

# Sprachgenerierung auf der Grundlage konzeptueller Strukturen

Matthias REHM

## 1. Einführung

In diesem Artikel wird ein kurzer Überblick über das System IPaGe (Inkrementeller **P**aralleler **G**enerator) gegeben, das Sprachgenerierung ausgehend von konzeptuellen Strukturen leistet. Diese Strukturen stellen eine günstige Ausgangsbasis für den Generierungsprozess dar. Zum einen erlauben sie in natürlicher Weise ein inkrementelles Eingabeverhalten, zum anderen bieten sie die Möglichkeit, den Generierungsprozess in einer Art zu parallelisieren, die eine einfache Adaption des Systems an unterschiedliche Domänen zuläßt. Dieser Aspekt wird zusätzlich durch das objektorientierte Paradigma der gewählten Programmiersprache unterstützt.

Zunächst wird die Applikation vorgestellt, die während der Entwicklung als Generierungskontext diente. Es folgt eine Beschreibung der konzeptuellen Strukturen, die als Schnittstelle und somit als Eingabe des Generierungssystems Verwendung finden. Als nächstes wird eine Definition für ein inkrementelles Verarbeitungsverhalten geliefert, das von solchen Strukturen unterstützt wird. Daraufhin wird gezeigt, in welcher Weise sich der Generierungsprozess parallelisieren läßt und welche Systemarchitektur und welche zusätzlichen Merkmale sich dadurch ergeben.

## 2. Generierungskontext

Zielvorgabe im Rahmen des hier vorgestellten Projektes war die Entwicklung eines applikationsunabhängigen Generierungssystems. Um einen Kontext für die Generierungsaufgabe zu schaffen, ist IPaGe an ein teilautonomes Robotersimulationssystem angebunden, in dem Baufixteile manipuliert werden (MILDE et al. 1997). In diesem Zusammenhang sollen Beschreibungen der Roboterhandlungen in der ersten Person generiert werden, beispielsweise "Ich lege die Schraube neben den Würfel". Da

innerhalb des Simulationssystems eine reaktive Steuerungskomponente realisiert ist, in der die Roboterhandlungen nicht explizit repräsentiert sind sondern situativ emergieren, muss zunächst eine Kategorisierung dieser Handlungen stattfinden. Zu diesem Zweck werden in einem schemabasierenden Ansatz sensomotorische und linguistische Daten integriert und das Ergebnis, die Repräsentation der aktuellen Handlung, als konzeptuelle Struktur zur Verfügung gestellt (GOECKE & MILDE 1998).

### 3. Konzeptuelle Strukturen

Als Schnittstelle zu einer möglichen Applikation und damit als Startpunkt des Generierungsprozesses dienen konzeptuelle Strukturen, wie sie von Jackendoff beschrieben und von Lobin adaptiert wurden (JACKENDOFF 1991, LOBIN 1996).

Der erfolgreiche Einsatz konzeptueller Strukturen in Prozessen der Sprachverarbeitung bietet eine gute Ausgangslage, diese als Schnittstelle zu Prozessen nichtlinguistischer Verarbeitung in Betracht zu ziehen (s. beispielsweise NOGIER & ZOCK 1992, DORR 1993).

Bei den konzeptuellen Strukturen handelt es sich um eine strukturelle Semantikepräsentation mit Hilfe eines festen Satzes konzeptueller Primitive. Zur Repräsentation einer bestimmten Bedeutung werden diese Primitive entsprechend kombiniert. Dabei sind komplexe konzeptuelle Strukturen aus anderen aufgebaut. Abbildung 1 zeigt den Aufbau einer Beispielstruktur aus dem hier zugrundegelegten Szenario, die in Sätzen wie "Ich greife den gelben Würfel" oder "Vorderen Gewindeblock greife ich" resultiert. Die einfachste konzeptuelle Struktur besteht aus einem Typ und dem Kopf (dem Wert) der Struktur (Abb. 1: [AGENT i], [COLOR yellow]). Die Typen spielen eine entscheidende Rolle bei der Parallelisierung des Generierungsprozesses. Beispielsweise beschreibt Jackendoff eine Typ-Phrasen-Korrespondenz, die während der syntaktischen Verarbeitung genutzt wird. Eine konzeptuelle Struktur kann durch weitere Strukturen modifiziert werden (Abb. 1: [OBJECT cube]) oder sie kann Argumente fordern, um bestimmte Bedeutungen zu repräsentieren (Abb. 1: [EVENT grasp]).

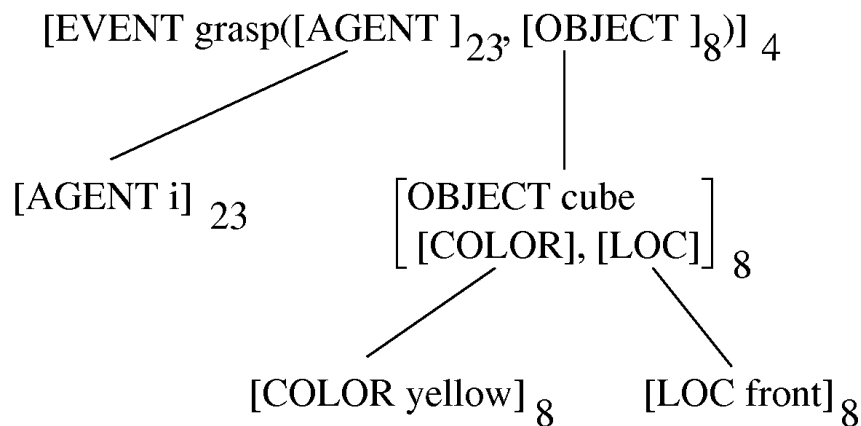


Abbildung 1: Komplexe konzeptuelle Struktur mit Unterstrukturen, die als Eingabeinkremente des Generierungsprozesses dienen können. Teile derselben Struktur sind koindiziert. Die relevanten Merkmale konzeptueller Strukturen sind: Typ des logischen Kopfes, der logische Kopf selbst, sowie Anzahl und Typen der Argumente.

Durch den rekursiven Aufbau dieser Strukturen kann auf einfache Weise ein inkrementelles und paralleles Verarbeitungsverhalten implementiert werden.

#### 4. Inkrementalität

Ein inkrementelles Verarbeitungsverhalten gilt als wichtiges Merkmal des Generierungsprozesses (LEVELT 1989, KEMPEN & HOENKAMP 1987). Dabei wird oft auf eine präzise Definition des Begriffs verzichtet. Unter einem inkrementell arbeitenden System versteht Finkler folgendes:

“Der **inkrementelle Verarbeitungsmodus** in einem System *S* ist durch eine Kombination aus Konsumierung der Eingabe und Produktion der Ausgabe definiert, wobei die drei folgenden Bedingungen erfüllt sein müssen:

Die Verarbeitung startet, bevor die Eingabe komplett vorliegt (...)

1. Die Ausgabe des Ergebnisses wird gestartet, bevor die Verarbeitung der Aufgabenstellung abgeschlossen ist (...)

2. Die Ausgabe des Ergebnisses des Systems wird gestartet, bevor die Eingabe für die aktuelle Aufgabenstellung komplett vorliegt (...)

Ein System heißt **inkrementelles System**, wenn es in der Lage ist, im inkrementellen Verarbeitungsmodus zu arbeiten.” (FINKLER 1997).

In welcher Weise erfüllt nun IPaGe diese Anforderungen? Als Eingabe wird eine konzeptuelle Struktur angesehen, die eine Roboterhandlung beschreibt und i.d.R. mit einem Satz ausgedrückt werden kann. Da jede Teilstruktur einer komplexen Struktur als Eingabeinkrement dient, ist der erste Punkt erfüllt. Die Verarbeitung startet, sobald ein Eingabeinkrement zur Verfügung steht.

Die Ausgabe ist dementsprechend definiert als ein gemäß der gegebenen Grammatik vollständiger Satz, welcher die durch die konzeptuelle Struktur beschriebene Handlung ausdrückt. Die Ausgabe beginnt, sobald die erste Konstituente der ausgewählten Satzstruktur bereitgestellt wird, so dass auch der zweite Punkt erfüllt ist.

Die Erfüllung des dritten Punkts kann nicht garantiert werden, was allerdings kein prinzipielles Problem des Systems ist, sondern in der Art der Verarbeitung und der Anbindung an mögliche Applikationen begründet ist. Sobald ein beliebiges Eingabeinkrement zur Verfügung steht, wird dieses verarbeitet. Da die Reihenfolge der Ausgabe zwar durch die Reihenfolge der Eingabeinkremente beeinflusst, aber nicht determiniert wird, kann es jedoch passieren, dass bereits die komplette Eingabe vorliegt, wenn mit der Ausgabe begonnen wird. Dieser letzte Punkt bezieht sich daher eher auf Systeme mit festgelegter Ein-/Ausgabereihenfolge. Zudem ist hier die Geschwindigkeit der Applikation von Bedeutung. Je schneller die einzelnen Eingabeinkremente zur Verfügung gestellt werden, desto weniger wahrscheinlich wird es, dass der dritte Punkt erfüllt wird.

Da es sich hier nicht um ein prinzipielles Problem handelt, kann IPaGe als ein im inkrementellen Verarbeitungsmodus arbeitendes System betrachtet werden.

## 5. Parallelisierung des Generierungsprozesses

Konzeptuelle Strukturen tragen Informationen sowohl für die lexikalische als auch für die syntaktische Verarbeitung. Auf lexikalischer Seite sind dies die Typen und Köpfe der Strukturen, die zur Auswahl geeigneter Lexeme benutzt werden, auf syntaktischer Seite sind es die Typen, die aufgrund der Typ-Phrasenkorrespondenz relevant sind.

Ein erster Schritt, um einen verteilten Generierungsprozess zu erreichen, ist daher die Trennung dieser Verarbeitungseinheiten. Ein zweiter Schritt besteht dann darin, die einzelnen Verarbeitungsschritte in diesen Einheiten zu parallelisieren. Zu diesem Zweck ist es notwendig, die Eigenschaften der jeweiligen Eingabestrukturen zu identifizieren, die eine solche verteilte Verarbeitung nahelegen.

Eine vollständige Trennung der lexikalischen und der syntaktischen Verarbeitungseinheit ist nicht möglich, da während der syntaktischen Verarbeitung die Ergebnisse der lexikalischen Seite benötigt werden. Um eine Satzstruktur zu determinieren, ist beispielsweise die Subkategorisierungsinformation des Verbs von entscheidender Bedeutung. Es besteht daher die Notwendigkeit der Kommunikation zwischen den Verarbeitungseinheiten. Diese Kommunikation wird über ein Blackboard realisiert, das als zentraler Datenspeicher zur Verfügung steht, so dass alle Prozesse Zugang zu den relevanten Daten haben. Es handelt sich hierbei nicht um eine klassische Blackboardarchitektur, da kein Controller existiert, der die Prozessreihenfolge festlegt und die Informationen verteilt. Da alle Prozesse eigenständig auf die benötigten Informationen zugreifen, ist die Verarbeitung einer gegebenen Eingabestruktur nicht deterministisch, sondern abhängig von der Reihenfolge, in der die Eingabeinkremente zur Verfügung gestellt werden, und von der Geschwindigkeit der einzelnen Prozesse.

### 5.1 Systemarchitektur

In den beiden Verarbeitungseinheiten der lexikalischen und der syntaktischen Verarbeitung können verschiedene Prozesse identifiziert werden (Abbildung 2). Auf lexikalischer Seite sind dies die Prozesse der Transformation, der Lexemauswahl und der morphologischen Verarbeitung. Auf syntaktischer Seite sind es Mapping und Ersetzung.

**Transformation:** Die lexikalische Verarbeitung befasst sich mit der Wahl geeigneter Worte, um die gegebenen semantischen Relationen auszudrücken. Zunächst müssen die vorhandenen Ambiguitäten in der Eingabestruktur aufgelöst werden. Diese Ambiguitäten ergeben sich durch die strukturelle Kodierung der Bedeutung. Der gleiche logische Kopf kann für einen gegebenen Typ unterschiedliche Bedeutung in verschiedenen strukturellen Kombinationen haben.

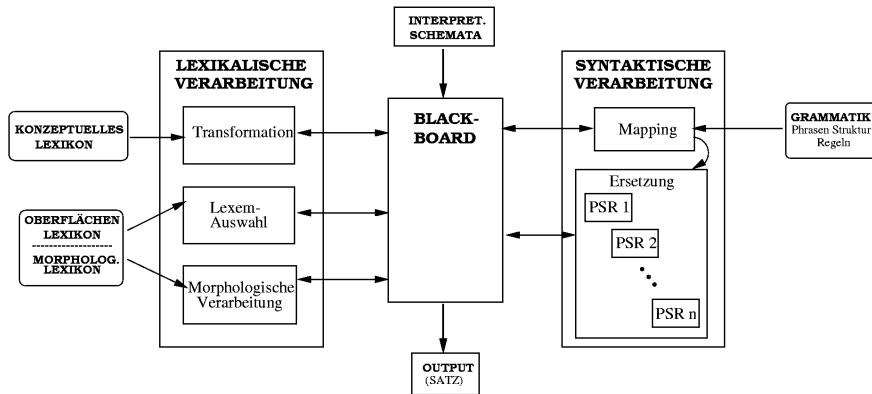


Abbildung 2: Die Architektur des Systems IPaGe. Die Verarbeitung ist aufgeteilt in eine lexikalische und eine syntaktische Verarbeitungseinheit, in denen verschiedene Prozesse unterschieden werden können. Diese kommunizieren über ein zentrales Blackboard miteinander.

**Lexemauswahl:** Als nächstes muss ein Lexem ausgewählt werden, das die intendierte Bedeutung kodiert. Für eine gegebene Bedeutung existieren oft mehrere Lexeme, von denen ein bestimmtes auszuwählen ist. Beispielsweise kann ein Baustein als Block, Würfel, Quader, Klotz, Baustein, Quadrat, etc. bezeichnet werden.

**Morphologische Verarbeitung:** Ist der Auswahlprozess beendet, muss das entsprechende Lexem passend flektiert werden. Dieser Prozess ist im aktuellen System noch nicht realisiert, stattdessen findet eine Auswahl der flektierten Form aus einem Vollformenlexikon statt.

**Mapping:** Während der syntaktischen Verarbeitung wird festgelegt, welche Phrasen die Äußerung konstituieren. Eine erste Einschränkung des Suchraums ist möglich, da eine Korrespondenz zwischen konzeptuellen Typen und Phrasenstrukturen besteht, welche die entsprechenden Strukturen realisieren.

**Ersetzung:** Phrasenstrukturen werden über klassische Phrasenstrukturregeln beschrieben, die die Konstituenten der Phrasen festlegen. Diese Konstituenten müssen durch die Worte ersetzt werden, die von der lexikalischen Verarbeitung geliefert werden.

## 5.2 Parallelisierung der Unterprozesse

Die einzelnen Unterprozesse innerhalb der lexikalischen und syntaktischen Verarbeitung können für verschiedene Eingabeinkremente parallelisiert werden. Dies wird am Beispiel des Mappingprozesses verdeutlicht.

Die vorhandene Typ-Phrasen-Korrespondenz legt eine Aufteilung der Verarbeitung auf der Grundlage der vorhandenen Typen nahe. Für jeden konzeptuellen Typen existiert also ein eigener Mappingprozess, der auf Strukturen dieses Typs arbeitet. Ein gegebenes Eingabeinkrement triggert dann seinen typspezifischen Mappingprozess ohne den Gesamtprozess zu blockieren. Erreicht eine weitere Struktur mit anderem Typ das Blackboard, kann die Verarbeitung dieser Struktur sofort starten.

Das objektorientierte Paradigma der benutzten Programmiersprache unterstützt direkt die hier gewählte Art, Prozesse zu parallelisieren. Dies wird wiederum am Beispiel des Mappingprozesses verdeutlicht. Für den Gesamtprozess wurde eine Klasse erzeugt, die die Verarbeitung innerhalb des Mapping implementiert. Um die Arbeit auf verschiedene Prozesse zu verteilen, wurden Instanzen dieser Klasse für die unterschiedlichen konzeptuellen Typen erzeugt und als eigenständige Threads gestartet. Auf diese Weise ist eine einfache Anpassung des System möglich. Werden neue konzeptuelle Typen benötigt, ist außer der Ergänzung der Lexika auf der Seite der lexikalischen Verarbeitung lediglich eine entsprechende typspezifische Instanz der jeweiligen Prozessklasse zu erzeugen.

Diese Art der Parallelisierung liefert zudem einen weiteren Vorteil, der am besten am Ersetzungsprozess veranschaulicht werden kann. Eines der Designkriterien für IPaGe war Flexibilität, hier in der Bedeutung, dass auf beliebige Eingabefragmente reagiert werden kann. Komplexe konzeptuelle Strukturen können letztendlich bis auf atomare Strukturen zurückgeführt werden, die nur noch aus einem Typen und einem Kopf bestehen.

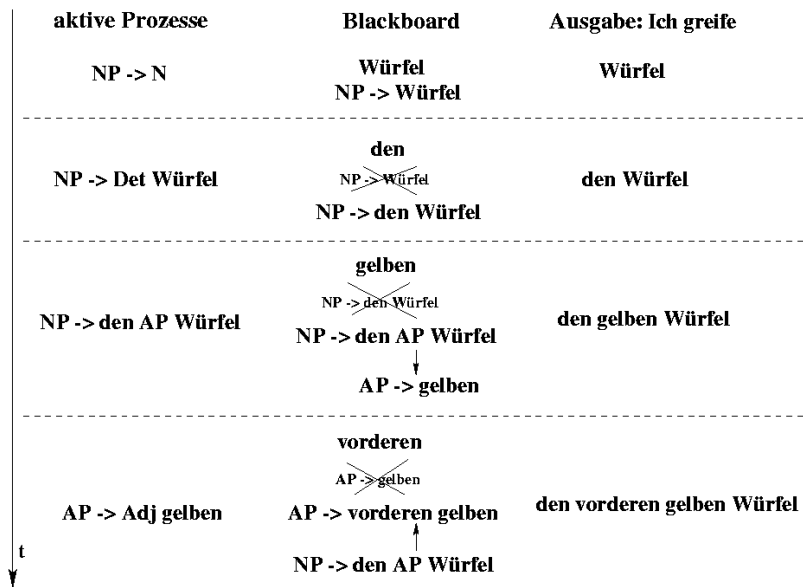


Abbildung 3: Verbesserung des Ergebnisses während der Laufzeit. In Abhängigkeit vom Ausgabeprozess werden mehr oder weniger stark ausgearbeitete Konstituenten geäußert.

Jede dieser atomaren Strukturen kann als Eingabeinkrement in den Generierungsprozess dienen. Die Reihenfolge ist dabei in keiner Weise vorgegeben. Da diese atomaren Strukturen aufgrund ihrer Typen zu bestimmten Phrasenstrukturen korrespondieren, und da die Reihenfolge verschiedener Phrasen in einer Äußerung zu einem bestimmten Maß flexibel ist, kann das Ausgabeverhalten diese Flexibilität reflektieren. Die Reihenfolge, in der die Phrasen der Äußerung ausgegeben werden, ist tatsächlich in hohem Maße abhängig von der Reihenfolge, in der die Eingabestrukturen bereitgestellt werden. Wenn beispielsweise eine konzeptuelle Struktur, die als direktes Objekt realisiert wird, als erstes das Blackboard erreicht, ist es sehr wahrscheinlich, dass die Äußerung mit eben jenem direkten Objekt beginnt: "Die Schraube lege ich neben den Würfel."

### 5.3 Qualitative Inkrementalität

Durch die Art der Parallelisierung der Prozesse ergibt sich ein weiterer, wünschenswerter Effekt. Grundsätzlich können nach Finkler zwei Typen von Inkrementalität unterschieden werden: quantitative und qualitative Inkrementalität (FINKLER 1997). Bisher war nur die Rede von quantitativer Inkrementalität. Die einzelnen Teile einer Eingabestruktur können nacheinander (hier sogar in beliebiger Reihenfolge) bereitgestellt und verarbeitet werden. Unter qualitativer Inkrementalität wird die Fähigkeit verstanden, ein bereits erreichtes Ergebnis zu verbessern, wenn neue Informationen zur Verfügung stehen. Dieses Phänomen wird in IPaGe während des Ersetzungsprozesses erreicht. Eine gegebene Phrase kann durch verschiedene mehr oder weniger komplexe Strukturen realisiert werden, eine Nominalphrase u.a. beispielsweise durch  $NP \rightarrow N$ ,  $NP \rightarrow \text{Det } N$  oder  $NP \rightarrow \text{Det } AP \ N$ . Alle Strukturregeln, die eine bestimmte Phrase realisieren, werden als eigenständige Prozesse gestartet, und versuchen, die Nonterminalsymbole ihrer rechten Seite durch Terminalsymbole zu ersetzen. Gelingt dies für eine Regel, so steht ein erstes Ergebnis zur Verfügung. Dies wird in Abbildung 3 veranschaulicht. Erreicht "Würfel" das Blackboard, so kann die erste Regel,  $NP \rightarrow N$ , ihre rechte Seite vollständig ersetzen und ist somit erfüllt. Die komplexeren Regeln versuchen weiterhin, ihre Nonterminalsymbole zu ersetzen. Ist im obigen Beispiel der entsprechende Artikel irgendwann verfügbar, so wird das bisherige Ergebnis überschrieben:  $NP \rightarrow \text{den Würfel}$ . Erreicht der Ausgabeprozess die Nominalphrase, so wird das aktuell vorhandene Ergebnis ausgegeben.

Die rechte Spalte der Abbildung 3 gibt an, welche Ausgabe das System zu den jeweiligen Zeitpunkten liefern würde. Dabei wird deutlich, dass die Ausgabe des Systems vom komplexen Zusammenspiel aller beteiligten Prozesse abhängt.

## 6. Zusammenfassung

Konzeptuelle Strukturen lassen sich erfolgreich als semantisches Repräsentationsformat im Sprachgenerierungsprozess einsetzen. Dies wurde am Beispiel des Generierungssystems IPaGe gezeigt. Aufgrund ihres Aufbaus unterstützen konzeptuelle Strukturen zum einen eine inkrementelle Verarbeitungsstrategie, die sogar eine arbiträre Reihenfolge der Eingabeinkremente erlaubt. Zum anderen erlauben sie in Zusammenhang mit dem objektorientierten Paradigma der für die Implementierung gewählten Programmiersprache eine Parallelisierung des Generierungsprozesses, bei der die Adaption des Systems an andere als die hier vorgestellte Domäne problemlos möglich ist. Zu diesem Zweck ist eine Anpassung der verwendeten Lexika an die jeweilige Domäne notwendig. Werden zudem andere konzeptuelle Typen verwendet, müssen lediglich weitere Instanzen der jeweiligen Prozessklassen erzeugt werden.

## Literatur

- DORR, B.J. (1993): Machine translation. MIT Press.
- FINKLER, W. (1997): Automatische Selbstkorrektur bei der inkrementellen Generierung gesprochener Sprache unter Realzeitbedingungen. Infix.
- GOECKE, K. U. und MILDE, J.-T. (1998): Talking About What I Do: Conceptualization and Robots. In HILDEBRAND, B. et al.: Architectures in Cognitive Robotics, Report SFB 360, Nr. 13, Universität Bielefeld, 11-18.
- JACKENDOFF, R. (1991): Semantic Structures. MIT Press.
- KEMPEN, G. und HOENKAMP, E. (1987): An incremental procedural grammar for sentence formulation. In: *Cognitive Science* 11, 201-258.
- LEVELT, W.J.M. (1989): Speaking: From intention to articulation. MIT Press.
- LOBIN, H. (1996): Handlungsanweisungen: sprachliche Spezifikation teilautonomer Aktivität. DUV.
- MILDE et al. (1997): J.-T. M., S. STRIPPGEN und K. PETERS (1997): Situated communication with robots. In: *First international workshop on human-computer conversation*.
- NOGIER, J.F. und ZOCK, M. (1992): Lexical choice as pattern matching. In: NAGLE et al. (eds.): *Conceptual Structure: current research and practice*. Ellis Horwood.