Sonderforschungsbereich 378
Ressourcenadaptive kognitive Prozesse

KI-Labor am Lehrstuhl für Informatik IV

Leitung: Prof. Dr. W. Wahlster

REAL

Universität des Saarlandes
FB 14 Informatik IV
Postfach 151150
D-66041 Saarbrücken
Germany

Tel. 0681 / 302-2363

Memo Nr. 64

Ein Modell für natürlichsprachlichen Zugang zu autonomen Robotern

Eva Stopp

Mai 1997

Sonderforschungsbereich 378
Ressourcenadaptive
Kognitive Prozesse

ISSN 0944-7822

64

Ein Modell für natürlichsprachlichen Zugang zu autonomen Robotern*

Eva Stopp
SFB 378 – Projekt REAL
Fachbereich 14 – Informatik
Universität des Saarlandes
D–66123 Saarbrücken
Eva.Stopp@cs.uni-sb.de

13. Mai 1997

1 Anforderungen

1.1 Allgemeine Anforderungen

Intelligente Roboter haben die Eigenschaft, sich autonom in ihrer Umgebung zu bewegen. Dieses Verhalten setzt voraus, daß ein Sensorsystem existiert, das die Erforschung der Umgebung und die Analyse der aufgenommenen Daten ermöglicht: Tast-, akustische und visuelle Sensoren. Sie benutzen diese Information, um ein Umweltmodell aufzubauen. Die so ermittelte Information ist jedoch nicht immer vollständig. In dieser Situation ist es von Vorteil, wenn Rückfragen an den menschlichen Operator bzgl. fehlender Information gestellt werden können. Eine natürlichsprachliche Schnittstelle sollte also natürliche Sprache nicht nur als reine Kommandosprache benutzen, sondern es sollte vielmehr ein Dialog zwischen Benutzer und autonomem System stattfinden. Auf diese Weise können Mehrdeutigkeiten und Mißverständnisse ausgeräumt werden.

Das Wissen des Roboters über seine Umgebung kann als wertvolle Grundlage dienen, den Dialog an die jeweilige Situation anzupassen. Es werden Mehrfachrepräsentation und redundante Berechnungen vermieden, die sich beispielsweise negativ auf die benötigte Antwortzeit auswirken können. Dies ist gerade in einem solchen reaktiven System wichtig, für das lediglich der augenblickliche Kontext relevant ist.

Möchte man eine natürlichsprachliche Schnittstelle für einen intelligenten Roboter entwerfen, so muß diese auf den Roboter zugeschnitten sein. Für einen andersgearteten Roboter muß sie neu implementiert werden. Gelingt es jedoch ein allgemeines Modell für solche natürlichsprachliche Schnittstellen zu entwerfen, das im Kern roboterunabhängig ist und durch entsprechende Wissensbasen an bestimmte Roboter angepaßt werden kann, so ist dies ein großer Schritt in Richtung Portabilität. Ein weiterer Schritt in diese Richtung ist größtmögliche Unabhängigkeit von der verwendeten Sprache. Den meisten Menschen fällt es leichter, sich in ihrer Muttersprache auszudrücken. Die Anpassung des Dialogsystems an solche Bedürfnisse zeugt von Benutzerfreundlichkeit.

1.2 Anwendungssituationen

Für ein solches Modell sind verschiedene Anwendungssituationen denkbar. Die folgenden vier Aufgaben einer natürlichsprachlichen Schnittstelle werden in dieser Arbeit als die wichtigsten insbesondere innerhalb von Montagedomänen angesehen [Stopp et al., 1994].

*Ich werde hier auf eine Literaturbetrachtung verzichten. Sie wird jedoch an anderer Stelle nachgeholt.

Aufgabenspezifikation

Gerade bei autonomen Robotern spielen Befehle eine zentrale Rolle, da sie Menschen bei bestimmten Tätigkeiten unterstützen bzw. ihnen diese Tätigkeiten sogar völlig abnehmen. Dabei spielt für den Menschen die Abstraktionsebene, die er in einem Befehl benutzt, keine Rolle. Er kann ohne Zögern von Metabefehlen wie *Baue das Objekt zusammen* zu Befehlen wechseln, die beispielsweise die Feinbewegung des Roboters steuern. Dies sollte auch in der natürlichsprachlichen Schnittstelle berücksichtigt werden.

Innerhalb eines Dialogs können Befehle nicht nur durch Imperativformulierungen gegeben werden [Lyons, 1977], sondern auch in deklarativen Sätzen und sogar Fragen enthalten sein. Auf diese Weise werden Dialoge zwar interessanter und vielfältiger, ihre Interpretation ist jedoch wesentlich schwieriger.

Hinzu kommt, daß ein Mensch, der die Situation, in der sich ein Roboter gerade befindet, beobachtet und aus dieser Situation heraus Befehle geben soll, i.a. nicht in der Lage sein wird, die räumliche Situation des Roboters in Koordinaten auszudrücken. Stattdessen werden räumliche Ausdrücke benutzt, die zunächst analysiert und interpretiert werden müssen, d.h. es muß eine Abbildung gefunden werden von dem räumlichen Ausdruck im Kontext auf konkrete Positionen in der Welt des Roboters. Eine solche Abbildung kann nicht immer eindeutig sein, eventuelle geringfügige Abweichungen können jedoch durch autonomes Verhalten ausgeglichen werden. Ähnliches gilt für die verwendeten Objektbezeichner: Eindeutige Bezeichner werden oft durch beschreibende Audrücke ersetzt, die der Sprecher in dieser Situation für eindeutig hält [Searle, 1994]. Auch hier können räumliche Ausdrücke eine Rolle spielen.

Ausführungserläuterung

Eines der Hauptmerkmale eines autonomen Roboters ist seine Fähigkeit, die Reihenfolge von Teilaufgaben in der Durchführung einer Montageaufgabe je nach Situation zu variieren. Eine solche Änderung der normalen Vorgehensweise kann zu Fragen seitens des Benutzers führen. Deshalb sollte dieser – sofern er es möchte – darüber informiert werden können. Je nach Kenntnisstand sollten solche Beschreibungen und Erklärungen unterschiedlich detailliert ausfallen.

Erklärungen zu Fehlerbehebung

Autonome Roboter können nicht nur die Reihenfolge von Teilaufgaben beeinflussen, sondern sind auch in der Lage, Fehlersituationen zu erkennen und sogar zu beheben. Montagepläne werden dabei zur Laufzeit dynamisch an die jeweilige Situation angepaßt. Diese Eigenschaft macht es jedoch für den Operator schwierig, das Verhalten des Roboters vorherzusagen und zu verstehen. Natürlichsprachliche Erklärungen und Beschreibungen, warum und wie ein Plan geändert wurde, würden die Kooperativität des intelligenten Roboters erhöhen. Sie müssen abhängig vom Kontext gegeben werden und unterschiedlich komplex ausfallen können. Gegebenenfalls müssen Fragen des Benutzers direkt beantwortet werden. Dabei sollte es parametrisierbar sein, ob die Erklärungen jedesmal bei Auftreten eines Fehlers gegeben werden oder ob sie durch Fragen des Benutzers ausgelöst werden.

Vielleicht sogar noch wichtiger sind solche Beschreibungen, wenn der Roboter nicht mehr in der Lage ist, mit einer Fehlersituation eigenständig fertigzuwerden. Dann muß der Benutzer informiert werden, damit er bei der Behebung behilflich sein kann.

Austausch von Umgebungsinformation

Aufgrund der dynamischen Natur der Umgebung kann die geometrische und visuelle Information nie vollständig sein, selbst für einen autonomen Roboter mit mehreren unterschiedlichen Sensoren. Deshalb ist ein Dialog zum Informationsaustausch ebenfalls wichtig. Roboter und Benutzer können sich gegenseitig helfen, indem sie Informationen bzgl. Umwelt und Weltmodell austauschen. Der Benutzer sollte beispielsweise das System nach der Position von Objekten fragen können.

1.3 Vorteile eines dialogbasierten Ansatzes

Wie schon erwähnt, ist es für die Interaktion mit einem autonomen Roboter nicht ausreichend, natürliche Sprache einfach als Kommandosprache zu benutzen[Stopp et al., 1994]. Flexible Interaktion mit einem solchen Roboter ist nur dann möglich, wenn dieser auch in der Lage ist zu antworten, d.h. etwas über seine Aufgaben zu berichten, sein Verhalten zu erklären und Informationen über seine Umgebung zu geben. Diese Fähigkeiten machen es für den Menschen leichter, sein Verhalten zu verstehen. Es können Mehrdeutigkeiten und Mißverständnisse ausgeräumt werden. Geschieht diese Kommunikation in natürlicher Sprache, so ist dies eine für den Menschen natürliche Form der Verständigung. Es muß keine komplexe Befehlssyntax erlernt werden, weil der Roboter lediglich diese versteht.

2 Ein roboter- und einzelsprachunabhängiges Modell

2.1 Architektur

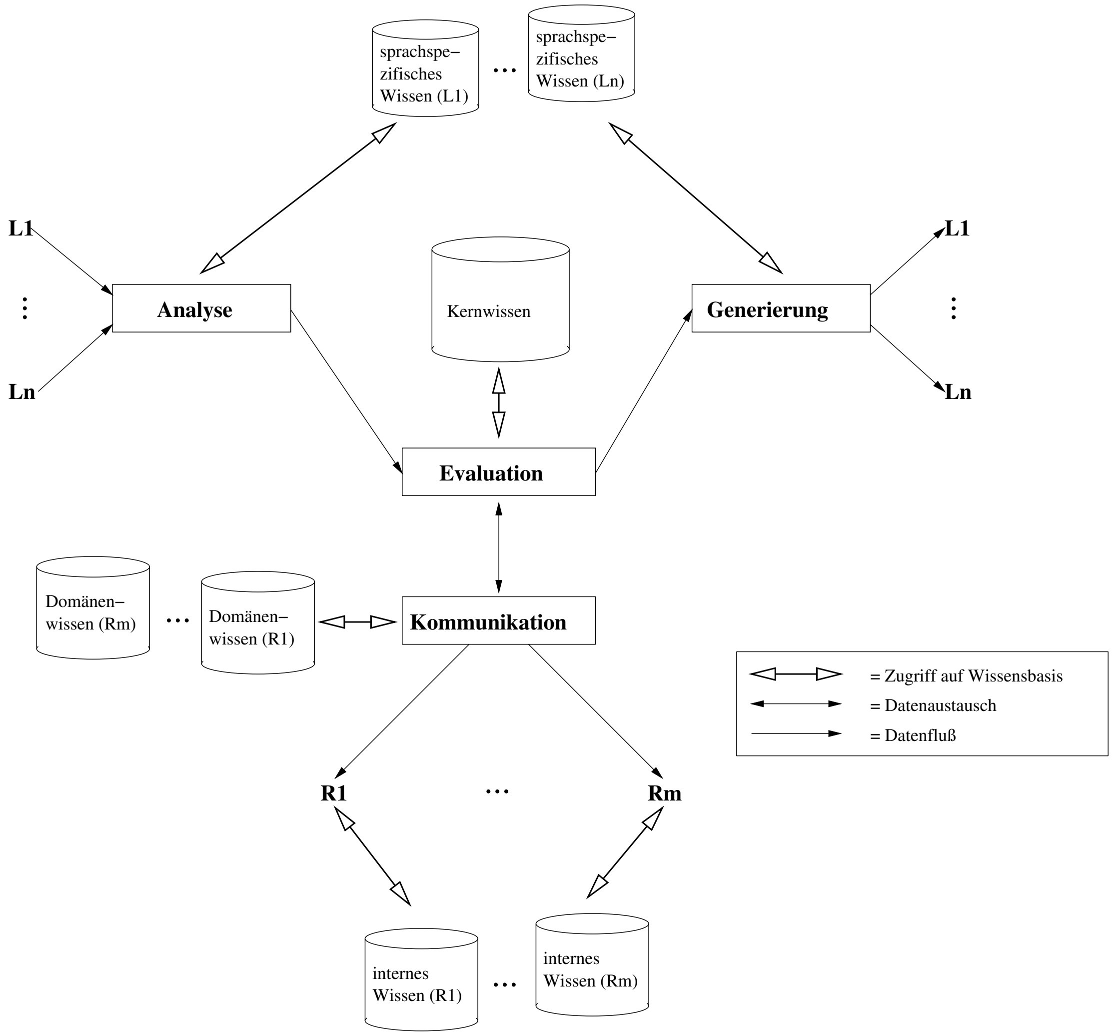

Abbildung 1: Einzelsprachunabhängiges Architekturmodell zur Ansteuerung autonomer Roboter

Um unabhängig von dem jeweils angesprochenen Roboter und außerdem einzelsprachunabhängig zu sein, müssen die Wissensquellen nach diesen Eigenschaften voneinander getrennt werden. Ansonsten ist in dem hier entwickelten Architekturmodell deutlich das allgemeine Verarbeitungsprinzip innerhalb von Dialogsystemen zu erkennen: Analyse, Evaluation, Generierung (siehe Abb. 1). Hinzugekommen ist eine weitere Komponente, die für die Kommunikation mit dem jeweils angesprochenen Roboter zuständig ist. Die Evaluation bildet das einzelsprach- und roboterunabhängige Kernsystem.

Es wird zwischen vier unterschiedlichen Arten von Wissensquellen unterschieden: sprachspezifisches Wissen, Domänenwissen, Kernwissen und roboterinternes Wissen. Das sprachspezifische Wissen ist einzelsprachabhängig, d.h. für jede Sprache gibt es eine eigene Wissensbasis, die morphosyntaktisches, konzeptuelles Wissen – z.B. Abbildung eines Verbs auf eine Aktion – und den sprachabhängigen Teil des semantischen Wissens – z.B. die Abbildung bestimmter räumlicher Ausdrücke auf räumliche Relationen – enthält. Das roboterabhängige Wissen wird hier Domänenwissen genannt. Hier gibt ebenfalls für jeden Roboter eine Wissensbasis, die Wissen über Umwelt, Aufgaben und Pläne enthält. Das Kernwissen ist der sprachunabhängige Teil der Wissensbasen des Modells. Es enthält ein Dialoggedächtnis, allgemeine Dialogkonzepte sowie den sprachunabhängigen Teil des semantischen Wissens – z.B. die Interpretation räumlicher Relationen. Der Roboter selber hat noch internes Wissen, beispielsweise die Interpretation von Kraft- und Drehmomentwerten.

Aufgrund der Trennung der Wissensbasen findet in der Analysekomponente lediglich die syntaktische Analyse und der einzelsprachabhängige Teil der semantischen Analyse statt. Die Evaluation ist dann für den einzelsprachunabhängigen Teil zuständig. Als zentralem Modul kommen ihr auch weitere Aufgaben zu: Erklärungskomponente, Fokussteuerung zur Interpretation und Erzeugung von Personalpronomen, Steuerung der Ausführungserläuterung, Steuerung der Kommunikation, Update von Domänen- und Kernwissen, Erzeugung von Textplänen usw. Die Kommunikationskomponente ist dann für den Datenaustausch mit dem Roboter zuständig, während die Generierungskomponente die von der Evaluation gelieferten Textpläne verbalisiert.

Der Datenfluß verläuft also immer über die Evaluation: bei Wo-Fragen etwa von der Analyse zur Generierung, bei Anweisungen von der Analyse zur Kommunikation und bei Erklärungen von der Kommunikation zur Generierung.

2.2 Umgebungsrepräsentation: Objektinformation

Es kann notwendig sein, daß das natürlichsprachliche System die Weltkoordinaten oder sonstige Merkmale benötigt, um sie an den Benutzer weiterzugeben oder um interne Berechnungen durchführen zu können. Solche Information setzen sich aus dem Namen des Objekts und seiner Beschreibung zusammen, welche aus den Koordinaten, einem Rotationswinkel usw. bestehen kann. Die genaue Beschreibung ist roboterabhängig.

$$< Objektinformation > \; = < Objektname >< Objektbeschreibung >$$
$$< Objektbeschreibung > \; = < Koordinaten >|< Rotationswinkel \mid ...$$

2.3 Aufgabenrepräsentation: Klassifikation von Roboteraktionen

Im Zusammenhang mit Robotern spielen v.a. Befehle, die man ihnen geben kann, eine Rolle. Das hängt damit zusammen, daß sie dem Menschen Arbeit abnehmen sollen: sie müssen also auf irgendeine Art vom Menschen instruierbar sein. Betrachtet man mobile Roboter, so kann man die Aktionen, die sie durchführen können, nach folgenden Fragen klassifizieren (siehe Abb. 2):

- *Wer führt die Aktion aus?*
  Das kann entweder der Roboter als ganzes sein, z.B. als Bewegung des Gesamtroboters, oder Teile des Roboters, z.B. bei Montageoperationen. Manipulierbare Roboterteile sind die Manipulatoren und die Sensoren. Die Fortbewegungsmittel selber, wie etwa die Räder, sollen zum Gesamtroboter gerechnet werden.

- *Was wird manipuliert?*
  Roboter und Manipulatoren können entweder sich selbst oder andere Objekte beeinflussen. Sensoren haben keinen direkten Einfluß auf andere Objekte, sondern werden selber manipuliert.

- *Wie wird manipuliert?*
  Eigenmanipulation von Roboter und Manipulatoren kann sich in Transformation bzw. Bewegung (Rotation, Translation) äußern. Das gleiche gilt für die Manipulation von Objekten durch den Roboter (z.B. Verschieben). Die Manipulatoren können zusätzlich noch Objekte greifen und loslassen, um ihre Montageoperationen durchführen zu können. Sensoren werden aktiviert, deaktiviert oder manipuliert.

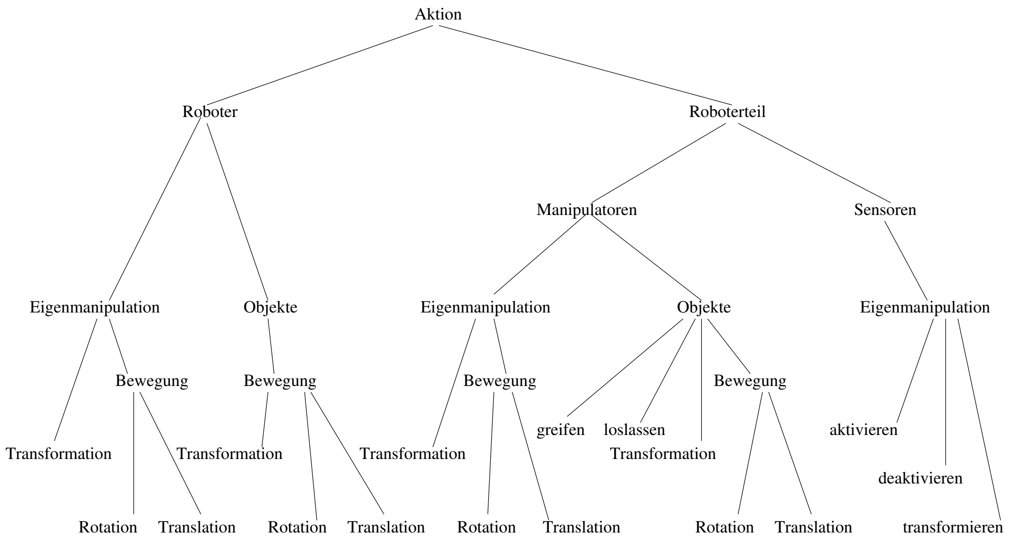

Abbildung 2: Eine Klassifikation von Roboteraktionen

Prägend für eine solche Operation sind also die Aufgabe, die ausgeführt werden soll, der Ausführende (= Agent) und ggfs., je nach Aktion, ein Objekt, das manipuliert wird (= Patient). Geht man nun davon aus, daß sich diese drei Hauptmerkmale jeweils durch ihren Namen und eine Beschreibung ausdrücken lassen, so ist man in der Lage eine reguläre Sprache zu definieren, die Abb. 2 entspricht:

$$
\begin{aligned}
&< Roboterkommando > &&= < Aktion >< Agent >< Patient > \\
&< Aktion > &&= < Aktionsname >< Aktionsbeschreibung > \\
&< Agent > &&= < Agentname >< Agentbeschreibung > \\
&< Patient > &&= < Patientname >< Patientbeschreibung > \\
&< Aktionsname > &&= < bewegen >|< verändern >|< greifen >|< loslassen >| \\
& &&\quad < aktivieren >|< deaktivieren > \\
&< Aktionsbeschreibung > &&= < Koordinaten >|< Rotationswinkel >|< Trajektorie >| \\
& &&\quad < Geschwindigkeit >|< Constraints >| \ldots \\
&< Agentname > &&= < Roboter >|< Manipulator_1 >| \ldots |< Manipulator_n >| \\
& &&\quad < Sensor_1 >| \ldots |< Sensor_m >, \qquad m, n \in N_0 \\
&< Agentbeschreibung > &&= < Koordinaten >| \ldots \\
&< Patientname > &&= < Agent >|< Objekt_1 >| \ldots |< Objekt_l >, \quad l \in N_0 \\
&< Patientbeschreibung > &&= < Koordinaten >| \ldots \\
&< bewegen > &&= < rotieren >|< verschieben >
\end{aligned}
$$

Basierend auf diesen Überlegungen läßt sich eine Aktionshierarchie definieren, die allgemein gehalten ist, so daß erst eine Instantiierung konkrete roboterabhängige Eigenschaften berücksichtigt. Es können im Montage- und Servicebereich i.a. also folgende Aktionen unterschieden werden (siehe Abb. 3): transformieren, Aktionen zur Aktivierungssteuerung (aktivieren, deaktivieren), bewegen (Rotation, Translation), aufnehmen (im Spezialfall: greifen), plazieren (loslassen, ablegen, fügen).

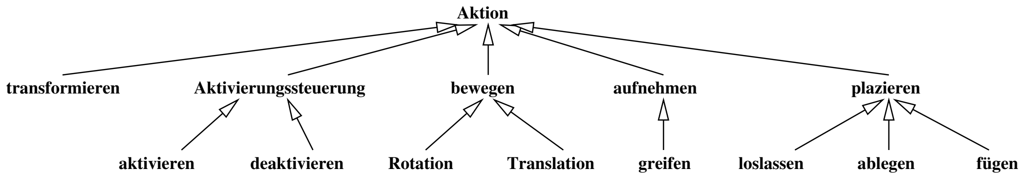

Abbildung 3: Aktionshierarchie

Die notwendigen Informationen, die sich ja auch schon in der oben aufgeführten Sprache widerspiegeln, werden in den Slots der jeweiligen Klassen der Hierarchie gespeichert:

- **Name**: Der Name der Aktion ermöglicht einen leichteren Lexikonzugriff.
- **Agent**: Dieser Slot enthält die Objektinstanz, die an der Aktion als Agent beteiligt ist, d.h. Agentname und Agentbeschreibung.
- **Patient**: vgl. Agent, nur daß es sich hierbei um das Objekt handelt, mit dem etwas geschehen soll.
- **Beschreibung**: Hier ist die Aktionsbeschreibung zu finden. Der Slot wird zur Laufzeit anhand des Domänenlexikons gefüllt, er ist also sowohl von der Aktion als auch von der Domäne abhängig. Auf die einzelnen Informationen – Parameter des Roboterbefehls – kann über Schlüsselwörter zugegriffen werden.

Um von einem Verb zu einem konkreten Roboterbefehl zu kommen, muß dieses mit Hilfe des sprachspezifischen Wissens – im folgenden soll es **Sprachlexikon** genannt werden – mit einer Aktionsklasse verbunden werden. Durch das sogenannte Domänenwissen – **Roboterlexikon** – wird die Klasse dann mit der konkreten Syntax des aktuellen Roboters assoziiert.

Aufgabe des Sprachlexikons ist es also, die Verben auf Aktionsklassen abzubilden. Es kann dem sprachspezifischen Wissen zugeordnet werden und ist ein Konzeptlexikon, das die Wörter der jeweiligen Sprache mit Konzepten des Systems in Verbindung bringt. Man könnte sich jedoch auch einzelne domänenabhängige Versionen vorstellen, so daß die Größe des Lexikons auf die jeweilige Domäne beschränkt werden könnte.

Das Roboterlexikon enthält eine Assoziation jedes (Domänen-) Befehls mit der zugehörigen Befehlssyntax des benutzten Roboters sowie eine Abbildung der in der Befehlssyntax vorkommenden Parameter auf Zugriffe innerhalb der Aktionsinstanz. Es hat folgende Aufgaben:

- Es dient einerseits zum Füllen der Aktionsbeschreibung in der aktuellen Instanz. Auf diese Weise ist garantiert, daß nur Parameter enthalten sind, die zu dem aktuell benutzten Befehl des angesprochenen Roboters gehören.
- Andererseits dient die Befehlssyntax dazu, einen Roboterbefehl in der verlangten Syntax zu erzeugen.
- Für jedes der vorkommenden Symbole existiert ein Eintrag im Lexikon, der erklärt, wo diese Information zu finden ist. Die Auswertung ist also rekursiv.
- Es kann auch Regeln enthalten, z.B. um verschiedene Varianten eines Befehls erzeugen zu können, abhängig davon, welches Objekt die Rolle des Patienten einnimmt.

2.4 Ausführungsrepräsentation: Pläne und Fehlerbehebung

Die Pläne des Roboters können zur Ausführungserläuterung genutzt werden. Hierzu muß der Name des jeweiligen Plans mit den einzelnen Planschritten ausgetauscht werden. Außerdem muß dem System der aktuelle Planschritt mitgeteilt werden, um eine Beschreibung im Gesamtzusammenhang geben zu können. Art und Umfang dieser Informationen sind jedoch stark domänenabhängig, gegebenenfalls muß auch noch das Sprachlexikon angepaßt werden.

$< Planname >< Planschritt_1 > ... < Planschritt_n >, n \in N_0$

Sofern der Roboter eine Fehlerdiagnosekomponente mit Fehlerbehebung besitzt, kann diese zum besseren Verständnis des Benutzers beitragen. Hierbei sind folgende Informationen wichtig:

- Wo ist der Fehler aufgetreten (während der Durchführung welcher Befehle)?
- Welcher Fehler ist aufgetreten?
- Welche Ursache wird vermutet?
- Wie soll der Fehler behoben werden (High-Level-Ziel, evt. auch gesamter Plan)?

Es sind noch weitere Informationen denkbar, die zwischen Roboter und natürlichsprachlichem System ausgetauscht werden können. So können Steuerbefehle auf höchster Ebene, z.B. allgemeine Initialisierungs- oder Rücksetzungsbefehle an den Roboter, wichtig sein:

$< refresh >|< restart >|< init >$

Literatur

[Lyons, 1977] **J. Lyons**. Statements, Questions and Commands. In: A. Zampolli (Red.), *Linguistic Structures Processing*, S. 255–280. North Holland, Amsterdam, etc, 1977.

[Searle, 1994] **J. R. Searle**. *Sprechakte: Ein sprachphilosophischer Essay*, Bd. 458 von *Wissenschaft*. Suhrkamp Taschenbuch, Frankfurt/Main, Deutschland, 6. Aufl., 1994.

[Stopp et al., 1994] **E. Stopp, K.-P. Gapp, G. Herzog, T. Längle und T. C. Lüth**. Utilizing Spatial Relations for Natural Language Access to an Autonomous Mobile Robot. In: B. Nebel und L. Dreschler-Fischer (Red.), *KI-94: Advances in Artificial Intelligence*, S. 39–50. Springer, Berlin, Heidelberg, 1994.