauf. Als Referenz dient ihm das alltägliche Gespräch mit Mitmenschen. Bittet er beispielsweise eine Person im Nebenzimmer, das Fenster zu schließen, erwartet er, dass dies – falls sein Wunsch richtig interpretiert wurde – geschlossen wird. Andernfalls erfolgt normalerweise eine Rückfrage („welches Fenster meinst du denn?“). Schließlich kann, wenn nötig, eine Statusmeldung (positiv oder negativ, eventuell mit weiteren Erklärungen) folgen.

Ein wichtiger Punkt ist dabei, dass bei einem Gespräch mit menschlichen Teilnehmern jederzeit die adressierte Ebene gewechselt werden kann. Ein Satz kann sich auf den Gegenstandsbereich beziehen, oder den Fortgang des Diskurses zum Inhalt haben. Die *Con-*

versation Acts Theory (CAT, [Tra94]) leistet einen wesentlichen Beitrag zur Analyse von Dialogen, da sie Äußerungen hinsichtlich ihrer Funktion in einem Diskurs klassifiziert. Bei der Interaktion zwischen Diskursteilnehmern ist (Selbst-)Reflexion jedes Diskursteilnehmers darüber notwendig, ob er die Inhalte des bisherigen Dialogs so verstanden hat wie die anderen Teilnehmer. Ergebnisse dieser der Synchronisierung der Dialogführung dienenden Überlegungen können in interaktiven Diskursen stets thematisiert werden. Der Zustand der Interaktion kann somit zum Gegenstandsbereich des Dialogs werden. Die Wichtigkeit der Einordnung einer Äußerung dahingehend, ob sie den Gegenstandsbereich oder den Diskurs betreffen, ist in [Lud03b] umfassend dargestellt.

Zusätzlich kann man ein menschliches Gegenüber jederzeit um *Erklärungen* bitten, in einer Antwort wird es ausführen, wie man zu einem gewünschten Ergebnis kommt. Ziel unserer weiteren Ausführungen ist es, aufzuzeigen, dass auch computergestützte Assistenzsysteme über die Möglichkeit verfügen sollten, solche Erklärungen abzugeben. Wir werden vorstellen, wie dies in unserem vorhandenen Dialogsystem implementiert werden kann. Die entstehende Schnittstelle ist auch in Einsatzbereichen anwendbar, in denen die Ausführung der Schritte zum Erreichen des Ziels vom Benutzer selbst ausgeführt werden müssen. Denkbar ist z.B. eine interaktive Montageanleitung für den Zusammenbau von Möbeln.

Bei genauerer Betrachtung von Abbildung 1 fällt auf, dass im Dialogsystemscenario Benutzerfragen, die Erklärungen erfordern, in zwei Kategorien eingeteilt werden können. Fragen des Typs „was kann ich *tun*?“, beziehen sich darauf, welche Funktionalität der Applikation wie angesteuert werden kann. Dabei kann es durchaus eine Diskrepanz zwischen den Vorstellungen des Benutzers und den tatsächlich zugänglichen Funktionen geben (beispielsweise, wenn die Möglichkeit, einen Dauerauftrag anzulegen, nicht implementiert ist). Wir werden uns hier auf die Behandlung dieser Fragetypen beschränken. Fragen der Form „was kann ich *sagen*?“ (d.h.: welche Eingaben kann das Dialogsystem selbst sinnvoll verarbeiten), stellen ebenfalls eine wichtige Assistenzfunktion dar, werden hier aber nicht adressiert.

Zunächst erfolgt jedoch eine kurze Vorstellung des Dialogsystems, innerhalb dessen die neue Funktionalität implementiert werden soll.

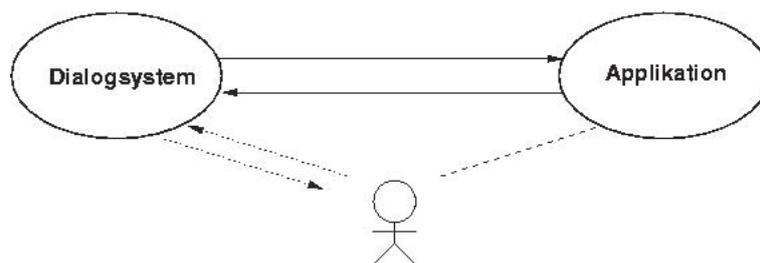


Abbildung 1: Schema eines Dialogsystems

2 Das Dialogsystem

Im Rahmen des BMBF-Leitprojekts Mensch-Technik-Interaktion EMBASSI (elektronische Multimodale Bedien- und Service Assistenz) wurde das modulare Dialogsystem CONALD entwickelt (siehe [FLB01, Lud03a]). Die einzelnen Komponenten des Systems, die eine Benutzeräußerung verarbeiten, sind innerhalb eines Multiagenten-Systems (MAS) realisiert. Der Benutzer spricht über ein Mikrophon(array), ein Spracherkennungserzeugt aus dem Sprachsignal einen ASCII-String. Der Parser transformiert die Äußerung in eine semantische Repräsentation, die in einer erweiterten Form der DRT ([KR93]) definiert ist. Aus dieser wird das Ziel abgeleitet, das der Benutzer (wahrscheinlich) erreichen will. Die Berechnung der einzelnen Schritte bis zum Ziel stellt eine Planungsaufgabe dar, wir verwenden für die Berechnung den FF Planer ([HN01]). Der initiale Zustand für den Planungsprozess ergibt sich aus dem Zustand der Applikation, ist also situationsabhängig (Situierung). Wird ein Plan gefunden, so wird dieser an die ausführenden Agenten geleitet, hier befindet sich also die Schnittstelle zur Applikation.

3 (Selbst-)Erklärung der Applikation

In komplexen Anwendungen stellt sich dem Benutzer oftmals die Frage, wie er ein bestimmtes Ziel erreichen kann. Idealerweise kann ihm das Dialogsystem darüber Auskunft geben, welche einzelnen Schritte er dafür ausführen muss. Der Gedanke, dass es hilfreich ist, ein System um solche selbstreflexiven Eigenschaften zu erweitern, ist spätestens seit Brian Smiths Arbeit ([Smi82]) bekannt. Er entwickelte eine Programmiersprache (3-LISP), die entsprechende Schnittstellen bereitstellte. Auch Winograds SHRDLU ([Win72]), eines der ersten Dialog- und Planungssysteme im Kontext der Blockswelt, konnte dazu befragt werden, warum es einen bestimmten Plan(schritt) ausgeführt hat. Im Unterschied zu heute gab es damals aber noch keine gut erforschten Repräsentationstechniken und Planungskalküle. SHRDLU führt keine Planung im klassischen Sinn durch. Die Operatorbeschreibungen sind implizit im System enthalten und es verfügt zudem nicht über eine explizite Planrepräsentation. Außerdem konnte lediglich ein bereits ausgeführter Plan im nachhinein erläutert werden.

Bei CONALD geht es darum, eine geeignete Verbalisierung für einen Plan zu finden, dessen Operatoren im wohldefinierten PDDL-Format ([FL01]) vorliegen. Die Berechnung eines Planes wird bislang nach einem Befehl des Benutzers angestoßen. Erforderlich ist eine Erweiterung des Dialogsystems dahingehend, dass auch entsprechende Benutzerfragen einen Plan generieren. Im Unterschied zum Ausführungsmodus wird aber in diesem Erklärungsmodus kein Befehl an die Ausführungsebene weitergeleitet, sondern es erfolgt eine Präsentation des Plans.

In PDDL sind Pläne durch einzelne Planoperatoren definiert (Abbildung 2 zeigt ein Beispiel): Ein Planschritt kann dann zur Ausführung kommen, wenn bestimmte Vorbedingungen erfüllt sind. Nach der Ausführung hat sich der Zustand der Welt geändert, die Änderungen sind in den Nachbedingungen beschrieben.

Bei einem Plan handelt es sich zunächst um ein abstraktes Objekt, das sich durch die Erklärung in der Vorstellung des Benutzers bilden muss. Die in CONALD vorhandene Komponente zur Textgenerierung ist in der Lage, aus der berechneten Abfolge und den vorhandenen Operator-Definitionen entsprechende Erläuterungen zu erzeugen. Eine mögliche Art der Präsentation des Gesamtplans wäre die Nennung der einzelnen Planschritte in einer geeigneten (meist linearen) Reihenfolge. Die Planschritte selbst können genauer erläutert werden:

- Der relevante Ausschnitt der Applikationssituation ist durch die Vorbedingungen gegeben. Eine Verbalisierung dieser Bedingungen verdeutlicht dem Benutzer, in welchem Zustand sich das System befinden sollte, damit dieser Schritt ausgeführt werden kann.
- Die erforderliche Handlung selbst ist im Planoperator durch den Namen der Aktion und die Liste der Parameter beschrieben.
- Der Zustand der Applikation nach Anwendung des Schrittes ist wiederum in den Nachbedingungen festgelegt.

Jedoch sind dabei einige Gegebenheiten zu berücksichtigen:

- Der Plan sollte auf einem gewissen, an einen Menschen angepassten, Abstraktionsniveau präsentiert werden. Oft werden einzelne, kleine Planschritte, die das System intern ausführt, für eine Erklärung nicht relevant sein und den Benutzer eher verwirren als informieren.
- Länge und Inhalt der Präsentation sollten auf die individuellen Vorkenntnisse, Fähigkeiten und Interessen des Benutzers abgestimmt sein. Dabei kann die Ausgabe zunächst an verschiedene Benutzergruppen (Anfänger, Experte) angepasst werden (Individualisierung). In einem weiteren Schritt ist die Adaptation auf den konkreten Benutzer möglich (Personalisierung). Das in CONALD vorhandene Modul zur Benutzermodellierung ist für die Speicherung solcher Informationen geeignet, eine Anpassung des Modells durch Beobachtung des Benutzers ist möglich.
- Ist es nicht möglich, zu einer Anfrage einen Plan zu generieren (das heißt, der Benutzer kann in der gegebenen Situation sein Ziel nicht erreichen), ist auch das in

```
(:action offerconfirmation
:parameters (?t - T_NEW_CATALOG
             ?p - PRICE
             ?o - offer)
:precondition (and (has-catalog ?o ?t)
                  (has-price ?o ?p)
                  (not (has-status ?o confirmed)))
:effect (has-status ?o confirmed))
```

Abbildung 2: Planoperator in Einkaufsdomäne.

geeigneter Form mitzuteilen. Optimal wäre eine Erklärung, warum eine Ausführung nicht möglich ist.

- Führt mehr als ein Plan zum Ziel, ist – wieder unter Berücksichtigung des Benutzermodells – entweder ein passender auszuwählen, oder eine geeignete Form der Präsentation einer Auswahl zu finden.

4 Umsetzung

Da im Dialogsystem CONALD Benutzereingaben als Ziele für einen Planungsprozess behandelt werden, der durch einen Planer berechnet und dann Schritt für Schritt umgesetzt wird, ist es möglich, die vorgestellte Funktionalität zu implementieren. Entscheidend ist dabei, dass das System erkennen muss, dass eine Eingabe in diesem Fall *nicht* bis hin zur Ausführung des Plans bearbeitet werden soll, sondern die Erklärungskomponente in Aktion treten soll. Dazu ist es nötig, den vorhandenen Dialogmanager so zu konfigurieren, dass er neben den in Abschnitt 1 vorgestellten Dialogebenen *Gegenstandsbereich* und *Diskurs* noch die dritte Ebene *Erklärung* kennt und die richtigen Aktionen auslöst.

Literatur

- [FL01] Maria Fox und Derek Long. *PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains*. <http://www.dur.ac.uk/d.p.long/competition.html>, 2001.
- [FLB01] Yves Forkl, Bernd Ludwig und Kerstin Bücher. Dialogue Management in the EMBASSI Realm. Using Description Logics to Reason about Ontology Concept. In Andreas Heuer und Thomas Kirste, Hrsg., *Intelligent Interactive Assistance and Mobile Multimedia Computing (Proceedings of the Workshop IMC2000)*, Seiten 150–153, Rostock, 2001. Neuer Hochschulschriftenverlag.
- [HN01] Jörg Hoffmann und Bernhard Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.
- [KR93] Hans Kamp und Uwe Reyle. *From Discourse To Logic*. Dordrecht, 1993.
- [Lud03a] Bernd Ludwig. Dialogue Understanding in Dynamic Domains. In Henk Zeevat Peter Kühnlein, Hannes Rieser, Hrsg., *Perspectives on Dialogue in the New Millennium*, number 114 in Pragmatics & Beyond New Series, Seiten 243–269. John Benjamins Publishing Company, 2003.
- [Lud03b] Bernd Ludwig. *Ein konfigurierbares Dialogsystem für Mensch-Maschine-Interaktion in gesprochener Sprache*. Dissertation, Erlangen-Nürnberg, 2003.
- [Smi82] Brian Cantwell Smith. *Procedural Reflection in Programmin Languages*. Dissertation, Massachusetts Institute of Technology, 1982.
- [Tra94] David R. Traum. *A Computational Theory of Grounding in Natural Language Conversation*. Dissertation, Rochester, 1994.
- [Win72] Terry Winograd. *Understanding Natural Language*. Academic Press, 1972.