

# Natürlichsprachliche Kommunikation im Rahmen der Datenbanksemantik

Roland Hausser  
14. September 2000

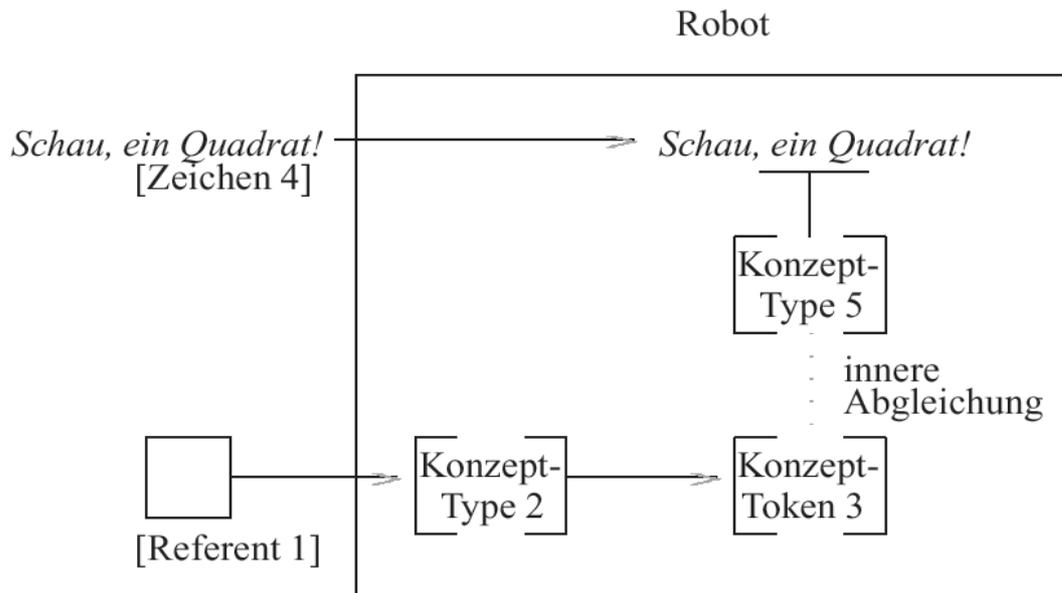
## Abstract

*Database semantics is a computational model of natural language communication based on the following innovations. One is a new data structure, called a wordbank, which stores content in the form of distributed, bidirectionally pointered feature structures; these code concatenated propositions and serve as the semantic representation for language interpretation and production. The other is the time-linear algorithm of left-associative grammar (LA-grammar); it is used for reading content coded in natural language into and out of the wordbank. The system is presented as a declarative model of a cognitive agent which functionally integrates the procedures of interpretation, conceptualization, production, query, and inference.*

– Datenbanksemantik ist ein Computermodell der natürlichsprachlichen Kommunikation, das auf folgenden Innovationen beruht. Die eine ist eine neue Datenstruktur, *Wordbank* genannt, in der Inhalte als verteilte, bidirektional verzeigte Merkmalstrukturen gespeichert werden; diese kodieren verkettete Propositionen und dienen als semantische Repräsentation zur Interpretation und Produktion von Sprache. Die andere ist der zeitlineare Algorithmus der linksassoziativen Grammatik (*LA-grammar*); dieser wird für das Ein- und Auslesen natürlichsprachlich kodierte Inhalte in die, bzw aus der, *Wordbank* verwendet. Das System wird als deklaratives Modell eines kognitiven Agenten präsentiert, das die Prozeduren der Interpretation, Konzeptualisierung, Produktion, Befragung und Inferenz funktional integriert.

## 1 Datenbankmetapher natürlichsprachlicher Kommunikation

Bei der Interaktion mit einem Standardcomputer ist das Verstehen der natürlichsprachlichen Zeichen fast ganz auf den Benutzer beschränkt. Wenn ein Benutzer z.B. in einer Datenbank nach einem roten Objekt sucht, wird das Wort **rot** vor der Eingabe in den Computer und später wieder bei der Ausgabe vom Benutzer verstanden. Innerhalb des Computers wird **rot** dagegen als ein Zeichen manipuliert, das bezüglich der denotierten Farbe uninterpretiert ist.



**Abb. 1:** Unmittelbare Referenz auf der Grundlage interner Abgleichung

Was für einen Standardcomputer gilt, trifft jedoch auf die Mensch-Maschine-Kommunikation nicht allgemein zu. Die heutige Technologie ermöglicht den Bau eines autonomen Roboters, den man beauftragen kann, einen Gegenstand zu bringen, dessen Typ ihm bekannt ist, den er aber bisher noch nie gesehen hat (z.B. das rote Buch auf dem Schreibtisch im Nebenzimmer). Wenn diese Maschine in der Lage ist, spontan eine offene Menge verschiedener Aufgaben dieser Art auszuführen, hat sie ein Sprachverständnis, das auf einer bestimmten Abstraktionsebene mit den entsprechenden kognitiven Vorgängen beim Menschen als funktional äquivalent angesehen werden kann.

Die Interaktion eines solchen Roboters mit seiner Handlungsumgebung basiert auf Mustern, die auf die herein- und herausgehenden Parameterwerte seiner Erkennungs- und Handlungskomponenten abgeglichen werden. Diese Muster konstituieren eine prozedurale Definition von Konzepten, die neben ihrer Funktion im nichtsprachlichen Erkennen und Handeln eine zweite Funktion als Sprachbedeutungen übernehmen können.<sup>1</sup>

Die Sprachinterpretation des Roboters muß funktionieren, ohne daß dafür eine externe (metasprachliche) Relation zwischen dem Zeichen und dem Referenten postuliert wird. Dies wird über das Prinzip der *internen Abgleichung* geleistet, das im folgenden mit der sprachlichen Referenz auf ein Quadrat illustriert wird.

Der Roboter erkennt das Referenzobjekt (1), indem ein geeigneter Konzept-Type (2) auf die hereinkommenden Parameterwerte (hier *bitmap outline*) abgeglichen und das Referenzobjekt intern als ein Konzept-Token (3) instantiiert wird. Die Oberfläche des Sprachzeichens (4) wird entsprechend erkannt. Der Oberfläche **Quadrat** ordnet das Lexikon denselben Konzept-Type wie (2) als wörtliche Bedeutung (5) zu.

Die Beziehung zwischen der Sprachebene und der Ebene der nichtsprachlichen Kognition (auch Kontextebene genannt) wird über das Abgleichen zwischen dem lexikalischen Konzept-Type (5) und dem referentiellen Konzept-Token (3) etabliert. Eine lexikalisch als Type definierte wörtliche Bedeutung kann auf beliebig viele Referenz-Token abgeglichen werden. Dies modelliert die Flexibilität, die die natürlichen Sprachen von den Logik- und den Programmiersprachen unterscheidet.

Die in Abb. 1 illustrierte Art der Referenz wird *unmittelbar* genannt, weil Zeichen und Referent in der aktuellen Handlungsumgebung präsent sind. Dagegen wird die Referenz auf Objekte, die nicht in der aktuellen Handlungsumgebung enthalten sind, z.B. auf die Person J.S. Bach, *mittelbar* genannt. Bei der mittelbaren Referenz interagiert der Agent mit seiner Handlungsumgebung allein auf der Sprachebene, wobei als Referenten Objekte im Erinnerungsspeicher verwendet werden.

Die interne Abgleichung wird in Abb. 1 an dem einzelnen Wort **Quadrat** illustriert. Das Grundprinzip kann jedoch auf voll entwickelte natürlichsprachliche Kommunikation generalisiert werden, und zwar mit Hilfe der *Datenbankmetapher*. Diese geht davon aus, daß das Wissen der Sprecher-Hörer in der Form von Datenbanken repräsentiert ist. Kommunikation ist erfolgreich, wenn der Sprecher eine bestimmte Teilstruktur seiner Datenbank in Sprache kodiert und der Hörer diese in seiner Datenbank *analog* rekonstruiert, und zwar sowohl bzgl. der korrekten Dekodierung des Inhalts als auch bzgl. der korrekten Speicherung an einer entsprechenden Stelle.

Die Datenbankmetapher geht von vornherein davon aus, daß die kommunizierenden Datenbanken in kognitive Agenten (z.B. Roboter) eingebettet und dabei Schnittstellen zum sprachlichen und nichtsprachlichen Erkennen und Handeln

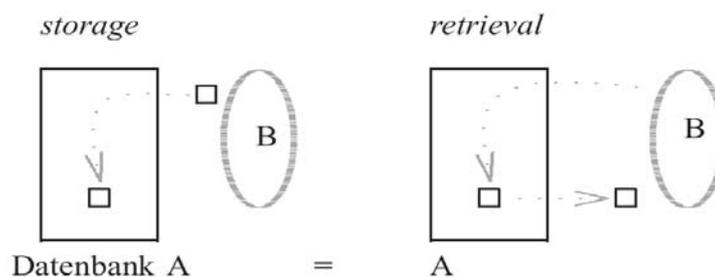
realisiert sind. Deshalb kann die Datenbankmetapher auf Aussagen, Fragen und Aufforderungen gleichermaßen angewendet werden. Bei Aussagen speichert der Hörer den vom Sprecher kodierten Datenbankinhalt an einer entsprechenden Stelle seiner Datenbank. Bei Fragen bestimmt der Hörer, wo die vom Sprecher erbetene Information gespeichert ist, bzw. gespeichert sein sollte. Bei Aufforderungen erstellt der Hörer eine kognitive Rekonstruktion der vom Sprecher erbetene Handlung.

Die programmiertechnische Umsetzung der Datenbankmetapher umfaßt folgende Aufgaben.

### 1.1 Realisierungsaufgaben der Datenbankmetapher

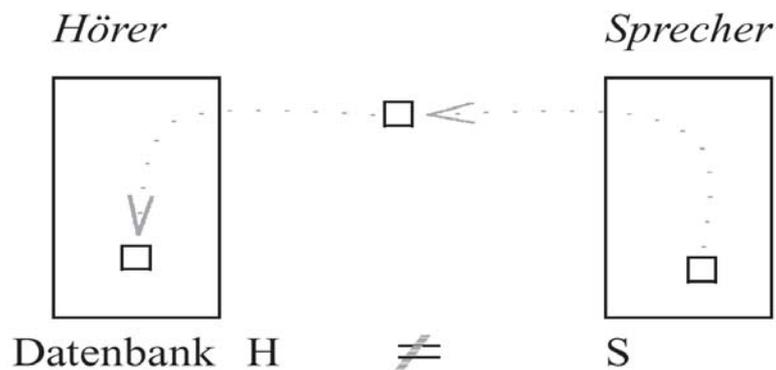
1. Kodierung von Datenbankinhalten in natürliche Sprache (Sprechermodus)
2. Dekodierung von Sprachzeichen in Datenbankinhalte (Hörermodus)
3. Befragung der Datenbank nach bestimmten Inhalten (Sprecher befragt den Hörer)
4. Inferenzen auf Datenbankinhalten zur Bestimmung der Identität zwischen Substantiven, der temporalen Relationen, der Interpretation nichtwörtlicher Verwendungen (z.B. Metaphern) etc.
5. Indizierung der Speicherstelle, von der die in Sprache kodierten Sprecherinhalte stammen, so daß der Hörer die entsprechenden Speicherstelle für die rekonstruierten Inhalte bestimmen kann.

Um zunächst die Natur des Indizierungsproblems zu erklären, vergleichen wir die Interaktion zwischen Benutzer und konventioneller Datenbank mit der zwischen Sprecher und Hörer.



**Abb. 2:** *Interaktion mit einer konventionellen Datenbank*

Die Interaktion findet zwischen zwei verschiedenen Entitäten statt, dem Benutzer (Ovale) und der Datenbank (große Vierecke), wobei die kleinen Vierecke die Sprachzeichen repräsentieren, die als Ein- und Ausgabe dienen. Der Benutzer steuert die Speicher- und Abrufoperationen mit Hilfe einer Programmiersprache, deren Befehle auf der Datenbank als elektronische Prozeduren ausgeführt werden.



*Abb. 3: Interaktion zwischen Sprecher und Hörer*

Hier repräsentieren die großen Vierecke kognitive Agenten, die natürlich oder künstlich sein können. Es gibt keinen Benutzer. Vielmehr findet die Interaktion zwischen zwei gleichen und gleichberechtigten Agenten statt, die ihren Informationsfluß gegenseitig steuern, indem sie abwechselnd den Sprecher- und den Hörermodus übernehmen (*turn taking*). Der Sprecher steuert die Sprachproduktion als autonomer Agent. Die Interpretation des Hörers wird von den hereinkommenden Sprachzeichen gesteuert.

## 2 Inhaltsdarstellung: verkettete Propositionen

Die beiden ersten Aufgaben zur Realisierung der Datenbankmetapher, also die Kodierung und Dekodierung, erfordern eine allgemeine, Sprach- und Domänen-unabhängige Methode zur Darstellung von Inhalten. Hierfür verwenden wir den klassischen Begriff einer Proposition. Im Einklang mit Aristoteles (384–322 v. Chr.) sind Propositionen einfache Zustandsdarstellungen. Sie sind so allgemein und abstrakt, daß sie gleichzeitig als Teilzustände realer oder möglicher Welten und als Bedeutung sprachlicher Sätze aufgefaßt worden sind.

Propositionen setzen sich aus drei Arten von Grundelementen zusammen, den *Argumenten*, den *Funktoren* und den *Modifikatoren*. Eine elementare Proposition besteht aus einem Funktor, der mit einer für ihn charakteristischen Anzahl von Argumenten kombiniert wird. Die Modifikatoren sind optional und können auf Funktoren oder Argumenten angewendet werden.

In der Welt sind die Argumente die *Objekte* (im allgemeinsten Sinn), die Funktoren die (intrapropositionalen) *Relationen* zwischen den Objekten und die Modifikatoren die *Eigenschaften* der Objekte oder Relationen. Mit diesen Grundelementen ist eine allgemeine Ontologie zur Darstellung kognitiver Zustände gegeben - im Unterschied zu spezielleren Ontologien, wie etwa zur Beschreibung der Welt aus Sicht der Physik (basierend auf Atomen, Masse etc.), der Biologie (basierend auf Stoffwechsel, Fortpflanzung etc.) oder der Ökonomie (basierend auf Angebot, Nachfrage, Geldmenge etc.).

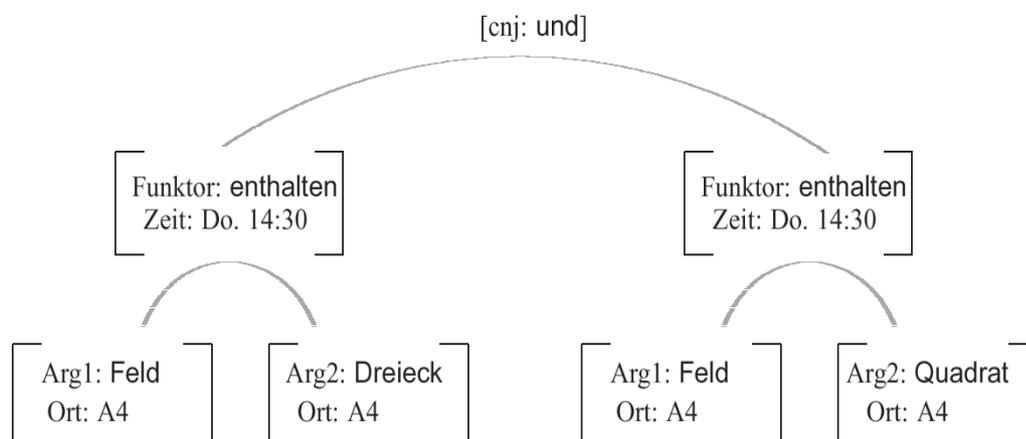
In den natürlichen Sprachen sind die Argumente die *Substantive*, die Funktoren die *Verben* und die Modifikatoren die *Adjektive* (wobei Adnominalen auf Substantive und Adverbialen auf Verben angewendet werden). Die Substantive, Verben und Adjektive umfassen die *Inhaltswörter* einer Sprache.

<i>Logik</i>	<i>Welt</i>	<i>Sprache</i>
1. Argument	Objekt	Substantiv
2. Funktor	Relation	Verb
3. Modifikator	Eigenschaft	Adjektiv

**Abb. 4:** Die Bausteine elementarer Propositionen

Elementare Propositionen können mit *Operatoren* verändert (z.B. Negation) oder miteinander verknüpft werden (z.B. Disjunktion). In den natürlichen Sprachen entsprechen den Operatoren die *Funktionswörter*. In der Welt entsprechen den Operatoren u.a. die *extrapropositionalen Relationen*.

Als Beispiel einer Inhaltsdarstellung mittels verketteter Propositionen betrachten wir einen Roboter, der gerade ein Dreieck und ein Viereck in Feld A2 seiner Handlungsumgebung findet und diese Situation automatisch analysiert. Wir gehen davon aus, daß die Konzepte **Feld**, **Dreieck**, **Viereck** und **enthalten** als Programme realisiert sind, die von der visuellen Wahrnehmung des Roboters gelieferte Bitmap-Umriss analysieren.



**Abb. 5:** Verkettung zweier elementarer Propositionen

Die elementaren I-Propositionen in Abb. 5 können mit **Feld enthält Dreieck** und **Feld enthält Viereck** paraphrasiert werden. Sie sind über die extrapropositionale Relation **und** miteinander verkettet.

Die Merkmalstrukturen für Argumente (Objekte, Substantive) bestimmen den Ort, an dem das System ihnen begegnet ist. Die Merkmalstrukturen für Funktoren (intrapropositionale Relationen, Verben) bestimmen den *Zeitpunkt*, an dem die Relation festgestellt wurde. Bei Modifikatoren (Eigenschaften, Adjektiven) stimmt die raum-zeitliche Bestimmung mit derjenigen der modifizierten Struktur überein.

### 3 Datenstruktur einer Wortbank

Das in Abb. 5 illustrierte Format verwendet *graphische* Mittel, um die Funktor-Argument-Struktur des Inhalts auszudrücken. Diese sind gut geeignet, um semantische Intuitionen auszudrücken weshalb Baumstrukturen in der Linguistik sehr beliebt sind. Für eine Implementierung der natürlichsprachlichen Kommunikation im Rahmen der Datenbankmetapher sind graphische Darstellungen jedoch ungeeignet. Denn Datenbanken kodieren Relationen abstrakt, wobei in Beziehung stehende Einheiten beliebig weit von einander gespeichert werden können.

Um eine einfache und effiziente Prozedur zum natürlichsprachlichen Einbetten und Extrahieren propositionaler Inhalte zu unterstützen, rekodieren wir das graphische Format von Abb. 5 als Tabelle aus Merkmalstrukturen, wobei die Funk-

tor-Argumentstruktur elementarer Propositionen (intrapropositionale Relationen) und die Verkettung elementarer Propositionen (extrapropositionale Relationen) mit Hilfe von bidirektional verzeigerten Attributen ausgedrückt wird. Diese neuartige Datenstruktur heißt *Wortbank*.<sup>2</sup>

TYPES	PROPLETS	
[Konzept: enthalten] [Rolle: Funktor]	[funkt: enthalten ARG:Feld Dreieck conj: 23 und 24 prn: 23]	[funkt: enthalten ARG:Feld Quadrat conj: 23 und 24 prn: 24]
[Konzept: Feld] [Rolle: Argument]	[arg: Feld FUNK: enthalten id: 7 prn: 23]	[arg: Feld FUNK: enthalten id:7 prn: 24]
[Konzept: Quadrat] [Rolle: Argument]	[arg: Quadrat FUNK: enthalten id: 9 prn: 24]	
[Konzept: Dreieck] [Rolle: Argument]	[arg: Dreieck FUNK: enthalten id: 8 prn: 23]	

**Abb. 6:** *Rekodierung von Abb. 5 als Wortbank*

Die Daten sind als alphabetisch geordnete Listen von Types angeordnet, wobei jedem Type eine offene Anzahl von Token folgt. Die Types drücken die Grundeigenschaften der Inhaltswörter aus, wie z.B. ihre grammatische Rolle. Die Token, *Proplets*<sup>3</sup> genannt, fungieren als die Elemente der einzelnen Propositionen. Eine Zeile, bestehend aus einem Type gefolgt von einer offenen Anzahl von Proplets, wird eine *Tokenzeile* genannt.

Proplets, die zu derselben Proposition gehören, werden durch ihre gemeinsame Propositionsnummer **prn** zusammengehalten. Merkmale, die zusammenhängende Proplets über intra- und extrapropositionale Relationen bestimmen, heißen *Fortsetzungsprädikate*.

In intrapropositionalen Relationen bestimmt ein Argument (Objekt, Substantiv) den zugehörigen Funktor, ein Funktor (Relation, Verb) die zugehörigen Argumente und ein Modifikator (Eigenschaft, Adjektiv) das zugehörige Modifikandum und umgekehrt. In extrapropositionalen Relationen kann ein Argument die Identität oder Nicht-Identität mit anderen Argumenten, und ein Funktor konjunktionale Relationen (z.B. **und**, **dann**, **weil**) mit anderen Funktoren bestimmen.

Die Tokenzeilen einer Wortbank entsprechen der Struktur einer klassischen *Netzwerkdatenbank*. Eine Netzwerkdatenbank definiert eine 1:n Relation zwischen zwei Arten von Verbänden, den *owner*- und den *member*-Verbänden. In einer Wortbank fungieren die Types als *owner* und die zugehörigen Proplets als *member records*.

Die Kodierung propositionaler Inhalte als verteilte, bidirektional verzweigte Merkmalstrukturen in der Datenbansemantik ist nicht nur eine technische Bequemlichkeit. Sie ist vielmehr entscheidend für eine einheitliche Behandlung der NL-Interpretation, Inferenz und NL-Produktion.

## 4 Intuitive Überlegungen zur Definition einer Datenbanksprache

Die Definition einer Datenbanksprache (DBS) zum automatischen Ein- und Auslesen von Inhalten in die Wortbank entspricht der Definition z.B. des Prädikatenkalküls. So wie der Prädikatenkalkül als künstliche Sprache entwickelt wurde, die bestimmte wahrheitskonditionale Eigenschaften der natürlichen Sprachen modelliert, so wird DBS dafür entwickelt, die automatischen Ein- und Ausleseprozeduren zu modellieren, die für die natürlichen Sprachen charakteristisch sind. Und so wie der Prädikatenkalkül bzgl. eines mengentheoretisch definierten Modells interpretiert wird, wird DBS bzgl. einer Datenbank (Wortbank) mit bidirektional verzweigten Merkmalstrukturen (Proplets) interpretiert.

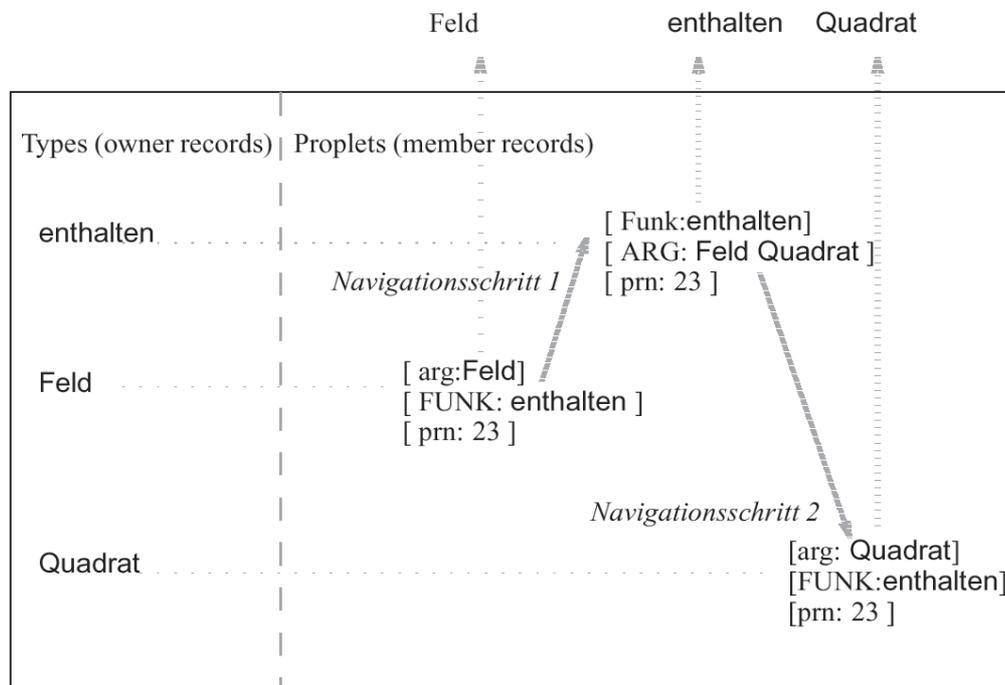
Wie ein Modell, ist eine Wortbank sprachunabhängig. Während jedoch ein Modell als Darstellung der Welt konzipiert ist, dient eine Wortbank zur Darstellung kognitiver Inhalte. Außerdem werden Elementarbedeutungen wie *rot*, *Dreieck* und *enthalten* in der logischen Semantik extensional behandelt, während sie



**Feld**, und die intrapropositionalen Fortsetzungsprädikate (Großbuchstaben) mit dem Wert NIL (hier durch den leeren Wert dargestellt) enthält. Die Interpretation der ersten syntaktischen Komposition kopiert den Propletnamen **Feld** in das ARG-Merkmal von **enthalten**, und den Propletnamen **enthalten** in das FUNK-Merkmal von **Feld**. Die Interpretation der zweiten Komposition kopiert den Propletnamen **enthalten** in das FUNK-Merkmal von **Dreieck** und den Propletnamen **Dreieck** in das ARG-Merkmal von **enthalten**.

Das Ergebnis ist eine (ungeordnete) Menge der drei koindizierten Proplets **Feld**, **enthalten** und **Dreieck**. Diese Proplets sind autonome Entitäten, die gemeinsam die Funktor-Argument-Struktur ‘ihrer’ Proposition über ihre jeweiligen Fortsetzungsprädikate spezifizieren. Das Indizierungsproblem wird automatisch gelöst, indem jedes Proplet am Ende seiner alphabetisch geordneten Tokenzeile in der Wortbank gespeichert wird. Die Datenbanksemantik verwendet also die Propletnamen als Primärschlüssel.

Als nächstes betrachten wir die Produktion eines Satzes aus der Wortbank:



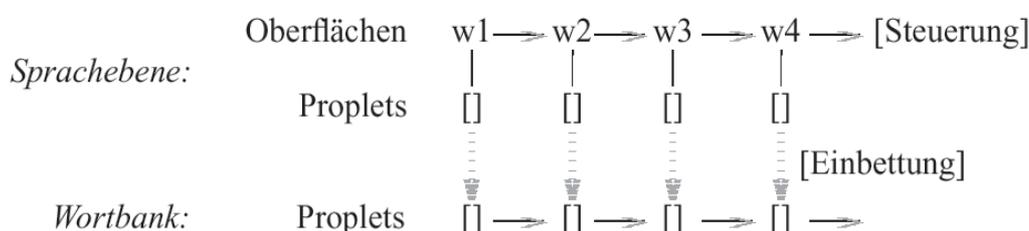
**Abb. 8:** DBS-Produktion von *Feld enthält Quadrat*

Eine Wortbank geht über eine klassische Netzwerkdatenbank insofern hinaus, als sie im Rahmen ihre Verbundstrukturen *mögliche Fortsetzungen* definiert. Diese bilden die Grundlage für Operationen, die konventionelle Datenbanken nicht zur Verfügung stellen, nämlich eine autonome, zeitlineare Navigation durch die verketteten Propositionen der Datenbank.

Die Navigation beginnt z.B. beim Proplet **Feld** mit der Propositionsnummer (prn) 23. Aufgrund des Fortsetzungsprädikats [FUNK: **enthalten**] sucht der Navigationsalgorithmus alphabetisch nach der Tokenzeile von **enthalten** und bewegt sich vom Type zum Token mit der prn 23. Das 'nächste' Proplet **enthalten** hat das Fortsetzungsprädikat [ARG: **Feld Quadrat**]. Der erste Wert **Feld** bestätigt die vorgehende Navigation. Der zweite Wert **Quadrat** ist das neue 'nächste' Proplet. Es wird gefunden, indem der Navigationsalgorithmus alphabetisch nach der Tokenzeile von **Quadrat** sucht und vom Type zum Token mit der prn 23 geht. Auf diese Weise wird Proposition 23 vollständig traversiert. Dabei kann der Inhalt automatisch herausgelesen werden, indem der Wert des ersten Attributes der traversierten Merkmalstrukturen automatisch kopiert und geäußert wird – wie die oberste Zeile von Abb. 8 zeigt.

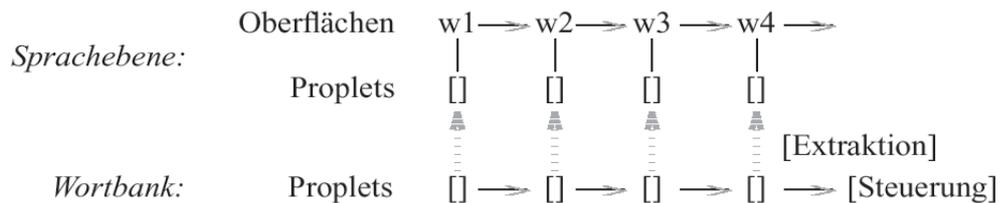
Nach einer intrapropositionalen Navigation kann als nächstes eine extrapropositionale zu einer anderen Proposition folgen, entweder auf der Grundlage einer Identität zwischen Argumenten (Objekten, Substantiven) oder einer Konjunktion zwischen Funktoren (Relationen, Verben). Auf diese Weise kann der Inhalt einer Wortbank traversiert werden, entweder zufällig im Sinne der freien Assoziation oder nach bestimmten Strategien oder Mustern. Diese Navigationen sind ein einfaches Model des Denkens. Wenn das System im Sprechermodus ist, äußert es Wörter, die den traversierten Knoten entsprechen.

Auf einer abstrakteren Ebene als Abb. 7 und 8 kann der zeitlineare Charakter der DBS-Interpretation und -Produktion folgendermaßen beschrieben werden:



**Abb. 9:** Schema der DBS-Interpretation (Analyse)

Das System folgt den Oberflächen der Zeichen, bestimmt ihre Bedeutungen (Proplets) über das Lexikon, ergänzt die Werte der Fortsetzungsprädikate bei jedem zeitlinearen syntaktisch-semantischen Ableitungsschritt und ordnet die vervollständigten Proplets in die entsprechende Tokenzeile der Wortbank.



**Abb. 10:** Schema der DBS-Produktion (Generierung)

Das System navigiert durch die verketteten Propositionen der Wortbank, paßt auf jedes traversierte Proplet einen entsprechenden Lexikoneintrag und äußert dessen Oberfläche.

Die Schemata 9 und 10 genügen der natürlichen Auffassung, daß Interpretation ( $\downarrow$ ) und Produktion ( $\uparrow$ ) inverse vertikale Prozesse sind. Trotzdem haben Interpretation und Produktion ihre eigentliche Hauptrichtung gemeinsam, nämlich die Richtung der horizontalen zeitlinearen Grundstruktur ( $\rightarrow$ ).

## 5 Begriff des Fragments

Die bisherigen Überlegungen zur Interpretation und Produktion natürlicher Sprache können als 'Fragment' formuliert werden. Der Fragmentbegriff wurde von R. Montague für die vollkommen explizite Definition eines formales Systems zur Analyse natürlicher Sprachen geprägt. Ein Fragment ist vollständig bzgl. seiner Funktion oder Funktionen, kann aber zunächst nur eine kleine, repräsentative Untermenge der fraglichen natürlichen Sprache behandeln. Nachdem ein Fragment im Grundsatz funktioniert, soll es Schritt für Schritt erweitert werden, bis es schließlich die ganze Sprache abdeckt.

Montagues Ansatz, auch Montague-Grammatik genannt, erforderte die Neuerung einer expliziten Übersetzung analysierter natürlichsprachlicher Ausdrücke in Formeln der intensionalen Logik, die mit den bekannten Methoden der Modell-

theorie wahrheitskonditional interpretiert werden können. Als Montagues Übersetzungsalgorithmus im Prinzip funktionierte, wurde er auf die Behandlung neuer Sprachkonstruktionen erweitert, die wiederum Innovationen bzgl. ihrer mengentheoretischen Interpretation erforderten.<sup>4</sup>

Während Montague eine maximal explizite und elegante Bedeutungsbeschreibung mit den Mitteln der Mengentheorie anstrebte, ist unser Ziel eine deklarative Definition, die für eine effiziente prozedurale Implementierung geeignet ist.

### Montague-Grammatik

natürliche Sprache  
 $\Downarrow$  *Übersetzung*  
 Logiksprache  
 $\Downarrow$  *Interpretation*  
 mengentheoretisches Modell  
*Welt*

### Datenbanksemantik

natürliche Sprache  $\Rightarrow$   
*Interpretation*  
 natürliche Sprache  $\Leftarrow$   
*Produktion*  
 (nsb-Erkennen)  $\Rightarrow$   
 (nsb-Handeln)  $\Leftarrow$   
*Welt*

DATENBANK

*kognitiver Agent*

**Abb. 11:** Montague-Grammatik und Datenbanksemantik im Vergleich

Ontologisch behandelt Montague die natürlichsprachlichen Zeichen und das mengentheoretische Modell als Teil der ‘Welt’, die vergangene, gegenwärtige, zukünftige und mögliche Zustände umfaßt. Seine semantische Methode basiert auf einer *metasprachlichen* Definition der Sprachzeichen, des Modells und der Beziehung zwischen Sprachzeichen und Modell.

Das DBS Fragment behandelt Bedeutungen dagegen ontologisch als kognitive Inhalte. Die semantische Methode basiert auf *kognitiven Prozeduren*, mit denen die Beziehung zwischen der Sprache und den Objekten der Welt etabliert wird. Dabei wird die Welt nicht definiert, sondern als die reale Handlungsumgebung des Agenten vorausgesetzt.

## 6 Rolle der nichtsprachlichen Kognition

Die schematische Darstellung der Datenbanksemantik in Abb. 11 unterscheidet zwischen der sprachlichen Ebene und der Kontextebene (vgl. auch Abb. 1). Die sprachliche Ebene umfaßt Sprachinterpretation (Erkennen) und Sprachproduktion (Handeln). Die Kontextebene umfaßt nichtsprachbasiertes Erkennen und Handeln.

Entwicklungsgeschichtlich entstand zuerst die kontextuelle Kognition. Die sprachliche Ebene entwickelte sich später als Spezialisierung des nichtsprachbasierten Erkennens und Handelns. für eine computerbasierte Modellierung ist die Kontextebene jedoch schwieriger als die sprachliche Ebene.

Dies liegt daran, daß heutige Standardcomputer weitgehend auf den Sprachkanal beschränkt sind, wobei die Eingabe über die Tastatur und die Ausgabe über den Bildschirm erfolgt. Eine Modellierung der Kontextebene setzt jedoch einen kognitiven Agenten voraus, der seine Umwelt selbstständig erkennen und in ihr Handeln kann. Dies geht über die Möglichkeiten von Standardcomputern hinaus und erfordert stattdessen die Technologie von Robotern.

Wenn das DBS Fragment im folgenden für Standardcomputer definiert wird, so ist dies vor allem ein Zugeständnis an die heutige Technologie. Dabei muß man sich jedoch im Klaren sein, daß dieses Zugeständnis zu schwerwiegenden Einschränkungen bei der Modellierung kognitiver Agenten führt, und zwar sowohl in der Theorie als auch in der Praxis.

Praktisch wird es dem resultierenden Systems unmöglich, natürlichsprachliche Befehle als Handlungen auszuführen oder eigenständige Beobachtungen zu machen, die dann natürlichsprachlich mitgeteilt werden können. Theoretisch entzieht dieses Zugeständnis dem resultierenden System zwei wesentliche Fundamente.

Das eine ist die prozedurale Definition elementarer *Konzepte*, die einerseits zum eigenständigen Aufbau kontextueller Inhalte und andererseits als Sprachbedeutungen dienen (vgl. wiederum Abb. 1). Das andere ist die Grundlegung der kognitiven *Kohärenz* bei kontextuellen Inhalten.

Zum Beispiel kann ein Inhalt wie **Peter überquert die Straße** direkt über die unmittelbare Wahrnehmung oder indirekt über Datenträger, z.B. Sprache, aber auch Film etc., in die Wortbank eingelesen werden. Beim direkten Einlesen über die unmittelbare Wahrnehmung folgt die Kohärenz der Inhalte aus der Kohärenz

der externen Welt – also der zeitlichen und räumlichen Abfolge von Ereignissen, den Teil-Ganzes-Beziehungen etc.

Beim indirekten Einlesen besteht dagegen die Möglichkeit, daß ein Autor die aus dem unmittelbaren Erkennen und Handeln vertrauten Elemente auf dem Datenträger neu gemischt und verkettet hat. Dies ist der Grund, warum datenträgerbasierte Inhalte inkohärent sein können.

Beispielsweise ist ein Schwimmer, der zuerst am Beckenrand steht, dann ins Wasser springt und schließlich unter dem aufbrausenden Wasser verschwindet, kohärent. Dagegen ist ein Schwimmer, der mit den Füßen zuerst aus dem aufbrausenden Wasser auftaucht und dann in hohem Bogen am Beckenrand landet, nicht kohärent – es sei denn, man macht die zusätzliche Annahme, daß dieser Inhalt mit Hilfe eines Datenträgers, z.B. einem rückwärtslaufenden Film, dargestellt wird.

Entsprechend sind Menschen, die sich miteinander auf Deutsch oder Englisch unterhalten, kohärent. Eine Biene, die sich mit einem Grashüpfer auf Deutsch unterhält, ist dagegen nicht kohärent – es sei denn, man macht die zusätzliche Annahme, daß dieser Inhalt mit Hilfe eines Datenträgers, z.B. Sprache, dargestellt wird.

Die Verwendung von Standardcomputern hat zur Folge, daß Inhalte nur indirekt über Datenträger, speziell Sprache, in die Wortbank eingelesen werden können. Deshalb liegt die Verantwortung für die Kohärenz der Datenbankinhalte allein bei dem Benutzer, der diese eingelesen hat und nicht beim System. Aus der Kohärenz bzw. Inkohärenz der Datenbankinhalte folgt wiederum die Kohärenz bzw. Inkohärenz der vom DBS Fragment produzierten Sprache. Denn die Konzeptualisierung bei Sprachproduktion beruht auf einer Navigation durch die vorhandenen Datenbankinhalte, die direkt in Sprache abgebildet wird (vgl. Abb. 8).

## 7 Simulierung der kontextuellen Kognition?

Anstatt auf die kontextuelle Kognition zu verzichten, um die Schwierigkeiten mit der Verwendung von Robotern zu vermeiden, könnte man eine Simulation der kontextuellen Kognition in Erwägung ziehen. Hierfür stehen die Möglichkeiten der kognitiven Modellierung (*cognitive modeling*) zur Verfügung.

Die kognitive Modellierung kreiert Trickfilm-Agenten (*animated agents*) in virtuellen Umgebungen auf Standardcomputern. Der Benutzer kann die Bewegung durch eine künstliche Welt steuern (*automated cinematography*, Funge, Tu, & Terzopoulos 1999) und mit künstlichen Agenten kommunizieren, die mit

ihrer virtuellen Handlungsumgebung interagieren (Rickel & Johnson 1999, Loyall & Bates 1993).

Kognitive Modellierung und Datenbanksemantik haben entgegengesetzte Ziele: der eine Ansatz arbeitet an der Interaktion eines natürlichen Agenten (dem Benutzer) mit einer künstlichen Welt, der andere an der Interaktion eines künstlichen Agenten mit der natürlichen Welt. Daher ist für die kognitive Modellierung eine Kontrollstruktur als *message dispatcher* oder *blackboard architecture* wahrscheinlich besser geeignet als die Wortbanknavigation der Datenbanksemantik. Umgekehrt ist die Verwendung virtueller Realität für die Datenbanksemantik aus folgenden Gründen problematisch.

Erstens führt sie zu einer Fehlleitung des Aufwandes: anstatt Kohärenz in einer wirklichen Interaktion mit der externen Welt zu begründen, richtet sich fast die gesamte Programmierarbeit auf eine möglichst realistische Simulierung von 3-D Welten mit Trickfilm-Agenten. Zweitens ist die Interaktion mit künstlichen Welten nicht glaubhaft als Fundierung der Wortsemantik. Drittens trägt die Simulation der sprachbasierten Referenz auf virtuelle kontextuelle Objekte wenig zur Modellierung der Referenz auf wirkliche Objekte bei (vgl. Abb. 1).

Die Unterschiede und Gemeinsamkeiten zwischen Datenbanksemantik und kognitiver Modellierung können wie folgt zusammengefaßt werden. Aus Gründen der Allgemeinheit wird die modelltheoretische Semantik mit in den Vergleich aufgenommen.

Kognitive Modellierung und Datenbanksemantik haben die programmier-technische Methode gemeinsam. Dagegen basiert die modelltheoretische Semantik auf metasprachlichen Definitionen, die für eine prozedurale Implementierung meist ungeeignet sind.

Modelltheoretische Semantik und kognitive Modellierung haben gemeinsam, daß sie mit Nachbildungen der Welt arbeiten. Datenbanksemantik setzt die externe Welt dagegen voraus und stützt sich auf eine prozedurale Interaktion mit ihr.

Datenbanksemantik und modelltheoretische Semantik haben gemeinsam, daß sie formale Definitionen des Gesamtsystems, sogenannte Fragmente, entwickelt haben. Dagegen ist die kognitive Modellierung möglicherweise zu anwendungsorientiert, um eine deklarative Speziokation – über die Definition von Modellierungssprachen wie VRML oder CML hinaus – zu erlauben.

## 8 Schematische Darstellung von DBS-FRAGMENT

Als Hintergrund zur schematischen Beschreibung eines datenbanksemantischen Fragments mit der Bezeichnung DBS-FRAGMENT, betrachten wir zunächst ein Fragment der Montague-Grammatik. Das in Montague 1974, Kapitel 8, auch PTQ genannte Fragment besteht aus folgenden Komponenten:

1. Eine rekursive Definition syntaktischer Kategorien und ihre intuitive Interpretation für eine endliche Untermenge.
2. Ein Lexikon, das diese Kategorien verwendet
3. Eine Kategorialgrammatik für die syntaktische Analyse der Sätze des Fragments.
4. Eine intensionale Logik mit einer modelltheoretischen Interpretation.
5. Eine Menge von Übersetzungsregeln, die syntaktisch analysierte Sätze in logische Formeln mit entsprechenden Wahrheitsbedingungen übersetzen.
6. Eine Menge von Bedeutungspostulaten.

*Abb. 12: Komponenten von Montagues PTQ-Fragment*

Dabei wird die Sprachproduktion als Generierungsprozedur der Kategorialgrammatik in Verbindung mit dem Lexikon behandelt. Die Sprachinterpretation hat das Ziel, die Wahrheitsbedingungen der generierten Ausdrücke zu charakterisieren.

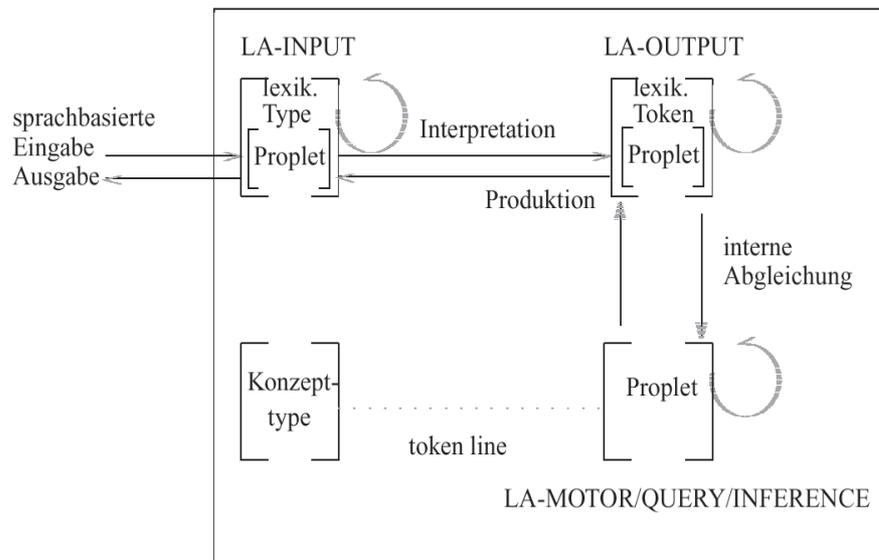
DBL-FRAGMENT hat dagegen die Aufgabe, kognitive Inhalte von einem kognitiven Agenten zum anderen zu transportieren. Diese Aufgabe wird von einer Reihe von Motoralgorithmen übernommen – Komponenten, die in der Montaguegrammatik keine Entsprechung haben. Aus praktischen Gründen ist DBL-FRAGMENT für Standardcomputer definiert.

1. DBS-LEXIKON: Eine kleine Anzahl lexikalischer Merkmalstrukturen, die jeweils mit einer Menge zugehöriger Wortoberflächen assoziiert sind, mit denen die Merkmalstrukturen instantiiert werden.
2. WORTBANK: eine verbundbasierte Netzwerkdatenbank, die bidirektional verzeigerte Proplets enthält.

3. LA-INPUT: eine LA-Grammatik für eine natürliche Sprache, deren syntaktische Regeln semantisch interpretiert sind, und zwar in der Form von Kopierregeln, die in die Fortsetzungspredikate der lexikalischen Merkmalstrukturen Werte eintragen.
4. LA-MOTOR: eine LA-Grammatik für die Navigation durch die verketteten Propositionen der Wortbank. Die Traversionszähler der Wortbank und vier Spurprinzipien verhindern Rückfall und Spaltung.
5. LA-OUTPUT: eine LA-Grammatik, die die von LA-MOTOR produzierte Propletsequenz in sprachspezifische Oberflächen abbildet, unter Berücksichtigung des Navigationstyps.
6. LA-QUERY: eine Menge von LA-Grammatiken, die Frageproplets auf die Tokenzeilen der Wortbank anwenden, um gewünschte Informationen abzurufen.
7. LA-INFERENCE: eine Menge von LA-Grammatiken, die aus dem Inhalt der Wortbank neue Propositionen ableiten.

*Abb. 13: Komponenten von DBS-FRAGMENT*

Die Motoralgorithmen sind die Komponenten 3–7. Sie interagieren wie folgt:



*Abb. 14: Schematische Struktur von DBS-FRAGMENT*

LA-INPUT wird von den hereinkommenden Sprachzeichen angetrieben und wandelt sie in Propletmengen, die in der Wortbank gespeichert werden. LA-MOTOR, LA-QUERY, und LA-INFERENCE motorisieren gemeinsam die Wortbanknavigation. LA-OUTPUT wird von der autonomen Navigation angetrieben, und reflektiert sie mit natürlichsprachlichen Ausdrücken.

Unter Berücksichtigung der algebraischen Definition der LA-Grammatik (Hausser 1992) und der Standarddefinition klassischer Netzwerkdatenbanken (siehe z.B. Elmasri & Navate 1989) ist DBL-FRAGMENT formal ebenso stringent wie PTQ. Auf die explizite Darstellung der LA-Grammatiken von DBS-FRAGMENT muß hier allerdings verzichtet werden, da sie über den Rahmen dieses Aufsatzes hinausgehen würde. Sie findet sich in Hausser 1999 und 2001b.

## 9 Schlußwort

Ein Fragment der Datenbanksemantik besteht aus einem Lexikon, einer semantischen Repräsentation und formalen Grammatiken. Die Grammatiken haben die Namen LA-INPUT, LA-OUTPUT, LA-MOTOR, LA-QUERY und LA-INFERENCE, und sind für die Interpretation und Produktion natürlicher Sprache sowie für die Konzeptualisierung, Befragung und Inferenz, respektive, definiert.

Die Integration dieser verschiedenen Funktionen basiert auf der Navigation durch verkettete Propositionen in einer Datenbank (siehe 4.1 und 4.2). Die Navigation wird mit Hilfe des zeitlinearen Algorithmus der LA-Grammatik realisiert. Das 'Schienensystem' der Navigation liefert die Datenstruktur der Wortbank, die verkettete Propositionen mit Hilfe von distribuierten Merkmalstrukturen darstellt. Sie drücken intrapropositionale Funktor-Argument-Strukturen und extrapropositionale Relationen mittels bidirektional kodierter Fortsetzungsprädikate aus.

Als Zugeständnis an die heutige Technologie wurde das formale Fragment für Standardcomputer definiert, und nicht für Roboter. Die generelle Theorie der Datenbanksemantik basiert jedoch entscheidend auf kognitiven Agenten, die in der Lage sind, auf der nichtsprachbasierten Ebene wahrzunehmen und zu handeln.

Aus diesem allgemeinen Ansatz folgt, daß das Fragment strukturell für eine Übertragung von Standardcomputern auf Roboter geeignet ist. Wenn die zusätzlichen Möglichkeiten einer prozeduralen Fundierung der Wortsemantik, einer Fundierung der Kohärenz in der kontextuellen Kognition, und einer Umsetzung von Sprache in externe Handlungen und von externen Wahrnehmungen in Sprache einmal zur Verfügung stehen, können sie in das gegenwärtige Fragment integriert werden.

Zudem steht die Verwendung von Standardcomputern dem grundsätzlichen Funktionieren des Kommunikationsmodells nicht im Wege (mittelbare Referenz). Sie verhindert auch nicht die Möglichkeit, das bestehende Fragment zu erweitern. Eine Erweiterung sollte mit der Definition eines größeren Lexikons und einer Syntax beginnen, die zusätzliche Konstruktionen analysieren und interpretieren kann.<sup>5</sup> Um das integrierte Funktionieren des Gesamtsystems zu gewährleisten, muß eine solche Erweiterungen konsequent durch entsprechende Erweiterungen aller anderen Komponenten ergänzt werden

Als programmiertechnisches System ermöglicht die Datenbanksemantik eine leistungsfähige und bequeme Verifikation an großen Mengen realer Daten. Dabei müssen nicht nur die empirische Abdeckung von Lexikon und Syntax verifiziert werden, sondern auch die semantische Korrektheit der Interpretation, die Gültigkeit der Inferenzen, die Kohärenz der Konzeptualisierung und die rhetorische Korrektheit der Produktion.

Systeme der Datenbanksemantik mit ausreichender Abdeckung haben eine Fülle praktischer Anwendungen. Sie ermöglichen neue Methoden z.B. bei natürlichsprachlichen Schnittstellen mit klassischen Datenbanken, bei Indizierung und Abruf in textuellen Datenbanken, beim Modellieren der Informationsübertragung in Dialogen, bei der Interlingua-basierten maschinellen Übersetzung und bei der Unterstützung der Spracherkennung mit Grammatik- und Domänenwissen.

## Bibliographie

- Elmasri, R. & S.B. Navathe (1989) *Fundamentals of Database Systems*, Benjamin-Cummings Publishing Company, Inc., Redwood City, CA.
- Funge, J., X. Tu, & D. Terzopoulos (1999) „Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters,“ *SIGGRAPH 99*, Los Angeles, CA.
- Hausser, R. (1992) „Complexity in Left-Associative Grammar,“ *Theoretical Computer Science*, Vol. 103, Elsevier.
- Hausser, R. (1996) „A Database Interpretation of Natural Language,“ *Korean Journal of Linguistics*, Vol. 21, No. 1,2:29–55.
- Hausser, R. (1999) *Foundations of Computational Linguistics*, Springer-Verlag, Berlin–New York. Hausser, R. (2001a) „The Four Basic Ontologies of Semantic Interpretation“, in H. Kangassalo et al. (eds.) *Information Modeling and Knowledge Bases XII*, IOS Press Ohmsha, Amsterdam.

- Hausser, R. (2001a) „The Four Basic Ontologies of Semantic Interpretation“, in H. Kangassalo et al. (eds.) *Information Modeling and Knowledge Bases XII*, IOS Press Ohmsha, Amsterdam.
- Hausser, R. (2001b) „Database Semantics for Natural Language“, *Artificial Intelligence*, Vol. 129, No. 1.
- Hausser, R. (2001c) „Spatio-Temporal Indexing in Database Semantics“, in A. Gelbukh (ed.) *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science Vol. 2004. Springer-Verlag, Berlin, New York.
- Lappin, S. (ed.) (1996) *The Handbook of Contemporary Semantic Theory*, Blackwell Publishers, Oxford, England.
- Loyall, A. B. & J. Bates (1993) „Real-time Control of Animated Broad Agents“, *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, Boulder, Colorado.
- Montague, R. (1974) *Formal Philosophy*, edited by R. Thomason, Yale University Press, New Haven.
- Nasr, A., O. Rambow, M. Palmer, & J. Rosenzweig (1997) „Enriching Lexical Transfer with Cross-linguistic Semantic Features“, *Proceedings of the Interlingua Workshop at the MT Summit*, San Diego, California, October.
- Rickel, J., & Johnson, W.L. (1999) „Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control“, *Applied Artificial Intelligence* 13:343–382.

## ENDNOTEN

- <sup>1</sup> Die prozedurale Behandlung der Konzepte ist eine grundsätzliche Alternative zur logischen Semantik, die Bedeutungen metasprachlich über Wahrheitsbedingungen zu beschreiben sucht. Dieser Unterschied wird in Hausser 2001a ausführlich thematisiert.
- <sup>2</sup> Ein möglicher Vorläufer ist die Verwendung der bidirektionalen Verzeigerungsmethode zur Beschleunigung beim Caching (persönliche Mitteilung von Jaime Carbonell, CMU, April 2000).

- <sup>3</sup> Der Begriff *Proplet* ist analog zu *droplet* gebildet und bezeichnet die Grundbausteine elementarer Proposition. Oberflächlich mögen Proplets den Merkmalstrukturen der HPSG ähneln. Letztere sind jedoch als Teil von Phrasenstrukturbäumen konzipiert, und nicht als Teil einer Datenbank; zudem fehlen die Kodierung der Funktor-Argumentstruktur über *bidirektionale* Zeiger, eine Verkettung von Propositionen über extrapropositionale Relationen, und somit die Basis für eine zeitlineare Navigation.
- <sup>4</sup> Viele dieser Extensionen sind in S. Lappin (ed.) 1996 überblicksartig zusammengestellt.
- <sup>5</sup> Eine LA-Grammatik, die einen maßgeblichen Umfang des Englischen abdeckt, findet sich in Hausser 1999, p. 458 f. Eine entsprechende LA-Grammatik für das Deutsche wird auf S. 364 f. definiert.