Automatische Generierung von mehrsprachigen Instruktionstexten aus einer Wissensbasis

Dietmar Rösner

Habilitationsschrift
Fakultät für Informatik
Universität Stuttgart
1994

Kurzfassung

Diese Arbeit untersucht die Frage, inwieweit aus einer im Computer abgelegten Wissensstruktur automatisch Texte in mehreren Sprachen generiert werden können. Als Untersuchungsgegenstand und Anwendungsbereich wurden Instruktionstexte aus mehrsprachigen Benutzerhandbüchern gewählt. Ein Korpus von Beispieltexten wurde detaillierten Untersuchungen unterzogen. Dabei ging es darum, ein Modell der in den Instruktionstexten verwendeten Diskurssstrukturen zu abstrahieren, die Gemeinsamkeiten und Unterschiede der einzelsprachlichen Ausprägungen der Texte zusammenzustellen und Vorgehensweisen für die Repräsentation des Sachgebietswissens zu entwickeln.

Diese Analysen legten die Grundlage für die Implementation eines prototypischen Systems, mit dem aus repräsentierten Plänen für Wartungsarbeiten automatisch Instruktionstexte in Deutsch, Englisch und — in eingeschränktem Umfang — auch Französisch generiert werden können. Das repräsentierte Wissen kann auch für die qualitative Simulation der Wartungsabläufe genutzt werden. Ein Autorenwerkzeug erlaubt die interaktive Erstellung und Edition der Planstrukturen, die den Ausgangspunkt für die Textgenerierung bilden. Die linguistischen Einheiten in den produzierten Texten sind mit den Wissensbasiselementen assoziiert. Die vom System produzierten Texte werden damit zu einem interaktiven Medium, auf dem Fragen gestellt und weitere Informationen angefordert werden können.

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Anwendung wissensbasierter Textgenerierung 2

1.2 Übersicht über die Arbeit 3

2 Zur automatischen Generierung von Texten **5**

2.1 Einführung 5

2.1.1 Wo ist Generierung sinnvoll? 5

2.1.2 Beispiele von Textgeneratoren 7

2.1.3 Die klassische Dichotomie: Inhalt vs. Form 8

2.1.4 Ausgangspunkt: Situation und Kommunikationsbedürfnis . . . 8

2.1.5 Generierung als wissensbasierter Prozeß 10

2.1.6 Inhaltsbestimmung über Schemata 10

2.1.7 Eingabestrukturen für die Generierung? 12

2.1.8 Perspektive in der Lexikalisierung 13

2.2 Exkurs: Generierung und Computeranimation 14

2.2.1 Präsentation einer natürlichen Form 14

2.2.2 Die Hauptaufgaben 14

2.2.3 Flexibilität durch Wissensbasierung 15

2.3 Grammatikformalismen für die Generierung 16

2.3.1 Systemische Grammatik 16

2.3.2 Systemische Grammatik als Basis für Generierungssysteme . . 21

2.3.3 PENMAN 21

2.3.4 Andere Implementationen systemischer Grammatiken 24

2.3.5 Baum–Adjunktions–Grammatiken 24

2.3.6 Andere Ansätze zur Generierung 28

3 Technische Dokumentation und Mehrsprachigkeit **29**

3.1 Aspekte technischer Dokumentation 29

3.1.1 Umfang 29

3.2 Mehrsprachigkeit 30

3.2.1 Mehrsprachigkeit im Informationszeitalter 30

3.2.2 Technische Unterstützung von Mehrsprachigkeit 31

3.3 Behandlung mehrsprachiger Dokumente 32

3.3.1 Der herkömmliche Ansatz . . . 32
3.3.2 Einsatz maschineller Übersetzungssysteme . . . 33
3.3.3 Das Problem der Wissensbasierung von MÜ . . . 33
3.3.4 FAHQTUT . . . 34
3.3.5 Piktogramme statt Mehrsprachigkeit? . . . 36
3.4 Der Ansatz in TECHDOC . . . 38
3.4.1 Diskussion . . . 40
3.5 Generierung und Multilingualität . . . 42
3.5.1 Multilingualität in Generierungssystemen . . . 42
3.6 Andere Arbeiten zur Generierung von Instruktionstexten . . . 43

4 Diskursanalyse und Textstrukturen 45
4.1 Einleitung . . . 45
4.2 Rhetorical Structure Theory (RST) . . . 46
4.2.1 Formale Elemente von RST . . . 47
4.2.2 Ein Beispiel . . . 49
4.2.3 Zur graphischen Darstellung . . . 50
4.2.4 Durchführung von RST–Analysen . . . 50
4.3 Exemplarische Textanalyse . . . 52
4.3.1 RST–Analyse des Beispieltexts „Unternehmen Zukunft“ . . . 53
4.4 Einige methodische Fragen der RST–Analyse . . . 56
4.4.1 Zur Bestimmung elementarer Einheiten . . . 56
4.4.2 RST–Analyse und Textgliederung . . . 57
4.4.3 Signale für RST–Relationen . . . 57
4.4.4 RST–Baum und Schema–Abstraktion . . . 58
4.5 Abschließende Bemerkungen . . . 59

5 Untersuchungen zu Strukturen und Inhalten eines Wartungshandbuchs 61
5.1 Das Material . . . 61
5.1.1 Aspekt Multimedialität . . . 61
5.1.2 Überblick . . . 62
5.2 Zündkerzen . . . 62
5.2.1 Zur Diskursanalyse . . . 62
5.3 Motoröl . . . 68
5.3.1 Inhaltliche Analyse . . . 68
5.3.2 Zur RST–Analyse des Abschnitts . . . 69
5.4 Ölwechsel . . . 70
5.4.1 Zur RST–Analyse . . . 70
5.5 Kontrollieren des Kühlmittelstands . . . 73
5.6 Aspekte der sprachlichen Realisierung . . . 75
5.6.1 Glossieren . . . 75
5.6.2 Signale für RST–Relationen . . . 76

5.6.3 Mehrfachanalysen 77
5.6.4 Nominalphrasen 79
5.6.5 Positive vs. negative Formulierungen 79
5.6.6 Anweisung vor Begründung 80
5.6.7 Systemische Metafunktionen in Handbuchtexten 81
5.6.8 Zum Detailliertheitsgrad von Handbuchtexten 81
5.6.9 Zur Unvollständigkeit der Anleitungen 83
5.6.10 Das abstrakte Schema 'Wartungsaktivität' 83
5.6.11 Exkurs: Textproduktion und Lesermodell 84
5.7 Zusammenfassung: Diskursrelationen in Instruktionstexten 86

6 Zur Repräsentation des Gegenstandsbereichs 89
6.1 Einführung 89
6.1.1 Zum Vorgehen bei der Repräsentation von Wissen 89
6.1.2 Anforderungen an die Wissensrepräsentation 90
6.1.3 Abstraktes technisches Modell 91
6.2 Die Beispieltexte aus der Sicht der Wissensrepräsentation 92
6.2.1 Modellierungsfragen am Beispieltext „Zündkerzen“ 92
6.2.2 Modellierungsfragen: Flüssigkeiten und Behälter 97
6.2.3 Modellierungsfragen am Beispieltext „Ölwechsel“ 99
6.2.4 Modellierungsfragen am Beispieltext „Motoröl“ 100
6.3 Grundlegende Unterscheidungen 104
6.3.1 Objekte 104
6.3.2 Zur Modellierung von Aktionen 105
6.3.3 Eigenschaften 106
6.4 Probleme der Wissensrepräsentation 107
6.4.1 Algebraische vs. analoge Repräsentation 107
6.4.2 Zum Problem der Modellierungstiefe 108
6.4.3 Berücksichtigung unterschiedlicher Begriffssysteme 109
6.5 Terminologische Logik 112
6.5.1 LOOM als Beschreibungssprache 112
6.5.2 Modellierung in terminologischer Logik — ein ausführliches Beispiel 113
6.5.3 Einführung in die Nutzung von LOOM 117
6.5.4 Elemente der TECHDOC–Wissensbasis 122

7 Das System TECHDOC 127
7.1 Zur Architektur des TECHDOC–Systems 127
7.1.1 Die Entwicklung der Demonstrations–Versionen 129
7.1.2 Schwache vs. starke Multilingualität 130
7.1.3 Wiederverwendung linguistischer Ressourcen 130
7.2 Die verwendeten Grammatiken 133
7.2.1 Die englische Grammatik 133

7.2.2 Die deutsche Grammatik . . . 138
7.2.3 Französische Grammatik . . . 143
7.2.4 Anforderungen des Textmaterials . . . 145
7.3 Behandlung textueller Phänomene . . . 147
7.3.1 Artikel in Nominalphrasen . . . 147
7.3.2 Referenz durch Abstraktion . . . 148
7.3.3 Substanzen . . . 148
7.3.4 Sprachspezifische Variationen . . . 149
7.3.5 Ellipsen als textuelles Phänomen . . . 149
7.4 Fragen der Lexikalisierung . . . 152
7.4.1 Lexikalisierung in PENMAN . . . 152
7.4.2 Abstrakte Aktionskonzepte . . . 153
7.4.3 Lexikalisierung durch Klassifikation . . . 155
7.4.4 Lexikalisierung und syntaktische Struktur . . . 161
7.4.5 Perspektive bei der Lexikalisierung . . . 161
7.5 Qualitative Simulation von Abläufen . . . 161
7.5.1 Verbindung von Simulation und Lexikalisierung . . . 163
7.6 Zusammenfassung . . . 165

8 Elemente eines integrierten Dokumentenmanagementsystems 167
8.1 Design und Implementierung eines Autorenwerkzeugs . . . 167
8.1.1 Hauptfunktionen . . . 167
8.1.2 Der Prototyp des Autorenwerkzeugs . . . 170
8.2 Der TECHDOC-Formatter . . . 170
8.2.1 Die Dokumentrepräsentation . . . 170
8.2.2 Exkurs: SGML . . . 172
8.3 Die Benutzungsoberfläche . . . 174
8.3.1 Fragenbeantwortung . . . 176
8.3.2 Sprachausgabe . . . 178

9 Studien zur Übertragbarkeit der Ansätze 179
9.1 Zum Wartungshandbuch der S-Klasse . . . 180
9.1.1 Das Material . . . 180
9.1.2 Der Text „Kühlmittelstand“ . . . 181
9.1.3 Der Text „Motorölstand“ . . . 182
9.1.4 Zur Rekonstruktion der Texte . . . 185
9.2 Wartungshandbuch des Falcon 900-Flugzeugs . . . 186
9.2.1 Strukturierung . . . 186
9.2.2 Definitionen . . . 188
9.2.3 Der Beispieltext ‘Replacement of filtering elements’ . . . 190
9.2.4 Schema-Abstraktion . . . 197
9.3 Fallstudie: Aspekte eines Betriebssystem-Handbuchs . . . 198
9.3.1 Exkurs: Instruktionstexte bei der Computer-Nutzung . . . 198

9.3.2 Die Einführung: „What Is DOS?“ 199
9.3.3 Der Abschnitt „About Diskette Drives and Diskettes“ 202
9.3.4 Der Abschnitt „Getting a Diskette Ready for Use“ 203
9.3.5 Diskursrelationen im DOS-Handbuch 205
9.3.6 Das Schema „Kommando-Beschreibung“ 208
9.3.7 Zur Modellierung des Gegenstandsbereichs 209

10 Zusammenfassung und Ausblick **211**

Index **213**

Literaturverzeichnis **219**

A Definition zusätzlicher RST-Relationen **233**
A.1 PRECONDITION 233
A.2 UNTIL 234
A.3 ALTERNATIVE 234
A.4 STEP-SEQUENCE 234

B PENMAN: Beispiele für SPL-Terme und Generierungsergebnisse **235**

C Eine Charakterisierung des benötigten Fragments anhand der HP–NL Test Suite **241**

D Technische Dokumentation **247**

E Beiträge zu TECHDOC **249**

Abbildungsverzeichnis

2.1 Ein alltäglicher „Dialog“ mit einem Automaten 6
2.2 Lückentexte . . . und ihre Probleme 7
2.3 Einige Systeme zur Textgenerierung 7
2.4 Beispiele von Primärdaten 8
2.5 Ein initialer Baum 25
2.6 Ein auxiliarer Baum 25
2.7 Das Adjunktionsergebnis 26

3.1 Die EG–Produkthaftpflicht–Richtlinie 29
3.2 Die ISO/IEC–Richtlinie 37 31
3.3 Eine Installationsanleitung (fast) ohne Worte 36
3.4 Die GENESIS als Piktogramm 37
3.5 Vergleich: Piktogramme vs. Texte 38
3.6 TECHDOC–Szenario: multilinguale Generierung technischer Dokumente 39
3.7 Vergleich: Generierung vs. wissensbasierte MÜ 40
3.8 Vergleich: Generierung vs. MÜ mit kontrollierter Eingabe 40
3.9 Mehrsprachige Generierung aus Wissensbasen 41

4.1 Liste der RST–Relationen nach Mann & Thompson 47
4.2 Definition der Relation ELABORATION (nach M&T) 49
4.3 Graphische Darstellung einer ELABORATION-Relation 51
4.4 Einige Definitionen von RST–Relationen (nach M&T) 52
4.5 Beispieltext ‘Unternehmen Zukunft’ 53
4.6 RST–Analyse von Unternehmen Zukunft 54
4.7 Vorgeschlagene Definition für PROGRESSION 56

5.1 Seite „Zündkerzen“ aus HONDA–Manual 63
5.2 Text „Zündkerzen“ mit Basiseinheiten für die RST–Analyse 64
5.3 Textstruktur der Handbuchseite „Zündkerzen“ 65
5.4 Seite „Motoröl“ aus HONDA–Manual 68
5.5 Elementare Einheiten für RST–Analyse „Motoröl“ 69
5.6 RST–Analyse „Motoröl“ aus HONDA–Manual 70
5.7 Seite „Ölwechsel“ aus HONDA–Manual 71

5.8 Text „Ölwechsel“ mit elementaren Einheiten für die RST–Analyse . . 71
5.9 RST–Analyse „Ölwechsel“ aus HONDA–Manual 73
5.10 Seite „Kontrollieren des Kühlmittelstands“ aus HONDA–Manual . . . 74
5.11 Unterschiedliche einzelsprachliche Realisierungen von RST–Relationen 75
5.12 Handbuchtexte: textuelle Dimension 81
5.13 Handbuchtexte: interpersonelle Dimension 81
5.14 Handbuchtexte: Dimension Detailliertheit 82
5.15 RST–Relationen in Handbuchtexten 86

6.1 Ausschnitt aus der Modellierung 98
6.2 Eine englische Originaldarstellung 110
6.3 . . . und eine inadäquate Übertragung ins Deutsche (nach [Schmitt 1992]).111

7.1 TECHDOC: Systemarchitektur 128
7.2 System ADVERB-INTENSIFICATION und zugehöriger Chooser . . 135
7.3 System PP-INTENSIFICATION und zugehöriger Chooser 136
7.4 Ein generierter Beispieltext in Englisch und Französisch 144
7.5 Lexikalisierungen eines abstrakten Aktionskonzepts für Deutsch 153
7.6 . . . und für Englisch 154
7.7 Lexikalisierungen von Herstellen-einer-Verbindung 155
7.8 Definition von Vor- und Nachbedingungen für eine Aktion 162
7.9 Ausschnitt aus einer qualitativen Simulation 166

8.1 Dokumentrepräsentation für Rekonstruktion des Texts ‘Checking coolant’ 171
8.2 ... und die daraus generierte und formatierte Dokumentseite 172
8.3 Beispiel einer Dokumenttypdefinition 173
8.4 Korrespondenzen zwischen SGML und LaTeX 174
8.5 Automatisch produzierte LaTeX– und SGML–Version eines Dokuments 175
8.6 TECHDOC-Benutzungsschnittstelle mit erfragter Zusatzinformation . 177

9.1 Seite „Kühlmittelstand“ aus dem Handbuch der S-Klasse 180
9.2 Seite ‘Coolant level’ aus dem Handbuch der S-Klasse 181
9.3 Seite „Motorölstand“ aus dem Handbuch der S-Klasse 183
9.4 Seite ‘Engine oil level’ aus dem Handbuch der S-Klasse 183
9.5 Eine generierte Seite aus dem Handbuch der S–Klasse 185
9.6 Beispielseiten aus dem Handbuch des FALCON 900 187
9.7 ‘Replacement of Filtering Elements’ aus dem Handbuch des Falcon 900 191
9.8 ‘Echange des éléments filtrants’ aus dem Handbuch des Falcon 900 . . 193
9.9 Eine Seite aus dem DOS User’s Guide 200
9.10 Einführung in den ‘DOS User’s Guide’ auf Englisch 201
9.11 ... und Einführung in das DOS Benutzerhandbuch auf Deutsch. . . . 201
9.12 Beispiele für Definitionen im DOS User’s Guide 202
9.13 „Formatieren mit dem Befehl FORMAT“ aus dem DOS Handbuch . . 206

Kapitel 1

Einleitung

Schon 1949, gerade fünf Jahre nachdem die MARK I in Betrieb genommen worden war, wandte sich Warren Weaver, der damalige Direktor der Naturwissenschaftlichen Abteilung der Rockefeller Foundation, mit einem Aufruf an rund 200 Wissenschaftler. Er stellte in diesem 'Weaver–Memorandum' seine Überlegungen vor, wie mit Hilfe von Elektronenrechnern Texte aus einer natürlichen Sprache in eine andere übersetzt werden könnten. Damit gab er den Anstoß zu einer ganzen Reihe von Forschungsprojekten zur Maschinellen Übersetzung (MÜ). Eine anfängliche Euphorie wich allerdings bald der Erkenntnis, daß man die Schwierigkeiten erheblich unterschätzt hatte.

Auch bei vielen späteren Bemühungen — zum Beispiel der „Yale–Schule“ von Schank und seinen Mitarbeitern [Schank, Abelson 1977]) und von Großprojekten wie LILOG [Herzog, Rollinger 1992] — wurde erneut deutlich, worin die zentrale Schwierigkeit liegt: Sprachverstehen in einem realistischen Anwendungskontext ist nur dann möglich, wenn das sprachverarbeitende System über eine enorme Fülle von Wissen aus den unterschiedlichsten Bereichen verfügt und zusätzlich in der Lage ist, dieses Wissen aufgrund seiner sprachlichen Interaktionen laufend zu aktualisieren und fortzuschreiben.

Bei einer etwas groben Unterteilung sind hier eher sprachbezogene und eher inhaltsbezogene Wissensquellen zu unterscheiden. Zur ersten Kategorie gehört insbesondere lexikalisches und grammatisches Wissen; die zweite Kategorie ist mit dem gleichzusetzen, was Menschen als Erfahrungsschatz besitzen, d.h. es gehören dazu nicht nur z.B. Problemlösungsstrategien, Schlußregeln und Wissen über Begriffsbeziehungen, sondern das gesamte Spektrum dessen, was mangels besserer Begrifflichkeit oft „Weltwissen“ genannt wird. Hierzu gehört dann Wissen über physikalische, chemische, biologische und andere Gesetzmäßigkeiten der uns umgebenden Welt ebenso wie Wissen über soziale Beziehungen, über Konventionen des Umgangs, über die Funktionen der Sprachverwendung u.v.a.m.

Die bisherigen Forschungen im Bereich der Sprachverarbeitung haben zu neuen Formalismen und Techniken sowie prototypischen Systemen zur Wissensrepräsentation geführt. Hier setzt diese Arbeit an, wenn sie die Generierung mehrsprachiger

Dokumente aus einer Wissensbasis als Ansatz zum Umgang mit Multilingualität entwickelt und ein entsprechendes System prototypisch realisiert. Anwendungsbeispiel sind mehrsprachige Instruktionstexte als Teil der Dokumentation zu technischen Produkten. Grundgedanke des Ansatzes ist, daß Informationen über das betrachtete Produkt in einer Wissensbasis vorliegen, die als Ausgangspunkt für die automatische Generierung zumindest von Rohversionen solcher Dokumente dienen können.

Modellierungen, die für die erforderliche Wissensrepräsentation ggf. genutzt werden können, liegen heute bereits in Ansätzen vor, wenn bei der rechnergestützten Produktentwicklung z.B. Produktdatenmodelle und Prozeßmodelle erstellt werden. Aber selbst wenn die benötigten Strukturen aufgebaut werden müssen — ggf. mit geeigneten Werkzeugen wie graphische Unterstützung und automatische Klassifikation —, ergeben sich immer noch erhebliche Vorteile gegenüber der herkömmlichen Vorgehensweise. Ändert sich ein Produkt — wie es typischerweise der Fall ist — nur in einigen Details, so müssen nur diese Änderungen in der Wissensrepräsentationsstruktur nachgeführt werden, und eine dem aktuellen Stand des Produkts entsprechende neue Dokumentation läßt sich sofort (und bei Vorliegen von Generatoren in mehreren Sprachen auch sofort mehrsprachig) erzeugen.

Weitere Vorteile einer solchen Vorgehensweise sind:

- Dokumentationstexte können für die jeweilige Aufgabe und den jeweiligen Adressaten maßgeschneidert werden. Ein Wartungstechniker sollte z.B. anders gestaltete Informationen zur Verfügung gestellt bekommen als ein Kunde.
- Die Repräsentation kann auch für interaktive Anwendungen genutzt werden, bei denen nur die Teile einer Dokumentation generiert werden, die für die aktuelle Situation relevant sind.
- Eine Verbindung mit dem Hyper–Media–Ansatz ist möglich.
- Die Wissensbasis kann auch für weitere Zwecke genutzt werden.

1.1 Anwendung wissensbasierter Textgenerierung

In der vorliegenden Arbeit wird ein neuer Ansatz zum Umgang mit mehrsprachigen Dokumenten vorgestellt und prototypisch realisiert. Ausgangspunkt für die Erstellung von Texten in verschiedenen Sprachen ist nicht mehr ein Text in einer natürlichen Sprache (Quelltext), sondern im System repräsentiertes Wissen über den Gegenstandsbereich. Damit vermeidet man die Probleme bei der Analyse von Quelltexten. Für Instruktionstexte gehören zum repräsentierten Wissen insbesondere Pläne für durchzuführende Aktivitäten, Wissen über die beteiligten Objekte, Wissen über die Aktionen, über die Beziehungen zwischen Objekten, Aktionen, Eigenschaften. Anstatt einen Text in einen anderen zu übersetzen, wird aus dem repräsentierten Wissen Text in verschiedenen Zielsprachen generiert.

Die Herstellung der Dokumentationstexte ist nicht isoliert von den anderweitigen Verwendungen des Wissens. Die Strukturen der Wissensbasis sind nicht lediglich im Hinblick auf Textproduktion ausgelegt, sondern auch nutzbar z.B. für Simulation und Deduktion. Der Ansatz wird nicht nur vorgestellt, sondern als prototypisches System implementiert und mit benötigten Werkzeugen in ein System zum Dokumentenmanagement integriert.

Anwendungsbereich ist die Dokumentation spezieller technischer Geräte oder Anlagen. Dieses Gebiet ist aus einer Reihe von Gründen interessant: Zum einen hat technische Dokumentation und ihre Verfügbarkeit in verschiedenen Sprachen eine hohe ökonomische Relevanz, zum anderen behandeln technische Texte konstruierte Artefakte; kulturelle Differenzen sind in diesen Texten von geringerer Bedeutung als in anderen Gebieten. Damit wird sowohl die Wissensrepräsentation als auch die sprachliche Realisierung erleichtert. Für den Ansatz unterstützend erweist sich die Tatsache, daß aus anderer Motivation heraus im Umfeld der CAx–Techniken (CAD, CAE, CAM, CASE, ...) ohnehin umfangreiche Informationsbestände entstehen, die den Anforderungen einer automatischen Informationsverarbeitung genügen müssen und die zumindest ein Ausgangspunkt für die hier vorgeschlagenen Verfahren sein können.

1.2 Übersicht über die Arbeit

Die Arbeit hat folgenden Aufbau: Kapitel 2 führt in diejenigen Aspekte des Forschungsgebiets Textgenerierung ein, die für den hier vorgestellten Ansatz relevant sind. So wird unter anderem verdeutlicht, welche Vorteile ein an den Ansätzen der systemischen Grammatik orientiertes Vorgehen für die Generierung hat. Mit PENMAN wird eine Komponente vorgestellt, die in die eigenen Arbeiten integriert werden konnte.

In Kapitel 3 werden spezielle Anforderungen des gewählten Anwendungsgebiets der Instruktionstexte für technische Erzeugnisse erörtert. Die wirtschaftliche Relevanz der computerunterstützten Behandlung von Mehrsprachigkeit wird verdeutlicht sowie andere Möglichkeiten zum Umgang mit dieser Thematik erläutert.

Kapitel 4 stellt den methodischen Apparat für wichtige Teile der durchgeführten Textanalysen bereit. Es erläutert die grundlegenden Annahmen und Definitionen der sogenannten Rhetorical Structure Theory (RST, [Mann, Thompson 1987]), illustriert deren Anwendung mit einer exemplarischen Textanalyse und entwickelt eine Methodik der Analyse von Diskursstrukturen.

Ausgangspunkt für den Systementwurf sind ausführliche Untersuchungen an exemplarischem Textmaterial aus dem mehrsprachigen Benutzerhandbuch für einen Pkw. In Kapitel 5 wird dieses Material vorgestellt und unter linguistischen Gesichtspunkten analysiert. Besonderes Augenmerk wird dabei auf Fragen der Diskursorganisation und auf die Frage gelegt, welche Gemeinsamkeiten und welche Unterschiede bei der einzelsprachlichen Realisierung identischer Inhalte vorliegen. Für die Makro-

struktur der Texte wird das Schema „Wartungsaktivität“ abstrahiert.

Die Analysen der Beispieltexte werden in Kapitel 6 fortgeführt. Hier geht es um die Frage, wie eine Wissensbasis gestaltet sein muß, aus der vergleichbare Texte automatisch generiert werden können. Die für die Implementation verwendeten Techniken werden erläutert.

Kapitel 7 erläutert zunächst die Architektur des prototypischen Systems TECHDOC zur mehrsprachigen Generierung und das Zusammenspiel seiner Komponenten. Dann wird vertiefend auf einzelne Komponenten eingegangen. Die Grammatiken für die derzeit unterstützten Sprachen Deutsch, Englisch und Französisch werden skizziert. Es wird dargestellt, wie textuelle Phänomene (wie Pronominalisierung und Referenz) behandelt werden und wie sich sprachunabhängig modellierten Aktionskonzepten einzelsprachliche Lexikalisierungen zuordnen lassen. Werden solche Aktionskonzepte mit Vor- und Nachbedingungen modelliert, so läßt sich ein Wartungsablauf in einer qualitativen Simulation nachvollziehen.

Um praktisch einsetzbar zu sein, muß sich ein System zur Textgenerierung in ein umfassendes System zur Dokumentenerzeugung, -edition, -distribution und -verwaltung integrieren lassen. Damit zusammenhängende Anforderungen werden in Kapitel 8 erörtert und die für die Benutzungsoberfläche von TECHDOC realisierten Komponenten vorgestellt. Dazu gehören z.B. ein Autorenwerkzeug für die Wissensbasis und ein Formatter zur automatischen Layout–Erzeugung aus Dokumentrepräsentationen. Die Benutzungsoberfläche ermöglicht, generierte Texte als interaktives Medium für den Zugang zu ergänzender Information aus der Wissensbasis zu nutzen, die entweder textuell oder mit anderen Medien präsentiert wird.

Wie sich die entwickelten Methoden übertragen und verallgemeinern lassen, wird in Kapitel 9 anhand weiterer Beispieltexte diskutiert. Untersucht werden Texte aus verwandten technischen Gebieten, aber mit zum Teil anderem Adressatenkreis, sowie Texte aus einem Betriebssystemhandbuch als Beispiel für Software–Dokumentation.

Kapitel 2

Zur automatischen Generierung von Texten

2.1 Einführung

Der Schwerpunkt der Forschung zur Verarbeitung natürlicher Sprache liegt traditionell bei der Analyse. Erst in den letzten Jahren hat sich die natürlichsprachliche Generierung als eigenständiges Gebiet etabliert.[1] Die Hauptimpulse für die Entwicklung des Gebiets kamen hierbei weniger aus der Linguistik als aus der Künstlichen Intelligenz. In jüngster Zeit wird der Frage möglicher Anwendungen von Generierung vermehrte Aufmerksamkeit zuteil.

2.1.1 Wo ist Generierung sinnvoll?

Wenn man das Beispiel eines alltäglichen „Dialogs“ in Abb. 2.1 betrachtet, der jedem in dieser oder ähnlicher Form geläufig sein dürfte, so wird deutlich, daß hier zwar sprachliche Formen eingesetzt werden, das Repertoire aber nur aus einer kleinen Zahl vorgefertigter Textstücke besteht. Für solche Anwendungen muß keine natürlichsprachliche Generierung eingesetzt werden.

Für andere Anwendungen kann es ausreichen, mit sog. Lückentexten zu arbeiten. Dies sind vorgefertigte Texte mit Leerstellen, in die dann passende Ergänzungen einzufügen sind. Wie das Beispiel in Abb. 2.2 zeigt, sollte man dann aber zumindest darauf achten, daß die inhaltliche Konsistenz gewährleistet ist.

Der Aufwand, der für den Einsatz von Generierung erforderlich ist, ist insbesondere dann gerechtfertigt, wenn die Eigenschaften der natürlichen Sprache in der Anwendung wirklich genutzt werden können, d.h. wenn Variationsbreite, Flexibilität, Kompaktheit und andere ihrer Eigenschaften zum Tragen kommen.

Anwendungsgebiete, in denen man sich dieses wünscht, liegen insbesondere im Bereich der Verbesserung der Mensch–Maschine–Kommunikation, also bei

[1] Was sich auch ablesen läßt an der regelmäßigen Veranstaltung internationaler Workshops, vgl. u.a. [Kempen 1987], [Dale et al. 1990], [Paris et al. 1991], [Horacek, Zock 1993].

Beispiel: Bankautomat

System: Bitte Karte einführen
Benutzer: $< führt\ Karte\ ein >$

System: Bitte warten
Ihre Karte wird geprüft
System: Bitte Geheimzahl eingeben
Benutzer: $< gibt\ Geheimzahl\ ein >$

System: Bitte wählen Sie
Auszahlung
Drucken Kontostand
Dienstleistungen
Benutzer: $< wählt\ Auszahlung >$

System: Bitte wählen Sie

DM100	DM150
DM200	DM300
DM400	DM500

Benutzer: $< wählt\ Betrag >$

System: Bitte warten
Ihr Vorgang wird bearbeitet
System: Bitte Karte entnehmen
Benutzer: $< entnimmt\ Karte >$

System: Bitte Geld entnehmen
Benutzer: $< entnimmt\ Geld >$

System: Vielen Dank
Auf Wiedersehen

Abbildung 2.1: Ein alltäglicher „Dialog“ mit einem Automaten

- Auskunftssystemen,
- Dialogsystemen,
- Hilfesystemen,
- Erklärungskomponenten oder
- tutoriellen Systemen.

Zu anderen Anwendungen gehört

kein Fall für Generierung: Lückentexte

"Auch in der vergangenen Nacht hat die Radioaktivität in Bayern weiter abgenommen. So hat sich an den südbayrischen Meßstationen in München von gestern mittag auf heute morgen die Gamma–Dosis–Leistung von [15] auf [17], in Gundremmingen von [23] auf [25] und in Landshut von [37] auf [36] Mikroröntgen pro Stunde verändert ..."

(Bayrischer Rundfunk, 7.5.1986)

Abbildung 2.2: Lückentexte ... und ihre Probleme

- die Erstellung von Berichten etwa aus statistischen Daten und deren Auswertung oder
- die natürlichsprachliche Beschreibung von Datenbankinhalten, aber auch — wie in dieser Arbeit —
- die Generierung von technischen Dokumenten.

2.1.2 Beispiele von Textgeneratoren

Von welcher Art sind die Texte, die von Generierungssystemen produziert wurden? Eine kleine Auswahl zeigt die Tabelle in Abb. 2.3.

Beispiele generierter Texte		
YH	[Gabriel 81]	Dokumentationstext zu einem LISP–Programm
TEXT	[McKeown 83]	Information über Datenbank–Schemata
ANA	[Kukich 83]	Börsenberichte
NAOS	[Novak 86]	Beschreibung einer Straßenszene
SEMTEX	[Rösner 86]	Zeitungsmeldungen über Arbeitsmarkt
RAREAS	[Kittredge 86]	Wetterberichte
SOCCER	[André et al. 87]	Simultan–Kommentar eines Fußballreporters

Abbildung 2.3: Einige Systeme zur Textgenerierung

Wichtiger als die Frage nach dem erzielten „Endprodukt" ist die Frage nach dem Ausgangspunkt: Womit begann der Generierungsprozeß? Abb. 2.4 zeigt hierfür Beispiele.

In Textgeneratoren verwendete Primärdaten Beispiele:	
* Code eines Programms	YH
* Datenbankschema	TEXT
* Ausgabe einer Szenenanalyse	NAOS, SOCCER
* statistische Daten	ANA, SEMTEX
* meteorologische Daten	RAREAS

Abbildung 2.4: Beispiele von Primärdaten

2.1.3 Die klassische Dichotomie: Inhalt vs. Form

Die Verarbeitungsprozesse, die ein Generierungsprogramm durchläuft, werden häufig den beiden folgenden Kategorien zugewiesen:

- Prozesse, die den Inhalt des zu produzierenden Textes bestimmen (das **„What to say“**) im Kontrast zu
- solchen Prozessen, welche die Form des zu generierenden Textes bestimmen und diese auch erzeugen (das **„How to say“**).

Oft wird auch von *„strategischer“* vs. *„taktischer“* Komponente gesprochen.

Der Schwerpunkt der Arbeiten im Bereich Generierung liegt in den letzten Jahren auf Fragen der Planung von Texten, wobei hier in erster Linie Planung des Inhaltes und der Abfolge der Präsentation gemeint ist.

Für die morphologisch und syntaktisch korrekte „Ausführung“ des geplanten Inhalts — oft auch *Realisierung* (engl.: realisation) genannt — gibt es eine Reihe von Komponenten für unterschiedliche Sprachen, die für einen vielseitigen Einsatz konzipiert sind (für Deutsch siehe z.B. [Emele, Momma 1985], [Emele 1987], [Finkler, Neumann 1988]).

Neuere Arbeiten (z.B. [Bateman, Hovy 1991]) betonen, daß eine dritte, eigenständige Ebene betrachtet werden sollte (**„Why to say it?“**). Auf dieser werden die Fragen von Intentionalität und Pragmatik behandelt: Warum wird etwas artikuliert und warum in der dann jeweils gewählten Weise?

2.1.4 Ausgangspunkt: Situation und Kommunikationsbedürfnis

Betrachtet man den Generierungsprozeß sehr abstrakt, so läßt er sich etwa in folgender Weise charakterisieren (vgl. [McDonald et al. 1987]): Ausgangspunkt ist eine Situation, in der das Bedürfnis zu kommunizieren entsteht. Zu denken ist dabei etwa an das Bedürfnis, eine Frage beantwortet zu bekommen, das Bedürfnis, etwas

mitzuteilen, aber auch an die Fehlersituation, in die ein Betriebssystem gerät und Situationen ähnlicher Art.

Ausgehend von der Situation ist dann in einem ersten Schritt der Inhalt der Äußerung festzulegen (Was soll gesagt werden?). In den allermeisten Fällen wird wesentlich mehr Information zur Verfügung stehen als ausgedrückt wird. Hier geht es also um eine Selektionsaufgabe.

Die inhaltliche Spezifikation muß dann in eine sprachliche Spezifikation umgewandelt werden: Wie soll der Inhalt ausgedrückt werden? Hier gibt es eine Fülle von Entscheidungsnotwendigkeiten. Schließlich ist dann diese sprachliche Spezifikation in den Text umzusetzen.[2]

Ein Beispiel: Report-Generator

Die geschilderte Gesamtstruktur läßt sich erläutern am Beispiel eines Report-Generators. Die Situation könnte etwa eine Menge statistischer Daten sein (Beispiele: Börsendaten, andere Daten aus dem Wirtschaftsbereich), die von einem Programm analysiert werden, d.h. es werden Berechnungen angestellt und Vergleiche gezogen mit Daten früherer Zeiträume. Die Ergebnisse der Berechnungen und Vergleiche liefern einen Pool mit inhaltlichen Spezifikationen (vgl. [Kukich 1983]).

Daran anschließend ist zu entscheiden, was relevant genug ist, um darüber zu berichten; dies könnten etwa Maximalwerte sein, besonders krasse Abweichungen vom Gesamttrend oder Vergleiche mit früheren Zeitperioden. Diese Informationen müssen dann auch noch geordnet werden: In welcher Reihenfolge soll die Information gegeben werden? Hier ist oft eine Heuristik, die wichtigste Information an den Anfang zu stellen. Dies wird z.B. in Agenturmeldungen praktiziert, die ja so gestaltet werden, daß beim Redigieren einfach vom Schluß her gekürzt werden kann.

Der damit entstandene Textplan ist dann Ausgangspunkt für die Überlegungen zur sprachlichen Umsetzung. Nach Inhaltsauswahl und Inhaltsanordnung ist eine Fülle weiterer Entscheidungen zu treffen, etwa:

- Wieviel Information kommt in einen einzelnen Satz?
- Mit welchen Wörtern wird der Sachverhalt ausgedrückt (Lexikalisierung)?
- Wie wird zwischen syntaktischen Varianten (etwa: aktiv, passiv, unterschiedliche Konstituentenstellung) entschieden?
- Dann speziell in mehrsätzigen Texten: Können anaphorische Bezüge (z.B. durch Personalpronomen, Demonstrativpronomen u.ä.) oder Auslassungen genutzt werden?

[2]Diese idealisierte Darstellung von Stufen bei der Generierung sollte nicht als Vorgabe einer strikt sequentiellen Kontrollstruktur mißverstanden werden; die Frage nach Architekturen, welche die wechselseitigen Abhängigkeiten der beteiligten Prozesse berücksichtigen können, ist ein wichtiges Forschungsthema (vgl. z.B. [Horacek, Zock 1993]).

Diese Entscheidungen sind insbesondere dadurch charakterisiert, daß sie nicht isoliert getroffen werden können, sondern wechselseitigen Abhängigkeiten unterliegen. Ein gewähltes Verb kann etwa bestimmte Anforderungen an notwendige Ergänzungen mit sich bringen oder bestimmte syntaktische Formen wie Passivierung nicht zulassen, so daß sich dieser Aspekt von Generierung auch charakterisieren läßt als das Problem einer möglichst bewußten Auswahl zwischen miteinander in wechselseitigen Abhängigkeiten stehenden Möglichkeiten.

Besonders mehrsätziger Diskurs schafft Möglichkeiten, aber auch Entscheidungsnotwendigkeiten. Dies liegt insbesondere daran, daß vorausgegangene Sätze einen Kontext schaffen, innerhalb dessen interpretiert wird. Damit sind dann Rückbezüge und außerordentlich kompakte Informationsvermittlung möglich. Ein Beispiel:

> *„Um zu reinstallieren, die obigen Schritte in umgekehrter Reihenfolge ausführen.“*

Diese Äußerung ist wesentlich kompakter als die Form, bei der die Einzelschritte aufgeführt werden müssen; sie ist andererseits aber eben nur interpretierbar und nutzbar, wenn der vorausgegangene Text verstanden wurde.

2.1.5 Generierung als wissensbasierter Prozeß

Charakterisiert man Generierung als wissensbasierten Prozeß, so sind insbesondere drei Arten von Wissensquellen zu nennen, die in die Generierungsentscheidungen einfließen.

- Zum ersten wird Wissen über die Sprache benötigt. Dazu gehören nicht nur Grammatik und Lexikon, sondern auch Wissen über typische Textstrukturen und typische sprachliche Ausdrucksformen. So ist eine Äußerung wie *„Über Ihr Angebot haben wir uns irre gefreut“* in einem Geschäftsbrief inadäquat, eine strukturell analoge Äußerung wie *„Über Deine Einladung habe ich mich irre gefreut“* in einem privaten Brief aber völlig akzeptabel.

- Eine weitere Wissensquelle ist Wissen über den Gegenstandsbereich, aus dem die Inhalte des Textes gewählt werden. Beides allein reicht aber auch noch nicht aus.

- Für viele Entscheidungen bei der Gestaltung von Texten ist auch Wissen über den Adressaten, den Leser, notwendig (vgl. Abschnitt 5.6.11).

2.1.6 Inhaltsbestimmung über Schemata

Bei mehrsätzigen Texten muß die Abfolge der Einzelsätze in kohärenter Weise erfolgen. Bei einigen Anwendungen von Textgenerierung kann man sich auf die Strukturierung des Anwendungsgebietes verlassen, z.B. wenn bereits eine temporale Abfolge

— wie bei der Generierung von Kochrezepten [Dale 1990] oder bei der Beschreibung von Spielen [Davey 1979] — vorgegeben ist.

Ein etwas allgemeinerer Ansatz sind die von McKeown in ihrem TEXT–System vorgeschlagenen Schemata [McKeown 1985]. Mit diesen kann beschrieben werden, wie ein bestimmter Typ von standardisiertem Text (z.B. Begriffsdefinition, Kochrezept, Wetterbericht) üblicherweise aufgebaut ist, d.h. welche Inhalte in welcher Reihenfolge anzusprechen sind. Für das TEXT–System wurden vier Schemata definiert:

- Identification,
- Attributive,
- Constituency,
- Compare / Contrast.

Das Constituency genannte Schema gibt z.B. an, wie ein Begriff über seine Konstituenten erläutert werden kann. Das Schema umfaßt vier Einzelschritte:

1. Identifiziere den zu beschreibenden oder zu definierenden Begriff als Element einer allgemeinen Klasse oder gib Eigenschaftsinformationen über ihn.
2. Präsentiere die konstituierenden Elemente des Begriffs.
3. Präsentiere charakteristische Informationen über jedes konstituierende Element.
4. Präsentiere Eigenschafts– oder Analogie–Informationen über den Begriff.

Das Constituency–Schema wurde verwendet, um die folgende Erläuterung zur Benutzerfrage nach dem Begriff „guided“ in Bezug auf die Datenbank des Office of Naval Research (ONR) zu geben (das erste Schema–Element war dabei für Satz 1, das zweite für Satz 2, das dritte für die Sätze 3 – 6 und das vierte für Satz 7 verantwortlich, vgl. [McKeown 1985]):

> *[1] A guided projectile is a projectile that is self-propelled. [2] There are 2 types of guided projectiles in the ONR database: torpedoes and missiles. [3] The missile has a target location in the air or on the earth's surface. [4] The torpedo has an underwater target location. [5] The missile's target location is indicated by the DB attribute DESCRIPTION and the missile's flight capabilities are provided by the DB attribute ALTITUDE. [6] The torpedo's underwater capabilities are provided by the DB attributes under DEPTH (for example, MAXIMUM OPERATING DEPTH). [7] The guided projectile has DB attributes TIME TO TARGET & UNITS, HORZ RANGE & UNITS and NAME.*

2.1.7 Eingabestrukturen für die Generierung?

Was bei Parsing–Systemen keine Frage darstellt, ist für Generierungssysteme von wesentlicher Bedeutung: *Mit welchen Eingabestrukturen beginnen diese Systeme?*[3]

Versteht man unter Generierung den gesamten Komplex aus Inhaltsbestimmung und Inhaltsvermittlung, so wäre die Eingabe zur Generierung der Informationspool oder die repräsentierte Situation, von der dieser Gesamtprozeß ausgeht, also etwa die Dow–Jones–Daten für ein Programm zur Generierung von Börsenberichten oder die Ergebnisse eines Programms zur Szenenanalyse als Eingabe in ein Programm zur verbalen Beschreibung der Szenen.

Versteht man unter Generierung nur die Realisierungskomponente, dann sind Eingabestrukturen typischerweise als semantische Repräsentationen angesehene sog. Prädikat–Argument–Strukturen wie *love (Hans, Maria)*.

Betrachtet man solche Prädikat–Argument–Strukturen isoliert, so ist deutlich, daß sie in erheblichem Maße unterspezifiziert sind. Dies läßt sich schon an dem einfachen Beispiel erläutern: Wenn ein Generierungsprogramm für die Eingabe 'love (Hans, Maria)' den deutschen Satz *„Hans liebt Maria“* produziert, so würde intuitiv dieser (simple) Satz als Generierungsergebnis wohl akzeptiert werden. Eine solche Akzeptanz gründet sich aber auf eine ganze Reihe von Annahmen.

Da ist zunächst die Annahme über die Semantik des Prädikats 'love', die in die Akzeptanz der sprachlichen Realisierung einfließt. Bei anderer Semantik könnte das Generierungsergebnis ja auch heißen *„Maria liebt Hans“* oder, wenn besonders die wechselseitige Beziehung in dem Prädikat ausgedrückt werden soll, *„Hans und Maria lieben einander“*.

Klarheit über die Bedeutung der semantischen Repräsentation allein reicht aber ebenfalls noch nicht aus. Wir haben stillschweigend den Satz im Präsens akzeptiert; in dieser isolierten Form lag eben keinerlei Information vor, die eine exakte Tempus–Bestimmung erlauben und dann Sätze wie *„Hans hat Maria geliebt“* oder *„Hans wird Maria lieben“* zur korrekten Ausgabe machen würde. Dort, wo keine Information vorliegt, muß sich ein Generierungsprogramm auf Voreinstellungen stützen (wie hier: Tempus Präsens).

Ist ein Prädikat eingebettet in einen komplexeren Zusammenhang, so ergeben sich daraus ebenfalls Konsequenzen für die Art der linguistischen Realisierung. In entsprechendem Verwendungszusammenhang kann dem Prädikt auch eine nominalisierte Form zugeordnet werden, z.B.: *„(er äußerte sich abfällig über . . .)“* oder *„(er zeigte sich erfreut über) die Liebe von Hans und Maria“*.

Schließlich spielt der vorangegangene Text ebenfalls eine wichtige Rolle für die Variabilität des Realisierungsergebnisses. Formen wie *„er liebt sie“* oder *„er liebt dieses Mädchen“* können dem Prädikat in einem passenden Kontext als Realisierungsergebnis zugeordnet werden.

Zusammenfassend: Es ist meist wenig sinnvoll, eine isolierte Prädikat–Argument–

[3] Dieses Thema ist so zentral, daß es beispielsweise auf dem europäischen Generierungs–Workshop im Jahr 1991 in einer eigenen Podiumsdiskussion behandelt wurde [Horacek, Zock 1993].

Struktur (oder andere semantische Struktur) als *die* Eingabe in ein Generierungssystem anzusehen und von dem ihr zugeordneten Generierungsergebnis zu sprechen. Situationeller und textueller Kontext sowie die im Generierungssystem expliziten oder impliziten Annahmen über Werte unspezifizierter Merkmale gehören mit zur Spezifikation der Eingabe.

Eine vergleichbare Situation liegt ja auch in der Text–Interpretation vor. Dort kann man einer Aussage, wenn man sie isoliert betrachtet, oft ebenfalls nur schwer eine Interpretation zuordnen und der Kontext in dem genannten Sinne muß bei der Analyse mit einbezogen werden.

2.1.8 Perspektive in der Lexikalisierung

Ein häufig diskutiertes und fast schon klassisches Beispiel für Fragen der Lexikalisierung ist das folgende (siehe auch [Jacobs 1984]): Ein Kauf- / Verkaufsvorgang wird modelliert als zwei sich wechselseitig bedingende Teilvorgänge, einen Vorgang der Übertragung von Geld und einen der Übertragung des gekauften Objekts.

Je nach Perspektive läßt sich dieses Ereignis auf sehr unterschiedliche Weise lexikalisieren. Beispiele:

- *„Hans kaufte von Fritz einen Gebrauchtwagen für DM 1.000,–. “*
- *„Hans bezahlte für den Gebrauchtwagen an Fritz DM 1.000,–. “*
- *„Fritz verkaufte seinen Wagen an Hans für DM 1.000,–.“*
- *„Fritz erhielt von Hans für den Gebrauchtwagen DM 1.000,–.“*
- ...

Bei diesen Beispielen haben wir uns im wesentlichen auf Lexikalisierungen konzentriert, die einen Prozeß ausdrücken und bei denen hier einer der Teilaspekte (Kauf oder Verkauf) im Vordergrund stand. Das Ereignis könnte aber noch zu ganz anderen sprachlichen Aussagen führen, zum Beispiel solchen, bei denen eine der beteiligten Personen oder eines der Objekte im Blickpunkt stehen. Beispiele:

- *„Das Auto, das Hans von Fritz kaufte, hat DM 1.000,– gekostet.“*
- *„Für (nur) DM 1.000,– hat Fritz sein Auto an Hans verkauft.“*
- ...

Liegen zusätzliche Aspekte vor, so ergibt sich ein noch größeres Spektrum an Aussagemöglichkeiten. Einige Beispiele:

- *„Fritz hat Hans für seinen alten Wagen noch DM 1.000,– abgeknöpft.“*
- *„Hans ließ sich den Wagen, den er von Fritz kaufte, immerhin DM 1.000,– kosten.“*

- *„Trotz neuem TÜV erbrachte der Wagen von Fritz beim Verkauf an Hans nur noch DM 1.000,–.“*
- *„Für den fast nicht mehr fahrtüchtigen Wagen von Fritz hat Hans immerhin DM 1.000,– berappen müssen.“*
- ...

Diese Beispiele mögen genügen, um deutlich zu machen, daß es ein hoffnungsloses Unterfangen wäre, eine vollständige Auflistung der sprachlichen Möglichkeiten zu versuchen. Sie unterstreichen noch einmal die Situationsbezogenheit sprachlicher Äußerungen (vgl. 2.1.7).

2.2 Exkurs: Generierung und Computeranimation

Fragt man nach anderen Gebieten der Informatik, mit denen „Generierung“ Gemeinsamkeiten aufweist, so kommt das Gebiet der Computeranimation ins Blickfeld.

2.2.1 Präsentation einer natürlichen Form

Eine offensichtliche Gemeinsamkeit beider Gebiete ist die, daß eine — wie auch immer geartete — computer–interne Repräsentation in eine „natürliche“ Form umgesetzt wird. „Natürlich“ heißt in diesem Zusammenhang, daß die Ausgaben der Systeme dem Menschen besonders leicht zugänglich sind; in einem Fall ein Bewegungsablauf, im anderen Fall ein Text. Damit wird auch die Anwendungsrelevanz beider Gebiete deutlich: Ihre Ergebnisse können zu einer erheblichen Verbesserung der Mensch–Maschine–Kommunikation beitragen. Dies gilt für Ergebnisse jedes einzelnen der Gebiete, erst recht könnten kombinierte Formen zu ganz neuen Möglichkeiten führen. So könnte z.B. ein auf dem Rechner — möglicherweise in Ergänzung eines Diagnose–Systems — verfügbares „Reparaturhandbuch“ nicht nur erklärenden Text enthalten, sondern auch Animationen, welche die auszuführenden Schritte veranschaulichen.

Die „Natürlichkeit“ der Ausgabe stellt hohe Ansprüche an die zugrundeliegenden Programme; ihre Ergebnisse sind auch für den Laien unmittelbar beurteilbar. Eine „unnatürliche“ Bewegung wird sofort als solche erkannt, auch wenn vielleicht nicht einmal verbalisiert werden kann, worin die Fehlerhaftigkeit begründet liegt. Ähnliches gilt für die Qualität von Texten.

2.2.2 Die Hauptaufgaben

Für die Computeranimation werden oft die folgenden drei Hauptaufgaben angeführt [Thalmann et al. 1988]:

- Objektbeschreibung,

- Bewegungsfestlegung und
- Bilderzeugung.

In erster Näherung lassen sich diese Aufgaben mit denen bei der Textgenerierung in Beziehung setzen: der Inhaltsauswahl, der Inhaltsanordnung und der Festlegung der syntaktischen Merkmale und ihrer Realisierung (vgl. 2.1.3).

2.2.3 Flexibilität durch Wissensbasierung

Ausgangspunkt für ein Fachgespräch bei der GI-Jahrestagung 1990 ([Reuter 1990], pp. 548 - 600) war die Feststellung, daß bei der Computeranimation derzeit jede Einzelheit noch explizit beschrieben werden muß. Bei der Veranstaltung wurde dann nach besseren Vorgehensweisen gesucht. So wurde vorgeschlagen, daß u.a. durch Ansätze aus der KI — wie objektorientiertes Programmieren — das Animationssystem in die Lage versetzt werden soll, alle vorkommenden Objekte mit ihren physikalischen und verhaltensmäßigen Eigenschaften zu „kennen“, um auf der Basis diesen Wissens dann in der Lage zu sein, Bewegungen selbständig aufgrund von Vorgaben zu generieren. Solche Möglichkeiten sind umso notwendiger, je mehr Interaktion zwischen den Elementen einer Animation stattfindet (z.B. Hindernisse, Rückwirkungen von Objekten aufeinander, u.ä.).

Die Notwendigkeit, alle Einzelheiten einer Animation festlegen zu müssen, kann verglichen werden mit der Verwendung von vorgefertigten Texten ('canned texts'), wie sie in fast jedem Programm und in vielen einfachen Benutzerschnittstellen vorkommen (z.B. als Hinweise oder Fehlermeldungen).

Bei der Verwendung von vorgefertigten Texten ist „Generierung“ letztendlich ein Indexierungsproblem. Für überschaubare Anwendungen (wie z.B. eine kleine Anzahl von Fehlermeldungen) lassen sich die benötigten Texte ablegen und für das Retrieval geeignet indizieren; für größere Anwendungen ist dieser Ansatz unrealistisch, da keine Verallgemeinerungen erfaßt und nutzbar gemacht werden können. Ferner ist der ausgedruckte Text dem System lediglich als Zeichnfolge und in keiner inhaltsbezogenen Weise zugänglich. Damit entfallen auch alle Möglichkeiten eines systemseitigen Kontextbezugs, z.B. in der Form *„wie bereits oben ausgeführt, ...“* .

Sowenig eine bloße Aneinanderreihung von Einzelsätzen einen akzeptablen Text darstellt, sowenig attraktiv erscheint für die Animierung das Vorgehen, eine Folge von einzelnen Bitmaps abzulegen und dann abzuspielen, so daß der Eindruck einer kontinuierlichen Bewegung entsteht. In beiden Fällen ist explizit verfügbares Wissen über die beteiligten „Objekte“ notwendig. Dieses Wissen beinhaltet auch die Information über die Art, wie sich diese Objekte dann im jeweiligen Generierungsprodukt darstellen. So muß bei Generierung, die den Kontext berücksichtigt, in einem Text nicht nur zur Verfügung stehen, daß ein bestimmtes Objekt in einem vorausgehenden Satz bereits erwähnt worden ist, sondern auch, mit welcher Nominalgruppe dies geschah. Diese Information fließt dann ein in die Entscheidung, ob ein Pronomen

zur Referenz verwendet werden kann, und wird dann auch zur Bildung des korrekten Pronomens benötigt.

In die Entscheidung kann weiter einfließen, ob konkurrierende Nominalgruppen im Umfeld existieren, auf die der pronominale Verweis fälschlicherweise vom Leser bezogen werden könnte. Diese Abwägung kann ggf. komplexe Überlegungen erfordern über die Möglichkeit zur semantischen Disambiguierung (Wird der Leser Referenzen, die zwar grammatisch aufgrund des gleichen Genus und Numerus anderer Nominalgruppen möglich wären, schon deshalb nicht als Ambiguität empfinden, weil sie aus semantischen oder anderen Gründen auszuschließen sind?).

Ähnliche Notwendigkeiten, andere beteiligte Objekte zu kennen, existieren in der Animation: Wechselwirkungen von beispielsweise sich anstoßenden Kugeln mit unterschiedlichen Geschwindigkeiten und Richtungen oder die Notwendigkeit von Richtungs- und Geschwindigkeitsänderungen aufgrund von Hindernissen oder anderen sich bewegenden Objekten.

2.3 Grammatikformalismen für die Generierung

Wenn man sich mit der Generierung natürlicher Sprache beschäftigt, so ergibt sich eine neue Sicht auf grammatische Theorien bzw. grammatische Formalismen. Es erweist sich beispielsweise, daß bestimmte grammatische Formalismen für die Aufgabe der Generierung geeigneter sind als andere und daß daher viele Generierungssysteme mit Grammatik–Theorien bzw. –Formalismen arbeiten, die in den Arbeiten zur Analyse natürlicher Sprache nicht die gleiche Aufmerksamkeit erfahren.

2.3.1 Systemische Grammatik

Grammatisch korrekte Sätze produzieren zu können, reicht für ein Generierungssystem nicht aus. Ziel ist es vielmehr, *für die jeweilige Kommunikationssituation adäquate* sprachliche Äußerungen zu erzeugen. Jeder nur auf Grammatikalität ausgerichtete Formalismus stößt für diese Aufgabe an Grenzen.

Es ist daher nicht verwunderlich, daß für Generierungssysteme besonders solche grammatischen Ansätze interessant sind, die eine weniger strukturelle, sondern stärker funktionale Sicht auf die Sprache haben und die Sprachverwendung in den Vordergrund des Interesses rücken. Darin gründet sich auch die besondere Bedeutung von systemischer Grammatik[4] für das Gebiet Generierung. Systemische Grammatik wird entweder direkt als Basis für die Entwicklung von Generierungssystemen verwendet oder beeinflußte zumindest bei einzelnen Systemen die Entwicklung sehr stark.

Systemische Grammatik hat ihre Wurzeln eher in der Anthropologie und Soziologie und viel weniger in Mathematik und formaler Logik ([Halliday 1978],

[4] Oft auch systemisch–funktionale Grammatik oder kurz SFG; s. z.B. [Halliday 1978], [Butler 1985]

[Halliday 1985]). Im Zentrum steht die Frage nach Sprache als sozialer Aktivität (vgl. [Winograd 1983]):

- Was sind soziale Funktionen von Sprache?
- Wie erfüllt Sprache diese sozialen Funktionen?
- Wie arbeitet Sprache?

Sprache wird betrachtet als Ansammlung syntaktischer Ressourcen, auf die sich ein Sprecher stützen kann, um in einer Kommunikationssituation erfolgreich eine Bedeutung zu vermitteln. Er benutzt die Ressourcen, um die unterschiedlichen Aspekte (Metafunktionen, s.u.) seiner Botschaft zu realisieren.

Systemische Grammatik versucht, jegliche syntaktische Alternation auf Unterschiede — und seien sie ggf. auch nur minimal — in der intendierten Bedeutung, im sozialen Kontext usw. zurückzuführen. Sie hat zur Beschreibung solcher Unterschiede ein reichhaltiges Repertoire an Merkmalen und Merkmalswerten entwickelt. Hat man solche Merkmale und Merkmalswerte zur Verfügung, so kann man sie umgekehrt zu einer bewußten Auswahl aus der Fülle der verfügbaren syntaktischen Ressourcen verwenden.

Auf Halliday geht die Unterscheidung in die folgenden drei hauptsächlichen Metafunktionen in der Sprache von Erwachsenen zurück (vgl. [Halliday 1985]):

- Die sog. ideationale Funktion ist inhalts- oder bedeutungsorientiert. Was sagt eine Äußerung über die Welt, über Objekte, Eigenschaften und Zustände?

  Die Transitivitätsstruktur von Sätzen bzw. Satzteilen dient in erster Linie dazu, die ideationale Funktion zu realisieren (vgl. 2.3.1).

- Die interpersonale Funktion hat mit der Frage zu tun, warum die Äußerung gemacht wird. Geschieht dies, um z.B. einen Sachverhalt mitzuteilen, eine Aktion zu verlangen, eine Reaktion auszudrücken oder irgendeiner Konvention Genüge zu tun?

  Die interpersonale Funktion muß mit geeigneten syntaktischen Strukturen mit der ideationalen Botschaft der Äußerung verbunden werden. Der Satzmodus (engl: mood) ist eine der Ressourcen zur Realisierung der interpersonalen Funktion.

- Die textuelle Funktion muß für eine kohärente Kommunikation Sorge tragen. Der Hörer oder Leser soll dabei unterstützt werden, die Beziehungen zwischen den einzelnen Elementen der Information zu erkennen. Typische Fragen sind hier die nach Thema und Rhema oder topic und focus: „Was ist neu und was kennt der Leser / Hörer bereits? Was ist wichtig und was sekundär?“.

  Zur Erfüllung der textuellen Metafunktion tragen insbesondere die thematische und die informationelle Struktur von sprachlichen Äußerungen bei.

Systeme in der systemischen Grammatik

Konstituenten eines Satzes erfüllen verschiedene Funktionen. Eine Schlüsselbeobachtung der systemischen Grammatik (daher bezieht sie auch ihren Namen) ist die, daß diese Funktionen als zu verschiedenen Systemen gehörend analysiert werden können. Die Struktur eines Satzes wird durch die simultane Operation dieser Systeme bestimmt.

Für die Analysen englischer Sätze[5] haben sich vier grundlegende Systeme herausgeschält. Dies sind:

- das System des Modus (mood),
- das System der Transitivität (transitivity),
- das thematische System
- und das informationelle System.

In jedem der Systeme gibt es Funktionen, die zu denen anderer Systeme orthogonal liegen.

Die Systeme im einzelnen:

- das System des Modus:
  Hier geht es um die Frage, ob die 'clause' eine Frage, einen Befehl, eine Aussage oder einen der unterschiedlichen Typen eingebetteter Sätze bildet. Dies drückt sich im wesentlichen in der relativen Positionierung der Elemente aus, insbesondere in der Position des Oberflächensubjekts.

- das System der Transitivität:
  Zu jedem Satz gehört ein Prozeß oder eine Beziehung — meist durch das Verb spezifiziert — und ein oder mehrere Partizipanten, die durch Nominalgruppen, Präpositionalphrasen oder Teilsätze spezifiziert sind. Die Transitivitäts–Struktur repräsentiert den propositionalen Gehalt eines Satzes (*„Wer machte was mit wem?“*). Die hauptsächlichen Partizipanten werden oft als Actor (Partizipantenrolle für den Durchführenden der Aktion) und Actee (Partizipantenrolle für Person oder Objekt, mit der/dem die Aktion durchgeführt wird) bezeichnet.

- das thematische System:
  In einer systemischen Analyse beginnt jeder Satz mit einer Konstituente, die als 'theme' fungiert, gefolgt vom Rest des Satzes, 'rheme' genannt. Dieser Aspekt der Satzstruktur ist in traditionellen und transformationellen Grammatiken weitgehend ignoriert worden; auch in vielen implementierten Systemen werden Variationen der Ordnung aufgrund thematischer Auswahl vernachlässigt (s.a. [Winograd 1983], p. 284).

[5] in der systemischen Benennung „clauses“

- das informationelle System:
  Ein Sprecher kann nicht nur ein Element als Thema eines Satzes hervorheben, er kann zusätzlich eine Struktur aufprägen, die deutlich macht, welcher Teil der Mitteilung als neue Information und welcher als bereits gegebene zu sehen ist, die dem Hörer schon bekannt sein sollte.

  Bei gesprochener Sprache interagiert dieses System besonders deutlich mit dem informationellen System: Eine Standard–Informationsstruktur ist durch die Ordnung der Konstituenten gegeben; diese läßt sich aber überschreiben durch die Auswahl von Betonung und Intonation. Das Ende des Satzes ist die unmarkierte Position für neue Information; unterschiedliche grammatikalische Formen erlauben es nun verschiedenen Konstituenten, diese Position einzunehmen, so etwa die Passivierung, die (im Englischen) den Aktor aus der Normalposition als Subjekt am Anfang des Satzes in eine optionale 'by'–Phrase am Ende bewegt.

Zur Unterscheidung zwischen 'groups' und 'phrases'

Abweichend von der normalerweise in der Linguistik verwendeten Terminologie werden in der systemischen Grammatik 'groups' als Formen von 'word complexes' betrachtet, d.h. „ *...as a combination of words built up on the basis of a particular logical relation*“ ([Halliday 1985], p. 159). Es werden drei Hauptklassen von Gruppen in diesem Sinne unterschieden: Nominalgruppe, Verbalgruppe und Adverbialgruppe.

Als 'phrase' werden dagegen strukturelle Bestandteile bezeichnet, die als 'contraction of a clause' zu sehen sind, d.h. im wesentlichen die Präpositionalphrase. „*Starting from opposite ends, the two achieve roughly the same status on the rank scale as units that lie somewhere intermediate between the rank of a clause and that of a word*“ ([Halliday 1985], p. 159).

Abhängigkeiten zwischen Systemen

Die Systeme in einer SFG sind in einem Netzwerk organisiert. Dabei ist es typisch, daß die Werte eines Systems darüber entscheiden, welche Folgesysteme überhaupt zugänglich sind. Ein einfaches Beispiel: Innerhalb der Nominalgruppe gibt es die Funktion 'DEICTIC'. Das deiktische Element kann entweder 'spezifisch' oder 'non–spezifisch' als Wert haben; abhängig davon gibt es für Englisch zwei unterschiedliche NUMBER–Systeme. Bei der spezifischen Deiktik ist die Unterscheidung im NUMBER–System die in 'non–plural' und 'plural'; dabei werden dann 'mass nouns' zusammen mit Singular in die Kategorie 'non–plural' gruppiert. Im Kontrast dazu gilt bei der non–spezifischen Deiktik im NUMBER–System die Unterscheidung 'singular' und 'non–singular'; in diesem Fall sind 'mass nouns' dann zusammen mit Plural in der Kategorie 'non–singular' klassifiziert (vgl. [Halliday 1985], pp. 161, 162).

Zur Analyse und Realisierung von Prozessen

In Halliday's Analysen hat ein Prozeß drei Komponenten [Halliday 1985]:[6]

- den Prozeß selbst,
- die Teilnehmer (participants) des Prozesses und
- die Umstände (circumstances), die mit dem Prozeß verknüpft sind.

Für die drei Komponenten gibt es typische sprachliche Realisierungen. Der Prozeß selbst führt im Normalfall zu einer Verbalgruppe. Die Partizipanten sind oft Objekte oder Personen und werden üblicherweise als Nominalgruppen realisiert. Es gibt allerdings auch Fälle (z.B. Kausalität oder mentale Prozesse, die sich auf Einstellungen usw. beziehen), bei denen Prozesse andere Prozesse und nicht nur Objekte verknüpfen.

Umstandsbestimmungen werden oft als Präpositionalphrasen realisiert. Sie liefern u.a. Informationen zur zeitlichen und räumlichen Plazierung des Prozesses, zur Art seiner Durchführung, zu seinem Zweck.

Die Partizipanten werden als stärker mit dem Prozeß verbunden gesehen als die Umstandsbestimmungen. Eine solche Unterscheidung in mehr essentielle und mehr periphere Elemente wird von den meisten syntaktischen Theorien getroffen.[7]

Halliday betont in [Halliday 1985] eine andere wichtige Unterscheidung, nämlich die zwischen einem *transitiven* Modell der Partizipanz in Prozessen im Kontrast zu einem *ergativen* Modell. Eine typische transitive Interpretation der Satzstruktur erfolgt in der Form, daß ein Handelnder (actor) im Hinblick auf ein Ziel (goal) agiert; hierbei wird eine Dimension von Extension im folgenden Sinne ins Blickfeld gebracht:

> *'The actor is engaged in a process; does the process extend beyond the actor, to some other entity or not?'* [Halliday 1985]

Halliday gibt hier das Beispiel: *'The lion chased the tourist'*, das er in Beziehung zu *'The lion ran ...'* sieht im Sinne *'The lion did some running'*. Bleibt es dabei, so liegt der intransitive Fall vor (*'The lion ran'*); wird die Handlung auf einen anderen Partizipanten ausgedehnt, so ist eine Möglichkeit der transitive Fall (*'The lion chased the tourist'*).

Die komplementäre Analyse ist dann in Begriffen eines ergativen Modells, bei dem der Fokus auf Verursachung (causation) statt Extension gelegt ist.

> *'Some participant is engaged in a process; is the process brought about by that participant or by some other entity?'* [Halliday 1985]

[6] Daran orientiert sich auch die Modellierung von Prozessen im 'upper model' (vgl. 6.5.4).

[7] Beispiele dafür sind 'bound elements', inhärente Rollen [Fawcett 1987], Theta-Rollen, thematische Beziehungen ([Jackendoff 1983], [Jackendoff 1991]), Prädikat–Argument–Strukturen [Bresnan 1982].

Am Beispiel von oben erläutert bedeutet dies, daß *'The lion chased the tourist'* nun weniger mit *'The lion ran ...'* als mit *'The tourist ran ...'* in Beziehung gesetzt wird. Die Frage ist dann, ob das Laufen vom Touristen selbst ausgelöst wurde (intransitiver Fall: *'The tourist ran'*) oder von einer externen Handlungsinstanz (transitiver Fall: *'The lion chased the tourist'*).

Werden Paare von ergativen und nicht–ergativen Sätzen gebildet (z.B. *'Mary sailed the boat'* und *'The boat sailed'*), so wird der Partizipant, der beiden Elementen gemeinsam ist, aus funktionaler Perspektive als *Medium* bezeichnet. Diese Funktion gilt als Schlüsselfigur in jedem Prozeß:

> *'The one through which the process is actualized and without which there would be no process at all.'* [Halliday 1985]

Zusammenfassend läßt sich sagen, daß für die transitive Interpretation die hauptsächliche Konfiguration der Partizipanten die von Actor und Goal ist, beim ergativen Standpunkt die von Agent und Medium; Actor und Agent wie auch Goal und Medium müssen nicht übereinstimmen [Halliday 1985].

2.3.2 Systemische Grammatik als Basis für Generierungssysteme

Als ein Forschungsziel in der Generierung kann gesehen werden, solche Lösungen zu finden, die es erlauben, möglichst vielseitig einsetzbare — d.h. leicht auf andere Themengebiete, Textsorten, Aufgabenstellungen, usw. übertragbare — und möglichst flexible Systeme zu ermöglichen. Die Arbeiten hierzu betreffen sowohl die benötigten Wissensquellen als auch die dafür geeignetsten Verarbeitungsmechanismen.

Die größte explizit zugängliche Flexibilität bieten Generierungssysteme, bei denen ein System von Merkmalen (features) und zugehörigen Werten zugrundegelegt ist, die es erlauben, *jede* sprachliche Variabilität und Variationsmöglichkeit durch eine Kombination von Merkmalen und ihren Werten zu beschreiben.

In der Tradition der systemisch–funktionalen Grammatik wurde ein umfangreiches Beschreibungsinventar dieser Art entwickelt und dient als Basis für das Generierungssystem PENMAN mit NIGEL als Grammatik–Komponente für Englisch (vgl. [Matthiessen, Bateman 1991], [PENMAN]).

2.3.3 PENMAN

PENMAN wurde als portierbare Komponente zur Textgenerierung entworfen.[8] Die gewählte Organisation soll das System von speziellen Wissensgebieten unabhängig machen. PENMAN läßt sich gliedern in die linguistische Komponente NIGEL und eine Umgebung (environment) genannte Komponente. NIGEL ist eine große (nach

[8] Das System wurde am Information Sciences Institute in Marina del Rey, Kalifornien / USA entwickelt und implementiert.

Aussagen der Autoren die größte) Grammatik im Stile von Hallidays systemisch–funktionaler Grammatik. NIGEL wiederum besteht aus der eigentlichen Grammatik und einer Komponente, die Grammatik und Umgebung miteinander zu kommunizieren erlaubt.

NIGEL ist ein Netzwerk mit ca. 600 Entscheidungspunkten (sog. 'systems'), wobei jedes System eine einzelne minimale grammatische Alternation repräsentiert. Bei der Generierung durchläuft PENMAN das NIGEL–Netzwerk und bestimmt dabei so lange Merkmale an den Entscheidungspunkten, bis genügend Information für einen voll spezifizierten Satz vorhanden ist.

Technisch realisiert werden Entscheidungen dadurch, daß jedem System in NIGEL eine Entscheidungsprozedur ('chooser' genannt) zugeordnet ist. In diesen Prozeduren wird insbesondere Information aus Wissensquellen, die extern zur Grammatik sind, für die Entscheidung über die grammatische Struktur herangezogen. Die Schnittstelle hierfür sind formalisierte Fragen, die sogenannten 'inquiries', die an die externen Wissensquellen gerichtet sind. Diese Systemfragen sind so definiert, daß sie mit abstrakten Zugriffsfunktionen arbeiten und so unabhängig von der speziellen Syntax der Repräsentationssprache sind, in der Wissensquellen in NIGELs Umgebung implementiert werden. Will man PENMAN an unterschiedliche Formalismen zur Wissensrepräsentation und an unterschiedliche Sachgebiete anpassen, so ist es nur notwendig, daß die von den Inquiries abgefragten Unterscheidungen in der Wissensrepräsentation darstellbar und modelliert sind.

Für einige der extern zur Grammatik liegenden Wissensquellen bietet PENMAN Strukturierungshilfen an. Hierzu gehört insbesondere das sogenannte 'upper model'; dies ist eine Begriffshierarchie von Beziehungen und Entitäten, mit der die jeweiligen Modelle des Sachgebiets geeignet verbunden werden müssen (vgl. [Bateman 1990]). Dadurch ist es möglich, die von den Systemfragen erforderlichen Unterscheidungen auch tatsächlich verfügbar zu haben.

Fragen an die Wissensbasen der Umgebung

Die systemisch–funktionale Grammatik betont, daß sich sprachliche Oberflächenformen danach unterscheiden, welche funktionalen Unterschiede ihren Ausdruck finden sollen. Die beim Durchlaufen der Grammatik gestellten Systemfragen an die Wissensbasen der Umgebung beziehen sich auf Unterschiede im Hinblick auf die drei Metafunktionen. Sofern nicht auf Voreinstellungen zurückgegriffen wird, sollen diese Fragen auch mit den in den Wissensbasen vorhandenen Informationen beantwortet werden.

Zu den Fragen zum Inhalt einer Äußerung (ideationale Metafunktion) gehören insbesondere Fragen zur Kategorisierung der beteiligten Prozesse, Objekte und Eigenschaften sowie ihrer Beziehungen (d.h. Fragen danach, welche Art von Subsumptionsbeziehung vorliegt). Realisiert ist die inhaltsbezogene Wissensbasis durch das 'upper model' und das darunter subsumierte Gebietsmodell.

Zur interpersonalen Metafunktion gehören Fragen wie: Wer macht die Äußerung

gegenüber wem? Welcher Sprechakt liegt zugrunde? Was sind die Ziele des Sprechers und Hörers?

Zur textuellen Metafunktion gehören alle Fragen, die sich auf den bereits produzierten Text beziehen, also die Fragen der Vorerwähntheit, der eindeutigen Referenzierbarkeit, usw..

Die Sentence Plan Language (SPL)

Mit Ausdrücken der Sentence Plan Language (SPL, [Kasper 1989]) kann der Inhalt spezifiziert werden, der durch den Sprachgenerator PENMAN ausgedrückt werden soll. SPL–Terme sind rekursiv aufgebaut. Sie bestehen aus einem Kopf und einer Liste von Paaren aus Rollen und weiteren SPL–Termen als Füllern. Im Kopf wird ein Identifikator als erstes Element eingeführt. Mit diesem kann auf den gesamten Ausdruck an anderer Stelle wieder Bezug genommen werden. Nach dem aus syntaktischen Gründen eingeführten '/' folgt dann als drittes Element des Kopfes der Bezeichner für den Typ des Objekts, Prozesses oder sonstigen dargestellten Konzepts. Dem schließt sich dann eine Liste von Rollen und ihren wiederum als SPL–Termen gegebenen Füllern an.

In SPL–Termen können auch sogenannte SPL–Makros verwendet werden. Damit lassen sich die Eingaben für das Generierungssystem kompakter schreiben. Diese Makros expandieren bei der Interpretation des SPL–Terms in ein Bündel von Merkmalen. Diese Merkmale gehören zur Menge der möglichen Antworten auf Systemfragen, die während der Netz-Traversierung gestellt werden können. Der Abarbeitungsmechanismus ist so organisiert, daß dann, wenn bereits im SPL–Term Antworten auf Inquiries angegeben sind, die Befragung der Umgebung entfällt. In den SPL–Makros werden solche Bündel von Merkmalen zusammengefasst, die als Kombination den durch das Makro angebenen Effekt auf der syntagmatischen Ebene nach sich ziehen.

SPL–Terme sind hybrid in dem Sinne, daß sie nicht nur inhaltliche Elemente des zu generierenden Satzes zu spezifizieren erlauben, sondern auch im Hinblick auf die syntagmatische Realisierung Festlegungen ermöglichen. Die SPL–Makros sind dabei eine — für einen direkt eingebenden Benutzer oder auch ein Anwendungsprogramm — bequeme Ausdrucksmöglichkeit für solche Festlegungen. Zu den SPL–Makros gehören zum Beispiel Möglichkeiten zur Festlegung des Artikels (definit, indefinit oder Null–Artikel) und zur Festlegung der grammatischen Zeit (expandiert dann in eine Konstellation aus Sprechzeit, Ereigniszeit und Referenzzeit, aus der sich dann wiederum die grammatische Zeit ergibt).

In Abschnitt 7.2.2 und im Anhang finden sich Beispiele für SPL–Terme und die zugehörige sprachliche Realisierung.

Andere Nutzungen von NIGEL

NIGEL wurde im Hinblick auf die Generierung entwickelt und implementiert. Kaspar konnte aber zeigen, daß sich in Verbindung mit einem Unifikationsansatz die NIGEL–

Grammatik auch für die Analyse einsetzen läßt (vgl. [Kasper 1988]). Bei einer Kooperation zwischen dem IMS Stuttgart und dem IPSI Darmstadt [Bateman et al. 1992] wurde für ein kleines Fragment von NIGEL eine Darstellung im TFS–System (Typed Feature Structures, [Emele, Zajac 1990]) erstellt. Damit wurden Experimente in bidirektionaler Sprachverarbeitung durchgeführt.

2.3.4 Andere Implementationen systemischer Grammatiken

Die NIGEL–Grammatik ist nach allen vorliegenden Informationen die systemische Grammatik des Englischen mit der größten Abdeckung. Der dafür eingesetzte Aufwand dürfte auch erheblich über 100 Personenjahren liegen. Es gibt aber weitere Implementationen systemischer Grammatiken; dazu gehören für Englisch die von Patten erstellte Implementation [Patten 1988] sowie die Grammatik des COMMUNAL–Projekts [Fawcett 1988], die aktiv weiterentwickelt wird (s.a. 8.3.2).

Im KOMET–Projekt der GMD wird eine deutsche systemische Grammatik entwickelt [Teich 1992]. Hier wurde zunächst auf der Basis von NIGEL statt des 'upper model' die auf Steiner zurückgehende Klassifikation der Prozeßtypen des Deutschen verwendet (vgl. Kap. 3 in [Steiner et al. 1988]). Die aktuellen Arbeiten verwenden ein erweitertes 'upper model' von PENMAN.

Von der Gruppe um Matthiessen in Sydney ist eine multilinguale Variante von PENMAN entwickelt worden, die neben Englisch Fragmente von Japanisch und Chinesisch abdeckt [Bateman et al. 1991]. In dieser Gruppe wird auch an einer systemischen Grammatik für Französisch gearbeitet [Caffarel 1992]; diese Arbeiten sind aber noch im theoretischen Bereich und die Implementierung befindet sich erst in den Anfängen [Caffarel 1993].

2.3.5 Baum–Adjunktions–Grammatiken

Für die „linguistische Komponente“ wird in einer Reihe von Generierungssystemen der Formalismus der Baum–Adjunktions–Grammatiken (engl.: Tree Adjoining Grammars; kurz: TAGs) verwendet oder Techniken, die mit TAGs sehr viele Gemeinsamkeiten haben (z.B. [McDonald et al. 1987], [McDonald, Pustejovsky 1985]).

Formale Aspekte von Baum–Adjunktions–Grammatiken

Im Unterschied zu Phrasenstruktur–Grammatiken sind TAGs keine string–erzeugenden Systeme, sondern Systeme zur Baum–Generierung. Eine TAG besteht aus einem Paar (I, A), wobei I, A endliche Mengen elementarer Bäume sind. Die Menge I enthält dabei *initiale Bäume*, die Menge A die sogenannten *auxiliaren Bäume*. Initiale Bäume haben als Wurzelknoten das Startsymbol S, und die Knoten an den Blättern sind alle terminal. Initiale Bäume stehen sozusagen für die „kleinsten“ vollständigen Satzstrukturen. Auxiliare Bäume haben an der Wurzel ein Non–Terminal X, und die

TAGs: Adjoining oder Adjunktion

Beispiel: (nach [Harbusch 1988])

initialer Baum α:

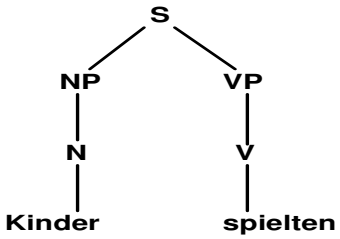

Abbildung 2.5: Ein initialer Baum

auxiliarer Baum β:

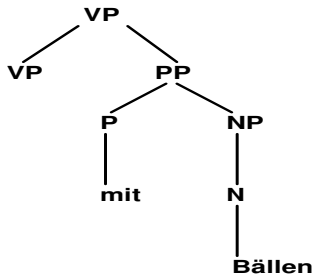

Abbildung 2.6: Ein auxiliarer Baum

Knoten an den Blättern sind alle terminal bis auf einen, den sogenannten „Fußknoten“, der ebenfalls mit X, d.h. der Marke des Wurzelknotens, markiert ist.

Für initiale und auxiliare Bäume gilt, daß sie auch in dem Sinne „minimal“ sind, daß sie keine rekursiven Vorkommen ihrer Non–Terminale enthalten.

Als Kompositionsoperation ist bei TAGs die Adjunktion definiert. Adjunktion erweitert einen beliebigen Baum γ um einen auxiliaren Baum auf folgende Weise: Sei γ ein elementarer oder bereits durch Adjunktion erweiterter Baum, β sei ein auxiliarer Baum mit X als Wurzelknoten. Die Adjunktion von β zu γ im Knoten n (mit Bezeichnung X) ergibt sich durch die folgenden Operationen:

1. Der von n in γ dominierte Unterbaum, hier t genannt, wird entfernt, wobei eine Kopie von n zurückbleibt.

2. Der Hilfsbaum β wird nun im Knoten n angebracht und sein Wurzelknoten mit

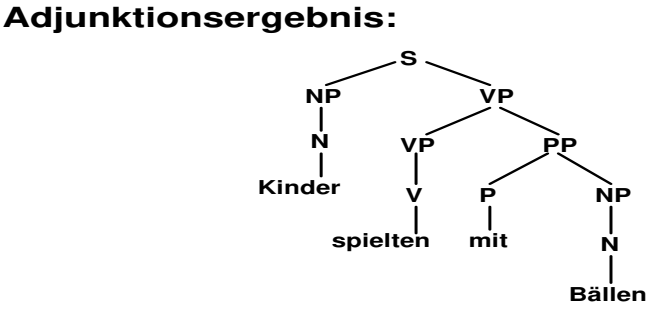

Abbildung 2.7: Das Adjunktionsergebnis

n identifiziert.

3. Der entfernte Unterbaum t wird im Fußknoten von β angebracht, und der Wurzelknoten von t wird mit dem Fußknoten von β identifiziert.

Die Adjunktionsoperation kann bestimmten Bedingungen (constraints) unterworfen werden. So kann für jeden Knoten n in jedem Baum t genau eine der folgenden Arten von Bedingungen für die Adjunktion angegeben werden:

1. Selektive Adjunktion (SA): Nur eine spezifizierte Teilmenge der Menge aller auxiliaren Bäume sind adjungierbar.

2. Null–Adjunktion (NA): Kein auxiliarer Baum ist adjungierbar im Knoten n.

3. Obligatorische Adjunktion (OA): Mindestens einer aus der Menge der in n adjungierbaren auxiliaren Bäume muß adjungiert werden.

Bedeutung von TAGs für die Generierung Die Bedeutung von TAGs für die Generierung wird offensichtlicher, wenn man die folgenden Beziehungen zwischen semantischen oder konzeptuellen Strukturen und den elementaren bzw. auxiliaren Bäumen zugrundelegt [Joshi 1987]:

Jeder elementare Baum sollte in etwa ein Prädikat und die Heads aller Argumente darzustellen erlauben. Initiale Bäume entsprechen in etwa einfachen Sätzen (einfachen Propositionen), auxiliare Bäume entsprechen minimalen rekursiven bzw. iterativen Konstruktionen. Bezüglich der durch sie erweiterten initialen Bäume haben sie die Struktur von Modifikatoren bzw. Komplementen.

Ein Beispiel In [McDonald, Pustejovsky 1985] wird auch die Frage diskutiert, wie mit TAGs Fragen generiert werden können. Illustriert wird dies am Beispielsatz: *'How many ships did Reuter's report that Iraq had said it attacked?'*

Der TAG-Mechanismus wird hier auf folgende Weise verwendet: Ein initialer Baum für den am weitesten eingebetteten Satz und die vorangestellte Wh–Phrase bilden den Ausgangspunkt, d.h. dies würde einem Satz wie *'How many ships did Iraq attack?'* entsprechen. Zu diesem Ausgangsbaum werden dann sukzessive Auxiliarphrasen adjungiert, die sich dann zwischen die Wh–Phrase und den eigentlichen Satz schieben, d.h. es entsteht die Abfolge der Sätze *'How many ships did Iraq say it had attacked?'* und *'How many ships did Reuter's report that Iraq had said it attacked?'*.

Vorteil bei dieser Vorgehensweise ist, daß die Beziehungen zwischen der Wh–Phrase und dem eingebetteten Satz als lokale Beziehung auf einem einzigen einfachen Baum angegeben werden können. Der Mechanismus zwingt daher nicht dazu, irgendwelche Mechanismen für sog. 'Movement' zu bemühen. Joshi unterstreicht, daß solche Lokalitätseigenschaften auch für andere Konstruktionen mit unbeschränkten Abhängigkeiten wünschenswert sind [Joshi 1987].

Diese Sicht auf den Ablauf der Generierung — ein Bottom–Up–Ansatz — hat auch Konsequenzen für die semantischen Strukturen, von denen hier der Realisierungsprozeß ausgeht. Logiker würden gegebenenfalls die Beispielfrage auf folgende Weise repräsentieren:

$$\lambda(\text{quantity-of-ships}) \;.\; \text{report}(\text{Reuters}, \text{say}(\text{Iraq}, \text{attack}(\text{Iraq}, \text{quantity-of-ships})))$$

Dieser Lambda–Ausdruck ist eine mehr oder weniger direkte Umsetzung der Abfolge in der Oberfläche, wird aber dem eben geschilderten kumulativen Prozeß nicht gerecht. Außerdem verlangt er nicht–lokale Beziehungen zwischen den beiden Vorkommen von quantity–of–ships.

Als Alternative wird in [McDonald, Pustejovsky 1985] vorgeschlagen, den logischen Ausdruck in einzelne Einheiten aufzubrechen und zuzulassen, daß diese benannten Einheiten wechselseitige Referenzen enthalten. Dies heißt dann:

$$\begin{aligned} &\text{U1} = \lambda(\text{quantity-of-ships}) \;.\; \text{attack}(\text{Iraq}, \text{quantity-of-ships}) \\ &\text{U2} = \text{say}(\text{Iraq}, \text{U1}) \\ &\text{U3} = \text{report}(\text{Reuters}, \text{U2}) \end{aligned}$$

Bei einer solchen Aufgliederung der semantischen Repräsentation in einzelne logische Terme ergibt sich natürlich die Frage, in welcher Reihenfolge der Realisierungsprozeß ablaufen soll. Ein Prinzip hierfür kann sein, mit den Grundeinheiten zu beginnen, d.h. mit solchen Einheiten, die keine Referenzen auf andere Einheiten enthalten.

TAGs für Deutsch Eine Baum–Adjunktions–Grammatik für Deutsch ist in Saarbrücken in Entwicklung [Harbusch et al. 1991]. Sie wird u.a. im Projekt WIP eingesetzt [Wahlster et al. 1993].

Verbindung von SFG und TAGs Ein System, bei dem versucht wurde, die Techniken der SFG mit Baum-Adjunktions-Grammatiken zu verbinden, wird in [McCoy et al. 1990] vorgestellt.

2.3.6 Andere Ansätze zur Generierung

Auch innerhalb der sogenannten Unifikationsgrammatiken gibt es Arbeiten zur Generierung. Diese werden oft unter der Bezeichnung 'semantic head driven generation' (vgl. [Shieber et al. 1990]) geführt. Das Motiv für diese Arbeiten ist oft weniger eine konkrete Anwendung als die Untersuchung, inwieweit entwickelte Grammatiken „reversibel“ und damit sowohl für Parsing wie Generierung einsetzbar gemacht werden können. Die Strukturen, aus denen dann die Generierung erfolgt, sind mit den Ergebnissen der Analyse identisch, d.h. es handelt sich oft um noch stark syntaktisch „gefärbte“ Tiefenstrukturen. Eine weitere Einschränkung ist häufig, daß es lediglich Einzelsätze sind, die verwendet werden.