Diplomarbeit im Studiengang Wirtschaftsingenieurwesen

Natürlichsprachliche Interaktion mit autonomen 3D-Charakteren

Konzeption und Implementierung
eines virtuellen Darstellers als dialogfähigen Agenten

Jens Piesk
Matr.-Nr. 3771340

Betreuer:

Prof. Dr. L. Nastansky
Lehrstuhl für Wirtschaftsinformatik
Universität-Gesamthochschule Paderborn

Prof. Dr. G. Trogemann
Fächergruppe Kunst- und Medienwissenschaften
Kunsthochschule für Medien Köln

29. Mai 1997

Inhaltsverzeichnis

Kapitel 1

Einleitung

Durch *Virtual Reality* und *Multimedia* wird der Kommunikationskanal zwischen Mensch und Maschine *multimodal*. Die Einsatzmöglichkeiten gehen dabei weit über die bekannten Paradigmen der Gestaltung von Benutzerschnittstellen hinaus. Derzeitige graphische Benutzeroberflächen reduzieren die Möglichkeiten der visuellen Kommunikation auf den Aspekt der Bildsprache. Graphisches Design und ikonenhafte Darstellungen von Objekten aus dem Lebensumfeld des Benutzers leisten einen großen Beitrag zur intuitiven Benutzerführung. Die Arbeit des Schnittstellengestalters ist ein entscheidender Faktor für die Güte dieser Intuitivität.

Virtuelle Akteure bieten die Möglichkeit, textuelle Inhalte - neben der auditiven Wiedergabe - visuell darzustellen. Gestik und Mimik ergänzen die verbale Sprache. Die Gesamtdarstellung wird einprägsamer, und die anthropomorphe Gestalt ermöglicht eine emotionale Beziehung des Benutzers zum 3D-Charakter.

Multimodale Dialogschnittstellen sind hauptsächlich aufgrund der folgenden drei Interessen Gegenstand aktueller Forschung auf dem Gebiet der Mensch-Maschine-Interaktion (vgl. [NT94b]):

- Verbindung von *direkter Manipulation* und *natürlicher Sprache*: Die fehlende Information in natürlichsprachlichen Ausdrücken (z.B. "Lösche dieses Objekt dort!") kann durch zeigende Gesten zum Zeitpunkt der Äußerung ergänzt werden. Das Gesagte wird dadurch eindeutig.

- Nutzung nonverbaler Information zur *Spezifizierung des Kontextes*: Durch Analyse nonverbalen Verhaltens können in natürlichsprachlichen Ausdrücken enthaltene Mehrdeutigkeiten entschlüsselt werden. Der Blick auf ein Auto läßt darauf schließen, daß das Gesagte im Kontext des Autos zu interpretieren ist.

- Einbeziehung *menschenähnlichen Verhaltens* in Dialogsysteme: Dieses Forschungsgebiet betrifft den Feedbackkanal. Nonverbales Verhalten wird in die Antwort der Computers auf Benutzereingaben einbezogen. Gestik und Mimik eines anthropomorphen Interfaces geben Auskunft über den Status des Systems. So kann "Wegschauen" z.B. symbolisieren, daß das System zur Zeit nicht empfänglich für neue Eingaben ist.

Die vorliegende Arbeit ist in den dritten Bereich einzuordnen.

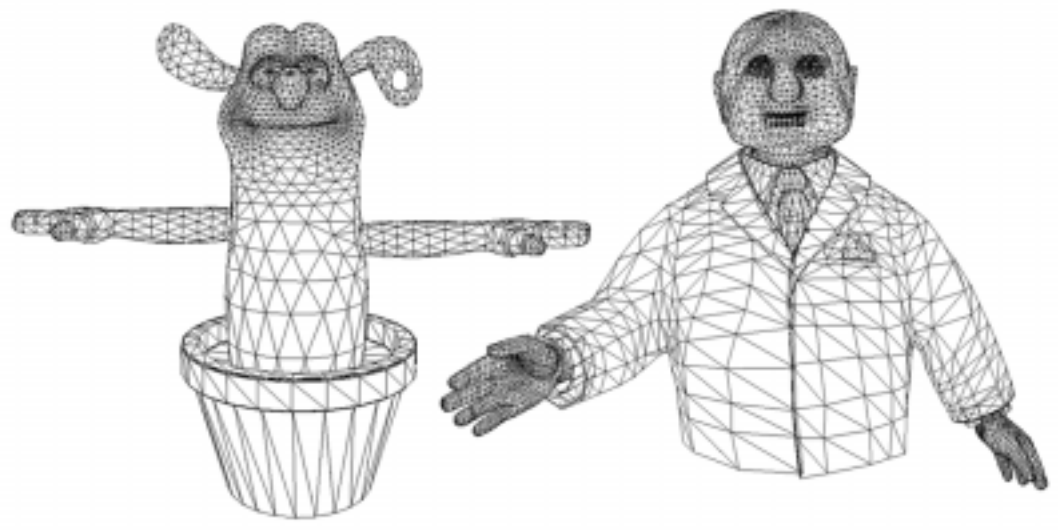


Abbildung 1.1: Dreiecksnetze von 3D-Charakteren

1.1 3D-Charakteranimation

An der Kunsthochschule für Medien Köln wird folgendes Verfahren zur Produktion und Animation von 3D-Charakteren verwendet:

Aus einem realen Modell einer Figur (z.B. aus Ton) wird mit einem 3D-Laserscanner ein Polygonmodell erzeugt. Der Laserscanner tastet mit einem Rotationsarm in 512 Winkelinkrementen und vertikal in 512 Ebenen die reale Figur ab. Für jeden dieser 512x512 Datenpunkte werden die Raum- (Entfernung zum Scandetektor) und Farbinformationen (RGB-Farbwert) aufgezeichnet. 3D-Koordinaten und Farbwerte werden getrennt verwaltet, so daß das 3D-Modell leicht mit neuen Texturen belegt werden kann, was z.B. im Fall eines Gipsmodells sinnvoll ist.

Aus den 3D-Koordinaten der Datenpunkte des Rotationsscanners wird ein für die Animation optimiertes Polygonmodell (Dreiecksnetz, Abbildung 1.1) erstellt [Bun96] und mit einer Textur versehen. Dieses statische 3D-Modell des virtuellen Akteurs ist Ausgangspunkt der Animation und stellt die Grundstellung der Figur dar. Mit einem Modellierungstool werden Basisausdrücke festgelegt. Ein *Basisausdruck* besteht aus einer Gruppe benachbarter Punkte des Polygonnetzes, für die von einem Animator in sechs Freiheitsgraden (drei translatorischen und drei rotatorischen) Abweichungen von der Ursprungsposition definiert werden. Die Ursprungsposition bezeichnet hierbei die jeweiligen 3D-Koordinaten des statischen 3D-Modells. Aus einem Basisausdruck entsteht durch Skalierung eine 3D-Trajektorie, die sich aus gleichzeitig stattfindender Translation entlang der Rotation um die drei Achsen des dreidimensionalen karthesischen Koordinatensystems zusammensetzt. Die 3D-Trajektorie beschreibt eine *Basisbewegung* von der Grundstellung zum Basisausdruck. Typische Basisbewegungen sind “linken Arm heben”, “Mund öffnen”, “rechte Augenbraue heben” etc.

Mit einem Animationstool wird dann für jede Basisbewegung eine Animationskurve erstellt, welche die Auslenkung des jeweiligen Basisausdrucks der Zeit zuordnet. Die Zeit wird in Bildern, sogenannten *Frames*, gemessen. Das Animationstool

Abbildung 1.2: Zwei 3D-Charaktere, die für die Echtzeitanimation konfigurierbar sind: ein sprechender Kaktus und ein sprechender Filzpantoffel

arbeitet mit der bei Videokameras üblichen Bildrate von 25 Bildern pro Sekunde. Ein Frame wird somit 40 Millisekunden lang dargestellt. Für komplexe Figuren oder für die Echtzeitanimation besteht die Möglichkeit, mehrere Basisausdrücke zu einem *Bewegungseffektor* zusammenzufassen, dessen Animationskurve dann die Auslenkung der zugeordneten Basisausdrücke steuert.

Abbildung 1.2 zeigt 3D-Charaktere, die mit diesem Verfahren entwickelt wurden. Das 3D-Polygonmodell des sprechenden Kaktus besteht aus ca. 4000 Polygonen (Dreiecken), die von 178 Basisausdrücken gesteuert werden. 36 Bewegungseffektoren steuern die Auslenkung dieser Basisausdrücke. Das 3D-Polygonmodell des Filzpantoffels besteht aus ca. 1800 Polygonen. Die Datenpunkte werden von 32 Basisausdrücken und 10 Bewegungseffektoren gesteuert.

Der Begriff *Echtzeitanimation* wird für ein Szenario verwendet, in dem Animationskurven nicht vorproduziert, sondern während der Darbietung generiert werden. Hierzu werden Datenhandschuhe und ein Motion-Capture-System verwendet, deren Sensoren reale Finger- und Körperbewegungen registrieren. Diese realen Auslenkungskurven werden auf die Animationskurven der für diese Konfiguration erstellten Bewegungseffektoren abgebildet. Die Animationssoftware übersetzt bei dieser Methode die zeitliche Änderung der Raumkoordinaten der einzelnen Sensoren - sowohl des Motion-Capture-Systems als auch der beiden Datenhandschuhe - auf die

Bewegungseffektoren, die zuvor von einem Animator für die jeweilige Figur angelegt wurden.

Der in Abbildung 1.2 dargestellte Kaktus ist als virtueller Moderator für das Jugendprogramm des WDR konzipiert. Dort soll er 'live' auftreten und unter anderem auf während der Sendung telefonisch gestellte Zuschauerfragen antworten. Gesteuert wird die Figur dabei von zwei Puppenspielern. Einer von beiden ist gleichzeitig der Sprecher. Dieser steuert mittels zwei Datenhandschuhen die Mundbewegungen und Mimik, der andere über ein Motion-Capture-System mit seinen Körperbewegungen die Gestik.

1.2 Bedeutung der Arbeit

Neue Technologien im Medienbereich ermöglichen neue Sende- und Marketingkonzepte. Das Buch zum Film gibt es schon, auch werden seit Bestehen der Film- und Fernsehindustrie Bücher verfilmt. Die Printmedien berichten über das Fernsehen und umgekehrt. Neue Medien erweitern dieses Spektrum nicht nur um neue Datenträger und interaktive Anwendungen, wie die CD-ROM oder das Internet, sondern integrieren auch bestehende Medien. Das digitale Fernsehen wird in Verbindung mit dem Internet in Zukunft an Bedeutung gewinnen. Mittelfristig ist die Vermarktung von virtuellen 3D-Charakteren im Mediendreieck Fernsehen - CD-ROM - Internet anzusiedeln. Langfristig werden sich synthetischen Darstellern in einer Welt mit zunehmend interaktiven und individualisierten Medien immer größere Einsatzpotentiale bieten. Ziel dieser Arbeit ist es, die technologischen Konzepte und Grundlagen für autonome 3D-Charaktere zusammenzustellen, sie für einen dialogfähigen Geschichtenerzähler weiterzuentwickeln und in einem Prototypsystem zu implementieren.

1.3 Aufgabenstellung

Ausgehend von einem vorhandenen echtzeitfähigen 3D-Charakter soll eine Konzeption für einen freien natürlichsprachlichen Dialog mit dieser Figur erarbeitet werden. Die Interaktion soll sich nicht auf eine natürlichsprachliche Steuerung beschränken. Die Figur soll also nicht nur Kommandos des Benutzers verstehen und ausführen, der Benutzer soll in der Figur vielmehr einen Gesprächspartner sehen. Der Dialog soll dem Charakter der Figur entsprechend unterhaltend sein. Weiterhin soll die Konzeption zu Experimentierzwecken in einem Prototypen implementiert werden.

1.4 Gliederung

Kapitel 2 analysiert das Interaktionsszenario zwischen einem virtuellen Akteur und seinem menschlichen Gesprächspartner. Ausgehend von einem langfristig zu realisierendem Zielszenario wird das Interaktionszenario der vorliegenden Arbeit definiert. Kapitel 3 beschreibt die linguistischen Grundlagen und Prinzipien der Computerlinguistik. Hier wird aufgezeigt, welche linguistischen Verarbeitungseinheiten

für die maschinelle Verarbeitung zur Verfügung stehen. In Kapitel 4 werden Prinzipien der hypertextuellen Strukturierung fiktionaler Texte und des nichtlinearen Geschichtenerzählens beschrieben. Kapitel 5 stellt die Problematik der Erzeugung sprachsynchronen nonverbalen Verhaltens dar. Die in den Kapiteln 3, 4 und 5 dargestellten Forschungsgebiete werden im Grundgedanken aufgeführt. Die Prinzipien, die für die in dieser Arbeit vorgestellte Konzeption relevant sind, werden in den einzelnen Kapiteln anhand von Beispielen und deren Systembeschreibungen näher erläutert. Kapitel 6 stellt die Konzeption des im Rahmen der vorliegenden Arbeit entwickelten Prototypsystems VISTA (Virtual Storytelling Actor) vor. Die Implementierung des VISTA-Systems wird in Kapitel 7 erläutert. Kapitel 8 stellt die während der Entwicklungsarbeit gewonnenen Erkenntnisse dar, bewertet das entstandene System und zeigt Tätigkeitsfelder für zukünftige Entwicklungen. Kapitel 9 ist eine Zusammenfassung der Arbeit.

Kapitel 2

Problemanalyse

Abschnitt 2.1 stellt das Zielszenario einer Interaktion zwischen Mensch/Realität und Maschine/Virtualiät dar und unterteilt den Interaktionsablauf in einzelne Phasen. Auf dieser Grundlage wird in Abschnitt 2.2 die Phasen auf der Seite der Virtualität unter dem Inhalt-Form Aspekt betrachtet. Die Unterschiede zwischen der verbalen, vokalen, nonverbalen und nonvokalen Kommunikationsform sind Gegenstand des Abschnitts 2.3. Abschnitt 2.4 gibt einen Überblick über die Ausdrucksformen eines synthetischen Darstellers. Der Bereich der nonverbal-vokalen Sprache wird kurz angesprochen, aber im Rahmen der vorliegenden Arbeit nicht vertieft.

2.1 Interaktion zwischen Realität und Virtualität

Es ist ein Szenario denkbar, in dem die Computerfigur als Eingabedaten neben der verbalen Sprache die optisch wahrnehmbaren Signale des menschliche Gesprächspartners in die Gestaltung der Interaktion einbezieht. Eine solche Apparatur wäre mit akustischen und optischen Sensoren ausgestattet. Ein System aus mindestens einer Kamera würde Gestik, Mimik und Körperhaltung des Benutzers registrieren. Diese Informationen würden dann zusammen mit dem über mindestens ein Mikrofon transformierten Sprachsignal interpretiert werden. Die Reaktion würde über Lautsprecher und Display in verbaler und nonverbaler Sprache, also durch audible Sprache und darauf abgestimmte Gesichts- und Körperbewegungen, dargestellt. Ein solches Interaktionsszenario kann in sechs Phasen unterteilt werden (Abbildung 2.1):

(1) Sensoren registrieren die Handlungen des Menschen.

(2) Der Computer erzeugt eine passende Reaktion, die dem Benutzer über

(3) Effektoren dargeboten wird.

(4) Der Mensch nimmt diese Darstellung über seine Sinne wahr,

(5) reagiert auf die Darstellung mit einer

(6) Handlung, die von den Sensoren des virtuellen Akteurs registriert wird.

Diese Arbeit beschäftigt sich mit den Phasen 1, 2 und 3.

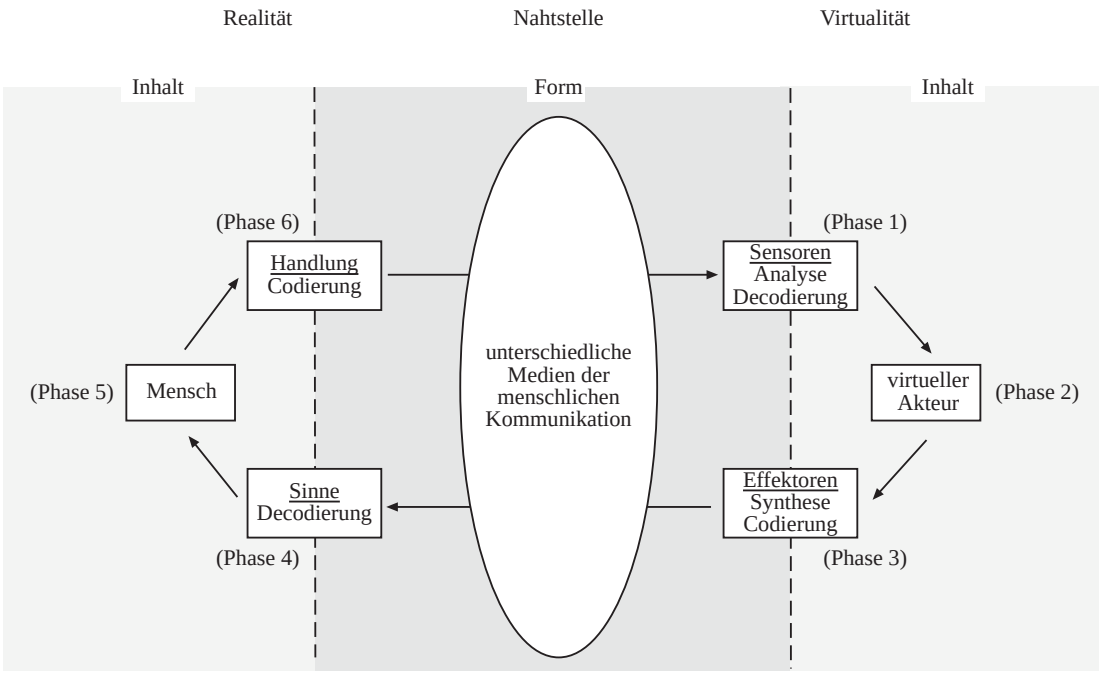


Abbildung 2.1: Phasen des Interaktionszenarios

2.2 Inhalt - Form

Die Phasen 1, 2 und 3 aus Abbildung 2.1 sind in Abbildung 2.2 noch einmal unter den Aspekten Form, Inhalt, Codierung und Decodierung dargestellt: Die Äußerung des Benutzers wird von der Maschine sensorisch wahrgenommen und analysiert (Phase 1). Die decodierten Informationen der verbalen und nonverbalen Sprache determinieren die inhaltliche Reaktion (Phase 2), die in verbale und nonverbale Sprache codiert und dem menschlichen Gesprächspartner dargeboten wird (Phase 3).

2.3 Verbal, vokal, nonverbal, nonvokal

Die Dichotomie *verbal-nonverbal* klassifiziert Reize und Zeichen unabhängig von der Ausdrucksform. Sprachzeichen können *akustisch* (vokal) und *visuell* (nonvokal) ausgedrückt werden. Aus diesen beiden Dichotomien ergeben sich die vier Modalitäten der verbal-vokalen, nonverbal-vokalen, verbal-nonvokalen und nonverbal-nonvokalen Kommunikationsformen (Abbildung 2.3).

In dieser Arbeit wird die *nonverbale* auf die *nonverbal-nonvokale Kommunikation* begrenzt. Nonverbale akustische Ausdrucksformen werden nicht näher analysiert und formalisiert.

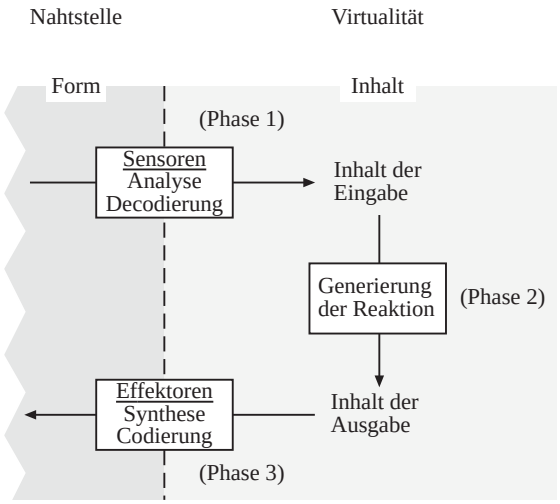


Abbildung 2.2: Interaktionsphasen

Codierung / Ausdruck	verbal	nonverbal
vokal	gesprochene Sprache	Paralinguistik, nichtsprachliche Lautzeichen
nonvokal	Schrift, Sprachsubstitute, Zeichensprache	visuelle Kommunikation, Gestik, Mimik, Körperhaltung, taktile Kommunikation

Abbildung 2.3: Klassifizierung von Kommunikationsformen nach Codierung und Ausdrucksform der Sprachzeichen

Definition des Interaktionsszenarios

Phase 1 wird auf Tastatureingaben beschränkt. Die sensorischen Informationen, die das System von der Äußerung des Benutzers zur Verfügung hat, ist die textuelle (verbal-nonvokale) Eingabe. Die vorliegende Arbeit hat zwei Schwerpunkte:

(a) Die Generierung einer zur Äusserung des Benutzers passenden Antwort. (Inhalt - Phase 2)

(b) Die auditive und visuelle Darbietung dieser Reaktion in verbaler und nonverbaler Sprache. (Form - Phase 3)

Der 3D-Charakter wird seinem menschlichen Gegenüber visuell und auditiv dargestellt. Diese Darstellungsform ist vom Inhalt zu trennen. Der Inhalt meint das, *was* gesagt wird. Die Form ist die Darbietung des Inhalts, seine Inszenierung.

2.4 Ausdrucksformen virtueller Akteure

Der Begriff *Charakter* bezeichnet die Gesamtheit der charakteristischen Eigenschaften eines Wesens. Im allgemeinen Sprachgebrauch *hat* eine Person einen Charak-

ter. Im Film wird ein bestimmter Charakter durch einen Schauspieler *dargestellt*. Ein Wesen besitzt für seine Charakteristika unterschiedliche Ausdrucksweisen. In einer Kommunikationssituation sind die Ausdrucksweisen, die für den Kommunikationspartner durch seine Sinne wahrnehmbar sind, relevant. Ein 3D-Charakter kann durch

- das äußere Erscheinungsbild,
- die Gesichts- und Körperbewegungen (Mimik, Gestik und Körperhaltung),
- die Stimme,
- die sprachliche Ausdrucksweise und
- den Inhalt der Äußerungen

dargestellt werden. Eine autonomer 3D-Charakter kann nur glaubwürdig erscheinen, wenn die audiovisuelle Darstellung und der Inhalt aufeinander abgestimmt sind. Beides muß sich sinnvoll in den Kontext des Gespräches einfügen. Die Qualität der Übereinstimmung von Sprache, Gestik, Mimik und Körperhaltung wird mit den Kriterien der menschlichen Wahrnehmung bewertet.

Nonverbal-vokale Sprache

Die menschliche Stimme enthält neben verbalen Komponenten zusätzliche Informationen wie Geschwindigkeit, Tonhöhe etc. Diese nicht verbalen Informationen sind einerseits charakteristisch für den Sprecher (Tonlage, Lautstärke, Sprechgeschwindigkeit), andererseits werden sie von situationsbedingten Faktoren und dem Redeinhalt beeinflußt. Nonverbale Komponenten der Stimme enthalten Informationen über die Gefühls- oder Gemütslage des Sprechers. Die Codierung erfolgt hier in gröberen Zeitabschnitten: Ist der Sprecher glücklich über einen bestimmten Sachverhalt, so wirkt sich das meist auf mehrere Sätze oder ganze Redeabschnitte aus. Die Codierung dieser Information erfolgt nach semantischen oder pragmatischen Regeln.

Auch die Satzmelodie wird von den semantischen und pragmatischen Faktoren der Gesprächssituation beeinflußt. Sie ist jedoch hauptsächlich von der Syntax einer Sprache bestimmt. Es existieren z.B. Betonungsregeln für einzelne Worte oder Silben. So wird z.B. im Spanischen im allgemeinen die vorletzte Silbe eines Wortes betont. Solche Ausspracheregeln beeinflußen das Sprachverstehen. Dies wird z.B. bei den Kommunikationsschwierigkeiten, die durch regional stark ausgeprägten Akzente und Dialekte einer Sprache entstehen, deutlich.

Kapitel 3

Verbale Mensch-Maschine-Kommunikation

Natürliche Sprachen (Deutsch, Englisch etc.) dienen der zwischenmenschlichen Kommunikation. *Formale Sprachen* (Programmiersprachen, musikalische oder mathematische Notationen etc.), sogenannte *Formalismen*, wurden für spezifische Anwendungsgebiete definiert. Eines der Hauptziele der linguistischen Datenverarbeitung ist die Konstruktion künstlicher Systeme, die in der Lage sind, mit dem Benutzer natürlichsprachlich zu kommunizieren. Hierzu müssen die menschlichen Fähigkeiten zur Produktion und Rezeption von Sprache formalisiert werden [LW86]. Das Phänomen "natürliche Sprache" besitzt einige wichtige Eigenschaften, die ihre maschinelle Verarbeitung erschwert [Bin91]:

1. *Fehlen von expliziten Definitionen*:
   Keine existierende natürliche Sprache ist vollständig und exakt definiert, obwohl Sprachwissenschaftler unaufhörlich dieses Ziel verfolgen. Natürliche Sprachen entwickeln sich in ihrer Anwendung stetig fort.

2. *Nichteinhaltung von Regeln*:
   Im allgemeinen Sprachgebrauch werden die Regeln der korrekten Sprachstruktur nicht immer eingehalten: Es treten häufig Fehler in Schreibweise und Syntax auf; letztere vor allem beim Sprechen, wo unvollständige und schlecht strukturierte Sätze gebraucht werden - oft ohne das Verständnis zu beeinflussen.

3. *Kontextabhängigkeit*:
   Phänomene wie Anaphern, Ellipsen und Speech Acts können nur unter Einbezug des Satzkontextes gelöst werden. Der Kontext ist durch den vorhergehenden Satz, die Entstehungsgeschichte und in manchen Fällen durch die Pläne und Intentionen der Gesprächspartner bestimmt. Nur in wenigen Fällen kann die Bedeutung eines Satzes kontextunabhängig bestimmt werden.

4. *Mehrdeutigkeiten*:
   Worte haben viele unterschiedliche Bedeutungen, Sätze unterschiedliche

Interpretationen - auf syntaktischer oder semantischer Ebene. Natürlichsprachliche Ausdrücke sind dadurch zwei oder mehrdeutig. Nicht alle dieser Mehrdeutigkeiten können im Kontext gelöst werden.

Dieses Kapitel behandelt *natürliche Sprache* auf textuell-verbaler Ebene. Text ist verschriftliche Sprache, eine Transkription der zeitlichen Abfolge von Lauten als geordnete Reihe von Buchstaben. Die Linguistik nutzt diese Form der Zeitreihen-Notation zu Analysezwecken. Die Ergebnisse dieser Analysen sind in umfangreichen Regelwerken (Grammatiken, Lexika etc.) dokumentiert. Ziel der Computerlinguistik ist die Nutzung der bekannten Regeln zur maschinellen Verarbeitung natürlicher Sprache.

3.1 Linguistische Grundlagen

Dieser Abschnitt gibt einen Überblick über die Methoden und Erkenntnisse der Linguistik und Eräutert in diesem Zusammenhang insbesondere die Begriffe *Syntax*, *Semantik* und *Pragmatik*.

3.1.1 Basiseinheiten der Sprachwissenschaft

Die kleinste bedeutungsunterscheidende Einheit einer Sprache ist das Phonem. Die kleinste Einheit der Schriftsprache ist das Graphem, der Buchstabe. Das Morphem ist die kleinste bedeutungstragende Einheit. Das Lexem - oder Wort - ist die kleinste begriffliche Einheit. Die Tabelle 3.1 gibt hierfür einige Beispiele. Die komplexe-

Linguistische Einheit	**Beschreibung**	**Beispiele**
Phonem	kleinste bedeutungsunterscheidende Einheit	siehe Abschnitt 7.2.6
Graphem/ Buchstabe	kleinste Einheit der Schriftsprache	a, b, c, ...
Morphem	kleinste bedeutungstragende Einheit	Wortstämme, Pre- oder Postfixe:"end" weist auf die Verlaufsform von Verben hin (laufend, sitzend etc.); "ung" bedeutet die Substantivierung (Unternehmung, Verschönerung etc.)
Lexem/ Wort	kleinste begriffliche Einheit	Adjektive (schön, hell), Substantive (Charakter, Persönlichkeit)

Tabelle 3.1: Sprachwissenschaftliche Basiseinheiten

ren Konstrukte einer Sprache setzen sich aus diesen Basiseinheiten zusammen. So werden z.B. Wörter zu Sätzen kombiniert und Sätze zu Texten. Das syntaktische Regelwerk einer Sprache enthält die Vorschriften, nach denen diese Kompositionen stattfinden. Die für die analytische Betrachtung des Sprachverstehens und der menschlichen Kognition wichtige Unterscheidung in Syntaktik, Semantik und Pragmatik entstammt der Semiotik, der allgemeinen Lehre von den Zeichen und Sprachen. Diese Dreiteilung wurde von [Mor38] und nach ihm von [Car42] vorgeschlagen.

3.1.2 Syntaktik

Syntax ist der Begriff für die impliziten Regeln einer natürlichen Sprache, nach denen einfache Sprachelemente (z.B. Morpheme, Wörter) zu komplexeren Konstrukten (z.B. Satz) kombiniert werden. Syntaktische Regeln werden in Grammatiken zusammengestellt.

Formale Sprachtheorien

Die theoretische Linguistik liefert mit formalen Sprachtheorien Regelsysteme für die natürlichsprachliche Kommunikation. Die Theorie der generativen Transformationsgrammatiken von [Cho57] ist eine der einflußreichsten Sprachtheorien und wird hier exemplarisch dargestellt. Sie beschreibt die Syntax einer Sprache anhand einer Anzahl von Regeln. Die folgenden Regeln zeigen eine vereinfachte Version der 1957 aufgestellten generativen Grammatiken:

(I)	S(sentence)	$->$ NP(noun phrase) + VP(verb phrase)
(II)	NP	$->$ N(noun)
(III)	NP	$->$ article + N
(IV)	NP	$->$ adjective + N
(V)	NP	$->$ pronoun
(VI)	VP	$->$ V(verb) + NP
(VII)	VP	$->$ V + adjective
(VIII)	N	$->$ *Jane, boy, girl, apples*
(IX)	V	$->$ *likes, hit, was hit, was, are cooking, are*
(X)	adjective	$->$ *good, unfortunate, cooking*
(XI)	article	$->$ *a, the*
(XII)	pronoun	$->$ *he, she, they*

Mit diesen *Satzstrukturregeln* (phrase structure rules) kann die syntaktische Struktur eines Satzes bestimmt werden, welche häufig in Form eines *syntaktischen Baumes* dargestellt wird (Abbildung 3.1). Generative Grammatiken sind kontextfreie Grammatiken, weil die Regeln für jede Satzkonstruktion der zugrundeliegenden Sprache gelten.

Transformationsregeln

In einer späteren Version seiner Theorie unterscheidet [Cho65] zwischen *Oberflächen-* (surface structure) und *Tiefenstruktur* (deep structure) einer Äußerung (Abb. 3.2). Letztere enthält die zur Interpretation des Satzes notwendigen Informationen. Die Oberflächenstruktur definiert die korrekte Reihenfolge der Worte in einem Satz. *Transformationsregeln* (transformational rules) bilden die Oberflächenstrukturen bidirektional auf die Tiefenstrukturen ab. So basiert beispielsweise die Passivkonstruktion in Abbildung 3.1 auf derselben Tiefenstruktur wie die aktive Formulierung. Die Tiefenstruktur repräsentiert die Bedeutung des Satzes, sein gedankliches Konzept. Zu jedem gedanklichen Konzept existieren nach diesem Modell mehrere äquivalente Formulierungen, die mittels der Transformationsregeln aus der Tiefenstruktur erzeugt werden können.

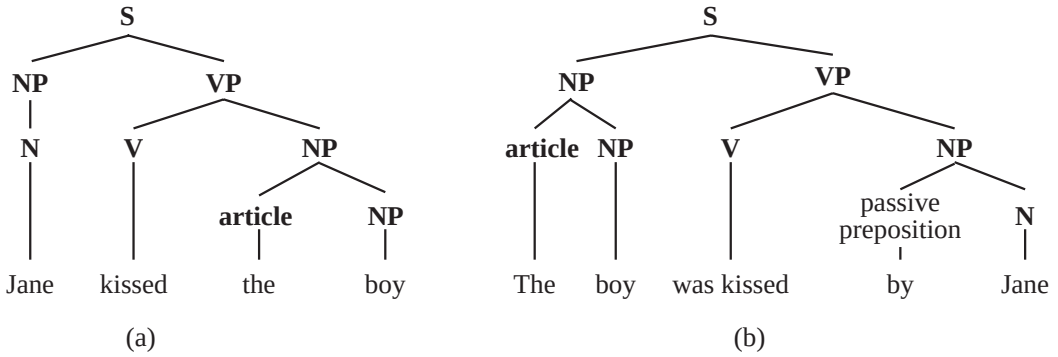


Abbildung 3.1: Baumstruktur eines Beispielsatzes

Beispielsatz "Jane kissed the boy"(a) und dessen Passivkonstruktion "The boy was kissed by Jane"(b)

Quelle: [Gre86], S. 68

Syntax alleine genügt jedoch nicht. Der Satz "Der zornige Bildschirm schmeckt, was sie verpaßt zu erinnern." ist syntaktisch korrekt, ergibt jedoch keinen Sinn.

3.1.3 Semantik

Für jeden sinnvollen Satz existiert neben der syntaktische Konstruktionsregel eine entsprechende semantische Vorschrift, die die Bedeutung der Satzbestandteile sinnvoll zu einer Bedeutung des Satzes kombiniert.

Der Begriff *Semantik* wird unterschiedlich aufgefaßt. In dieser Arbeit wird unter Semantik die Bedeutung eines Ausdruckes im Sinne einer lexikalischen Definition verstanden. Bei der Erstellung semantischer Definitionen und Beschreibungen sprachlicher Konstrukte treten folgende Probleme auf:

- Jede Ebene der in Abschnitt 3.1 beschriebenen linguistischen Objekthierarchie hat ihre eigenen semantischen Gesetzmäßigkeiten.

  > "Different kinds of linguistic object have a meaning. Morphemes, words, phrases, sentences, utterances and stories can each be said to have a meaning but it is not necessarily the same thing in each case. The answer to the question *What is the meaning of the morpheme 'ing'?*, for example, might be a different kind of thing from the answer to the question *What is the meaning of the story you have just read?*"
  > ([BLH91], S.115)

- Natürliche Sprachen enthalten Konstrukte mit mehr als einer Bedeutung (z.B. die Worte *legen, ziehen, gehen*).

- Die unterschiedlichen Bedeutungen eines sprachlichen Konstruktes lassen sich oft nicht klar abgrenzen: *zur Schule gehen, zum Sport gehen, zur Vorlesung gehen, es sich gut gehen lassen*

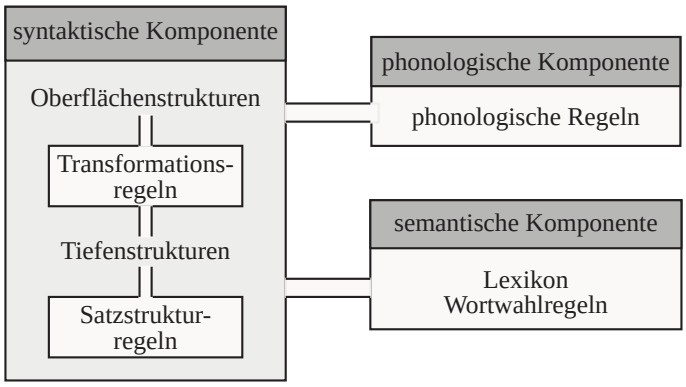


Abbildung 3.2: Chomskys Sprachtheorie

Quelle: [Gre86], S. 71

- Mehrere unterschiedliche natürlichsprachliche Konstrukte können die gleiche Semantik enthalten.

- Semantische Definitionen von sprachlichen Konstrukten enthalten selbst Konstrukte der betreffende Sprache (Worte werden mit Worten erklärt).

3.1.4 Pragmatik

Ein semantischer Ansatz bestimmt die Bedeutung eines Satzes reduktionistisch aus der Bedeutung der Satzteile und Worte. Umfassendes Verständnis natürlicher Sprache geht aber über die semantische Bedeutung einzelner Wörter oder Wortgruppen hinaus. Der Kontext, in dem ein Satz eingebettet ist, beeinflußt entscheidend seine korrekte Interpretation. Ein Satz wird von einem bestimmten Sprecher, in einer bestimmten Situation, zu einem bestimmten Zweck gesagt und bezieht sich - implizit - auf ein bestimmtes Hintergrundwissen. Die *Pragmatik* untersucht den Einfluß des Kontextes auf die korrekte Interpretation eines Satzes.

Die Pragmatik einer Texteinheit kann sich sowohl auf einzelne Worte als auch auf komplexere Ausdrücke oder ganze Textabschnitte auswirken. Auf der Wortebene bezieht sich z.B. “ich” auf den Sprecher und “jetzt” auf den Augenblick der Äußerung. In Abgrenzung zur semantischen Bedeutung von “ich” (z.B. ich = derjenige, der den Satz äußert) füllt die Pragmatik die abstrakte Beschreibung mit einer konkreten Person aus dem Kontext - dem Sprecher.

Die semantische Bedeutung von Wörtern und Sätzen kann im Sinn der semantischen Übereinkunft (Abschnitt 3.1.3) lexikalisch definiert werden. Dies ist im Fall der pragmatischen Bedeutung nicht möglich. Das Geäußerte hat für jeden eine individuelle Bedeutung. Sie ist für zwei Personen in der gleichen Situation und für eine Person in unterschiedlichen Situation anders. Da mentale Zustände nicht beobachtbar sind, kann diese pragmatische Bedeutung nur anhand des Verhaltens bestimmt werden. Der pragmatische Aspekt der Kommunikation ist folglich die Beeinflußung ihrer Teilnehmer ([WBJ96], S.22).

Eine pragmatische Analyse und Sichtweise verlangt nach größeren sprachlichen Verarbeitungseinheiten. Wurden auf der semantischen und syntaktischen Ebene Lexeme und Morpheme definiert und zueinander in Beziehung gesetzt, so befindet sich die Pragmatik auf der Ebene komplexer gedanklicher Konzepte. Diese sind nur durch komplexere sprachliche Elemente wie Sätze, Satzfolgen oder größere Texteinheiten darstellbar.

3.2 Strukturen höherer Verarbeitungsebenen

In den bisherigen Abschnitten wurde die Analyse und Repräsentationsformen isolierter Sätze betrachtet. In Abschnitt 3.1.4 ist gezeigt worden, daß zur korrekten Interpretation pragmatische Informationen des Kontextes nötig sind. Diese Analyseebene betrachtet nicht mehr das Wort, sondern den Satz als Basiseinheit. Gesucht sind Gesetzmäßigkeiten, die der Kombination von Sätzen zu Dialogen, Texten oder Erzählungen zugrundeliegen. Diesbezügliche Theorien und Forschungsgebiete können in zwei Gruppen aufgeteilt werden: Die *lokale Kohärenz* beschäftigt sich mit inhaltlichen Verweisen und Zusammenhängen zwischen benachbarten Sätzen, Vertreter der zweiten Gruppe mit der *globalen Struktur* von Texten.

3.2.1 Lokale Kohärenz

Ein Hauptaspekt der lokalen Kohärenz benachbarter Sätze ist die Referenzierung von Pronomen (er, sie etc.) mit inhaltlichen Informationen benachbarter Sätze. Dies soll hier anhand eines Textausschnittes aus der Geschichte "Alice im Wunderland" [Car89] verdeutlicht werden. Der Textausschnitt wurde zu diesem Zweck leicht modifiziert:

> Alice saß neben ihrer großen Schwester im Gras und langweilte sich. Ein paarmal hatte **sie** in ein Buch geschaut.
> (Modifikation A)

Das Wort "**sie**" kann sich auf "Alice" oder "die Schwester" beziehen. Der Textausschnitt bietet nicht genügend kontextuelle Informationen zur Auflösung dieser Doppeldeutigkeit. Bei der Interpretation der Sätze

> "Alice saß neben ihrer großen Schwester im Gras und langweilte sich. Ein paarmal hatte **sie** in ein Buch geschaut, das ihre Schwester las."
> (Originaltextausschnitt, [Car89])

wird die Situation klarer. In den Sätzen

> Alice saß neben ihrer großen Schwester im Gras und langweilte sich. Ein paarmal hatte **sie** in ein Buch, das Alice ihr geschenkt hatte, aber die meiste Zeit lag sie einfach da und sonnte sich. Sie hatte keine Lust, mit Alice zu spielen.
> (Modifikation B)

wird Alice zur Beobachterin. Das "**sie**" des zweiten Satzes bezieht sich auf ihre Schwester.

3.2.2 Dialoggrammatiken und die Sprechakt-Theorie

Die *Sprechakt-Theorie* ordnet jede mögliche Äußerung einer der folgenden Kategorie zu: repräsentativ, deklarativ, expressiv, direktiv und kommissiv. "Die Sonne scheint!" ist eine repräsentative Äußerung, eine Behauptung. Sie unterscheidet sich von der deklarativen ("Sie sind hiermit gefeuert!") insofern, als erst durch den Sprechakt selbst der ausgedrückte Sachverhalt eintritt. Expressive Sprechakte drücken eine psychische Einstellung aus ("Es freut mich, daß sie gekommen sind!"). Mit einer direktiven Äußerung möchte der Sprecher seinen Gegenüber zu etwas veranlassen ("Wie spät ist es?", "Bitte geben sie mir ein Vollkornbrot!"). Aus dieser Klassifizierung können Gesetzmäßigkeiten von aufeinanderfolgenden Äußerungen eines Dialoges aufgestellt werden. Sogenannte *Adjazenzpaare* reflektieren Heuristiken wie "Auf eine *Frage* folgt normalerweise eine *Antwort*." oder "Auf einen *Vorschlag* folgt entweder eine *Annahme* oder eine *Ablehnung* (negative Annahme) des Vorschlags." Vergleichbar mit den syntaktischen Satzstrukturregeln der generativen Transformationsgrammatiken (Abschnitt 3.1.2), die die grammatikalische Korrektheit eines Satzes bestimmen, entscheiden die Regeln der Dialoggrammatiken darüber, ob eine Sequenz von Äußerungen einen akzeptablen Dialog darstellt oder nicht. Dieses Prinzip soll nun anhand der folgenden von [GWF90] aufgestellten Regeln verdeutlicht werden:

(I) Adjazenzpaar– > Frage, Antwort
(II) Adjazenzpaar– > Frage, Einfügung, Anwort
(III) Adjazenzpaar– > Aufforderung, Reaktion
(IV) Adjazenzpaar– > Angebot, Annahme
(V) Adjazenzpaar– > Angebot, Einfügung, Annahme
(VI) Einfügungen – > Einfügung, Einfügungen
(VII) Einfügungen – > Einfügung
(VIII) Einfügung – > Frage, Einfügungen, Antwort
(IX) Einfügung – > Frage, Antwort

Mit Regelbasis kann z.B. eine Dialogsequenz der Form

$$F_1\ F_2\ F_3\ F_4\ A_4\ F_5\ A_5\ A_3\ A_2\ A_1$$

erkannt werden ("F" und "A" stehen für "Frage" und "Antwort").

> "These simple rules ... cleary do not constitute anything like a complete set of rules for adjacency pairs, but they are a beginning which we can add to."[1]
> *Anmerkung der Entwickler, [GWF90], S.248*

3.2.3 Geschichtsgrammatiken

Einige Psychologen vermuten, daß die verschiedenen Oberflächenstrukturen von Geschichten und Erzählungen auf einer tieferen, universellen Struktur basieren, die

[1] "Diese einfachen Regeln stellen kein vollständiges Regelwerk für Adjazenzpaare dar. Sie sind jedoch eine Grundlage, auf der wir aufbauen können." (Übers. durch Verf.)

allen Geschichten eines Typs gemeinsam ist. Diese Tiefenstruktur kann durch eine sogenannte *Geschichtsgrammatik* (story grammar) definiert werden. Eine Geschichtsgrammatik enthält Regeln für den Aufbau einer Geschichte. [Tho77] gibt für einfache Geschichten folgende Regeln an (vgl. [Gre86], S. 45 ff.):

(Regel I)	Geschichte	$->$ Schauplatz + Thema + Handlung + Lösung
(Regel II)	Schauplatz	$->$ Charaktere + Ort + Zeit
(Regel III)	Thema	$->$ Ziel
(Regel IV)	Handlung	$->$ Episode(n)
(Regel V)	Episode	$->$ Zwischenziel + Versuch(e) + Ergebnis
(Regel VI)	Versuch	$->$ Ereignis(se)
(Regel VII)	Lösung	$->$ Ereignis und/oder Zustand
(Regel VIII)	Ziel	$->$ erwünschter Zustand

Einen *Schauplatz*, ein *Thema*, eine *Handlung* und eine abschließende *Lösung* ergeben eine *Geschichte* (Regel I). Der Schauplatz wird durch die beteiligten *Charaktere*, den *Ort* und die *Zeit* der Handlung bestimmt (Regel II). Das Thema der Geschichte kann in Form des *Ziels*, welches der Hauptakteur in der Geschichte verfolgt, angegeben werden (Regel III). Die Handlung ist in *Episoden* untergliedert (Regel IV). Jede Episode besteht aus Unterzielen, den *Versuchen*, diese zu erreichen, und einem *Ergebnis* (Regel V). Jeder Versuch des Akteurs, die Unterziele zu erreichen, umfaßt *Ereignisse* (Regel VI). Das abschließende Ereignis führt dann zu einem *Zustand*. Dies ist entweder der *gewünschte Zustand*, wie er in der Zielsetzung der Handlung definiert worden ist (Happy End), oder er ist es nicht. In Abbildung 3.3 wird anhand eines Beispiels aus [Gre86] das Prinzip dieser Geschichtsgrammatik verdeutlicht. Je enger sich eine Erzählstruktur an dieses kanonische Format anlehnt, desto verständlicher ist die Geschichte. Andererseits ist eine vorhersehbare Struktur - zumindest für Erwachsene - meist langweilig (vgl. [Gre86]). Dem widerspricht allerdings der Erfolg der "Daily Soaps" und "Sitcoms", bei denen jede Folge auf dem selben Schnittmuster basiert. [Tru96] unterscheidet in diesem Zusammenhang "sieben Schritte einer gut erzählten Geschichte":

(1) Problem/ Bedeutung (problem/ meaning):
Der Held hat ein Problem, das er nicht lösen kann.

(2) Bedürfnis (need):
Dem Helden und den anderen Charakteren fehlt etwas zum "Glück".

(3) Wunsch (desire):
Der Held hat ein Ziel, das er unter allen Umständen erreichen will. Auf diesem Wunsch des Helden basiert die Geschichte.

(4) Antagonist (oponent):
Der Antagonist verfolgt dasselbe Ziel wie der Held. Es kommt zu einem Konflikt zwischen beiden.

(5) Plan (idem):
Der Held benutzt zur Erreichung seines Ziels einen Plan.

(6) Schlacht (battle):
Die Schlacht zwischen Held und Antagonist ist der finale Konflikt.

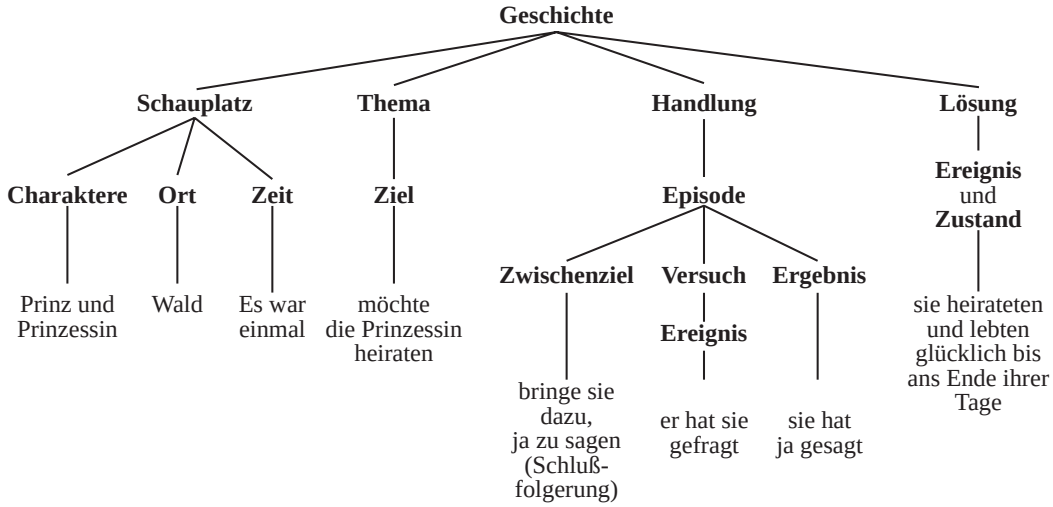


Abbildung 3.3: Baumstruktur einer einfachen Geschichte

Eine einfache Geschichte: "Es war einmal ein Prinz, der mit einer Prinzessin im Wald spazieren ging. Der Prinz hatte die Absicht, die Prinzessin zu heiraten. Also fragte er sie, ob sie seine Frau werden wolle. Sie sagte ja. Sie heirateten und lebten glücklich bis ans Ende ihrer Tage."

Quelle: [Gre86], S.46 f., (Übers. durch Verf.)

(7) Selbsterkenntnis (self-revelation):
Der Konflikt endet in der Selbsterkenntnis, in der das Bedürfnis gestillt wird. Ist das Ergebnis für den Helden positiv, befindet er sich auf einem "höheren Niveau" als zuvor. Ein negatives Ergebnis bedeutet die Zerstörung des Helden.

3.2.4 Planbasierte Theorien

Die Teilnehmer von Gesprächen und Dialogen verfolgen ebenso wie die Charaktere in Erzählungen mit ihrem Handeln bestimmte Ziele. Die Kenntnis dieser Absichten ist wichtig für das Verständnis. Die Intentionen der einzelnen Personen sind ein Teil der pragmatischen Information, die die Interpretation des Geäußerten entscheidend beeinflußt. [Leh82] formalisiert beispielsweise Erzählungen anhand der Ziele und Aktionen der darin vorkommenden Personen. Einige der von ihm definierten *primitiven Handlungseinheiten* (primitive plot units) sind in Tabelle 3.2 dargestellt. Diese Basishandlungseinheiten können zu komplexeren Abfolgen, die Ziele, Pläne und deren Ergebnisse (Scheitern oder Gelingen) umfassen, zusammengesetzt werden.

3.3 Maschinelles Sprachverstehen

Die *Computerlinguistik* versucht, die menschliche Fähigkeit der Rezeption und Produktion natürlicher Sprache zu modellieren. Sie ist ein Teilgebiet der Künstlichen

Handlungs-einheiten	Beispiele
Problem (problem)	Du wirst entlassen und brauchst einen Job. Dein Hund stirbt, und du wünschst dir einen Begleiter.
Erfolg (success)	Dir wird die erbetene Gehaltserhöhung bewilligt. Du flickst eine platten Reifen.
Versagen (failure)	Dein Heiratsantrag wird abgelehnt. Dein Kreditantrag wird abgelehnt. Du kannst Deine Geldbörse nicht finden.
Erleichterung (resolution)	Der Dieb deiner Geldbörse wird gefaßt. Dein defektes Radio geht plötzlich wieder.
Verlust (loss)	Die Frau, die du liebst, verläßt dich. Die Steuerrückzahlung war ein Irrtum.
Beharrlichkeit (perseverance)	Du möchtest (nochmal) heiraten. Du bewirbst dich nochmal in Harvard, nachdem du abgelehnt wurdest.
Glück im Unglück (hidden blessing)	Ein Verwandter stirbt, und du erbst eine Million. Nach einer Steuerprüfung bekommst du eine Rückzahlung.
Beweggrund (motivation)	Du brauchst einen Rat und fragst einen Freund. Du möchtest mit einem Kunden sprechen und rufst ihn an.

Tabelle 3.2: Beispiele für primitive Handlungseinheiten (primitive plot units), aus [Leh82], S.381

Intelligenz.

Die *Spracherkennung* assoziiert mit der akustischen Repräsentation linguistische Einheiten wie Phoneme, Silben, Worte, Sätze. Der Text ist als verschriftlichte Sprache eine von Spracherkennungs- und Sprachsynthesesoftware häufig verwendete linguistische Beschreibungsform der Aus- bzw. Eingabe. Die Phonemschreibweise (Lautschrift) der Fremdwörterlexika oder die SAMPA-Notation (Abschnitt 7.2.6) sind alternative Darstellungsformen.

3.3.1 Repräsentationsformen gesprochener Sprache

Die naturgetreueste Repräsentation gesprochener Sprache im Computer ist die Form eines digitalisierten Mikrophonsignals in n Werten pro Sekunde. Jeder dieser n Werte wird mit einer Genauigkeit von m-bit klassifiziert. Typische Abtastraten hierbei sind 8kHz, 16kHz oder 32kHz. Typische Auflösungsschärfen sind 8, 16 oder 32 bit.

Ein analoges oder digitales Audiosignal kann durch frequenz- oder amplitudenorientierte Verfahren analysiert und in eine akustisch-phonetische Repräsentationsform umgesetzt werden. Spracherkennung übersetzt diese Darstellung in die nächste Abstraktionsebene, die linguistisch-phonemische Repräsentation. Syntheseverfahren kehren diese Prozesse um. Abbildung 3.4 stellt diesen Ablauf dar. Detailliertere Darstellungen sind in der Literatur der Spracherkennung und -synthese zu finden. Einen Überblick gibt [Isa86].

3.3.2 Maschinelle Sprachanalyse

Ein Schema zur linguistischen Sprachanalyse ist in Abbildung 3.5 dargestellt. Gesprochene Sprache wird zunächst durch *phonologische Analyse* in Text umgesetzt.

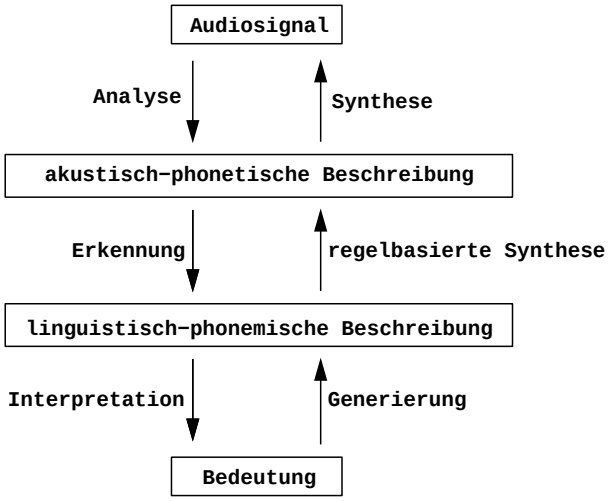


Abbildung 3.4: Umsetzung der sensorischen Eingabedaten in verschiedene Repräsentationsebenen

Quelle: [Isa86]

Dieser Schritt entspricht der in Abschnitt 3.3.1 angesprochenen Spracherkennung. Wird geschriebene Sprache als Eingabe verwendet, entfällt dieser erste Verarbeitungsschritt bzw. wird - bei Handschrift - durch die Schrifterkennung ersetzt. In der *morphologischen Analyse* wird jedes Wort in seine Wurzel und die Flexionselemente zerlegt (z.B. "geh-st"). Die *lexikalische Analyse* ordnet den Wörtern eine lexikalische Kategorie (Verb, Adjektiv, Substantiv etc.) und Merkmale (Vergangenheit, Plural etc.) zu. Die *syntaktische Analyse* wird als *Parsen* bezeichnet. Durch Anwendung der grammatischen Regeln wird die Struktur des Satzes bestimmt. Die *semantische Analyse* erzeugt eine Darstellungsform, die Schlußfolgerungen auf die Bedeutung des Satzes erlaubt. Semantische Mehrdeutigkeiten werden in der *pragmatischen Analyse* aufgelöst. Hier wird der Satz im Kontext, in dem er geäußert wurde, interpretiert. (vgl. [Win90], S. 92 f.)

Der Schwerpunkt der Computerlinguistik liegt in der syntaktischen und semantischen Analyse. Die Problematik, einen geeigneten Parser zu konstruieren, liegt einerseits in der Formulierung der Grammatik, eines präzisen Regelwerks, welches die möglichen Sätze einer Sprache erzeugt, andererseits im Parsen selbst. Ein *Parser* interpretiert die Grammatik und setzt eine Wortsequenz in eine syntaktische oder semantische Repräsentation um. Ein *Generator* produziert aus einer semantischen Repräsentation einen natürlichsprachlichen Text. Sowohl Parser als auch Generator benutzen Grammatiken und Lexika.

3.3.3 Strategien für syntaktisches und semantisches Parsen

Für die Analyse und das Zusammenfügen von Sätzen existieren unterschiedliche Strategien. Bottom-Up-Parser beginnen mit lokalen Wortkombinationen und arbeiten sich bis zur Satzebene vor. Top-Down-Verfahren suchen von Anfang an nach möglichen Sätzen. Parallele Verfahren bearbeiten alternative Möglichkeiten simul-

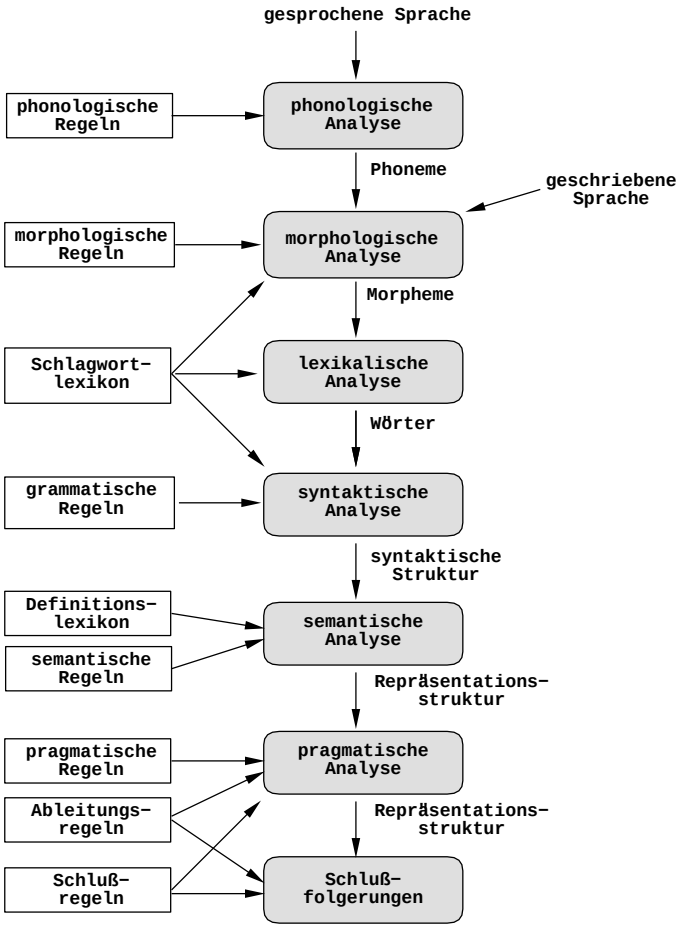


Abbildung 3.5: Maschinelle Sprachanalyse

Quelle: [Win90], S. 92 f.

tan, während sequentielle Verfahren erst dann Alternativen bearbeiten, wenn sich eine Möglichkeit als falsch herausgestellt hat. Ein Typus von Parsern basiert auf den in der theoretischen Linguistik entwickelten generativen Grammatiken (Abschnitt 3.1.2). *ATN-(Augmented Tranistion Network)-Grammatiken* und *lexikalisch-funktionale Grammatiken* sind speziell zur maschinellen Sprachverarbeitung entwickelt worden. ATN-Grammatiken interpretieren die Struktur eines Satzes als Folge von Übergängen. Lexikalisch-funktionale Grammatiken erzeugen eine funktionale Struktur des Satzes. Den einzelnen Wörtern und Ausdrücken wird die grammatische Funktion (Subjekt, Objekt etc.) zugeordnet.

3.3.4 Semantische Analysen

Die semantische Analyse transformiert die syntaktische Struktur eines Satzes in eine logische Form, die der Computer für Schlußfolgerungen nutzt. Ebenso umstritten wie die Wahl der Grammatik oder der syntaktischen Parsing-Strategie ist die Frage nach der geeigneten logischen Form zur Bedeutungskodierung sprachlicher Ausdrücke. Prädikatenlogik, semantische Repräsentationssprachen und semantische

Netzwerke sind verschiedene Methoden, die zur semantischen Kodierung benutzt werden.

3.3.5 Pragmatische Ansätze

Die letzte Stufe der maschinellen Sprachanalyse, die pragmatische Analyse, bezieht den Kontext in die Bedeutungserfassung ein. Pragmatische Ansätze zum Verstehen natürlicher Sprache finden sich z.B. in Winograds *SHRDLU system*, dem *Frame*-Konzept von Minsky [Min75] oder Schanks *Script*-Ansatz [SA77].

> "Die Beschränkungen, denen die Formalisierung der kontextabhängigen Bedeutung unterworfen ist, machen es gegenwärtig - und vielleicht für immer - unmöglich, Computerprogramme zu entwickeln, die das menschliche Sprachverständnis wirklich imitieren können." *[Win90], S. 97*

Beispiel: Frame Representation

[Min75] schlägt mit dem *Frame* eine Verarbeitungseinheit vor, die aus einem Kern und sogenannten Steckplätzen besteht. Ein *Frame* verkörpert ein gedankliches Konzept. Der Kern ist das Gerüst. Die Steckplätze werden kontextabhängig angepaßt. Sie stellen Facetten des in dem *Frame* codierten Konzeptes dar oder zeigen auf andere Konzepte, die mit dem *Frame* oder dem aktuellen Kontext in Bezug stehen. *Frames* eignen sich neben der Modellierung von Kontexten auch zur Codierung stereotypen Verhaltens (vgl. [Wal82], [BLH91]).

3.3.6 Computational Behaviourism

Die Linguistik untersucht die Gesetzmäßigkeiten natürlicher Sprachen. Die Computerlinguistik entwickelt aus den gefundenen Regeln algorithmische Beschreibungen für eine und - bei Übersetzungsszenarien auch - mehrere natürliche Sprachen. Es wird versucht, eine Grammatik zu entwickeln, die präzise genug ist, um als Computerprogramm implementiert werden zu können.

Eine weiterer Ansatz der maschinellen Verarbeitung und des Verstehens natürlicher Sprache entstammt dem philosophischen Ansatz des *Behaviorismus*[2]. Vertreter des *Computational Behaviourism* behandeln die maschinelle Verarbeitung natürlicher Sprache nicht als linguistisches Problem, sondern als behavioristisches. Hierbei gewinnt der invariante Teil verbalen Verhaltens an Bedeutung. Invariant bezeichnet hier den Teil einer Äußerung, der die Kernbedeutung codiert.

> Whereas linguistics as the study of the nature of language, psychology is the study of the nature of behaviour. In particular, the philosophical approach called "behaviourism" concentrates solely on the study of behaviour to the exclusion of mental processes. Rather than attacking natural language interactions as a linguistic problem, I have attacked it as a behavioural problem. This is a fundumental change in approach which has far-reaching consequences. The essential question is no longer 'How does language work?' but rather, 'What do people say?' In fact, I ask that question in a more specific form: "What are the invariant parts of verbal behaviour

[2] siehe Abschnitt A.

(i.e. questions) that form a specific operant (i.e. that lead to a specific answer)?" [3] *[Wha96]*

Dieser Ansatz hat bei der Generierung von Antworten eine Vergröberung der linguistischen Verarbeitungseinheit zur Folge. Anstatt einzelne Worte zu Sätzen und Sätze zu Texten zu kombinieren, werden ganze Sätze und Satzfolgen in einer Datenbank gespeichert und in einem *Information-Retrieval*-Prozess abgerufen.

Beispiel: CHAT Natural Language System

In [Wha96] ist die natürlichsprachliche Benutzerschnittstelle CHAT (Conversational Hypertext Access Technology) beschrieben. CHAT wurde für den Zugriff auf elektronische Informationen aus Datenbanken entwickelt und versetzt den Benutzer in die Lage, Anfragen in seiner Muttersprache zu formulieren, was insbesondere für jemanden, der wenig Computerkenntnisse besitzt, eine Erleichterung gegenüber traditionellen *Frontends* wie Menüsystemen darstellt.

Das System basiert auf einem Hypertextmodell, in dem die einzelnen Informationsparagraphen so verknüpft sind, daß der Benutzer durch die Verbindung zu der gewünschten Information navigieren kann. Das Navigieren wird von der Software übernommen und der Benutzer sieht nur den natürlichsprachlichen Dialog. Das CHAT-System basiert auf dem Prinzip des *Computational Behaviourism*. Der invariante Teil einer Äußerung wird erkannt und in das Programm codiert. Während des Dialogs wird jede neue Frage mit einer Reihe von Templates verglichen.

Dieses Prinzip soll an einem Beispiel verdeutlicht werden. Jemand, der wissen möchte, was NLP ist, hat viele Möglichkeiten danach zu fragen - z.B. "Was ist NLP?", "Definiere NLP!" oder "Erzähl mir etwas über NLP." Alle Fragen haben eins gemeinsam: Sie enthalten den Begriff "NLP". Diese Information kann zum Auffinden einer geeigneten Antwort mit der gewünschten Information verwendet werden.

3.4 Textbasierte Konversationsprogramme

In diesem Abschnitt werden einige Computerprogramme vorgestellt, die entwickelt wurden, um mit dem Benutzer einen textbasierten Dialog zu führen. Die Verfahren zur Sprachanalyse basieren nicht auf der Computerlinguistik, sondern sind mit dem Ansatz des *Computational Behavourism* vergleichbar.

[3] "Während die Linguistik die Natur und Struktur der Sprache zum Forschungsgegenstand hat, beschäftigt sich die Psychologie mit der Natur des Verhaltens. Der philosophische Ansatz *Behaviorismus* konzentriert sich - ohne Berücksichtigung mentaler Prozesse - auf die Verhaltensforschung. Anstatt natürlichsprachliche Mensch-Maschine-Interaktion als linguistisches Problem anzugehen, erforsche ich es als behavioristisches Problem. Dieser fundamentale Wechsel des Forschungsansatzes hat weitreichende Konsequenzen. Die grundlegende Frage ist nicht mehr 'Wie funktioniert Sprache?', sondern 'Was drücken die Menschen mit dem, was sie sagen, aus?'. Genauer gesagt, stelle ich die Frage in einer konkreteren Form: 'Was ist der invariante Teil einer verbalen Verhaltensweise (z.B. einer Frage), die einen bestimmten Operanden ergeben (z.B. die zu einer bestimmten Antwort führen)?' " (Übers. durch Verf.)

3.4.1 Turing Test

Der Turing Test beschreibt ein Szenario, bei dem eine Testperson als Gesprächsteilnehmer mit Bildschirm, Tastatur und vernetztem Rechner in geschriebener, natürlicher Sprache mit zwei Dialogpartnern kommuniziert. Einer ist menschlich, der andere eine Maschine, ein Computer, speziell programmiert auf dieses textbasierte Frage-Antwort-Spiel. Nach einer bestimmten Zeitspanne soll die Testperson dann bestimmen, welcher Dialogpartner menschlich und welches die Maschine ist. Alan Turing, der diesen Test 1950 als ein Meßverfahren für künstliche Intelligenz vorschlägt, prophezeite, daß sich innerhalb der nächsten fünfzig Jahre - also bis zur Jahrtausendwende - die Leistungsfähigkeit von Hard- und Software so steigern würde, daß für 70 % der Testpersonen bei einer Interaktionsdauer von 5 Minuten die maschinell generierten Antworten nicht von denen eines menschlichen Benutzers unterscheidbar wären (vgl. [Tur50]).

3.4.2 Eliza

Eliza gilt als das erste Computerprogramm zur textbasierten, natürlichsprachlichen Konversation mit Menschen. Die Grundidee von Eliza ist die Simulation der *Rogerian Method*, eine psychotherapeutischen Methode, deren Ziel ist, einen Dialog zu erzeugen, in dem primär der Patient erzählt. Der behandelnde Arzt hat dabei die Aufgabe, den Erzählfluß des Patienten durch entsprechendes Nachfragen aufrechtzuerhalten. (vgl. [Wei66])

3.4.3 Parry

In [Col75] ist ein Konversationsprogramm (Parry) beschrieben, das paranoides Verhalten simuliert. Parry basiert auf demselben Prinzip wie Eliza, benutzt jedoch zusätzliche Tricks: Parry kann im Gegensatz zu Eliza mit “I don’t know!” auf eine Frage antworten und somit ignorantes Verhalten zeigen. Mit Gegenfragen wie “Why do you ask that?” kann es einen Wechsel des Gesprächsthemas herbeiführen. Parry enthält eine Wissensdatenbank mit kurzen Geschichten über die Mafia und versucht diese Geschichten in einer starren Reihenfolge zu erzählen. Es kann sowohl stur auf dem letzten Gesprächsthema beharren als auch spontan ein neues einführen. Letzteres simuliert paranoides Verhalten gut, kann aber die Chance auf eine reguläre Konversation stark einschränken. Parry konnte die Testpersonen unter anderem deshalb täuschen, weil paranoides Verhalten oft solche unangemessenen Antworten zeigt. Eine gewisse Logik, die in diesen zunächst nicht passenden Reaktionen steckt, wird von Parry gut simuliert. Parry stellt insofern einen Fortschritt gegenüber Eliza dar, als es eine Persönlichkeit besitzt. Die *Rogerian Method*, die von Eliza simuliert wird, versucht die Persönlichkeit des Therapierenden zu verbergen.

3.4.4 Chatterbots

In sogenannten MUDs (Multiuser Dungeon) können sich weltweit Benutzer über das Internet einwählen und in verschiedenen Räumen mit anderen Benutzern, die sich zur gleichen Zeit in dem entsprechendem Raum befinden, textbasiert über

meist raumspezifische Gesprächsthemen unterhalten. Solche Institutionen bieten für Konversationsprogramme die Möglichkeit eines unerwarteten Turing-Tests: Der Gesprächspartner des Konversationsprogramms glaubt, mit einem humanoiden Gesprächspartner zu kommunizieren.

Für solche Konverstionsprogramme haben sich in der angelsächsischen Literatur die Bezeichungen *Chatterbots* und *Knowbots* etabliert. *Chatterbots* erweitern die Funktionalität von Eliza und Parry um die folgenden Eigenschaften (vgl. [Mau94]):

- In einem *Activation Network* sind Fragmente aus Konversationssequenzen gespeichert, wodurch der Zusammenhang der einzelnen Programmäußerungen erhöht wird.
- Mit kontroversen Behauptungen wie "People don't own cats!" wird versucht, dem Benutzer die Gesprächsleitung zu entziehen.
- Mit humorvollen Äußerungen wird versucht, das Programm menschlich erscheinen zu lassen.
- Von Zeit zu Zeit stimmt der Chatterbot mit dem Benutzer überein.
- Die als *Activation Network* implementierte Wissensdatenbank mancher *Chatterbots* enthält Informationen und Auszüge aus *Newsgroups*, die allerdings per Hand eingegeben und nacheditiert wurden.
- Ein speziell für textbasierte Konversation entwickelter Trick ist das simulierte Tippen: Die einzelnen Buchstaben werden mit realistischen Verzögerungen angezeigt. Dadurch wird der menschliche Tipprythmus imitiert. Bei manchen *Chatterbots* sind absichtlich Tippfehler eingebaut, die für den Benutzer sichtbar korrigiert werden.

Beispiel: Julia - ein Chatterbot

[Mau94] beschreibt die Architektur des Chatterbots "Julia". Die wichtigsten Eigenschaften werden hier zusammengefaßt. Das Konversationsmodul von Julia ist als priorisiertes Schichtenmodell von Miniexperten implementiert. Jedes Eingabemuster ist mit einer Alternative von möglichen Antworten gekoppelt. Direkte Befehle von dem Chatterbotentwickler haben höchste Priorität. Des weiteren werden vier Prioritätsstufen unterschieden:

I *High Priority Responses* reagieren auf allgemeine Fragen, die durch Schlüsselwortsuche behandelt werden können;

II *Topic Oriented Responses* sind im *Activation Network* gespeicherte Antworten, die auf das aktuelle Gesprächsthema abgestimmt wurden (Abbildung 3.6 zeigt einen Ausschnit aus dem Konversationsnetz.)

III *Low Priority Responses* umfassen eine Reihe von rudimentärem Allgemeinwissen und Kenntnisse, die der Roboter über sich selbst haben sollte ("Where do you live?", "What's 2 times 23?", "What color is your hair?").

IV *Sorry Responses* werden benutzt, wenn die Eingaben zu keinem der Muster aus den höheren Schichten paßt ("Go On!", "So?").

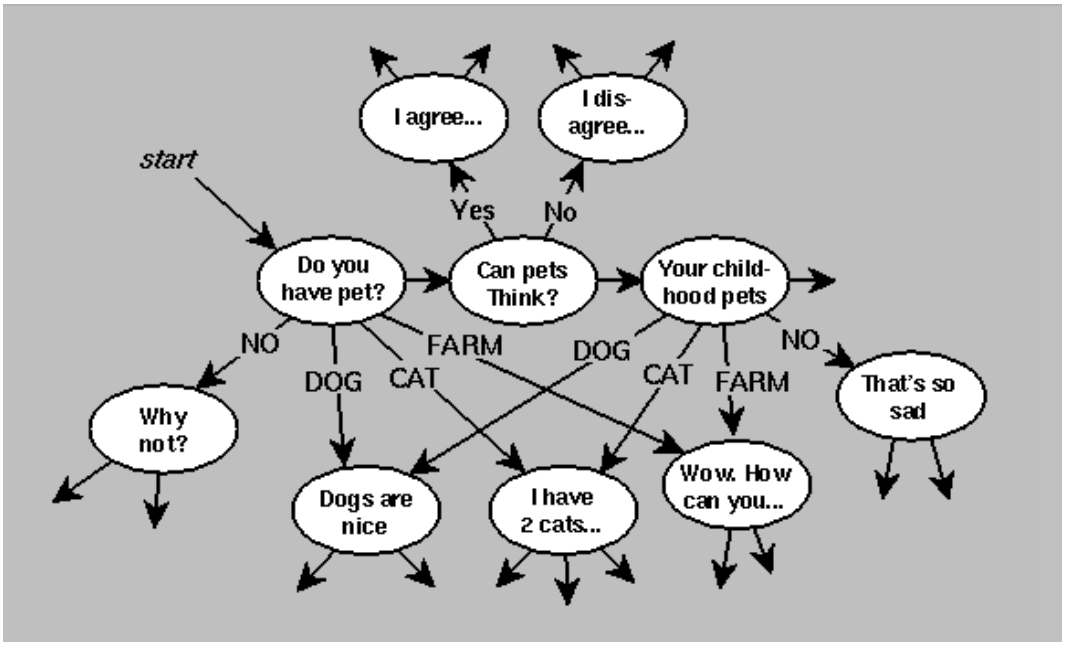


Abbildung 3.6: Ausschnitt aus einem Konversationsnetz des Chatterbots "Julia"

Quelle: [Mau94]

Eine Eingabe wird also solange in Reihenfolge der römischen Numerierung in den einzelnen Schichten nach passenden Wortmustern durchsucht, bis eine passende Antwort gefunden wurde. Die für die vorliegende Arbeit interessanteste Schicht ist das *Activation Network*, die die *Topic Oriented Responses* enthält. Ein Knoten dieses Graphen wird als Konversationsknoten bezeichnet. Jeder Konversationsknoten besitzt fünf Attribute:

- *Activation*
  Jeder Knoten startet mit einem initialen Aktivierungswert zwischen 0,0 und 1,0.

- *Patterns*
  Ein oder mehrere gewichtete Muster werden mit der Benutzereingabe verglichen. Bei positiven Resultaten wird die Aktivierung erhöht.

- *Response*
  Ein Zeichenkette, die als Antwort ausgewählt wird, falls der Knoten die höchste Aktivierung hat.

- *Enhancement*
  Wenn der Knoten als Antwort benutzt wird, dann wird die Aktivierung der in der *Enhancement*-Liste aufgezählten Knoten erhöht.

- *Inhibition*
  Wenn der Knoten als Antwort benutzt wird, dann wird die Aktivierung der in der *Inhibition*-Liste aufgezählten Knoten gehemmt.

Bei einem ausreichend großen Graphen von Konversationsknoten reduziert sich das Konversationsproblem zu einem *Retrieval*-Problem der geeigneten Auswahl einer passenden Antwort aus einer gegebenen Alternative.

3.5 Zusammenfassung

Die Abschnitte 3.1 und 3.2 haben gezeigt, daß das Phänomen *natürliche Sprache* auf Gesetzmäßigkeiten beruht, die auf verschiedenen Verarbeitungsebenen aufgestellt werden können. Die Computerlinguistik (Abschnitt 3.3) beschäftigt sich mit der Entwicklung von Verfahren, die die Regeln der natürlichen Sprache für die maschinelle Verarbeitung nutzen. Grundsätzlich läßt sich feststellen, daß die Konzepte der Computerlinguistik umso unklarer werden, je mehr die Verarbeitungsebene von der Transkription abstrahiert. Abschnitt 3.4 beschreibt die für die vorliegende Arbeit relevanten Entwicklungen auf dem Gebiet der textbasierten Konversationsprogramme und stellt einige Programme exemplarisch vor. Die Sprachanalyse der in der vorliegenden Arbeit vorgestellten Konzeption (Abschnitt 6) basiert auf den Prinzipien der textbasierten Konversationsprogramme. Insbesondere die Methoden von Eliza und Julia haben einen Einfluß auf die Konzeption des VISTA-Systems.

Kapitel 4

Hypertext und nichtlineare Erzählformen

Geschichten *interaktiv* oder *nichtlinear* zu erzählen, birgt auf den ersten Blick einen Widerspruch. Der Erzählende beginnt mit *seiner* Geschichte. Interaktivität impliziert jedoch, daß derjenige, der die Geschichte erfährt, den Verlauf und in manchen Fällen auch den Ausgang beeinflußen kann. Um diesen Konflikt zu lösen, muß der Autor nichtlinearer Erzählungen entweder die Einflußmöglichkeiten des Adressaten auf die Geschichte oder die Geschichte selbst einschränken - oder beides. Der Adressat kann auf solche Aktivitäten begrenzt werden, die keine Auswirkungen auf den Verlauf der Erzählung haben: Beispielsweise kann das Spielen mit Gegenständen in einem Raum zugleich unterhaltend und unbedeutend für den Werdegang der Geschichte sein. Erzählungen dieser Kategorie sind nur marginal interaktiv. Die Nichtlinearität bezieht sich auf Ad-Hoc-Ausnahmen und Wahlmöglichkeiten innerhalb der jeweiligen Situation, deren Eintritts- und Austrittspunkt vom Autor definiert ist. Die Abfolge dieser Interaktionsschauplätze ist nicht vom Adressaten beeinflußbar und somit linear. Die Geschichte kann auf die ursprüngliche Idee begrenzt werden. Für den Verlauf ist der Benutzer verantwortlich. Dies ist z.B. in vielen Computerspielen der Fall: Die Geschichte ist auf den Schauplatz, ein vorgegebenes Ziel und zwei unterschiedliche Charaktertypen - den Spieler und seine Widersacher - reduziert; der Spieler befindet sich z.B. in einem Schloß, hat die Aufgabe, eine Prinzessin zu befreien und tötet dabei alles, was ihm in die Quere kommt. (vgl. [Hip95])

In den meisten interaktiven Erzählungen finden sich Einschränkungen, die sowohl die Geschichte als auch deren Beeinflußung durch ihren Adressaten betreffen. Strategien, dies zu erreichen, gibt es viele. Der Kreativität des Autors sind in diesem Genre, das irgendwo zwischen Buch, Film und Computerspiel anzusiedeln ist, keine Grenzen gesetzt.

4.1 Grundtypen von Erzählstrukturen

Grundlegend für interaktive Erzählformen sind die zwei Elemente: Die *Episoden* und *Verzweigungsstellen*, die die Episoden miteinander verbinden. Innerhalb der einzelnen Episoden ist die Erzählstruktur linear. An den Verzweigungspunkten ist der

Rezipient vor verschiedene Wahlmöglichkeiten gestellt. Eine Sequenz von Episoden wird als *Pfad* bezeichnet. (vgl. [Bol91], S. 121 ff.)

[Pla95] unterscheidet fünf Grundtypen von Erzählstrukturen:

1. *ein Pfad, keine Wahlmöglichkeit:*
   Die traditionellen Erzählformen Film, Buch, Theater etc. lassen dem Publikum keine Einflußmöglichkeiten.

2. *ein Pfad, kleinere Abweichungen:*
   Der Benutzer beginnt am Anfang, schweift zwischendurch an vereinzelten Stellen ab, folgt aber stets einem vorgeschriebenen Verlauf und gelangt an ein vordefiniertes Ende.

3. *mehrere Pfade mit vordefinierten Verzweigungspunkten:*
   Bei diesem "Wähle-Dein-Eigenes-Abenteuer-Modell" wird der ansonsten passive Zuschauer an vordefinierten Stellen in der Geschichte aufgefordert, mit der Wahl eines Pfades aus einem Auswahlmenü den Verlauf der Geschichte zu beeinflussen. Er wird somit entlang vorprogrammierter Pfade zu einem von mehreren vorbestimmten Ausgängen geleitet.

4. *mehrere Pfade, unaufgeforderte Auswahl:*
   Verlaufsbeeinflussende Entscheidungen können jederzeit getroffen werden, der Benutzer wird hierzu nicht mehr aufgefordert, das Programm antwortet z.B. mit einer kontextabhängigen, vorproduzierten Auswahl von Szenen.

5. *Entdeckungsreise:*
   Der Benutzer bewegt sich durch eine Welt ohne vorproduzierte Sequenzen.

Jede Hypertextstruktur und jede interaktive oder nichtlineare Erzählung läßt sich ganz oder partiell in eine dieser fünf Kategorien einordnen. Zwei Parameter bestimmen den Grad der Interaktivität:

- die *Länge der Episoden* und
- die *Vernetzung der Episoden.*

Die Adjektive "interaktiv" und "nichtlinear" werden häufig und in den unterschiedlichsten Zusammenhängen verwendet. Im Kontext von Erzählformen ist ihre Bedeutung immer im Zusammenhang mit dem verwendeten Medium (papier- oder computerbasierte Schrift, analoge oder digitale Ton- oder Bildträger) zu interpretieren. Die in der angelsächsischen Literatur geprägten Begriffe *Interactive Fiction*, *Hyperfiction* und *Nonlinear Storytelling* werden häufig synonym gebraucht. Bei näherer Betrachtung können nichtlineare Erzählungen danach klassifiziert werden, ob der Rezipient durch Umschaltmechanismen Einfluß auf den Verlauf der Geschichte hat oder nicht. Kann der Benutzer den *Erzählverlauf* beeinflussen, ist die nichtlineare Erzählung interaktiv. Kann der Benutzer den *Geschichtsverlauf* beeinflussen, ist die Erzählung hochgradig interaktiv.

4.2 Interaktive Geschichten

[BHRM95] gibt in fünf Dimensionen einige Beispiele für unterschiedliche strukturelle Eigenschaften von interaktiven Erzählformen (Übers. durch Verf.):

- Eine interaktive Geschichte entwickelt sich aus
  (a) dem autonomen Verhalten der Charaktere,
  (b) dem semi-autonomen Verhalten der Charaktere, eingeschränkt durch eine abstrakte Handlungsstruktur oder aus
  (c) einer Echtzeitsteuerung der Agenten durch einen *Expert Story Master*.
- Die Charaktere können auf eine bestimmte Art und Weise handeln, weil
  (a) das ihr normales Verhalten ist,
  (b) das der Eindruck ist, den sie auf das Publikum machen möchten, oder weil
  (c) sie einem Skript folgen.
- Handlungsstruktur und Charaktere werden erschaffen von
  (a) professionellen Autoren,
  (b) einem computerbasierten Autor oder
  (c) dem Benutzer.
- Die Benutzer partizipieren zur Laufzeit als
  (a) Charaktere in der Geschichte,
  (b) Ratgeber von sonst autonomen Charakteren oder als
  (c) Regisseur der Inszenierung.
- Die Benutzer werden geleitet von
  (a) elaborierten Anweisungen oder
  (b) dem natürlichen Lauf der Ereignisse.

[Hip95] unterscheidet zwischen "Nonlinear Narratives in Practice" und "New Nonlinear Approaches". Zur ersten Kategorie der bereits praktizierten nichtlinearen Erzählungen zählen *Hypertext*, *lineare Erzählformen mit nichtlinearen Elementen* und das *Labyrinth Modell*. Zu der zweiten Gruppe der neuen Ansätze für interaktiven Erzählformen zählen das *Charakterverfolgungsmodell*, das *Modell der multiplen Realität* und das *Modell der verschachtelten Erzählungen*. Im folgenden wird jeweils der Grundgedanke der einzelnen Modelle erläutert. Eine ausführliche Diskussion der Vorteile und Probleme der einzelnen Modelle findet sich in [Hip95].

4.2.1 Hypertext

Der Begriff Hypertext bezeichnet eine Menge von Informationsobjekten, die durch eingebettete Verbindungen, sogenannten *Links*, untereinander vernetzt sind.

Beispiel: Hypertextroman "Afternoon"

Eines der ersten Beispiele des Genres *Hyperfiction* ist der Hypertextroman "Afternoon" [Joy87]. Die Benutzerschnittstelle besteht aus einem Textdisplay und einer Texteingabezeile. Im Textdisplay erscheinen die Episoden. Die Episoden enden mit Fragen, auf die der Benutzer in der Texteingabezeile mit "yes" oder "no" antworten kann. Aus dieser Benutzereingabe wird dann die entsprechende Verbindung zur nächsten Episode aktiviert. Weiterhin hat der Rezipient von "Afternoon" die Möglichkeit, einzelne Worte direkt aus dem Textdisplay als Verbindung zu aktivieren. Für den Fall, daß der Rezipient die *Return*-Taste aktiviert, ohne vorher eine Auswahl getroffen zu haben, ist für jede Episode ein Standardnachfolger definiert. "Afternoon" besteht aus vernetzten Episoden. Die Geschichte "Afternoon" existiert als die Summe der gelesenen Episoden. (vgl. [Bol91], S. 123)

4.2.2 Lineare Erzählformen mit nichtlinearen Elementen

In vielen interaktiven Erzählungen (z.B. animierte Kinderbücher) konvergieren die Erzählpfade so stark, daß die Erzählstruktur linear ist. Der Benutzer kann in den Geschichtsverlauf eingreifen, indem er durch Klicken auf Objekte isolierte Medien (Videosequenzen, Zusatzseiten, Animationen etc.) aktiviert. Dies sind die nichtlinearen Elemente.

4.2.3 Labyrinthmodell

Eine stark verbreitete Erzählform in interaktiven Medien ist die Form des Irrgartens. Dieses Genre hat mit dem *Textadventure* und dem *graphischen Labyrinth* zwei Ausprägungsformen. Im Gegensatz zu Hypertextstrukturen, in denen die Verbindungen durch Worte oder Objekte dargestellt werden, sind die Links in einem Labyrinth metaphorisch. Es existiert ein virtuelles physikalisches Modell von Verbindungen, welches den Entscheidungsbaum in einen virtuellen Raum (eine Höhle, eine Insel, ein Schloß etc.) abbildet. Alle Pfade sind physikalische Pfade in einem geschlossenen Raum, wodurch deren Konvergenz gewährleistet ist.

4.2.4 Charakterverfolgungsmodell

In dem Charakterverfolgungsmodell hat der Benutzer als Betrachter einer Szene die Möglichkeit, die Handlungen eines Charkaters seiner Wahl zu verfolgen. Er bewegt sich mit diesem Charakter durch die virtuelle Welt. Dies ist eine Variante des Labyrinthmodells, in dem die Charaktere die Räume als Bezugsobjekte ersetzen.

4.2.5 Modell der multiplen Realität

Eine weitere Variation des Irrgartenmodells ist das Modell der multiplen Realität. Der Autor muß für dieses Modell verschiedene Sichtweisen einer Geschichte oder Situation beschreiben, die nicht widersprüchlich, aber dennoch unterschiedlich sind. Diese verschiedenen Realitäten müssen so konstruiert werden, daß sie während der Inszenierung auf irgendeine Weise miteinander interagieren. Der Benutzer wechselt mit seinem Standpunkt im Verlauf der Interaktion die Realitäten.

4.2.6 Modell der verschachtelten Erzählungen

Die Navigationsreferenzen des Modells der verschachtelten Erzählungen sind die Geschichten selber. Die Verbindungen der Geschichten sind gemeinsame Situationen, Charaktere etc., durch die die Geschichten ineinander verschachtelt sind.

Kapitel 5

Dialogfähige virtuelle Charaktere

Die Synchronisation von nonverbalem Verhalten und verbaler Sprache eines autonomen 3D-Charakters ist Schwerpunkt dieses Kapitels. In den Abschnitten 5.1 und 5.2 werden die nonverbalen Komponenten des multimodalen Dialogs eines 3D-Charakters in Bezug zur natürlichsprachlich-verbalen Komponente (Kapitel 3) gesetzt. Abschnitt 5.3 behandelt den Agenten-Begriff und die Anforderungen an Agentensoftware.

5.1 Psychologische Grundlagen

In Kapitel 3 wurde die verbale Komponente des Phänomens *natürliche Sprache* behandelt. Ausgehend von linguistischen Grundlagen wurden Möglichkeiten aufgezeigt, verbal-natürliche Sprache als Medium zur Mensch-Maschine-Kommunikation zu nutzen. Dieser Abschnitt behandelt Gesetzmäßigkeiten nonverbaler Sprache und Strategien diese in den Interaktionsprozeß einzubeziehen.

Die Erstellung nonverbaler Lexika und Verhaltensregeln ist sowohl aus psychologischer Sicht wie auch im Hinblick auf eine automatische Generierung der Animationsdaten wünschenswert. Zunächst müssen (vergleichbar mit den Basiseinheiten der Linguistik Phonem, Morphem, Lexem, Graphem etc.) die Grundelemente der Gestik und Mimik erkannt werden. Dies sind - im Sinn der menschlichen Wahrnehmung - atomare Codierungselemente der nonverbalen Sprache. Weiterhin müssen geeignete Beschreibungsverfahren und Kombinationsregeln dieser Grundeinheiten der nonverbalen Sprache gefunden werden. Zur Erstellung von nonverbalen Lexika ist dies jedoch nicht ausreichend. Die implizite psychologische Semantik der Gestik, Mimik und Körperhaltung muß ermittelt werden. Dies ist Aufgabe und Ziel der nonverbalen Kommunikationsforschung: Es

> "müssen zunächst Zusammenhänge zwischen spezifischen Verhaltensvariationen und je spezifischen Wirkungen auf den Rezipienten (Beobachter oder Interaktionspartner) empirisch erschlossen werden. Hierbei stellen nun wiederum Verfahren der 3D-Computersimulation, wie sie in der VR zum Einsatz kommen, ein äußerst leistungsfähiges Forschungswerkzeug dar. Virtuelle Realitäten erscheinen hierbei nicht als Nutzanwendung kommunikationspsychologischen Grundlagenwissens, sondern

vielmehr als experimentalpsychologische Methodenplattform zur Generierung dieser Wissensbasis." ([BO96], S.230)

Entscheidend für die Bedeutung eines gestischen oder mimischen Ausdrucks ist nicht die Intention des Ausdrückenden, sondern die Wirkung auf den Betrachtenden. In diesem Kapitel wird auch für die Analyseebenen der nonverbalen Kommunikation die Dreiteilung der Semiotik in Syntaktik, Semantik und Pragmatik verwendet (Abschnitt 3.1.1). Abschnitt 5.1.1 behandelt syntaktische, semantische und pragmatische Funktionen der Mimik, Abschnitt 5.1.2 die der Gestik.

5.1.1 Mimik

Mit dem *Facial Action Coding System (FACS)* [EF78] steht ein Verfahren zur Mimikcodierung zur Verfügung, welches Gesichtsausdrücken den hervorrufenden Muskelgruppen zuordnet. Ekman und Friesen haben hierfür durch umfangreiche Analysen zeitreihenbasierter Datenprotokolle 46 Muskelgruppen (Tabelle 5.1), sogenannte *Action Units (AU)*, für das Gesicht identifiziert, mit denen alle unterscheidbaren menschlichen Gesichtsausdrücke codiert werden können. Eine *Action Unit* ist eine "anatomisch abgeleitete minimale Bewegungseinheit"([Ekm88], S.184). Ca. 55000 unterscheidbare Gesichtsausdrücke wurden katalogisiert. Auf der Ebene erkennbarer semantischer Unterschiede wurden diese in 30 Gesichtsausdrücken geteilt. (vgl. [EF75] und [EF77])

AU Nummer	**Name in FACS**	**Muskuläre Grundlage**
1	Inner brow raiser	Frontalis, pars medialis
2	Outer brow raiser	Frontalis, pars lateralis
4	Brow lowerer	Depressor glaballae; depresseor supercilii
5	Upper lid raiser	Levator palpebrae superioris
27	Mouth streches	pterygoids; digastric
42	Eyes slit	Orbicularis oculi
43	Eyes close	Relaxation of *levator palpebrae superioris*
45	Blink	Relaxation of *levator palpebrae* and contraction of *orbicularis oculi, pars palpebralis*
46	Wink	*Orbicularis oculi*

Tabelle 5.1: Auswahl einiger *Action Units* des FACS

Quelle: [Ekm88], S.191

[Ekm88] empfiehlt vier Schritte zur Codierung eines Gesichtsausdruckes bzw. einer Gesichtsbewegung:

- Festlegen der beteiligten *Action Units*,

- Codierung der Bewegungsintensität der einzelnen AUs in drei Stufen: niedrig, mittel und hoch,

- Klassifizierung der Bewegung in einseitig, beidseitig und asymmetrisch,

- Codierung der Kopf- und Augenstellung während der Gesichtsbewegung.

Eine ausführliche Beschreibung dieses Verfahrens gibt [Ekm88].

Funktionen der Mimik

Interessant für diese Arbeit ist neben der reinen Codierung von Gesichtsausdrücken und - bewegungen vor allem die Wechselwirkungen mit der verbalen Sprache. Die kommunikative Rolle von Gesichtsausdrücken kann anhand der Bedeutung in der multimodalen Informationsübertragung klassifiziert werden [Sch80]. [CPB+94] unterscheidet *syntaktische Funktionen*, *semantische Funktionen* und *Dialogfunktionen* der Gesichtsausdrücke:

- Gesichtsbewegungen erfüllen *syntaktische Funktionen*, die den Sprachfluß begleiten und auf verbaler Ebene synchronisiert werden. Augenbrauenheben oder Kopfnicken betonen Silben oder Pausen.

- Die *semantischen Funktionen* von Gesichtsausdrücken können das Gesagte unterstreichen, Worte ersetzen oder Gefühle ausdrücken.

- Mimische Äußerungen können auch *Dialogfunktionen* übernehmen. Eine Begrüßung kann z.B. durch Kopfnicken oder Heben der Augenbrauen ohne verbale Äußerung stattfinden. Die Bedeutung wird im Kontext der Äußerung interpretiert und ist somit pragmatischer Natur.

Diese drei Funktionen der Mimik werden durch die folgenden Parameter beeinflußt [CPB+94]:

- *Sprecher- und zuhörerspezifische Eigenschaften* wie soziale Identität, Gefühlszustand, Einstellung, Alter etc. werden sowohl in syntaktischen und semantischen Funktionen als auch in den Dialogfunktionen der Mimik codiert. So schauen sich Freunde im Gespräch z.B. häufiger an, während der Sprecher bei der Äußerung einer Lüge den Blick seines Zuhörers meist vermeidet.

- Speziell der Zuhörer codiert seine *Reaktionen* (Übereinstimmung, Aufmerksamkeit, Interesse, Verständnis etc.) auf das soeben vom Sprecher geäußerte in seiner Mimik.

In [NT94b] wird zwischen syntaktischer Mimik, Sprecher- und Zuhörermimik unterschieden. Abbildung 5.1 zeigt die Gesichtsausdrücke, die in der in [NT94b] beschriebenen Implementierung benutzt werden.

Syntactic Display	
1. Exclamation mark	Eyebrow raising
2. Question mark	Eyebrow raising or lowering
3. Emphasizer	Eyebrow raising or lowering
4. Underliner	Longer eyebrow raising
5. Punctuation	Eyebrow movement
6. End of an utterance	Eyebrow raising
7. Beginning of a story	Eyebrow raising
8. Story continuation	Avoid eye contact
9. End of a story	Eye contact
Speaker Display	
10. Thinking/Remembering	Eyebrow raising or lowering, closing the eyes, pulling back one mouth side
11. Facial shrug: "I don't know"	Eyebrow flashes, mouth corners pulled down, mouth corners pulled back
12. Interactive: "You know?"	Eyebrow raising
13. Metacommunicative: Indication of sarcasm or joke	Eyebrow raising and looking up and off
14. "Yes"	Eyebrow actions
15. "No"	Eyebrow actions
15. "Not"	Eyebrow actions
17. "But"	Eyebrow actions
Listener Comment Display	
18. Backchannel: Indication of attendance	Eyebrow raising, mouth corners turned down
19. Indication of loudness	Eyebrows drawn to center
Understanding levels	
20. Confident	Eyebrow raising, head nod
21. Moderately confident	Eyebrow raising
22. Not confident	Eyebrow lowering
23. "Yes"	Eyebrow raising
Evaluation of utterances	
24. Agreement	Eyebrow raising
25. Request for more information	Eyebrow raising
26. Incredulity	Longer eyebrow raising

Abbildung 5.1: Kommunikative Gesichtsausdrücke

Quelle: [NT94b]

5.1.2 Gestik

Ähnlich wie die Codierung von Gesichtsausdrücken sind Körperbewegungen untersucht worden. Ein Verfahren zur Notation von Körperbewegungen ist das *Berner System der Zeitreihennotation für Bewegungen*. Auf Grundlage dieses Gestikcodes werden in der Kommunikationspsychologie für empirische Studien Körperbewegungen in Computermodellen simuliert.

Eine weitere Untersuchung von Körperbewegungen wurde von [CO71] durchgeführt. Hier werden elementare Körperbewegungen und deren Kontrolle durch *Muscle Action Units* charakterisiert. Ein interessantes Ergebnis dieser Studie ist, daß Körperbewegungen neben den internen Synchronisationsregeln auch von gleichzeitig stattfindender Sprache beeinflußt werden, sowohl beim Sprecher als auch beim Zuhörer.

[McN92] schätzt, daß 90 % der Gesten während des Sprechens stattfinden. Die

folgenden vier Typen von Gesten geschehen nur in Verbindung mit verbaler Sprache und sind in jedem Kulturkreis wiederzufinden [CPB+94]:

- *Ikonisierung*:
  Der Sprecher ikonisiert das Objekt aktuelle Objekt. Zur Frage: "Hast Du ein Blatt Papier?" skizziert er z.B. die rechteckigen Umrisse.

- *Metaphorische Gesten*:
  Metaphorische Gesten repräsentieren eine abstrakte Eigenschaft des Gesprächsthemas.

- *Deiktische (zeigende) Gesten*: Zeigende Gesten spezifizieren z.B. ein Objekt ("Dieses Blatt Papier dort!").

- *Rythmische Schläge mit den Händen*: Rythmische Schläge mit den Händen begleiten stark betonte Worte.

Gesten orientieren sich an den Informationen der verbalen Äußerung, die für den Benutzer neu sind. Ikonisierende Gesten werden häufig zur Erläuterung verbaler Ausdrücke verwendet, die nicht dem täglichen Sprachgebrauch entstammen (vgl. [Bri]).

Gestik und verbale Sprache

Die beiden Kommunikationsmedien Gestik und verbale Sprache liefern auf semantischer und pragmatischer Ebene nicht immer die gleiche Information über ein gedankliches Konstrukt, wie dies z.B. bei Mundbewegungen und verbaler Sprache der Fall ist (Abschnitt 5.2.4). Gestik und verbale Sprache ergänzen sich vielmehr in den Beschreibungen der zu übermittelnden Idee.

Auf der lokalsten Ebene werden einzelnen Gesten phonologisch synchronisiert. So passiert z.B. der "Schlag", der energischste Teil der Geste, entweder parallel, oder kurz vor der phonologisch deutlichsten Silbe des begleitenden Sprachsegmentes (vgl. [CPB+94] [Ken80] [McN92]). Auf der globalsten Ebene enden die Handgestiken am Ende einer Erzählsequenz. In der mittleren Ebene wird die Gestik zur Betonung und Erläuterung des Gesagten eingesetzt.

5.1.3 Mehrdimensionalität der Gestik und Mimik

Syntaktische Regeln, die Phoneme und Buchstaben zu Worten (Abschnitt 3.1.2), Lexeme zu Sätzen (Abschnitt 3.1.3), Äußerungen von Gesprächspartnern zu Dialogen (Abschnitt 3.2.2) oder Episoden zu Geschichten (Abschnitt 3.2.3) kombinieren, enthalten ausschließlich Zuordnungsvorschriften für zeitliche Abfolgen. Das Codierungsproblem von Gestik und Mimik ist räumlicher und zeitlicher Natur, also mehrdimensional. Die Anzahl der Dimensionen wird durch die Detaillierungsgenauigkeit der Analyse bestimmt. Mehrere *Action Units* können gleichzeitig relevant sein und einen Gesichtsausdruck bilden. Für die verbalen Einheiten gilt die Ausschlußbedingung: Nur eins zu einem Zeitpunkt. Das FACS kann man z.B. auch zur statischen Codierung von Gesichtsausdrücken in Standbildern verwenden. Der Schwerpunkt

des FACS liegt aber in der - räumlichen *und* zeitlichen - Codierung von Gesichtsbewegungen (vgl. [Ekm88], S.200).

5.1.4 Nonverbales Verhaltensrepertoire von Zeichentrickfiguren

Virtuelle Akteure haben zusätzlich zum nonverbalen Verhaltensrepertoire des Menschen die Ausdrucksmöglichkeiten der Zeichentrickfiguren. Die Menge der zu codierenden Ausdrucksformen vergrößert sich dadurch. Dieser Sachverhalt soll hier am Beipiel der Gesichtsausdrücke verdeutlicht werden.

Beipiel: Gesichtsausdrücke von Zeichentrickfiguren

Neben den Gesichtsausdrücken, die Menschen erzeugen können, besitzen Zeichentrickfiguren mehr Ausdrucksmöglichkeiten. Der Umfang der Ausdrucksmöglichkeiten ist vom Charakter der Figur abhängig. So können bei einem Hund z.B. die Ohren in die Gestaltung der Gesichtsausdrücke mit einbezogen werden. Abbildung 5.2 zeigt einen Teil der Gesichtsausdrücke einer Zeichentrickfigur.

5.1.5 Grundemotionen

Gestik, Mimik und Körperhaltung sind als Medium der emotionalen Kommunikation von Bedeutung. Das Verhalten eines Gesprächsteilnehmers erzeugt bei seinem Gegenüber Emotionen (Interesse, Heiterkeit, Verachtung etc.). Emotionen können als pragmatische Interpretation gedeutet werden.

Die Forschungsergebnisse der Emotionspsychologie liefern erste Fakten für die Erstellung eines nonverbalen Lexikons: [Ekm88] und [Iza91] geben für emotionale Zustände Ausdrucksbeschreibungen an. [Iza91] unterscheidet die folgenden neun Grundemotionen:

- Interesse
- Freude
- Überraschung
- Kummer
- Zorn
- Ekel
- Geringschätzung
- Furcht
- Scham

Die Ausdrucksbeschreibungen von Grundemotionen können aus Sicht der Kommunikationspsychologie als Klassifizierung von Gesichtsausdrücken nach ihrer emotionalen Bedeutung interpretiert werden. Eine solche Zuordnung von Gesichtsausdrücken und den dargestellten Emotionen könnte als Eintrag in ein nonverbales Lexikon verwendet werden.

Abbildung 5.2: Gesichtsausdrücke einer Zeichentrickfigur

Quelle: [Bla94], S. 24 f.

5.1.6 Charakteranimation und Kommunikationspsychologie

Die komplexen Gesetzmäßigkeiten nonverbaler Sprache sind weitestgehend unerforscht. Die Kommunikationspsychologie leistet mit der Erforschung des nonverbalen Verhaltens das, was die Linguistik auf dem Gebiet der verbalen Sprache mit Lexika und Grammatiken größtenteils erreicht hat. Sie erstellt Regelwerke der nonverbalen Sprache. In der nonverbalen Kommunikationforschung wird zur Zeit intensiv an der "Definition eines nonverbalen Verhaltensrepertoires und der Bereitstellung spezifischer Displayregeln" ([BO96], S.230) gearbeitet.

Ähnliche nicht minder zu bewertende Schwierigkeiten treten bei der Analyse der von einem Animator während der Erstellung von Figur und Animationssequenzen intuitiv benutzten Mechanismen auf. Zwei unterschiedliche Wissenschaftsgebiete

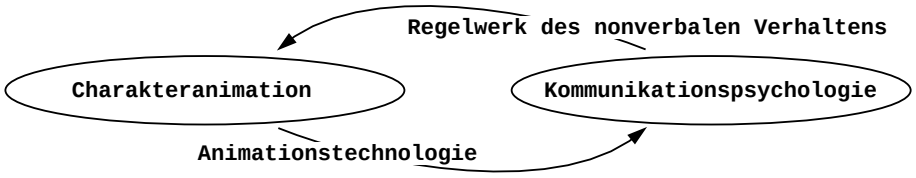


Abbildung 5.3: Wechselwirkung zwischen Charakteranimation und Kommunikationspsychologie

haben dabei ähnliche, sich teilweise ergänzende Aufgaben:

- Die nonverbale Kommunikationspsychologie beschäftigt sich mit der Decodierung der in Mimik, Gestik und Körperhaltung verschlüsselten Information. Zunehmend werden in diesem Wissenschaftsgebiet Computersimulationen menschlichen Verhaltens für empirische Studien verwendet, in denen die Wirkung der synthetisch erzeugten mimischen und gestischen Bewegungen auf Versuchspersonen untersucht wird.

- Die 3D-Charakteranimation geht den entgegengesetzten Weg: Der Animator hat eine bestimmte Intention und codiert diese in der Figur. Er benutzt dabei die innerhalb eines Kulturkreises implizit vorhandene, historisch entstandene (und zunehmend durch audiovisuelle Medien beeinflußte) semantische Übereinkunft der nonverbaler Sprache. Für autonom agierende Charaktere müssen die Kreativ-Mechanismen eines Animators in Hard- oder Software implementiert werden.

Die Mechanismen menschlicher Kognitionsprozesse zur Entschlüsselung der in einem 3D-Charakter codierten Information sind nicht ausreichend formalisiert. Das Handwerkszeug der Linguistik sind Buchstaben - Zeichen, in denen verbale Sprache codiert ist. Als Transkriptionsverfahren der nonverbalen Sprache existiert für die Mimik FACS (Facial Action Coding System) [EF78] (Abschnitt 5.1.1) und für die Gestik das Berner Zeitreihennotationssystem [FHPD81] (Abschnitt 5.1.2). Die Psychologie nutzt bei der Anwendung dieser Verfahren zunehmend die Technologien der 3D-Charakteranimation für empirische Untersuchungen. Umgekehrt können die Ergebnisse bei der Entwicklung und Animation von synthetischen Charakteren hilfreich sein.

Ein Animator kann zwar fehlende Regeln für die Gestik- und Mimikanimation durch seine Intuition ersetzen, autonom agierende Charaktere sind jedoch auf ein implementiertes Regelwerk angewiesen. Charakteranimation und Kommunikationspsychologie ergänzen sich gegenseitig (Abbildung 5.3).

Es sei hier noch erwähnt, daß sowohl in der Kommunikationspsychologie als auch in der Charakteranimation nicht die Erforschung bzw. Simulation mentaler Zustände die Zielsetzung ist. In beiden Wissenschaftsgebieten ist die Wirkung auf den Betrachter von Interesse.

5.2 Sprachsynchrones nonverbales Verhalten

Um den emotionalen Kommunikationskanal effektiv einsetzen zu können, muß der virtuelle Akteur konsistent agieren - in Inhalt und Form. Für die audiovisuelle Darbietung der Lehrinhalte bedeutet dies: Synchronisation von verbalem und nonverbalem Kommunikationskanal. Die Generierung von Bewegungsdaten verlangt eine intensive Analyse des Zusammenspiels zwischen verbaler und nonverbaler Kommunikation beim Menschen. Handelt es sich bei dem Animationsobjekt um eine Phantasiefigur, kommen noch die verbalen und nonverbalen Interaktionsmuster der Trickfilmanimation hinzu. Aus den Ergebnissen der Analyse müssen allgemeingültige Regeln aufgestellt werden, die zur Steuerung der Gesichts- und Körperbewegungen implementiert werden können. Die Wirkung auf den Benutzer des autonomen 3D-Charakters ist maßgebend für die Bewertung des Animationsmoduls.

In den folgenden Abschnitten wird aus bekannten - zumeist heuristischen - Gesetzmäßigkeiten menschlicher Kommunikation eine Regelbasis zur sprachsynchronen Gesichts- und Körperanimation von virtuellen Charakteren aufgestellt. Ausgangsbasis ist ein bereits vorhandener Antworttext, der von einem virtuellen Akteur dargeboten wird. Hierbei werden syntaktische, semantische und pragmatische Regeln gesucht, die die nonvokal-nonverbale Darstellung mit der vokal-verbalen synchronisieren.

5.2.1 Synchronisationsmodell

Im Rahmen der vorliegenden Arbeit wurde ein theoretisches Modell entwickelt, an dem sich Konzeption und Implementierung orientieren. Abbildung 5.4 stellt schematisch einen Überblick dar. Dieses Modell lehnt sich an die Sprachtheorie von Chomsky (Abschnitt 3.1.2) an und erweitert das in Abbildung 3.2 dargestellte Modell um die Bereiche der nonverbalen Sprache. Es wird hierbei von der Annahme ausgegangen, daß die Synchronisation von verbaler Sprache und nonverbalem Verhalten auf Gesetzmäßigkeiten der Phonologie, Syntax, Semantik und Pragmatik beruht. Die "?" in diesem Schema (Abbildung 5.4) kennzeichnen die nicht erforschten Gebiete. Die Aufgabe der Charakteranimation ist es, die heuristischen Regeln in einem autonomen 3D-Charakter zu implementieren und in Experimenten zu verifizieren.

Charakteristische Mundstellungen ausgewählter Phoneme oder rhythmische Handschläge auf den Vokalen betonter Worte sind Beipiele für phonembasierte Synchronisationsregeln. Auf der Ebene der Syntax sind u.a. Satzmelodie und Beenden der Handgestiken am Satzende zu nennen.

5.2.2 Koordinationsraum

Für die Wechselwirkungen zwischen Gestik, Mimik, Körperhaltung und Sprache gelten Gesetzmäßigkeiten, die zum Teil klar ersichtlich und einfach sind. Der weitaus größere Bereich ist jedoch subtil, komplex und dadurch nur durch aufwendige empirische Versuche aufzustellen. Abbildung 5.5 setzt die Freiheitsgrade des nonverbalen Verhaltens mit den verbalen Sprachelementen in Beziehung. Es wird bei diesem Synchronisationsmodell davon ausgegangen, daß die verbale Sprache die begleiten-

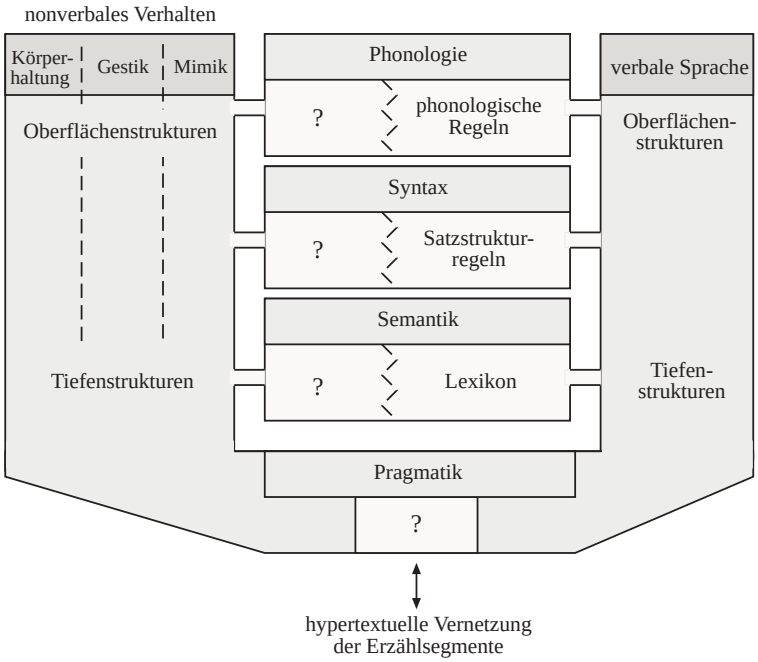


Abbildung 5.4: Synchronisationsmodell für verbales und nonverbales Verhalten eines autonomen 3D-Charakters

de Gestik und Mimik determiniert. Die Sprachelemente werden deshalb als Steuerungsvariablen sv_i bezeichnet. Die Bewegungseffektoren der Mimik und Gestik als Animationsvariablen av_j (Mimik) bzw. av_k (Gestik) bezeichnet. Beispiele für die s Steuervariablen sv_i sind "Phonem", "Wort", "Ausdruck", "Satz", "Episode", "Geschichte" etc. Die m Freiheitsgrade der Figur werden in

- n Bewegungseffektoren av_j für die Mimik ("Mund_auf_au", "Mund_breit_schmal", "linkes_Auge_auf" etc.) und
- $m-n$ Bewegungseffektoren av_k für die Gestik ("linker_Arm_hoch", "linken_Ellenbogen_knicken" etc.) aufgeteilt.

Jedes Tripel (sv_i, av_j, av_k) steht für eine Synchronisationsregel. Dieses Denkmodell kann sowohl für die Analyse der Benutzeräußerungen als auch zur Generierung der Systemreaktion angewandt werden. Der Analyseaspekt wird in dieser Arbeit nicht betrachtet. Die heuristischen Synchronisationsregeln aus Abschnitt 5.1 sind Koordinatenpunkte dieses Modells.

Beispiel: Ausdruck von Unwissenheit

Für den gestischen und den mimischen Ausdruck von *Unwissenheit* kann eine Synchronisationsregel zwischen *Schulterzucken* und bestimmten Mundbewegungen aufgestellt werden. Die Synchronisation Unwissenheit-Schulterzucken ist semantischer Natur. In Kombination mit unterschiedlichen mimischen Ausdrücken können pragmatische Regeln aufgestellt werden. So symbolisieren aufgeblähte Backen ein spon-

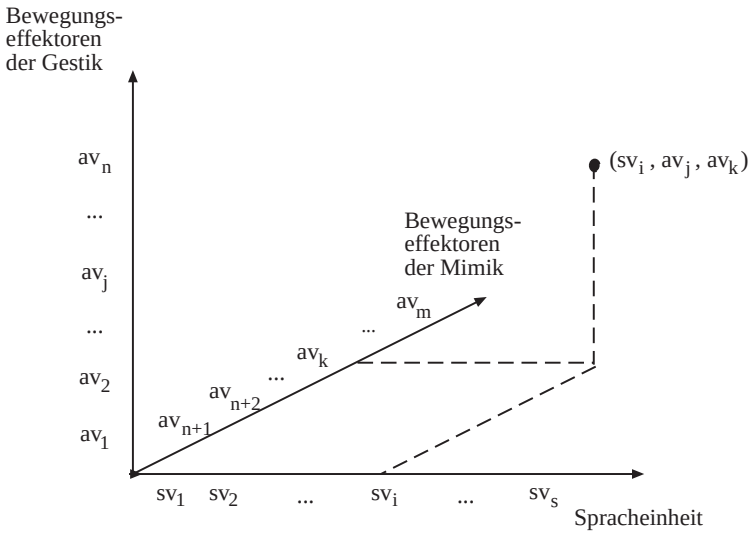


Abbildung 5.5: Gegenüberstellung von Gestik, Mimik und verbalen Sprachelementen

tan erkannte Unwissenheit, Kopfschüteln eine auf Unwissenheit beruhende verneinende Antwort auf eine Frage wie etwa "Kannst Du mir sagen ... ?".

Aus dem Beipiel "Ausdruck von Unwissenheit" wird klar, daß dieses Gedankenmodell nur eine Approximation an die real existierenden Synchronisationsregeln der Ausdrucksformen menschlicher Kommunikation sein kann.

5.2.3 Änderungszyklen linguistischer Einheiten

Die linguistischen Einheiten haben als diskrete Steuerungsvariablen sv_i die Eigenschaft, daß die Ausprägung sich periodisch ändert. Als Periodendauer kann für jede Spracheinheit ein Durchschnittswert angegeben werden. Abbildung 5.6 stellt diesen Sachverhalt am Beispiel der acht Steuerungsvariablen Phonem, Silbe, Ausdruck, Satzteil, Satz, Episode und Geschichte. Für jede Spracheinheit kann eine mittlere Taktfrequenz tf angegeben werden. Der Zustand der Steuerungsvariablen "Phonem" ändert sich im Millisekundenbereich, der eines Wortes im Zehntelsekundenbereich. Eine typische Episode der Geschichte dauert mehrere Minuten. Es stehen somit zur Kontrolle der Animationsvariablen av_j und av_k diskrete Steuerungsvariablen sv_i mit unterschiedlichen Grundfrequenzen zur Verfügung. Aus dieser Konstellation wird im Abschnitt 5.2.4 ein theoretischer Steuerungsmechanismus entwickelt.

5.2.4 Steuerung der sprachsynchronen Animation

Dem Benutzer soll ein Antworttext in einer sprachsynchronen Animation von Gestik, Mimik und Körperhaltung dargeboten werden. Der hierfür notwendige Regelmechanismus setzt den textuellen Input in verbale und nonverbale Kommunikationsformen um. Eine abstrakte Darstellung einer Steuerung zur Umsetzung von Text in nonverbale Sprache ist in Abbildung 5.7 dargestellt.

Zu einem bestimmten Zeitpunkt befindet sich die Animation an einer bestimm-

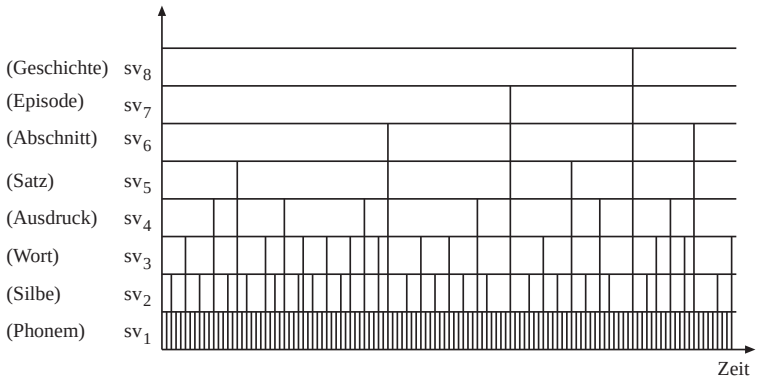


Abbildung 5.6: Spracheinheiten mit unterschiedlichen Änderungszyklen

ten Stelle des zu animierenden Textes. Zur Berechnung der Animationsdaten stehen u.a. Phonem, Morphem und Wort sowie Ausdruck und Satz, in dem das Wort geäußert wird, als diskrete Steuerungsvariable sv_i zur Verfügung. Der Kontext der Episode ist auch von Bedeutung und kann in die Steuerung der Animation mit einfließen. Der Einfluß der verbalsprachlichen Steuerungsvariablen sv_i auf die Animationskurven ist unterschiedlich groß. Die Gewichtungsfaktoren r_{ji} reflektieren die Stärke der Beeinflußung in Form von Zahlenwerten. Die Steuerungsvariablen sv_i beeinflußen nicht nur die Animationskurve, sondern auch die Animationsregeln anderer Steuerungsvariablen. So wechseln sich z.B. die Emotionen *Angst* und *Interesse* bei gleichzeitigem Auftreten in periodischen Abständen ab (vgl. [Bau96], S. 73). Angst kann auch den Sprechfluß unterbrechen. Die Gewichtungsfaktoren r_{ji} sind also keine Konstanten, sondern endogene Transformationsvariablen. Es sind für diese endogenen Variablen Funktionvorschriften der Form

$$r_{ji} = f_j(sv_1, sv_2, \ ... \ sv_i, \ ... \ sv_s)$$

denkbar, die diese gegenseitige Beeinflußung der linguistischen Eingabeparameter formalisieren. Diese Wechselbeziehungen sind jedoch nicht hinreichend genau untersucht worden, als daß eine solche Funktion explizit angegeben werden könnte (Abschnitt 5.1).

Der Definitionsbereich der Funktionen f_j ergibt sich aus den möglichen Ausprägungen der Steuerungsvariablen sv_i (Phonem, Silbe, Morphem, Wort, Ausdruck, Satz, Episode, Geschichte etc.). Die Ausprägungen der Steuerungsvariablen sv_i sind z.B. alle Phoneme einer Sprache, alle Silben einer Sprache etc. Aufgrund der kombinatorischen Explosion der Ausprägungsanzahl der komplexeren Elemente sind (ab der Ebene des Wortes) Klassifizierungen empfehlenswert. Bei Worten sind z.B. grammatikalische (Adjektiv, Substantiv etc.) oder semantische Klassifizierungen sinnvoll. Letzteres gilt auch für Ausdrücke oder Sätze. Bei Episoden oder Textabschnitten bieten sich die pragmatische Funktionen wie Einleitung, Überleitung, Haupteil, Schluß an.

Für jede Steuerungsvariable sv_i ist ein entsprechendes Lexikon zu implementieren, die den einzelnen Ausprägungen Animationsregeln zuordnen. Diese Animationsregeln bestimmen aus den Eingabedaten die konkreten Werte r_{ji}.

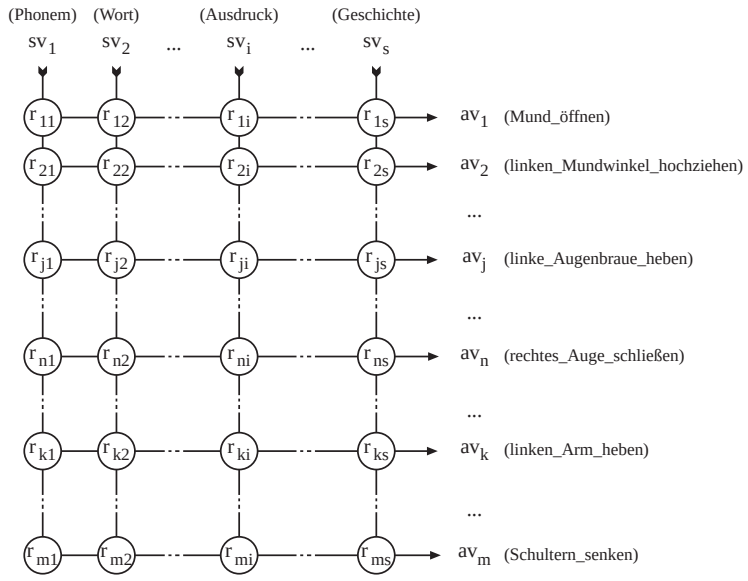


Abbildung 5.7: Umsetzung der verbalen Steuerungsvariablen auf die Animationsvariablen

Der in diesem Abschnitt dargestellte Steuerungsmechansimus soll anhand der Synchronisation von Mundbewegungen mit der Sprache verdeutlicht werden.

Beispiel: Synchronisation von Mundbewegungen und Sprache

Beim Sprechen bewegt sich der Mund nach Regeln der Phonetik. Die Mundbewegung des Sprechers werden vom Zuhörer zum Entschlüsseln der audiblen Informationen unterstützend benutzt. Ist der Kontext klar umrissen und wird die Kommunikation durch Gestik und Mimik unterstützt, können einzelne Wort oder Sätze auch ohne die akustische Information, nur durch Ablesen von den Lippen, entschlüsselt und verstanden werden. Mundbewegungen könne also auch verbal interpretiert werden. Manche Gehörlose sind in der Lage, Sprache von den Lippen abzulesen, was darauf schließen läßt, daß zum Verständnis ausreichend Informationen der gesprochenen Phoneme, Wörter und Sätze durch die Mundbewegung übertragen wird.

So ist z.B. der Mund bei dem Vokal “a” geöffnet. Ein Wort definiert als Phonemfolge eine Funktionsvorschrift

$$f_{Mund}(t, r_{Phonem,Mund}, r_{Morphem,Mund}, ...r_{Episode,Mund}, r_{Geschichte,Mund})$$

für eine Animationskurve mit n Freiheitsgraden, die die Mundöffnung steuert (t ist eine Variable der zeitlichen Dimension). Die Gewichtungsparameter r_{ji} bestimmen leichte Veränderungen der Animationskurve für dieses Wort in Abhängigkeit von der semantischen Bedeutung im Satzzusammenhang und der pragmatischen Interpretation des Wortes im Kontext. r_{ji} stellen kontextuelle Betonungsregeln dar.

Mundbewegungen werden in erster Linie auf Phonembasis mit dem Sprachsignal synchronisiert. Hierfür existieren phonologische Regeln, welche die Vokale (a, e, i, o,

u), Umlaute (ä, ö, ü) und Schließlaute (b, m, p) betreffen. In Abschnitt 7.2.6 wird dieser Zusammenhang näher erläutert. Das Tripel

$$[sv_i, av_j, av_k] = [Phonem, Mundbewegung, Handbewegung]$$

beschreibt einen Punkt in dem von den Dimensionen *Bewegungseffektor der Mimik*, *Bewegungseffektor der Gestik* und *linguistische Einheit* aufgespannten Raum (Abbildung 5.5).

[CPB+94] ordnen den Phonemen charakteristische Mundformen mit Freiheitsgraden in der Verformbarkeit zu. Die zeitliche und räumliche Verformung der einzelnen Mundformen wird dem Kontext angepaßt.

So könnte zum Beispiel die Animationsregeln $rule_{ji}$ mit sv_{Phonem} = "a"lauten:

$$rule_{Phonem,Mund} = \text{“auf”} \text{ und}$$
$$rule_{Phonem,av_j} = \text{“neutral”}$$

$$\forall \, \mathrm{j} \in \mathbb{N}^{+} | av_j \in \{\text{Mund, Auge, Hände, Arme, Bein, ... }\}$$

Jede Animationsvariable av_j der Figur interpretiert den Wert der Animationsregel $rule_{ji}$ individuell. So wird z.B. $rule_{ji}$ = neutralïn $r_{ji} = 0$ übersetzt, wenn r_{ji} additiv in der Funktionsvorschrift für av_j verknüpft ist. Bei einer multiplikativen Verknüpfung wäre eine Übersetzung in $r_{ji} = 1$ sinnvoll. Für den Mund wäre z.B. eine Animationfunktion in Form von

$$av_{Mund} = r_{Phonem,Mund} * (r_{Silbe,Mund} + r_{Wort,Mund} + r_{Episode,Mund} + r_{Geschichte,Mund})$$

denkbar. Da die Mundanimation eng mit der vokale Sprache verbunden ist, sind spezielle Verknüpfungsregeln mit den Variablen zu beachten, die die Sprachsynthese beeinflußen.

Wird die in Abschnitt 5.2.1 verwendete Dreiteilung in Syntaktik, Semantik und Pragmatik für dieses Steuerungsmodell angewandt, so läßt sich mit der Vergröberung der linguistischen Einheit eine Verschiebung von der Syntaktik zur Semantik und Pragmatik vermuten.

Das hier vorgestellte Verfahren zur Synchronisierung eines Antworttextes mit der Animation der Gesichts- und Körperbewegungen vernachlässigt, daß der Antworttext selbst schon eine Umsetzung von Inhalt in Form darstellt. So beeinflußt zum Beispiel der sich im Kontext ergebende Gefühlszustand den Text selbst genauso wie dessen Darbietung durch Sprachsynthese und Animation. Die Produktion des Textes ist also nicht zwangsläufig als Zwischenschritt zu implementieren, sondern in einem parallel verlaufenden Prozeß mit Wechselwirkungen zwischen verbaler und nonverbaler Darbietung.

5.2.5 Vorproduzierte Animationsdaten

Die Ausführungen in Abschnitt 5.2 basieren auf der Annahme, daß die Animationsdaten während der Interaktion mit dem Benutzer anhand der in dem Animationsmodul implementierten Regelbasis berechnet werden. Wie in Abschnitt 5.1 erwähnt, ist eine umfangreiche, empirisch fundierte Regelbasis nicht vorhanden.

Die Verwendung einer Bibliothek von Animationssequenzen stellt eine alternative Möglichkeit zur Echtzeitgenerierung der Animationsdaten dar. Die Animationssequenzen können von einem Animator vorproduziert werden. Hierbei stellt sich die Frage, ob die Animationssequenzen fest mit einem bestimmten Text verbunden sind oder ob das Animationsmodul zur Laufzeit anhand implementierter Regeln einem Text eine Animationssequenz zuordnet. Letzteres würde z.B. zur sprachsynchronen Lippenanimation Modifikationen der Animationsdaten erfordern. Ersteres hat einerseits Vorteile in der Qualität der Animation, andererseits aber die Verwaltung großer Datenmengen zur Folge.

5.3 Agenten-Paradigma

Synthetische Darsteller, die in der Lage sind, mit dem Benutzer einen natürlichsprachlichen Dialog zu führen, sollten ein gewisses Maß an Autonomie besitzen. Die Fähigkeit, im Auftrag des Benutzers bestimmte Aufgaben auszuführen, macht einen 3D-Charakter zum Agenten. Der Dialog dient dann der Spezifikation der Aufgaben und dazu, dem Benutzer mitzuteilen, was der Agent gerade macht. Der Benutzer sollte durch den Dialog in der Lage sein, die Aufgabenstellung auch dann noch zu ändern, wenn der Agent bereits mit der Bearbeitung begonnen hat. Eine solche Änderung kann die Aufgabenstellung, aufgrund der Informationen, die der Agent liefert, dem neuen Wissensstand anpassen.

Der Begriff *Agent* ist zum Modewort für moderne Softwaretools avanciert. Meist ist ein Programm mit dieser Bezeichnung *intelligent* oder *autonom*. Manche Systeme werden ihrer Namensgebung gerecht, viele jedoch nicht. Welche Kriterien sollte ein Softwareagent erfüllen? [Fon93] schlägt folgende Bewertungskategorien für Agenten vor:

1. *Autonomie*:
   Ein Agent sollte spontane und benutzerunabhängige Reaktionen zeigen.

2. *Personalisierbarkeit*:
   Die Fähigkeit, sich auf seinen Benutzer einzustellen, kann durch Lernkomponenten implementiert werden. Der Benutzer sollte den Agenten nicht explizit programmieren, sondern ihm seine Aufgaben "zeigen". Der Agent kann sich - z.B. durch "Beobachten" der vom Benutzer durchgeführten Tätigkeiten - dem Benutzer anpassen. Er sollte weiterhin in der Lage sein, das Erlernte zu speichern, so daß bei einem erneuten Programmstart der Zustand der letzten Interaktion wieder hergestellt werden kann. Somit kann der Agent bei jedem Lernzyklus auf der Intelligenz des letzten aufbauen.

3. *Kommunikationsfähigkeit*:
   Die dem Agenten übertragene Aufgabe muß vom Benutzer spezifiziert werden. Dies sollte in einem Gespräch geschehen, in dem der Benutzer und sein Agent sich gegenseitig ihre Absichten und Fähigkeiten kundtun. Hierbei sollte in einer Art Aufgabenteilung geklärt werden, wer was erledigt. Es sollte auch in gewissen Abständen Statusmeldungen über den Fortschritt der Tätigkeit gemacht werden.

4. *Risiko und Vertrauen*:
   Dem Agenten wird eine Aufgabe anvertraut. Das Risiko, das hierbei eingegangen wird, muß in Relation zu den Kosten, die ein mögliches Scheitern nach sich ziehen würde, gesetzt werden.

5. *Aufgabenbereich*:
   Die Kosten eines möglichen Versagens des Agenten sind stark abhängig von dem Aufgabenbereich. In der Unterhaltungsindustrie sind die Risiken relativ gering. Eine Flugzeug- oder Kernreaktorsteuerung muß absolut verläßlich sein.

6. *Sanfter Leistungsabfall*:
   Sollte der Agent nicht in der Lage sein, seine Aufgabe vollständig zu erfüllen, so sollte er die Aufgabe zumindest teilweise ausführen. Der Agent kann in einem solchen Fall den Benutzer um Rat fragen. Dies schafft Vertrauen in die Leistungsfähigkeit des Agenten.

7. *Kooperation mit dem Benutzer*:
   Der Agent sollte sowohl bei der Spezifikation als auch bei der Durchführung der Aufgabe Auskünfte über seine Leistungsfähigkeit und eventuelle Zwischenergebnisse erteilen. Das Gespräch dient dazu, die Aufgabe so abzuändern, daß sie vom Agenten erfolgreich durchgeführt werden kann.

8. *Anthropomorphismus*:
   Ein Agent muß nicht zwangsläufig menschähnliche Züge aufweisen. Manche tun es (Eliza, Parry, Julia etc.), andere wiederum nicht (z.B. Agenten zur Sortierung elektronischer Post, die die Benutzergewohnheiten speichern und diese dann nachahmen).

9. *Erwartungshaltung des Benutzers*:
   Agenten sind dort erfolgreich, wo sie die an sie gestellten Erwartungen erfüllen können. Die Erwartungshaltung des Benutzers bestimmt demzufolge das Einsatzgebiet.

In Abschnitt 8.6 wird die Implementierung des im Rahmen der vorliegenden Arbeit entstandenen VISTA-Systems mit diesen Kriterien bewertet.

Autonomie

Das Begriff "autonom" bedeutet wörtlich - aus dem Griechischen übersetzt - "nach eigenen Gesetzen lebend" und wird im allgemeinen Sprachgebrauch in der Bedeutung von "selbständig" und "unabhängig" verwendet [Mül82]. In [CHS94] wird der Autonomiebegriff ausführlich im Spannungsfelde einer minimalistischen und einer maximalistischen Definition diskutiert. Die minimalistische sieht Autonomie als die Fähigkeit eines Systems, sich selbst zu definieren. Es ist somit von einem Referenzsystem (dem Umfeld, Entwickler, Benutzer etc.) nicht kontollierbar. Die maximalistische Definition umfaßt Selbstorganisation. In dieser Arbeit wird die minimalistische Definition von Autonomie verwendet.

5.4 Forschungsprojekte

Autonome virtuelle Akteure werden als anthropomorphe Benutzerschnittstelle oder zu Unterhaltungszwecken entwickelt. Die folgenden Abschnitte beschreiben derzeitige Forschungsprojekte und Entwicklungen auf dem Gebiet der autonomen Charakteranimation.

5.4.1 Peedy - der persönliche Assistent zur Bedienung eines CD-Wechslers

In [BLK+97] ist ein sprechender Papagei namens *Peedy* beschrieben, der die musikalischen Wünsche seines menschlichen Gesprächspartners durch die Steuerung eines CD-Wechslers erfüllt. In diesem Abschnitt wird die Funktionalität und Systemarchitektur von Peedy beschrieben.

Systemarchitektur

Auf jede gesprochene Eingabe reagiert Peedy mit einer Kombination aus visueller und akustischer Ausgabe. Das System kann funktional in drei Teilsysteme aufgeteilt werden:

(1) Der Sprachverarbeitung sind die Module *Whisper*, *Names*, *NLP* und *Semantic* zuzuordnen. Die Spracheingabe wird in eine Ereignisbeschreibung übersetzt.

(2) Das Dialogmanagementmodul *Dialogue* entscheidet, wie der Charakter auf die jeweilige Eingabe reagiert.

(3) Die Video- und Audio-Ausgabemodule *Player/ReActor* und *Speech Controller* generieren die Animationsbewegungen, die Sprache und die Geräusche, die notwendig sind, um mit dem Benutzer in einer lebensähnlichen Art und Weise zu kommunizieren.

Die natürlichsprachlich-akustischen Kommandos des Benutzers werden von dem Spracherkennungssystem *Whisper* aufgenommen und in textuelle Äußerungen umgesetzt. Hierzu wird unter anderem - in Kombination mit einer Namensdatenbank - das Namenssubstitutionsmodul *Names* verwendet, welches ambivalente Bezeichnungen (z.B. Namen von Musikern, Liedern oder Alben) durch dem System bekannte Objekte ersetzt.

Linguistische Analyse

Die textuelle Äußerung wird dem NLP-(Natural Language Processing)-Modul übergeben, welches eine dreistufige linguistische Analyse der Eingabe durchführt. Die drei Schritte werden in [BLK+97] mit *Syntactic Sketch*, *Reassignment of Syntactic Ambiguities* und *Logical Form*.

(1) *Syntactic Sketch*:
In einem syntaktischen Analyseverfahren, welches auf dem Prinzip der *Augmented Phrase Structure Rules* basiert, wird eine syntaktische Skizze der Eingabe erzeugt. Das *Augmented-Phrase-Structure-Rules*-Verfahren ist eine

Bottom-Up-Methode, die die verschiedenen Möglichkeiten einer Alternative parallel in Erwägung zieht.

(2) *Reassignment of Syntactic Ambiguities*:
Hier werden syntaktische Mehrdeutigkeiten mit Hilfe semantischer Informationen aus on-line-Wörterbuchdefinitionen aufgelöst.

(3) *Logical Form*:
Im dritten und letzten Schritt der linguistische Analyse wird schließlich ein semantischer Graph erzeugt, der Prädikat-Argument-Beziehungen repräsentiert. Hierbei werden einzelnen Satzelemente funktionalen Rollen zugewiesen.

Der entstehende Graph codiert die semantische Struktur der englischen Äußerung, wobei jeder Knoten die semantische Ursprungsbedeutung (*Root Form*) eines Eingabewortes repräsentiert. Die Kanten entsprechen den zugehörigen funktionalen Rollen. Die logische Form wird von dem Semantikmodul weiterverarbeitet. Diese benutzt Kenntnisse des Interaktionsszenarios und des Aufgabengebietes zur Durchführung verschiedener Graphtransformationen. Diese anwendungsspezifischen Transformationen berücksichtigen

- allgemeine und alltägliche Sprachartefakte,
- geeignete Sprachinterpretationen im Kontext von benutzerassistierender Konversation,
- aufgabenspezifisches Vokabular,
- umgangssprachliche Ausdrücke und spezielle Grammatikkonstruktionen der *Task Domain* und
- Objektbeschreibungen in der Anwendung

und konvertieren den Graph in eine normalisierte, aufgabenspezifische, semantische Repräsentation. Die anwendungsspezifischen Transformationsregeln übersetzen die sprachbasierte Repräsentation in eine anwendungsspezifisch eindeutige Repräsentation. Das grundlegende Prinzip, nach dem dieser Regeln entwickelt wurden, berücksichtigt die Notwendigkeit alle rechtmäßigen englischen Paraphrasen einer Anfrage auf eine einzelne kanonische Struktur zu reduzieren. Mit dieser kanonischen Form kann Peedy auf eine einzelne, genau spezifizierte Bedeutungsrepräsentation zurückgreifen, während der Benutzer die Freiheit hat, diese Bedeutung in fast jeder denkbaren Weise auszudrücken.

Dialogmanagement

Nachdem der Dialogmanager *Dialogue* von der Sprachverarbeitung eine Eingabebeschreibung erhalten hat, generiert er unter Berücksichtigung der momentanen Dialogsituation Peedys Reaktion, eine geeignete Kombination aus Animation, verbalen Antworten und Anwendungsaktion. Die Dialogsituation hängt ab von dem Konversationszustand und einer Reihe von Kontextvariablen.

Konversationszustand

Der Konversationszustand wird durch einen einfachen begrenzten Automaten dargestellt, der eine Sequenz von Interaktionen modeliert, die während der Konversation auftreten. Für jede Kombination aus Konversationszustand und *Input Event* hat der Automat ein auszuführende Aktion. Zur Zeit besitzt der Automat fünf Konversationszustände und siebzehn *Input Events*, woraus sich ca. hundert verschiedene Transitionsmöglichkeiten ergeben. Jede Transition des Automaten kann Befehle enthalten, die Animationssequenzen oder gesprochene Reaktionen auslösen oder die Zielanwendung, den CD-player, betreiben. Zusätzlich zum Konversationszustand verwaltet der Dialogmanager eine Reihe von Kontextvariablen, die benötigt werden, um Parameter und Objektbeschreibungen zu speichern, die Peedys Verhalten beeinflussen. (vgl. [KL95])

5.4.2 Ein sozialer Agent

[NT94a] implementieren einen sozialen Agenten, der in Gruppendiskussionen mit Menschen soziales Verhalten vorweist. Diese multimodalen Mensch-Maschine-Schnittstelle kann in folgende drei Zielsetzungen zusammengefaßt werden:

(1) Multimodale Konversation wird durch Verknüpfung von natürlicher Sprache, Gesichtsausdrücke etc. erreicht.

(2) Der Computer interagiert in seiner Rolle als autonomer sozialer Agent mit einer Gruppe menschlicher Interaktionspartner.

(3) Der Computer ist ständig bemüht, Kommunikationsfehler zu erkennen und den Benutzern mitzuteilen.

Die Systemkonfiguration ist in Abbildung 5.8 dargestellt. Der soziale Agent wird durch Gesicht und Stimme dargestellt, die von zwei Subsystemen, der Gesichtsanimation (*Facial Animation Subsystem*) und dem Dialogmodul (*Speech Dialogue Subsystem*), gesteuert werden. Das Gesichtsanimationsmodul basiert auf einem Muskelmodel. Das Dialogmodul umfaßt die Spracherkennung, die syntaktische und semantische Analyse, die Erkennung der Intention, die Generierung einer Systemantwort und die Sprachsynthese.

5.4.3 Improvisierende Charaktere in virtuellen Welten

In [HRvGH96] wird der Begriff *Improvisational Character* geprägt: Synthetische Darsteller leben in virtuellen Welten und interagieren untereinander und mit menschlichen Teilnehmern. Nicht die konkreten Handlungen der improvisierenden Charaktere, sondern deren Verhaltensweisen werden implementiert. Einzelne Handlungen ergeben sich aus den jeweiligen Interaktionen. Ein ähnliches Szenario wird von [BLR91] mit dem OZ-Projekt verfolgt.

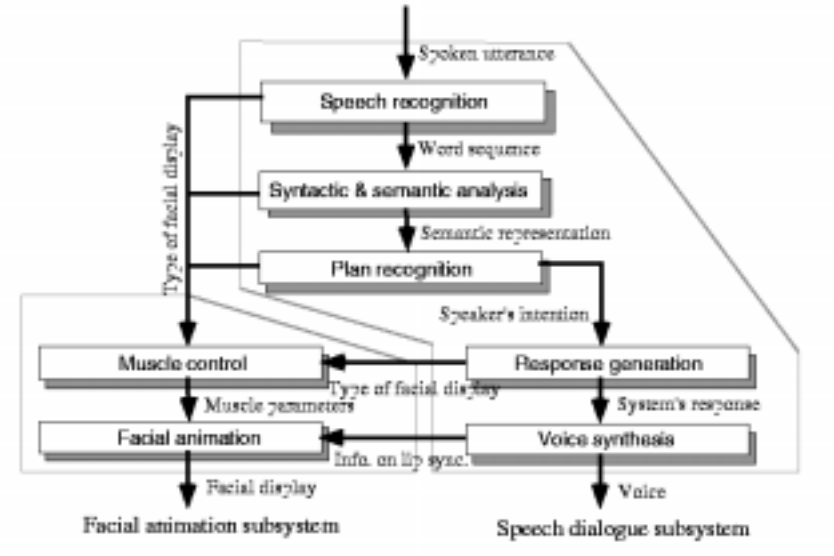


Abbildung 5.8: Systemkonfiguration des sozialen Agenten

Quelle: [NT94b]

5.5 Zusammenfassung

In Abschnitt 5.1 wurden die bekannte Transkriptions- und Codierungsverfahren für nonverbales Verhalten dargestellt. Weiterhin wurden bekannte Regeln zur sprachsynchronen Animation zusammengestellt. Für diesen Bereich gibt die Literatur Heuristiken an. Eine Theorie fehlt. In Abschnitt 5.2.1 ist ein Modell beschrieben, was die Grundlage einer Synchronisationstheorie für verbales und nonverbales Verhalten bildet. Dieses Modell wird mit Hilfe von Versuchen auf Basis der im Rahmen dieser Arbeit entwickelten Experimentierplattform VISTA (Abschnitt 7) zu einer Theorie ausgebaut.

Kapitel 6

Konzeption

In den vorherigen Kapiteln wurden die Regeln aufgezeigt, auf denen verbales und nonverbales Verhalten basiert. Weiterhin wurden theoretische Verfahren und Implementierungen dieses Problemkreises diskutiert. Aus der Menge dieser Möglichkeiten wurde für den im Rahmen der vorliegenden Arbeit implementierten Prototypen VISTA (*Virtual Storytelling Actor*) folgende Ansätze und Verfahren gewählt:

(a) Computational Behaviourism (Abschnitt 3.3.6)

(b) Das Analyseverfahren von Eliza (Abschnitt 3.4.2)

(c) Prinzip der priorisierten Schichten von Miniexperten der Chatterbots (Abschnitt 3.4.4)

(d) Hypertextnavigation (Abschnitt 4.2.1)

(e) 3D-Charakteranimationsverfahren der KHM (Abschnitt 1.1)

Die Grundlage der verbalen Kommunikation bilden die Punkte (a), (b), (c) und (d). Die Verbindung dieser drei Verfahren bildet das im Rahmen dieser Arbeit entwickelte Konzept des *interaktiven Dialogskriptes* (Abschnitt 6.3.1). Für die Verbindung mit (e) wurden Teile des in Abschnitt 5.2.1 dargestellten Synchronisationsmodells implementiert. Hauptbestandteil der Synchronisation bildet die Implementierung eines Emotionszustandes (Abschnitt 6.3.2).

Die in Kapital 7 dargestellte Implementierung integriert die Konzeptionen, die auf folgenden Ebenen erarbeitet wurden:

- Die *inhaltliche Konzeption* verbindet die systeminterne Informationsverarbeitung mit Prinzipien der maschinellen Verarbeitung natürlicher Sprache (Kapitel 3) und der Hypertextstrukturierung (Kapitel 4) von Wissen und Inhalten.
- Die *technologische Konzeption* befaßt sich mit der benutzten Hard- und Software, den Ein- und Ausgabedevices und der Informationsverarbeitung innerhalb des Systems (Datenstrukturen, Prozess- und Modulkommunikation etc.).
- Die *Konzeption zur Synchronisierung verbaler mit nonverbaler Sprache* verbindet die in Kapitel 3 erarbeiteten Grundlagen der Linguistik mit psychologischen Erkenntnissen nonverbalen Verhaltens aus Kapitel 5.1. Die wissenschaftlichen Erkenntnisse auf diesem Gebiet beschränken sich - sowohl in der Charakteranimation als auch in der Psychologie - auf Heuristiken. Auf Grundlage der im Rahmen dieser Arbeit aufgestellten Synchronisationskonzeption

und des implementierten Prototypes soll in den nächsten Jahren eine Theorie erarbeitet werden.

6.1 Anwendungsszenarien

Im Rahmen dieser Arbeit wurden ausgehend von einem virtuellen Fernsehmoderator (Abschnitt 1.1) einer Jugendsendung zwei Szenarien für einen autonomen Dialog mit den Mitgliedern der Zielgruppe entwickelt: der virtuelle Geschichtenerzähler und der virtuelle Pädagoge. Die Konzeption des virtuellen Geschichtenerzählers wurde in dem in Kapitel 7 dargestellten VISTA-(*Virtual Storytelling Actor*)-System implementiert.

6.1.1 Der virtuelle Pädagoge

Die Mitglieder der Zielgruppe bauen zu ihrem "Fernsehstar" eine Beziehung auf, die zur Vermittlung von Lehrinhalten genutzt werden kann. Die Fähigkeit eines virtuellen Akteurs, Emotionen auszudrücken, kann in einem Lehr-/Lernszenario funktionalisiert werden. Dieses Konzept basiert auf den folgenden Punkten:

(a) Durch sich wechselseitig ergänzende verbale und nonverbale Sprache soll die Mensch-Maschine-Kommunikation entscheidend verbessert werden.

(b) Ein anthropomorpher Charakter ermöglicht eine emotionale Beziehung zum Lehrmedium, die für die Vermittlung des Lehrinhaltes funktionalisiert werden kann.

(c) Die Vermittlung des Lehrinhaltes in Dialogform zwingt und motiviert den Lernenden zum aktiven Lernen.

Die in diesem Szenario eingesetzten 3D-Charaktere sind Cartoonfiguren, konzipiert für das Kinder- und Jugendfernsehen. Die Zielgruppe dieser Fernsehsendungen, Kinder im Alter von 8-13 Jahren, sind die potentiellen Benutzer des virtuellen Pädagogen.

Emotionsbasiertes Lernen

Mit Gestik, Mimik und Körperhaltung kann ein 3D-Charakter Emotionen nonverbal effizient kommunizieren. Auf Grundlage der Beziehung, die der Lernende zum virtuellen Pädagogen aufbaut, kann die Emotionsdarstellung gezielt funktionalisiert werden [OW75] [Ipf74]. Der Lernende möchte seinem Lehrer - dem virtuellen Pädagogen - Freude bereiten [SWZ90] [IL79]. Unterbewußtes Ziel des Lernenden ist es also, einen glücklichen Ausdruck im nonverbalen Verhalten des 3D-Charakters zu erzeugen. Dazu muß er eine bestimmte Anzahl von Lektionen erfüllen und auf Kontrollfragen korrekt antworten. Emotionsdarstellungen in Gestik und Mimik können bei der Vermittlung von Lehrinhalten folgende Funktionen erfüllen:

(a) *Motivationsfunktion*: Der unterhaltsame Charakter des virtuellen Akteurs motiviert zum Gespräch und animiert zum Lernen.

(b) *Feedbackfunktion*: Richtige Antworten werden mit glücklicher Mimik belohnt, falsche mit einer wütenden oder traurigen bestraft.

(c) *intermittierende Verstärkungsfunktion*: Der Lerneffekt wird dadurch verstärkt, daß an wichtigen Stellen Spannungsmomente eingebaut werden und der Lehrinhalt so einprägsamer vermittelt wird.

6.1.2 Der virtuelle Geschichtenerzähler

In dem Dialog mit dem virtuellen Geschichtenerzähler geht es um die Präsentation einer Geschichte. Die Eigenschaften dieses Szenarios sind (sortiert nach dem Klassifikationsschema aus Abschnitt 4.2):

- Die interaktive Geschichte entsteht während des Dialoges mit dem Benutzer aus dem halbautonomen Verhalten des virtuellen Akteurs. Die Autonomie der Figur ist durch die nichtlineare Geschichtsstruktur begrenzt.
- Der virtuelle Geschichtenerzähler folgt in seinen Handlungen einem Skript. Freiheitsgrade sind durch die Wahl der Wege in der nichtlinearen Handlungsstruktur und durch das Ausweichen in "Smalltalk" gegeben.
- Handlungsstruktur, "Smalltalk" Themen und 3D-Charakter werden von professionellen Autoren entworfen.
- Der Benutzer interagiert als Gesprächspartner mit dem virtuellen Geschichtenerzähler.
- Der Benutzer wird von den Äußerungen des virtuellen Geschichtenerzählers geführt.

6.2 Anforderungen

Aus den in den vorherigen Kapiteln vorgestellten Verfahren und Systemen wurde eine Interaktionsmethode entwickelt, die auf die Anforderungen eines interaktiven Geschichtenerzählers abgestimmt wurde.

Das VISTA-System muß neben dem langfristig angestrebten praktischen *Einsatz in computerbasierten On- und Offline-Medien* auch für experimentelle Zwecke geeignet sein. Letzteres meint nicht nur die Weiterentwicklung der Technologie, sondern auch die kreative Nutzung der bereitgestellten Software. Aus diesen drei Anwendungsszenarien der bereitgestellten Software ergeben sich folgende Anforderungen:

- Die Interaktion zwischen Benutzer und Figur basiert auf natürlicher Sprache. Der virtuelle Geschichtenerzähler unterstützt die auditive Wiedergabe mit Gestik und Mimik. Der Benutzer ist in der vorliegenden VISTA-Version auf die textuelle Eingabe beschränkt.
- Das Genre des nichtlinearen Geschichtenerzählers verlangt nach einem individualisierten, interaktiven Medium wie dem Internet oder dem digitalen Fernsehen. Zur Zeit ist das Internet sowohl technologisch als auch in der Verbreitung ausgereifter, so daß eine Softwarearchitektur für dieses Medium kurz und mittelfristig sinnvoll erscheint.

- Das System sollte in für den Benutzer akzeptabler Zeit eine Reaktion auf die Benutzereingabe erzeugen.
- Es sollte bei einem Interaktionsverlauf nicht mehrmals die gleiche Reaktion generiert werden - weder textuell noch in bezug auf die Animationssequenz.
- Das System sollte auch auf nicht verstandene Eingaben reagieren.
- Das Skript des Interaktionsdialoges soll für den Autor der Geschichte einfach zu erstellen sein.
- Der Autor muß nachträgliche Änderungen in den Texten des Skriptes einfach vornehmen können.
- Der Autor muß auch nach Skriptentwurf die Topologie des Interaktionsraumes verändern können. Der Interaktionsraum entsteht aus den einzelnen Erzählsegmenten und deren Verbindungen.
- Das System sollte sowohl für den Rezipienten als auch für den Autor benutzerfreundlich gestaltet sein.

6.3 Lösungsansätze

Die Konzepte *interaktives Dialogskript*, *Emotionszustand*, *Wechsel zwischen Erzähl- und Plaudermodus* und *Modifikation vorproduzierter Animationsdaten* stellen die grundlegenden Lösungsansätze dar.

6.3.1 Interaktives Dialogskript

Ein interaktives Dialogskript ist ein Dialogskript, in dem die Rolle eines Gesprächspartners - des späteren Benutzers - zum Zeitpunkt des Skriptschreibens unbekannt ist. Der Skriptautor schreibt eine Äußerung des virtuellen Akteurs, antizipiert mögliche Reaktionen, klassifiziert sie nach ihrer Bedeutung und identifiziert in den einzelnen Kategorien gemeinsame Wortmuster, die den invarianten Teil dieser Kategorie darstellen. Der invariante Teil einer Gruppe von Reaktionen gleicher Bedeutung stellt eine Verbindung in der Hypertextstruktur des Systems dar. Der Autor programmiert dadurch die Analyse der Benutzereingabe und die Hypertextstruktur. Die Analyse ist kontextabhängig, d.h. die analyserelevante Stichwortmenge ändert sich bei jeder Benutzereingabe. Wichtiger Bestandteil eines interaktiven Dialogskriptes sind die durch den Autor antizipierten Benutzerantworten. Der Autor muß herausfinden, was ein typischer Benutzer wahrscheinlich äußern wird, und darin für jede Gruppe von Fragen und Äußerungen, die dieselbe Antwort hervorrufen, ein gemeinsames Muster entdecken (vgl. [Wha96]). Es ist die Aufgabe des Autors, den Rezipienten an den Dialogverlauf zu binden, so daß er die Äußerungen des Rezipienten antizipieren kann. Zum Aufbau der Hypertextstruktur ist im Rahmen dieser Arbeit ein maschinenlesbares Skriptformat entwickelt worden (Abschnitt 7.1.1).

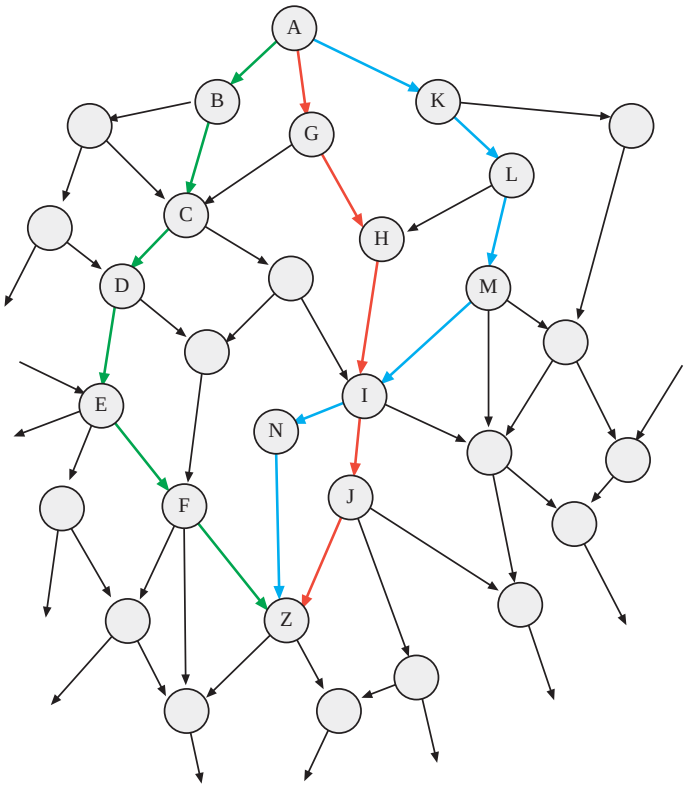


Abbildung 6.1: Aktivierungspfade in einer Hypertextstruktur

6.3.2 Emotionszustand

Der Emotionszustand wird aus den Angaben über die Gefühle und deren Intensität berechnet, die der Autor im Skript bei jedem Erzählsegment macht. Diese Angaben werden jedoch nicht als absoluter Emotionszustand interpretiert sondern als Beitrag des entsprechenden Segmentes zum aktuellen Emotionszustand. Der aktuelle Emotionszustand entwickelt sich aus der Akkumulation der Beiträge aller aktivierten Segmente. Der Einfluß eines einzelnen Segmentes wird mit zunehmender zeitlicher Entfernung vom Segment geringer.

Bei Aktivierung eines Erzählsegmentes verändert sich zunächst der Emotionszustand des virtuellen Darstellers. Hierzu hat der Autor beim Schreiben des Skriptes an jedem Erzählknoten die Gefühle, die die Situation hervorruft, und deren Intensität angegeben. Diese segmentbezogenen Emotionen werden vom Programm als Impulse interpretiert. Der jeweils aktuelle Emotionszustand der Figur ergibt sich dann aus dem Verlauf der Interaktion. Der Emotionszustand wird aus den bis zur aktuellen Situation aktivierten Erzählsegmente berechnet. Die Emotionsimpulse der im Verlauf der Interaktion aktivierten Segmente werden entsprechend ihrer zeitlichen Entfernung zum aktiven Segment gedämpft und aufsummiert. Die Ursprungsamplitude eines Impulses ist durch die Intensität der Emotion definiert.

Abbildung 6.1 stellt eine fiktive Hypertextstruktur von Erzählsegmenten dar. Ausgehend vom Startsegment A kann z.B. das Segment Z durch unterschiedliche Interaktionsverläufe auf unterschiedlichen Pfaden erreicht werden. Exemplarisch sind hier drei solcher Pfade herausgegriffen und rot (Segmentreihenfolge AGHIJZ), grün

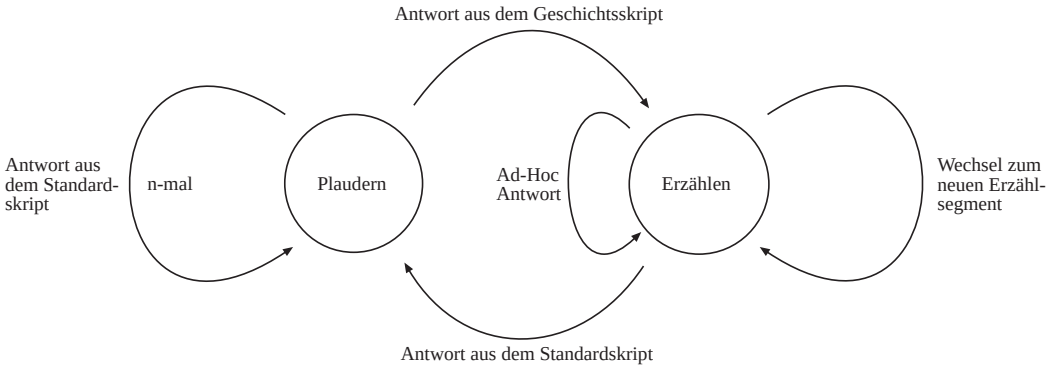


Abbildung 6.2: Zwei Zustände des VISTA-Systems: Geschichtenerzählen und Plaudern

(Segmentreihenfolge ABCDEFZ) und blau (Segmentreihenfolge AKLMINZ) markiert. Bei den drei Pfaden werden unterschiedliche emotionale Impulse aktiviert und somit ergibt sich für jeden Pfad bei Erreichen des Segmentes Z ein anderer Emotionszustand. Die Reaktion des Textmoduls besteht aus dem emotionalen Zustand der Figur und der entsprechenden textuellen Antwort. Der emotionale Zustand ist in einem Vektor aus Grundemotionen einschließlich der entsprechenden Intensität codiert. Die Anzahl der Grundemotionen ist programmiertechnisch beliebig und wird vom Autor des interaktiven Dialogskriptes definiert. Die Länge des Vektors ist dynamisch, die Anzahl der Grundemotionen somit im Geschichtsverlauf erweiterbar. In der Prototypimplementierung wurden zur Codierung die von Izard identifizierten neun Grundemotionen verwendet (Abschnitt 5.1.5). Dargestellt wird der Emotionszustand der Figur

(a) sprachlich in unterschiedlichen Formulierungen des Erzählinhaltes und

(b) visuell in Gestik, Mimik und Körperhaltung.

6.3.3 Wechsel zwischen Erzähl- und Plaudermodus

Das VISTA-System versteht zwei Skriptformate, eins für Geschichtsskripte und ein weiteres für Plauderdialoge. Falls das System den Benutzer im Kontext der Geschichte nicht versteht, plaudert die Figur mit dem Benutzer und kehrt dann nach n Interaktionen zur Geschichte zurück. Dieser Zustandswechsel ist in Abbildung 6.2 dargestellt.

6.3.4 Modifikation vorproduzierter Animationsdaten

Im Rahmen dieser Arbeit wurde bezüglich der Generierung der Animationsdaten (Abschnitt 5.2.5) eine hybride Konzeption aus Vorproduktion und regelbasierte Echtzeitgenerierung entwickelt:

(a) Komplexes, nicht formalisiertes Verhalten wird in einer Bibliothek vorproduzierter Animationssequenzen gespeichert.

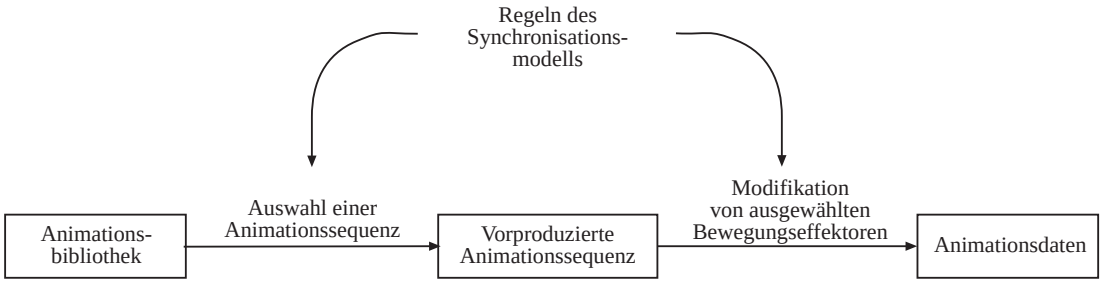


Abbildung 6.3: Hybride Konzeption zur Erzeugung der Animationsdaten

(b) Einzelne Bewegungseffektoren der ausgewählten Sequenz werden nach bekannten Regeln modifiziert.

Abbildung 6.3 stellt dieses hybride Animationsdatengenerierungskonzept schematisch dar. Aus der Animationsbibliothek wird eine Animatiossequenz ausgewählt. Die Animationskurven der Bewegungseffektoren, für die Regeln bekannt sind, werden entsprechend modifiziert. Eine genaue Beschreibung dieses Verfahrens befindet sich in Abschnitt 7.2.3.

6.4 Systemarchitektur

Unter Berücksichtigung der Anforderungen (Abschnitt 6.2) sind die Lösungsansätze in folgender Softwarearchitektur implementiert.

6.4.1 Funktionaler Systementwurf

Die funktionale Struktur des Systems ist in Abbildung 6.4 dargestellt. Die textuelle Eingabe des Benutzers wird anhand der Wortmuster, die zusammen mit den Erzählsegmenten von einem Autor definiert wurden, analysiert. Die resultierende Reaktionsregel aktiviert einen Link in der Hypertextstruktur. Daraufhin wird der Emotionszustand der neuen Situation angepaßt. Im Skript hat der Autor Name und Intensität der Emotionen der jeweiligen Erzählsegmente und Antwortsätze definiert. Der in neun Grundemotionen [Iza91] codierte Emotionszustand wird in Gestik, Mimik und Körperhaltung dargestellt, beeinflußt aber auch die Auswahl des Antwortextes. Die Sprachsynthese erzeugt aus nonvokal-verbaler Sprache die Audiodaten der vokal-verbalen Sprache, die vom Audioplayer dem Lernenden dargeboten wird. Der ursprüngliche Text und seine zwischenzeitliche Repräsentation in Phonemnotation beeinflußen die Körperbewegungen. Im Animationsmodul werden aus der Animationsdatenbibliothek, dem Emotionszustand und der Phonemliste der Sprachsynthese die Animationsdaten erzeugt. Der 3D-Player steuert mit diesen Animationsdaten das 3D-Polygonmodell des virtuellen Darstellers. Das Textdisplay zeigt den Antworttext zur Kontrolle auf dem Bildschirm an.

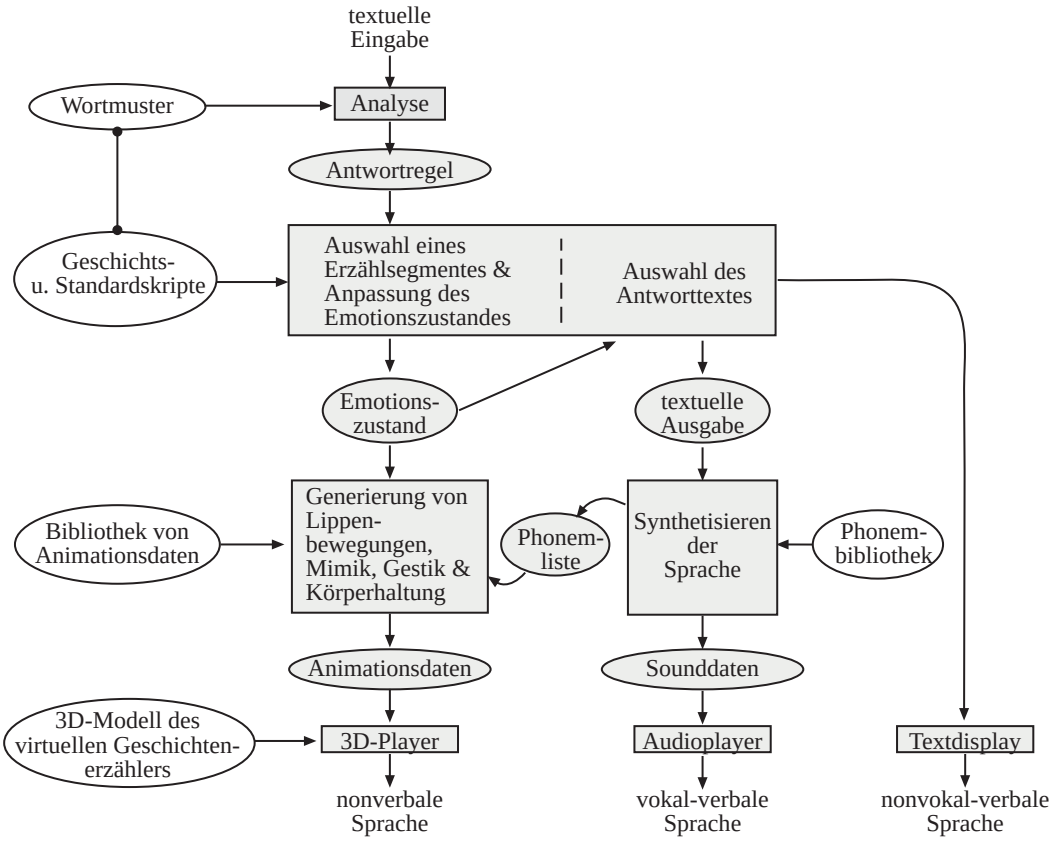


Abbildung 6.4: Systemüberblick

6.4.2 Entwurf der Softwarearchitektur

Aus den in Abschnitt 6.2 gestellten Anforderungen ergeben sich für die Softwarearchitektur (Abbildung 6.5) folgende Designentscheidungen:

- Das System läuft auf SGI-Workstations unter IRIX 5.3 oder höher.
- Die Architektur ist modular. Dadurch können Einzelkomponenten schnell und einfach an die technologische Entwicklungen angepaßt werden. Als 3D-Player kann sowohl die Animationssoftware *Maniac* als auch der *Trick17-Player* verwendet werden. Für beide existieren Schnittstellen.
- Das Textmodul ist vom Animationsmodul getrennt. Beide kommunizieren über einen TCP-Socket (Abschnitt 7.1.12).
- Das Textmodul ist im Hinblick auf eine zukünftige Anwendung im Internet in der Programmiersprache *Java* implementiert. Für ein solches Szenario können 3D-Player, Animationsmodul und Sprachsynthese einmalig lokal installiert werden. Das Textmodul liefert dann die Inhalte des Dialoges über das Internet.
- Zur textuellen Benutzereingabe wird die Java-Klasse *TextField* [Jav96] verwendet.
- Als Textdisplay wird die Java-Klasse *TextArea* [Jav96] verwendet.

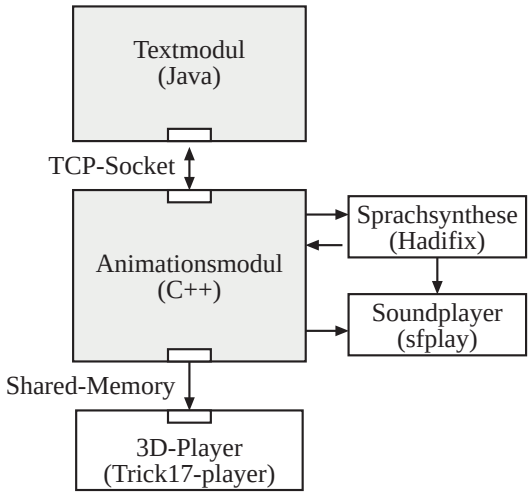


Abbildung 6.5: Softwarearchitektur des VISTA-Systems

- Das Animationsmodul ist in der Programmiersprache *C++* implementiert. Es steuert die Sprachsynthese und den Audioplayer mit dem *system*-Befehl.
- Zur Synthetisierung der Sprache wird das Sprachsynthesesystem *Hadifix* verwendet.
- Als 3D-Player wird der *Trick17-Player* verwendet, welcher mit dem Animationsmodul über einen gemeinsamen Speicherbereich (*Shared Memory*) kommuniziert.
- Als Audioplayer wird des UNIX-Programm *sfplay* verwendet.

Kapitel 7

Das VISTA-System

Die in Kapitel 6 erstellten Konzepte sind im Prototypsystem VISTA (*Virtual Storytelling Actor*) implementiert. VISTA ist ein virtueller Geschichtenerzähler, der im Dialog mit seinem Benutzer die Geschichte "Alice im Wunderland" [Car89] erzählt. Das VISTA-System ist eine Experimentierplattform, in der die Synchronisation von nonverbalem Verhalten mit der verbalen Sprache getestet werden kann. Dieses Kapitel beschreibt die Implementierung. Abschnitt 7.1 erläutert das Textmodul. Das Animationsmodul wird in Abschnitt 7.2 beschrieben.

7.1 Textmodul

Das Textmodul ist das Medium der Benutzereingabe und Verarbeitungsmodul der textuellen Sprache. Abbildung 7.1 zeigt die Benutzeroberfläche des Textmoduls. Die Funktionalität der Benutzeroberfläche ist auf eine Eingabezeile, ein Testdisplay und zwei Kontrollboxen reduziert. In die Eingabezeile schreibt der Benutzer mittels der Tastatur seine Äußerung und aktiviert abschließend die Verarbeitung mit der *Return*-Taste. Das Textdisplay stellt die Benutzereingabe und den Antworttext des VISTA-Sytems zur Kontrolle auf dem Bildschirm dar. Die Kontrollbox *answer rule mode* steuert die Auswahl einer Antwortregel aus der Listenstruktur (Abschnitt 7.1.5). Der Status der Kontrollbox *save dialogue* bestimmt, ob der Dialog zusätzlich in einer Datei gespeichert wird. In diesem Abschnitt die Funktionalität des Textmoduls erläutert. An wichtigen Stellen werden die Algorithmen anhand ihrer Implementierung in Java-Klassen verdeutlicht. Das VISTA-Textmodul basiert auf der Eliza-Implementierung in Java von [Hay]. Die dort implementierten Algorithmen wurden modifiziert und um die im Rahmen der vorliegenden Arbeit implementierten Konzepte erweitert.

7.1.1 Design eines Erzählknotens

Der Autor segmentiert seine Geschichte in Erzählknoten. Ein solches Erzählsegment enthält:

- den Erzählinhalt,
- eine Liste von Emotionen und deren Intensität und

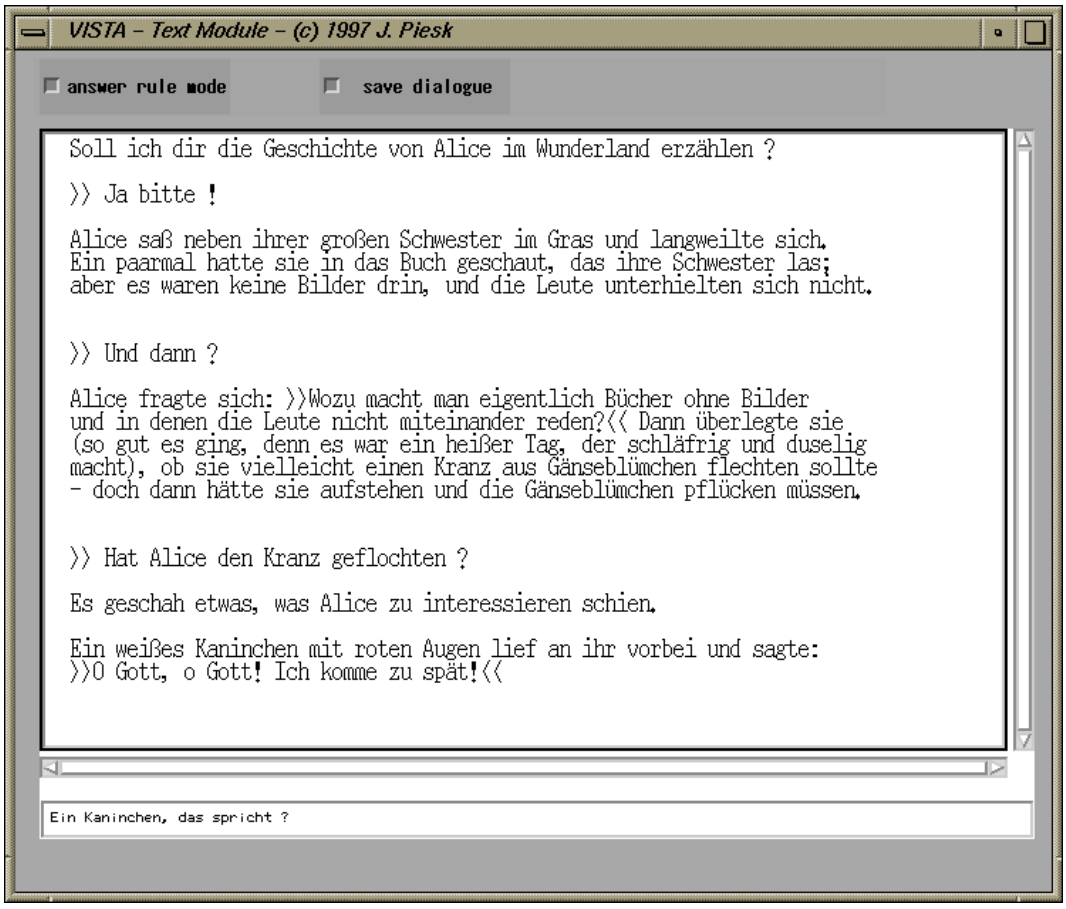


Abbildung 7.1: Benutzeroberfläche des Textmoduls

- jeweils eine Liste mit Eintritts- und Austrittsschlüsselworten, die zur Textanalyse (Abschnitt 7.1.3) benötigt werden.

Die Austrittsschlüsselworte enthalten Verbindungen zu Nachfolgesegmenten. Der Eingabetext wird nach diesen Austrittsschlüsselworten durchsucht, wenn das Segment aktiviert ist. Eine in dieser Schlüsselwortliste gefundene Antwortregel aktiviert ein Nachfolgesegment. Diese Liste der Austrittsschlüsselworte stellt also die Verbindungen dar, die von dem Knoten auf andere verweisen.

Die Liste der Eintrittsschlüsselworte wird von dem VISTA-Gedächtnis (Abschnitt 7.1.7) genutzt. Die Struktur der Listen der Eintritts- und Austrittsschlüsselworte entspricht dem Aufbau der Schlüsselwortlexika (Abschnitt 7.1.3).

7.1.2 Einlesen der Skripte

Es gibt zwei unterschiedliche Skripttypen - einer für Skripte mit Geschichtsstruktur und einer zum Einlesen von Schlüsselwortlexika. Die skripteinlesenden Objekte sind Instanzen der Java-Klassen *ScriptStory* und *ScriptKeyLex*. Die Methoden der Klasse *ScriptStory* sind für das Einlesen von Geschichtsskripten bestimmt, die Algorithmen der *ScriptKeyLex*-Klasse für Schlüsselwortelexika von allgemeinen Gesprächsthemen.

Segmentstruktur	Erläuterungen
segment: <ID1> <ID2> default: <dateiname> <text> (...) interested: <dateiname> <text> (...) surprised: <dateiname> <text> (...)	Antworttexte bei Aktivierung des Segmentes, sortiert nach Emotionszuständen
root: <dateiname> <text> (...)	Antwortexte für Rückkehr aus dem "Plaudermodus"
retro: <dateiname> <text> (...)	Antwortext für Gedächtnisantworten
kret: <wort> decomp: * (...)	Eintrittsschlüsselworte: Liste von Schlüsselworten und Schlüsselwortkombinationen zur Gedächtnisaktivierung
emotion: <name> <wert> (...)	Liste von Emotionen und deren Intensität
key: <wort> <priorität> decomp: * <wort> <wort> * next: <ID> response: <dateiname> <text> next: <ID> (...) decomp: * <wort> <wort> * next: <ID> <ID> (...) decomp: * response: <dateiname> <text> next: <ID> (...) (...)	Austrittsschlüsselworte: Liste von Schlüsselworten und Schlüsselwortkombinationen, die eine neues Segment aktivieren

Abbildung 7.2: Skriptform eines Erzählknotens

Alle Skripte werden zeilenweise eingelesen. Am Anfang jeder Zeile kennzeichnet eine bestimmte Buchstabenreihenfolge, die mit einem Doppelpunkt abschließt und in diesem Abschnitt mit Anführungsstrichen gekennzeichnet ist. Die Skripteinlese-Objekte reagieren auf einen Satz von selbstdefinierten Befehlen, mit denen der Geschichtsautor die eingegebenen Texte und Emotionen definiert.

Die Klasse **ScriptStory** ist für das Einlesen des Geschichtsskriptes zuständig. Folgende Befehle sind hierfür reserviert (Abbildung 7.2):

(a) Die Buchstabenfolge “segment:” erzeugt ein Erzählsegment mit den folgenden Listen als Objekte der Klasse **TextList**: *BodyList, InterestedList, JoyfulList, SurprisedList, SorrowfulList, AngryList, DisgustedList, ContemptuousList, FearfulList, AshamedList, RootList, RetroList.* Weiterhin wird eine Emotionsliste *EmotionList* als Objekt der Klasse **SegmentEmotionList** und zwei Schlüsselwortlisten *KeyList* und *RetroKeyList* als Objekt der Klasse **KeyList** erzeugt. Diese Listen werden solange mit Inhalt gefüllt, bis die nächste Buchstabenfolge “segment:” das nächste Segment erzeugt. Das Erzählsegment wird der **SegmentList** zugefügt.

(b) Die Buchstabenfolge “default:” trägt den nachfolgenden Text in die aktuelle *BodyList* ein. Die Buchstabenfolgen “root:”, “retro:”, “interested:”, “joyful:”, “surprised:”, “sorrowful:”, “angry:”, “disgusted:”, “contemptuous:”, “fearful:” und “ashamed:” tragen den jeweils nachfolgenden Text in die entsprechende *RootList, RetroList, InterestedList, JoyfulList, SurprisedList, SorrowfulList, AngryList, DisgustedList, ContemptuousList, FearfulList und AshamedList.*

(c) Die Buchstabenreihenfolge “emotion:” fügt der *EmotionList* dieses Segmentes die nachfolgend angegebene Emotion und Intensiät hinzu.

(d) Die Buchstabenreihenfolge “key:” trägt das nachfolgende Schlüsselwort und dessen Prioritätskennziffer in die *KeyList* ein und erzeugt eine Liste von Zerlegungsregeln.

(e) Jede Zerlegungsregel, die in diese List eingetragen werden soll, ist mit der Buchstabenreihenfolge “decomp:” gekennzeichnet. “decomp:” trägt die nachfolgende Zerlegungsregel in die Liste ein und erzeugt ein Liste von Antwortregeln, die durch “next:” oder “response:” gefüllt werden.

(f) “next:” kennzeichnet eine Antwortregel (Abschnitt 7.1.5) zur Navigation in der hypertextuellen Erzählstruktur.

(g) “response:” kennzeichnet einen Antwortsatz oder ein Antworttemplate (Abschnitt 7.1.5).

(h) “kret:” trägt den nachfolgenden Text in die *RetroKeyList* ein.

Die Klasse **ScriptKeyLex** erzeugt nach dem gleichen Verfahren aus den Skripten mit Standardantworten die Schlüsselwortlexika (Abschnitt 7.1.3). Die reservierten Buchstabenfolgen sind hierbei reduziert auf “key:”, “decomp:” und “response:”. Die Buchstabenfolgen zum Aufbau der Hypertextstruktur werden hier nicht benötigt. Die Klasse **ScriptSyn** liest Skripte mit Synonymlisten ein. Die Buchstabenfolge “synon:” kennzeichnet dabei alle nachfolgende Worte in der Zeile als Synonyme.

Abbildung 7.3 zeigt die Datenstruktur eines Erzählsegmentes nach dem Einlesen des Geschichtskriptes. Die Austrittsschlüsselworte verweisen entweder auf das nächste Erzählsegment (g) oder auf Antworttemplates (Abschnitt 7.1.5) innerhalb dieses Erzählsegmentes (f). Die Eintrittsschlüsselworte aktivieren den “Retrotext”(Abschnitt 7.1.7). Der Erzählinhalt sollte in unterschiedlichen Emotionszuständen ausgedrückt werden (interessiert, freudig, traurig etc.). Je nach Emotionszustand (Abschnitt 7.1.10) der Figur wird bei Aktivierung der Erzählsegmentes der entsprechende Text ausgewählt. Der “Getbacktext” wird aktiviert, wenn das System vom Plaudermodus in den Erzählmodus wechselt (Abschnitt 6.3.3). Dabei wird der Erzählinhalt dieses Segmentes zusammengefaßt.

7.1.3 Analyse der Benutzereingabe

Die Abbildung der textuellen Eingabe auf die Antwortregel erfolgt - mit einigen Modifikationen nach einem in [Wei66] beschriebenen Algorithmus:

(a) Durchsuche den Text nach Schlüsselworten.

(b) Überprüfe sequenziell alle vorhandenen Schlüsselworte in der Reihenfolge ihrer Prioritätsstufe.

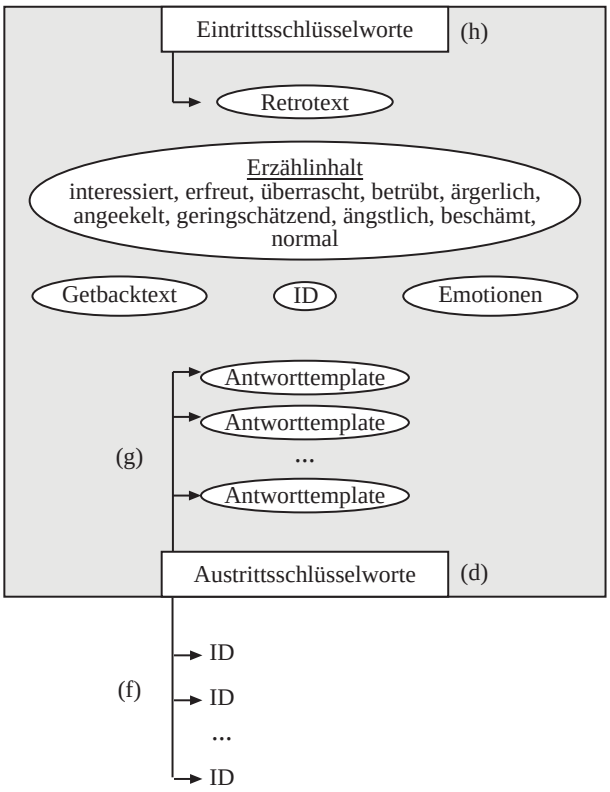


Abbildung 7.3: Datenstruktur eines Erzählsegmentes nach dem Einlesen des Geschichtsskriptes

(c) Jedem Schlüsselwort ist eine Liste von Zerlegungsregeln zugeordnet: Überprüfe bei jedem Schlüsselwort, ob eine dieser Regeln für den vorliegenden Text anwendbar ist. Falls keine Zerlegungsregel zutrifft: Gehe zum nächsten Schlüsselwort der Prioritätsreihenfolge.

(d) Jeder Zerlegungsregel ist eine Liste mit Antwortregeln zugeordnet. Falls eine Zerlegungsregel zutrifft: Wähle die erste Antwortregel der Liste. Sollte im weiteren Gesprächsverlauf dieselbe Zerlegungsregel nochmal aktiviert werden: Wähle die zweite Anwortregel. Bei der n-ten Aktivierung wird die n-te Antwortregel der Liste ausgewählt. Ist das Ende der Liste erreicht, wird sie wieder von vorn abgearbeitet. Befinden sich die Listen im Zufallsmodus, welcher durch die *List Random* Checkbox kontrolliert wird, wird die Antwortregel zufällig aus der Liste gewählt.

Mit dem Finden einer Antwortregel ist der Algorithmus beendet.

Schlüsselwortlexika

Das VISTA-System arbeitet nach dem in Abschnitt 3.4.4 beschriebenen *priorisierten Schichtenmodell von Miniexperten*. Jeder Miniexperte wendet den obigen Algorithmus mit seinem Lexikon von Schlüsselworten an. Jedem Schlüsselwort ist eine Liste mit Zerlegungsregeln zugeordnet, jeder Zerlegungsregel eine Liste mit Antwortregeln. Die Gesamtheit der Schlüsselworte einer Schicht bildet ein Schlüssel-

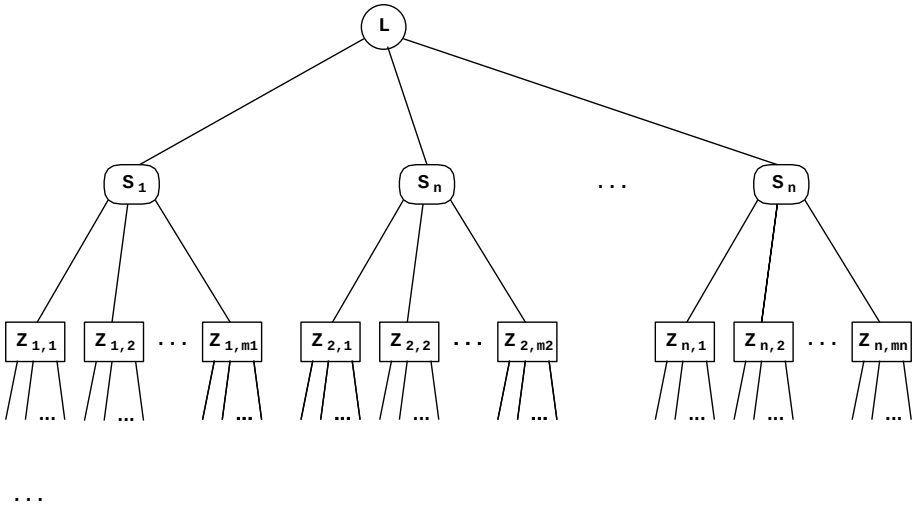


Abbildung 7.4: Struktur eines Schlüsselwortlexikons

Quelle: [Wei66])

wortlexikon. Die Struktur eines solchen Lexikons ist in Abbildung 7.4 dargestellt. Die Listenstruktur eines Lexikoneintrages ist in 7.5 abgebildet. S ist das Schlüsselwort, Z_i die i-te Zerlegungsregel des Schlüsselwortes und $A_{i,j}$ die j-te Antwortregel der i-ten Zerlegungsregel von S.

7.1.4 Zerlegungsregel

Eine Zerlegungsregel enthält ein oder mehrere Wörter bzw. Wortsequenzen, nach denen der Eingabetext durchsucht wird. Sind alle Wortsequenzen (ein Wort gilt als Sequenz der Länge eins) in dem Text enthalten, zerlegt die Zerlegungsregel den Satz in die angegebenen Wortsequenzen und die Wortgruppen dazwischen. Das folgende Beispiel verdeutlicht dieses Verfahren: Gibt der Benutzer z.B. den *Satz*

"Alice hat den Schlüssel auf dem Tisch vergessen."

ein, so wird der Miniexperte, dessen Lexikon das *Schlüsselwort*

"Schlüssel"

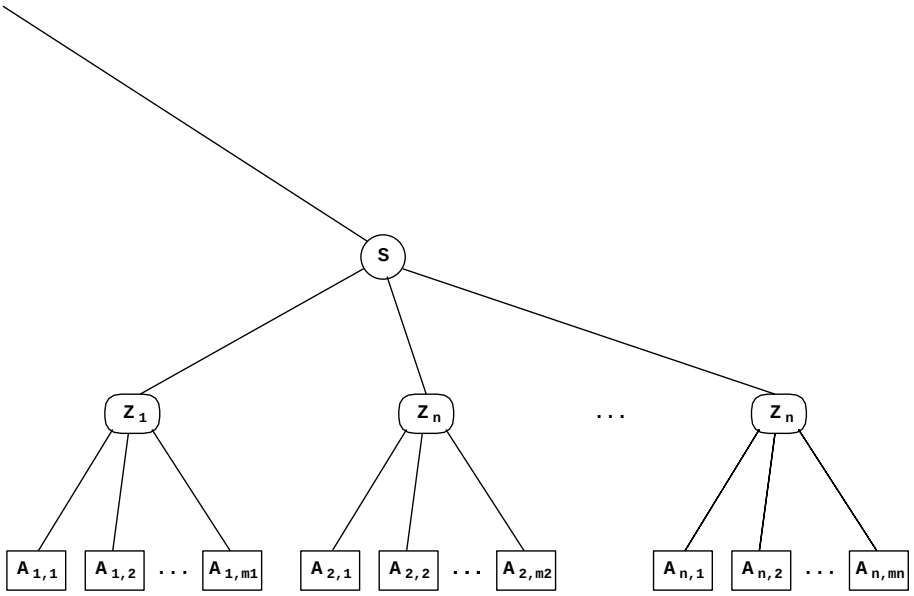
mit einer passenden Zerlegungsregel enthält, eine Antwort liefern. Die *Zerlegungsregel*

"* Schlüssel * Tisch *"

würde die Kriterien erfüllen. Enthält also der Lexikoneintrag "Schlüssel" diese Zerlegungsregel, so wird der Satz in die fünf Teile

(1) Alice hat den

(2) Schlüssel



...

Abbildung 7.5: Organisation der Zerlegungs- und Antwortregeln

Quelle: [Wei66]

(3) auf dem

(4) Tisch

(5) vergessen

zerlegt und die zugehörige Antwortregel aktiviert.

7.1.5 Antwortregel

Der aktuelle Modus (Abschnitt 6.3.3) bestimmt den Typ der Antwortregel. Im Plaudermodus besteht die Antwortregel aus einem Template, das den Antwortsatz ganz

"Ich weiß, sie hat den Schlüssel dort liegen lassen." (Antwortregel)

oder teilweise

"Ich weiß, (1) Schlüssel dort (5)." (Antwortregel)

definiert. Letztere produziert die Antwort "Ich weiß, Alice hat den Schlüssel dort vergessen.", erstere "Ich weiß, sie hat den Schlüssel dort liegen lassen." Eine ausführliche Diskussion dieses Verfahrens ist in [Wei66] zu finden. Diese beiden Antwortregeln können sowohl im Plauder- als im Erzählmodus definiert werden.

Die Objekte, in denen die Anwortregeln nach dem Einlesen der Skripte verwaltet werden, sind Instanzen der Klasse **Rule**. Für jede Skriptzeile, die entweder mit "response:" oder mit "next:" beginnt, wird eine solche *Rule-Instanz* erzeugt. "response:" erzeugt eine Antwortregel des Types *Antworttemplate*, "next:" speichert eine *ID-Folge* in der Antwortregel. Ein Antworttemplate wird als Objekt der Klasse **Text** gespeichert, eine ID-Folge als Objekt der Klasse **WordList**. Die Klasse *Text* speichert zusätzlich zum Antworttext Namen und Pfad der Audiodatei, in der der Antwortext als Sprachsignal einer menschlichen Stimme gespeichert ist. Ist keine Audiodatei vorhanden, wird anstatt des Namens das Schlüsselwort "noaudio" gespeichert. Die Klasse WordList speichert die Segment-IDs in einem Vektor. Eine Rule-Instanz stellt eine $A_{i,j}$ dar.

Die Antwortregeln sind in Objekten der Klasse **RuleList** gespeichert. Bei wiederholter Aktivierung derselben Zerlegungsregel Z_i werden andere Antwortregeln $A_{i,j}$ dieser Liste aktiviert. Das System hat für die Reihenfolge der Antworten einen *Zufallsmodus*, der über die Checkbox *List Random* der Benutzeroberfläche aktiviert wird. Befindet sich das System nicht im Zufallsmodus, wird bei der ersten Aktivierung von Z_i $A_{i,1}$ aktiviert, bei der zweiten Aktivierung $A_{i,2}$ etc. Übersteigt die Anzahl der Aktivierungen n die Anzahl der Antwortregeln n_j, dann wird die Liste wieder von vorne abgearbeitet.

Erzählmodus

Im Erzähl- oder Geschichtsmodus kommt zu beiden Template-Antwortregeln aus Abschnitt 7.1.5 die Möglichkeit hinzu, eine Sequenz von Nachfolgesegmenten als Antwortregel zu definieren. Eine solche Sequenz besteht aus einem oder mehreren Erzählsegementen, welche von dem virtuellen Geschichtenerzähler dargeboten werden. Insgesamt gibt es vier verschiedene Typen von Antwortregeln:

(a) einen Antwortsatz,

(b) ein Antworttemplate, das mit Worten oder Satzteilen der Benutzereingabe gefüllt wird,

(c) eine ID, die das nächste Erzählsegment identifiziert oder

(d) eine ID-Sequenz, die eine Reihenfolge von Erzählsegmenten definiert, dessen Inhalt wiedergegeben wird, ohne daß zwischen den Segmenten eine Benutzereingabe erwartet wird. Diese wird wieder im Anschluß an die Aktivierung des letzten Erzählsegmentes der Sequenz erwartet.

Die Makrostruktur der Geschichte besteht aus den vernetzten Erzählsegmenten (c und d), die Mikrostruktur aus Verzweigungen innerhalb der Erzählsegmente (a und b).

Für die Verwaltung der Segmente ist ein Objekt der Klasse **SegmentList** zuständig. SegmentList stammt von der Klasse *Hashtable* aus dem Java Packet *java.util* ab. Beim Einlesen der Skripte werden alle Segmente mit der Segmentidentifikation (ID) als Zugriffsschlüssel in diese Hashtabelle eingetragen. Liefert eine Antwortregel eine ID, wird diese mit dem Aufruf *SegmentList.get(ID)* als Zugriffsschlüssel für die Hashtabelle verwendet. Die Aktivierung eines Erzählsegmentes hat

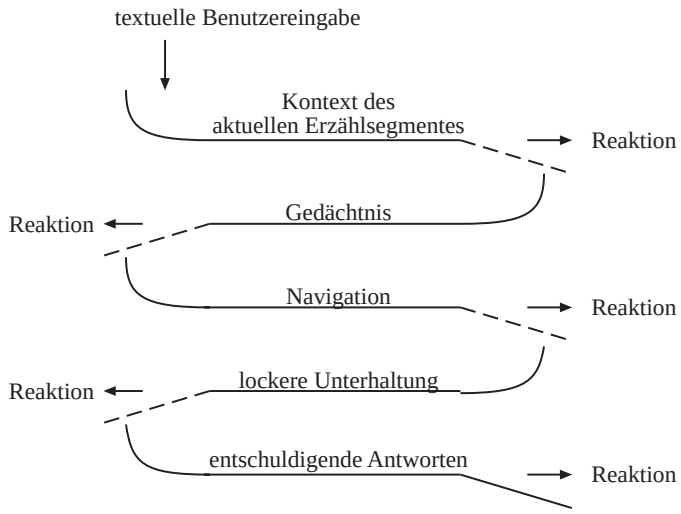


Abbildung 7.6: Verarbeitungsschichten des VISTA-Systems

neben der Auswahl eines Antworttextes auch eine Änderung des emotionalen Zustands der Figur zur Folge.

7.1.6 Verarbeitungsebenen

Die Benutzereingabe wird von Miniexperten analysiert, die in priorisierten Schichten angeordnet sind. Der Autor programmiert jede Schicht, indem er Skripte erstellt - für jede Schicht eins. Die Miniexperten werden dann vom VISTA-System nach den in dem jeweiligen Skript angegebenen Spezifikationen erzeugt. Die Routine zur Analyse der Benutzereingabe ist in fünf Verarbeitungsebenen organisiert (Abbildung 7.6): *Kontext des aktuellen Erzählsegmentes*, *Gedächtnis*, *Navigation*, *lockere Unterhaltung* und *entschuldigende Antworten*. Der Eingabetext des Benutzers wird zunächst nach den im aktuellen Erzählsegment angegebenen Wortmustern der Austrittschlüsselwortlisten durchsucht.

Ist im *Kontext des aktuellen Erzählsegmentes* (Ebene 1) kein passender Link oder Antworttext vorhanden, wird der Eingabetext auf niedrigeren Verarbeitungsschichten analysiert.

Die *Gedächtnisebene* (Ebene 2) durchsucht die Liste der Eintrittsschlüsselworte aller bis zum aktuellen Erzählsegment aktivierten Segmente.

Die *Navigationsebene* (Ebene 3) reagiert z.B. auf Anworten wie “Kann ich das nochmal hören?” oder “Kannst Du nochmal von vorne anfangen?” und aktiviert in der Hypertextstruktur das entsprechende Segment.

Die *Plauderebene* (Ebene 4) liefert Antworten aus dem ins Deutsche übersetzten Eliza-Skript. Diese Ebene ist eine Implementierung der in [Wei66] beschriebenen Algorithmen des Eliza-Programms.

Die unterste Ebene der *entschuldigenden Antworten* (Ebene 5) liefert Antworten wie “Ich glaube, ich habe Dich nicht verstanden?” oder “Können Sie das bitte anders ausdrücken?”.

Die obersten drei Ebenen (1-3) sind nur aktiv, wenn sich das System im Erzählmodus befindet. Im Plaudermodus beginnt die Analyse der Benutzereingabe auf

Ebene 4.

7.1.7 Gedächtnis

Ist ein Segment im Interaktionsverlauf einmal wiedergegeben worden, trägt das VISTA-System es in die Liste der bereits aktivierten Segmente ein. Mit Hilfe dieser Liste wird der Eingabetext nach den Eintrittsschlüsselwörtern von jedem bereits aktivierten Segment durchsucht, wenn in der darüberliegenden Schicht keine Antwortregel gefunden wurde. Wird in dieser Gedächtnisschicht eine Antwortregel gefunden, so bedeutet das nicht, daß das betreffende Segment nochmal aktiviert wird. Sein Inhalt wird lediglich in zusammengefaßter Form wiedergegeben. Die Aktivierung bleibt bei dem aktuellen Segment. Durch Aktivierung wird ein Segment zum aktuellen Segment. Diesen Status behält es solange, bis ein Nachfolgesegment gefunden und aktiviert wird.

Das VISTA-Gedächtnis wird von einem Objekt der Klasse **Memory** verwaltet. Wichtigste Methode dieser Klasse ist die Methode **lookUp(s)**. Sie bekommt als Argument die Benutzereingabe s. Das VISTA-Gedächtnis durchsucht die aktivierten Segmente, indem es bei jedem Segment die Methode **lookForRetro(s)** aufruft. Diese Methode gehört zur Klasse **Segment** und gibt einen Wert vom Typ *boolean*. Sie durchsucht die Eintrittsschlüsselwörter eines Segmentes. Paßt eine dieser Schlüsselwortkombinationen zur Benutzereingabe, gibt **lookUp(s)** dieses Segment zurück.

7.1.8 Wechsel in den Plaudermodus

Ein Wechsel vom Erzählmodus (TELL_MODE) zum Plaudermodus (CHAT_MODE) findet statt, wenn in den obersten drei Verarbeitungsschichten keine Antwortregel gefunden wurde. Der Modus ist in der Klasse **ChatSpace** in der Variablen *mode* gespeichert und bei einem Wechsel zum Plaudermodus auf CHAT_MODE (Konstante) gesetzt. Die Variable *chatDuration* wird auf einen Wert n=2 gesetzt und bei jeder Benutzeräußerung um eine ganze Zahl reduziert. Ist der Wert auf "0" reduziert, leitet eine Antwort aus dem *scriptGetBack* (z.B. "Ich sollte vielleicht einfach weitererzählen!") in den Erzählmodus über. Bei der Einstellung n=2 werden drei Antwortzyklen im Plaudermodus durchlaufen. Der Hypertextmechanismus der Erzählstruktur wird in diesem Modus nicht aktiviert. Die Algorithmen sind auf die in Abschnitt 7.1.3 beschriebenen Zerlegungs- und Antwortregeln reduziert.

7.1.9 Aktivierung eines Erzählknotens

Eine konsequente und ausgereifte Umsetzung des Anspruchs, einen Erzählinhalt in der sprachlichen Darbietung an die emotionale Vorgeschichte anzupassen, verlangt Methoden der automatischen Textgenerierung, die den Rahmen dieser Arbeit überstiegen hätten. Das Problem wurde auf einem einfacheren, aber effektvollen Weg gelöst: Der Autor schreibt für verschiedene Gefühlslagen unterschiedliche Textversionen des Segmentinhaltes. Als Grundlage dienen hier neun Grundemotionen - Interesse, Freude, Überraschung, Kummer, Wut, Ekel, Geringschätzung, Angst und Scham. Für jede dieser Emotionen gibt der Autor einen passenden Antworttext,

der von der Figur bei Aktivierung des Segementes gesprochen wird. Zur Auswahl des emotionsadäquaten Textes werden nach Berechnung des aktuellen Emotionszustandes die Grundemotionen nach ihrer Intensität sortiert. Die intensitätsstärkste Emotion bestimmt dann den Antworttext. Als Sortierungsalgorythmus wird *Quicksort* verwendet.

7.1.10 Berechnung des aktuellen Emotionszustandes

Nachdem die Emotionsimpulse des aktivierten Segmentes em_{i0} in das emotionale Gedächtnis eingetragen sind, werden die Emotionswerte em_i der aktuelle Emotionszustands E nach folgender Formel berechnet:

$$em_i = \sum_{j=0}^{n} g(j, em_{ij})$$

$$\forall i, j \in \mathbb{N}_0^+ \text{ mit } j \geq 0 \text{ und } em_i \in E \ .$$

Für die Gewichtsfunktion $g(i, em_i)$ gilt:

$$g(j, em_{ij}) > g(k, em_{ik})$$

$$\forall j, k \in \mathbb{N}_0^+ \text{ mit } k < j \leq n \text{ und } em_i \in E$$

Diese Funktion gewichtet also die einzelnen Emotionsimpulse em_{ij} nach ihrer Entfernung j zum aktuellen Segment. j gibt die Anzahl der aktivierten Segmente an. Die folgende Funktion erfüllt diese Kriterien:

$$g(j, em_{ij}) = em_{ij} e^{-di}$$

Der Dämpfungsfaktor d bestimmt die Stärke des Einflußes der zuvor aktivierten Segmente auf den aktuellen Emotionszustand.

Der Emotionszustand wird von zwei Klassen verwaltet: **EmotionMIX** und **EmotionTrail**. EmotionMix stellt eine Erweiterung der Klasse *Hashtable* aus dem Java-Packet *java.io* dar. Bei der Aktivierung eines Segmentes werden die entsprechenden Emotionen dieser *Hashtable* zugefügt. Ist eine Emotion in der *Hashtable* noch nicht enthalten, wird für diese Emotion ein neues Objekt der Klasse *EmotionTrail* erzeugt. Diese Klasse ist als dynamisch erweiterbarer Vektor implementiert, in den bei jeder Segmentaktivierung die Intensität, den das Segment zum Emotionszustand beiträgt, eingetragen wird. Ist die Emotion im Segment nicht enthalten, wird die Intensität "0" eingetragen. Der Index i des Vektors bestimmt somit die Entfernung des jeweiligen Zahlenwertes vom aktuellen Segment.

Zur Speicherung des berechneten Emotionszustandes enhält EmotionMix ein Objekt der Klasse **SegmentEmotionMix**, die ebenfalls als Vektor implementiert ist und auch für die Verwaltung der Emotionslisten der Segmente zuständig ist. Mit der Methode **qsort()** hat jedes Objekt dieser Klasse den *Quicksort-Algorithmus* implementiert, mit dem die Emotionen des aktuellen Emotionszustandes nach ihrer Intensität sortiert werden.

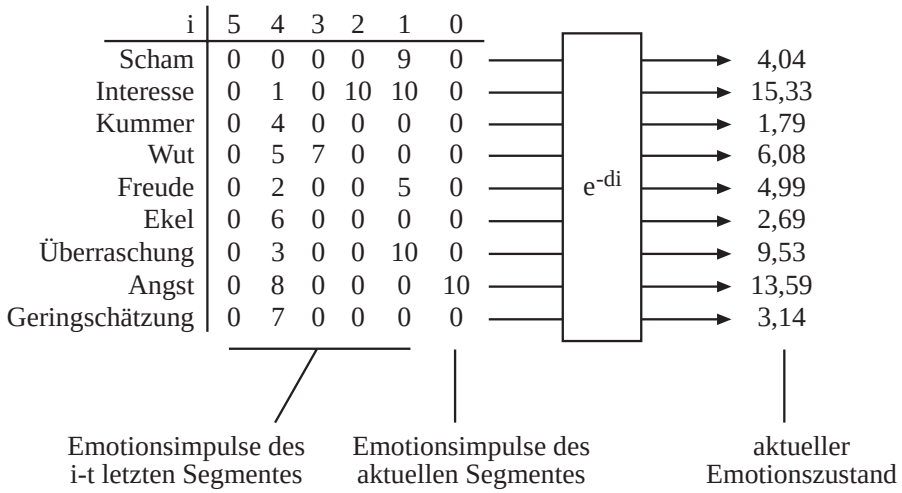


Abbildung 7.7: Berechnung des aktuellen Emotionszustands

Beispiel: Aktivierung des fünften Segmentes

Das fünfte Segment hat z.B. die folgenden Emotionsimpulse:

```
interest:           0
joy:                0
surprise:           0
sorrow:             0
anger:              0
disgust:            0
contemptuousness:   0
fear:              10
shame:              0
```

Diese Segmentemotionen werden in das emotionale Gedächtnis eingetragen bevor der aktuelle Emotionsvektor berechnet wird (Abbildung 7.7). Die Emotionen dieses Vektors werden dann nach ihrer Intensität sortiert:

```
interest:          15.33
fear:              13.59
surprise:           9.53
anger:              6.08
joy:                4.99
shame:              4.04
contemptuousness:   3.14
disgust:            2.69
sorrow:             1.79
```

In diesem Beispiel überwiegt die Emotion “Interesse”. Diese Emotion bestimmt Antworttext und Animationssequenz.

7.1.11 Sonderemotionszustände

Für bestimmte Ereignisse sind in der VISTA-Konzeption *Sonderemotionszustände* vorgesehen. Für jeden Sonderemotionszustand werden Bedingungen definiert. Werden diese Bedingungen im Interaktionsverlauf erfüllt, deaktiviert das VISTA-System den aktuellen Emotionszustand und ersetzt ihn durch den entsprechenden Sonderemotionszustand.

In der derzeitigen Implementierung ist der Sonderemotionszustand "Wut" (0 0 0 0 10 0 0 0 0) vorgesehen. Die Grundemotion "Wut" ist bei diesem Sonderemotionszustand mit der Intensität "10" aktiviert. Alle weiteren Grundemotionen sind mit der Intensität "0" bewertet und somit nicht beteiligt. Die Bedingung zur Aktivierung des Sonderemotionszustandes "Wut" ist: "Der Benutzer antwortet nicht." Dieser Emotionszustand beinflußt nicht die Wahl des Antworttextes im Textmodul, sondern nur die Wahl der Animationssequenz im Animationsmodul. Der Text wird zufällig aus den folgenden Anworttexten ausgewählt:

(1) Findest Du es nicht langweilig, Dir einfach etwas erzählen zu lassen. Sag auch mal etwas.

(2) Wenn Du nicht mehr mit mir sprechen willst, erzähl ich Dir auch nichts mehr!

(3) Interessiert Dich überhaupt, was ich hier erzähle?

Ist der Zufallsmodus (Abschnitt 7.1.5) werden diese Antworten bei wiederholter Aktivierung des Sonderemotionszustandes (0 0 0 0 10 0 0 0 0) in der Reihenfolge (1)-(2)-(3), in der sie im Skript angegeben sind, ausgewählt. In Abschnitt 8.1 zeigt die Aktivierung dieses Sonderemotionszustandes anhand eines Beispiels.

7.1.12 Socketverbindung zum Animationsmodul

Der Textmodul ist *Client* einer Socketverbindung zum Animationsmodul. Das Animationsmodul ist der *Socketserver*. Bei Systemstart wird der *TCP-Socket* von der *socketServer*-Routine errichtet und gebunden. Danach erwartet das Animationsmodul die Daten des Textmoduls. Abbildung 7.8 zeigt das Datenprotokoll dieser Verbindung. Vom *Java-Client* (Textmodul) werden aufeinanderfolgend vier Zeichenketten in den Socket geschrieben. Zur Synchronisation sendet der *C++-Server* (Animationsmodul) nach Empfang jeder dieser Zeichenketten ein Rücksignal. Die erste Zeichenkette wird als Antworttext in ASCII-Format interpretiert. Die zweite beinhaltet Pfad und Namen der Audiodatei, welche den Antworttext als vorproduziertes, gesprochenes Audiosignal enthält. Ist keine solche Datei vorhanden, wird das Schlüsselwort *noaudio* übermittelt. In der dritten Zeichenkette dieser Übertragung sind die Intensitäten der Grundemotionen (Kapitel 5.1.5) in der Reihenfolge Interesse (in), Freude (joy), Überraschung (sur), Kummer (sor), Wut (ang), Ekel (dis), Geringschätzung (con), Angst(fea) und Scham(sha) als Integerwerte enthalten. Das Leerzeichen " " zwischen den Werten wird von der auswertenden Routine, einer Methode der *stream*-Klasse aus der C++-Standardbibliothek, als Trennungsmarker benutzt. Die vierte und letzte Zeichenkette eines Übertragungsvorgangs enthält den Namen der intensitätsstärksten Emotion.

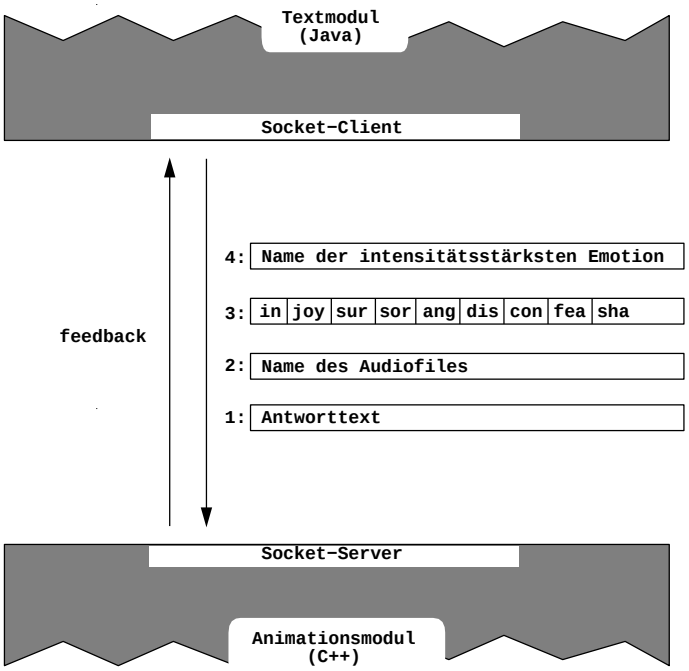


Abbildung 7.8: Protokoll des TCP-Sockets, der das Textmodul mit dem Animationsmodul verbindet

7.2 Animationsmodul

Die Generierung der Körperbewegungen basiert auf vorproduzierten Animationssequenzen, die in einer Bibliothek abgelegt sind.

7.2.1 Generierung der Animationsdaten

Unter Koordination der Animationssoftware *Maniac* werden mit einem *Motion-Capture*-System und zwei Datenhandschuhen die Animationsdaten aufgezeichnet. Während der Aufnahme wird die Figur von zwei Spielern gesteuert: einer kontrolliert mit an Armen, Kopf und Rücken befestigten Sensoren die Körperbewegungen, ein zweiter mit zwei Datenhandschuhen die Gesichtsmimik. Die Aufnahme erfolgt nach einem vorher angefertigten Drehbuch, welches im wesentlichen die den neun Grundemotionen zugeteilten Aufnahmezeiten enthält. Als brauchbares Maß hat sich eine halbe Minute pro Grundemotion erwiesen, da ein typischer Antworttext in der Regel kürzer ist.

Nach Abschluß der Aufnahme werden die Daten bei Bedarf nachbearbeitet und in einer Projektdatei gespeichert. Diese Projektdatei enthält neben den Animationsdaten auch die Konfigurationsdaten der Figur. Für das VISTA-System müssen Konfigurationsdaten und Animationsdaten getrennt werden. Hierfür stehen die beiden alleinstehenden Konvertierungsprogramme *AnimData* und *KonfigData*. *AnimData* speichert den Teil der Projektdatei, der die Animationsdaten enthält, in der Datei *anim.proj*. *KonfigData* filtert die Animationsdaten aus der Projektdatei und speichert das Ergebnis in der Datei *konfig.proj*. Die Konfigurationsdaten der *kon-*

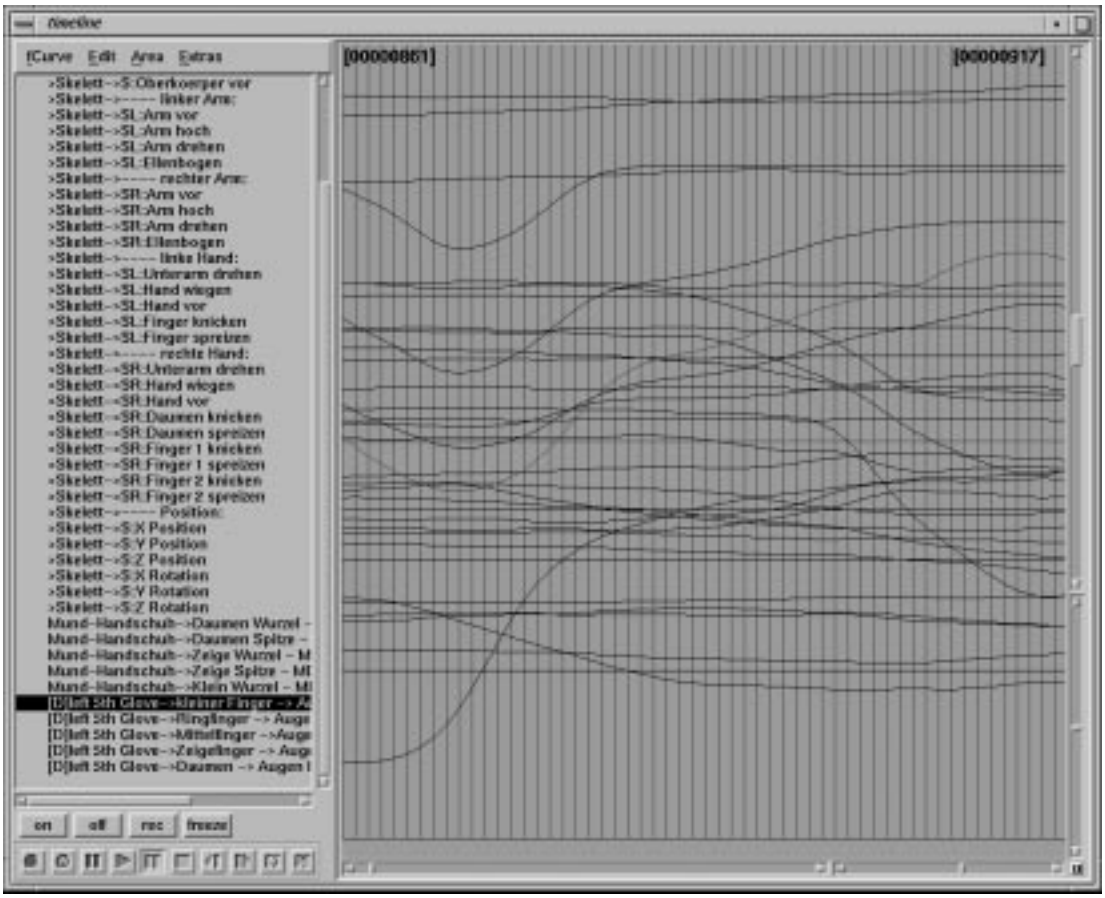


Abbildung 7.9: Animationskurven der 36 Bewegungseffektoren der Echtzeitkonfiguration mit Motion-Capture-System und Datenhandschuhen

fig.proj werden vom Visualisierungsmodul, dem 3D-Player, eingelesen, *anim.proj* vom Animationsmodul.

7.2.2 Animationsbibliothek

Eine Animationsbibliothek besteht aus einer Animationssequenz der Länge m Frames. In dieser Animationssequenz ist das Bewegungsrepertoire von Gestik und Mimik enthalten. Die Gesamtsequenz ist in n Untersequenzen unterteilt. Im System ist von jeder Untersequenz i der Startframe sf_i gespeichert. Abbildung 7.10 zeigt die Struktur einer solchen Animationsbibliothek. Für jede Animationsbibliothek, die das VISTA-System zu Programmstart einliest, wird ein Objekt der Klasse **Animdata** erzeugt. Die derzeitige Implementierung arbeitet mit einer Animationsbibliothek mit einer Länge von 3474 Frames. In der Gesamtanimationssequenz sind drei Grundemotionen gespeichert. *Wut* beginnt bei Startframe $sf_{Wut} = 275$, *Freude* bei $sf_{Freude} = 1490$ und Gestik, Mimik und Körperhaltung die einen traurigen Eindruck machen beginnen bei $sf_{Trauer} = 2627$. Die Werte der Startframes wird vom Autor nach Aufzeichnung der Animationssequenz durch Ansehen der Aufnahme ermittelt. Abbildung 7.11 zeigt aus jeder Grundemotion der Animationsbibliothek einen Ausschnitt.

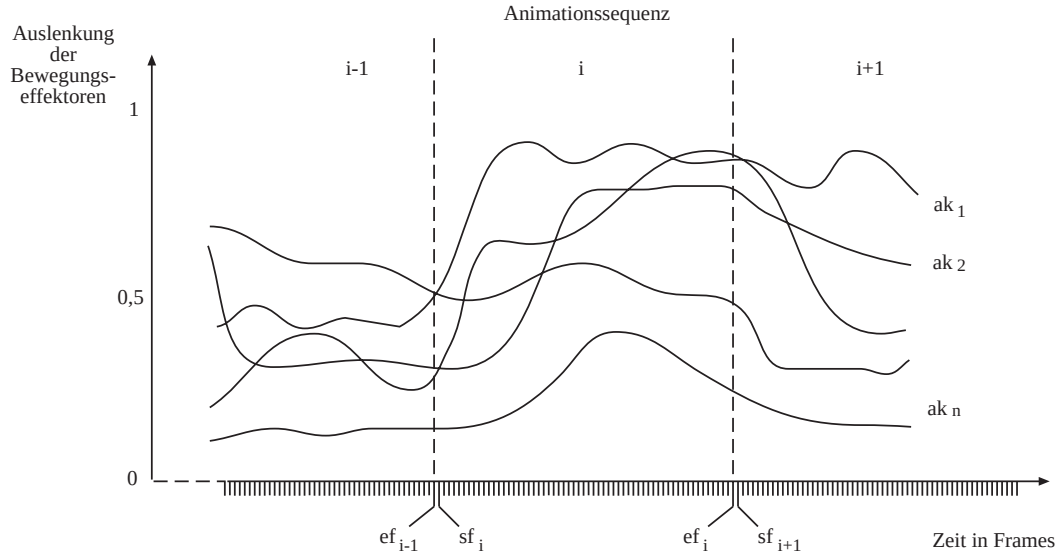


Abbildung 7.10: Datenstruktur der Animationsbibliothek

7.2.3 Datenstruktur

Bei Systemstart liest das Animationsmodul die Animationsdaten aus der Datei *anim.proj* ein und speichert die animationsrelevanten Daten der *anim.proj*-Datei in einer zweidimensionalen Matrix. Implementiert ist diese Matrix in dem *moeff[f][m]*-Array. Die Laufvariable f der ersten Dimension indiziert den *Frame* der Animationssequenz. Die Dimension stellt somit eine Aufteilung der Zeit in einzelne Bilder dar. Die Laufvariablen m indiziert die zweite, räumliche Dimension. Die vom Animator bei der Entwicklung der Figur angelegten Basisausdrücke werden hierzu auf höheren Ebene der Animation zu *Bewegungseffektoren*, kurz *moeffs* (*Motion Effectors*), zusammengefaßt. Die Einleseroutine für die Animationsdaten ist in der Klasse **Animdata** implementiert. Für jede Animationssequenz wird beim Einlesen ein Objekt dieser Klasse erzeugt, in dem die Daten verwaltet werden. Die Methode *readMoef(MoefName)* wird unter Angabe des Namens des Bewegungseffektors im Argument beim Einlesen der Animationsdaten für jeden Bewegungseffektor aufgerufen. Sie speichert die Daten in der entsprechenden Zeile in dem *moeff[f][m]*-Array (Abschnitt 7.2.3). Zum Abspielen der Sequenz liefert die Methode *getMoef(f,m)* den Wert des Bewegungseffektors m in Frame f. Das VISTA-System steuert das Polygonnetz des 3D-Charakters über die Ausprägungen von 36 Bewegungseffektoren. Die Werte der Tabelle 7.2.3 entstammen einer Kontrollausgabedatei des VISTA-Systems. Dies sind die auf drei Dezimalstellen gerundeten Werte, die von den Inputdevices (Motion-Capture-System und Datenhandschuhe) bei der Aufnahme ausgegeben, von der Animationssoftware *Maniac* aufgezeichnet, in der Projektdatei *record.proj* zwischengespeichert, von dem *AnimData* extrahiert, in *anim.proj* gespeichert und vom VISTA-System eingelesen wurden.

Abbildung 7.11: Ausschnitte aus Animationssequenzen der drei Grundemotionen *Wut* (links), *Freude* (mitte) und *Trauer* (rechts)

i		0	1	2	3	4	5	...
frame[i]		frame[0]	frame[1]	frame[2]	frame[3]	frame[4]	frame[5]	...
j	Name moeff[j]	moeff[i][j]						
1	Kopf drehen	0.041	0.069	0.084	0.089	0.089	0.089	...
2	Kopf wiegen	-0.044	-0.075	-0.088	-0.092	-0.092	-0.091	...
3	Kopf vor	0.137	0.234	0.281	0.299	0.306	0.312	...
...	...	...	...	...	...	...	...	
36	Augen links/rechts	0.160	0.285	0.362	0.412	0.451	0.49	...

Tabelle 7.1: Aufgezeichnete Werte von Bewegungseffektoren

7.2.4 Interpolation

An der Nahtstelle zwischen zwei Animationssequenzen werden die Animationskurven der einzelnen Bewegungseffektoren mit Hilfe einer trigonometrischen Funktion interpoliert. Als Interpolationsfunktion $I(f)$ wird

$$I(f) = \frac{1}{2}(1 - \cos(\pi * \frac{f}{AIF})$$

$$\forall f \in \{0, 1, 2, ..., AIF\}$$

mit AIF = Anzahl der Interpolationsframes

verwendet. Die Variable f zählt dabei die Bilder bis zum Anfang der Interpolation. Die Funktion hat die gewünschten Eigenschaften $I(0) = 0$ und $I(AIF) = 1$. Für die Animationskurven ak der einzelnen Bewegungseffektoren gilt dann:

$$ak(f) = aw + \delta * I(f)$$

mit $\delta = sw - aw$. Zur Berechnung der Differenz δ zwischen dem aktuellen Wert aw und dem Startwert sw der nächsten Animationssequenz werden jeweils die moeff-Werte des letzten Frames einer Animationssequenz in dem eindimensionalen Array $currentFrame[m]$ gespeichert und zur Interpolation mit den Werten der Startframe

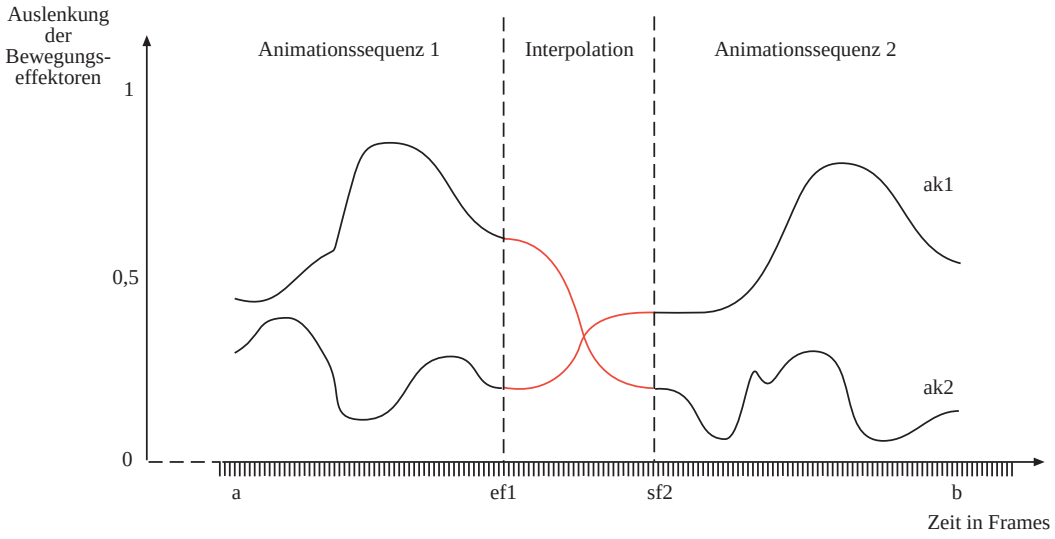


Abbildung 7.12: Interpolation der Auslenkungswerte zweier Bewegungseffektoren an der Nahtstelle zweier Animationssequenzen

der nächsten Animationssequenz verglichen. Abbildung 7.12 zeigt die Nahtstelle von zwei Animationssequenzen anhand der Animationskurven von zwei Bewegungseffektoren $ak1$ und $ak2$. Die Interpolation findet zwischen dem Endframe der ersten Animationssequenz $ef1$ und dem Startframe der zweiten Animationssequenz $sf2$ statt.

7.2.5 Auditive Wiedergabe

Zur sprachlichen Wiedergabe des Antworttextes gibt es zwei grundsätzlich verschiedene Strategien: vorproduzierte Audiodateien und Synthetisierung des geschriebenen Textes. Das VISTA-System verfolgt als Experimentierumgebung beide Strategien. Der Autor gibt im Skript vor dem Antworttext den Namen der Audiodatei an, welche den nachfolgenden Text als vorproduzierte Sprachaufnahme enthält. In diesem Fall synthetisiert das VISTA-System den Text nicht, sondern spielt die angegebene Audiodatei. Existiert für eine Antwortregel kein vorproduziertes Audiofile, so ersetzt der Autor im Skript den Namen der Audiodatei durch das Schlüsselwort *noaudio*. Das System synthetisiert in diesem Fall den Antworttext und gibt die von der Sprachsynthese zur Laufzeit produzierte Audiodatei wieder.

Das Nichtvorhandensein einer vorproduzierten Audiodatei kann zwei Gründe haben:

- Während der Entwicklungsphase des Skriptes kann es vorkommen, daß die Audiodateien noch nicht gesprochen wurden.
- Wenn die Antwortregel ein Template angibt, welches mit Worten oder Satzteilen aus der Benutzereingabe ergänzt wurde, wird der Antworttext zur Laufzeit zusammengesetzt. Eine Vorproduktion des gesamten Textes ist in diesem Fall nicht möglich.

Eine Strategie, die Vorproduktion von Audiodateien auch für zusammengesetzte Antworttexte tauglich zu machen, ist die einer Verkleinerung der vorproduzierten Einheit. Anstatt der Aufnahme ganzer Sätze werden Satzteile oder Worte vorproduziert. Eine Zusammensetzung der auditiven Antwort aus vorproduzierten Worten entspricht einer Sprachsynthese auf Lexembasis. Eine konsequente Verfolgung dieser Strategie bedeutet im nächsten Schritt eine Sprachsynthese auf Morphembasis, um die Mehrfachaufnahme von Wörtern in unterschiedlichen grammatikalischen Fällen zu vermeiden. Die nächste Stufe ist dann eine Sprachsynthese basierend auf Phonemen und Dyaden (Phonempaaren). Diese wird von dem Sprachsynthesesystem *Hadifix*, welches im VISTA-System verwendet wird, verfolgt.

Sprachsynthese

Das Sprachsyntheseprogramm *Hadifix* generiert aus dem Antworttext ein Sprachsignal. Die textuelle Eingabe wird zunächst in Lautschrift umgesetzt. Betonung und Phrasierung werden hierbei symbolisch erzeugt [PKS95]. Bei der Synthese werden kleine Abschnitte natürlicher Sprache (Phoneme und Phonempaare) aus einer Bibliothek aneinandergehängt [PHH94]. Bei der Betonung und Phrasierung wirkt im wesentlichen die Tonhöhe als Parameter [HP96]. Bei jeder betonten Silben steigt die Tonhöhe. Im Verlauf einer Äußerung fällt die Tonhöhe ab (Deklination). Bei den meisten Fragen steigt sie am Ende an.

7.2.6 Synchronisation der Mundbewegungen

Die Synchronisation der Lippenbewegungen mit dem Audiosignal der synthetisierten Sprache wird mittels einer Phonemliste erreicht, die die Sprachsynthese in die Datei *sounds.lst* schreibt. Jedem Phonem wird eine charakteristische Mundstellung zugeordnet. Dabei werden drei Mundstellungen unterschieden:

- breiter, geöffneter Mund bei den Vokalen “a” und “e” und dem Umlaut “ä”
- schmaler, geöffneter Mund bei den Vokalen “i”, “o” und “u” und den Umlauten “ö” und “ü”,
- geschlossener Mund bei allen anderen Phonemen und in Pausen.

Die Phoneme sind im SAMPA-Code (Speech Assessment Methods Phonetic Alphabet Code) [SAM97] notiert. Die Phonemliste liefert die Zeitpunkte der Phoneme in Millisekunden. Diese Zeitskala wird in eine Frameskala umgerechnet, so daß jedem Bild ein Phonem zugeordnet werden kann.

Beispiel: Phonemnotation

Die Verwendung der Phonemnotation zur Synchronisation der Mundbewegungen soll hier an einem Beispiel verdeutlicht werden. Angenommen die Figur spricht den Satz:

Dann geschah etwas, was für Alice zunächst nichts Besonderes war.

Die Sprachsynthese *Hadifix* verwendet intern die Phonemnotation des Satzes:

dan g@Sa: QEtvas, ma: vas fy:6 Qali:s@ tsunE:Cst nICts b@zOnd@R@s va:6.

Diese Phonemreihenfolge wird mit der jeweiligen Phonemdauer als Phonemliste in der Datei *sounds.lst* geschrieben (Tabelle 7.2.6).

	SAMPA	Dauer		SAMPA	Dauer		SAMPA	Dauer
1	pau	1	21	s	97	41	n	66
2	_d	44	22	f	95	42	I	71
3	d	13	23	y:	104	43	C	64
4	a	120	24	6	51	44	_t	36
5	n	127	25	Q	33	45	t	27
6	_g	44	26	a	112	46	s	85
7	g	12	27	l	100	47	_b	81
8	@	106	28	i:	147	48	b	7
9	S	85	29	s	95	49	@	106
10	a:	149	30	@	56	50	z	104
11	Q	33	31	_t	36	51	O	109
12	E	117	32	t	27	52	n	128
13	_t	36	33	s	81	53	_d	5
14	t	27	34	u:	83	54	d	13
15	v	67	35	n	63	55	@	106
16	a	107	36	E:	107	56	R	55
17	s	122	37	C	61	57	@	53
18	pau	5	38	s	83	58	s	90
19	v	48	39	_t	36	59	v	53
20	a	86	40	t	27	60	a:	204
						61	6	60
						62	pau	5

Tabelle 7.2: Phonemliste eines Beipielsatzes

Das Sprachsynthesesystem *Hadifix* wird mit dem *system*-Befehl aktiviert, nachdem ein Antwortsatz in die Datei *response* gespeichert wurde. Aus dieser Datei liest sie den zu synthetisierenden Text und schreibt das Sprachsignal in die Datei *response.raw* und die Phonemliste in die Datei *sounds.lst*. Die Datei *sounds.lst* wird vom Animationsmodul zur Synchronisation der Mundbewegungen eingelesen (Abschnitt 7.2.6). Die Datei *response.raw* wird abgespielt, wenn das Animationsmodul den Audioplayer *sfplay* startet.

7.2.7 Übergabe der Animationsdaten an den 3D-Player

Der 3D-Player, das Visualierungsmodul des VISTA-Systems, ist über eine *Shared-Memory*-Schnittstelle mit dem Animationsmodul verbunden. Er liest bis zu 25 mal pro Sekunde *n float*-Werte aus diesem gemeinsamen Speicherbereich. *n* ist die Anzahl der Bewegungseffektoren (Abschnit **??**). 25 Hz ist die maximale Datenrate des 3D-Players. Läuft das System auf einem Rechner, dessen Prozessorleistung für eine solche Bildwiederholungsfrequenz nicht ausreicht, so sinkt die Datenrate. Die Bilder, die der 3D-Player dann nicht mehr darstellen kann, werden ausgelassen. Anstelle der ausgelassenen Bilder verweilt das jeweils vorherige Bild auf dem Display. Bei einer Bilderwiederholungsfrequenz unter 12 Hz wirkt sich dies allerdings auf die Darstellungsqualität aus. Das Bild ruckelt. Das Animationsmodul schreibt unabhängig von der momentanen Datenrate des 3D-Players alle 40 Millisekunden (25 Hz) die *n moeff*-Werte in den gemeinsamen Speicherbereich. Die Datenrate ist auf der Seite des Animationsmoduls nicht kritisch, da die Werte aus dem einem Array gelesen werden.

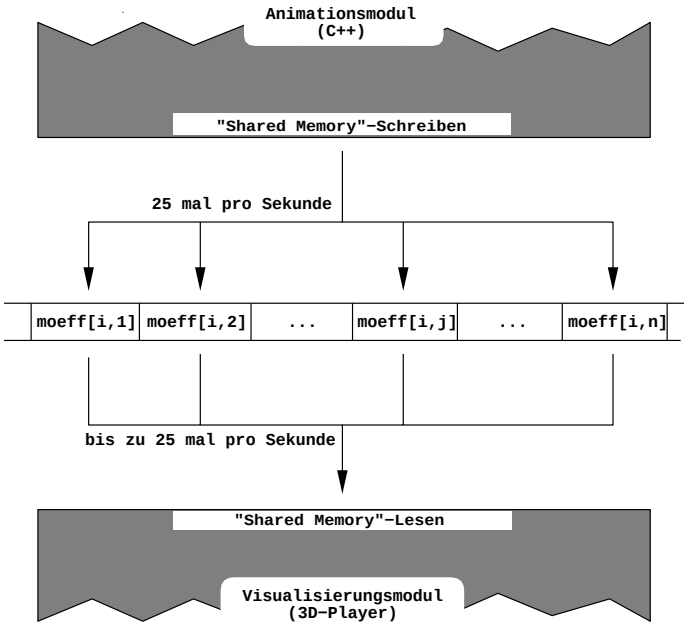


Abbildung 7.13: Protokoll der *Shared-Memory*-Schnittstelle, die das Animationsmodul mit dem 3D-Player verbindet

Kapitel 8

Diskussion und Perspektiven

In diesem Kapitel werden die im Rahmen der Diplomarbeit entwickelten Konzepte und deren Implementierungen kritisch diskutiert. Die in Abschnitt 8.1 dargestellten Dialogsequenzen geben einen Eindruck über die verbale Interaktion des VISTA-Systems mit dem Benutzer.

8.1 Beispieldialoge

Die Dialogsequenzen in diesem Abschnitt sind während einer Interaktion zwischen einem Benutzer und dem VISTA-System aufgezeichnet worden.

Dialogsequenz A:

1. VISTA: Möchtest Du die Geschichte von Alice und ihren Abenteuern im Wunderland hören?
2. USER: Ja.
3. VISTA: Alice saß neben ihrer großen Schwester im Gras und langweilte sich. Ein paarmal hatte sie in das Buch geschaut, das ihre Schwester las, aber es waren keine Bilder drin, und die Leute unterhielten sich nicht.
4. USER: Und warum unterhielten sich die Leute nicht?
5. VISTA: Genau das hat sich Alice auch gefragt!
6. VISTA: Alice saß also neben ihrer Schwester auf einer Wiese.
7. USER: Und was geschah dann?
8. VISTA: Alice fragte sich: "Wozu macht man eigentlich Bücher ohne Bilder und in denen die Leute nicht miteinander reden?" Dann überlegte sie, so gut es ging, denn es war ein heißer Tag, der schläfrig und duselig macht, ob sie vielleicht einen Kranz aus Gänseblümchen flechten sollte, doch dann hätte sie aufstehen und die Gänseblümchen pflücken müssen.
9. USER: Und darum blieb sie sitzen?
10. VISTA: Plötzlich lief ein weißes Kaninchen mit roten Augen an ihr vorbei. Das war nichts Besonderes. Alice fand es nicht einmal merkwürdig, daß das Kaninchen sagte "O Gott, o Gott! Ich komme zu spät!" Später allerdings wunderte sie sich, warum ihr das nicht gleich sonderbar vorgekommen war.
11. USER: Ist Alice immer noch schläfrig?
12. VISTA: Kannst du dir ein konkretes Beispiel vorstellen?
13. USER: Was?

14. VISTA: Warum fragst du?
15. USER: Weil ich Dich nicht verstehe!
16. VISTA: Ist das der wahre Grund?
17. USER: Natürlich.
18. VISTA: Ich sollte vielleicht einfach mal weitererzählen.
19. VISTA: Es lief also plötzlich dieses Kaninchen an ihr vorbei.
20. USER: Und dann?
21. VISTA: Alice rannte hinter dem Kaninchen her und sah, wie es in einem Kaninchenloch verschwand.
22. USER: Da konnte Alice ihm nicht folgen?
23. VISTA: Im nächsten Augenblick kroch Alice hinter ihm hinein, ohne lange zu überlegen, wie sie da je wieder herauskommen sollte. Zunächst führte der Gang wie ein Tunnel in die Erde.
24. USER: Und dann?
25. VISTA: Doch dann ging es so plötzlich senkrecht nach unten, daß ...

Wenn der Benutzer nichts eingibt und nur die *Return*-Taste drückt, erzählt der Kaktus die Geschichte in linearer Form. Allerdings wird nach n Interaktionen der Sonderemotionszustand *Wut* aktiviert. Mit dem entsprechenden Ausdruck in Gestik und Mimik fragt der virtuelle Geschichtenerzähler dann nach, ob den Benutzer die Geschichte nicht interessiert. Die folgende Dialogsequenz stellt dieses Szenario dar. Die Zahlenangaben in Klammern hinter dem Wort VISTA stellen den Emotionszustand bei der jeweiligen Äußerung dar. Die Angaben entsprechen den auf eine Dezimalstelle gerundeten Intensitäten der Grundemotionen in der Reihenfolge "Interesse, Freude, Überaschung, Kummer, Ärger, Ekel, Geringschätzung, Angst, Scham" an.

Dialogsequenz B:

1. VISTA: Möchtest Du die Geschichte von Alice und ihren Abenteuern im Wunderland hören?
2. USER:
3. VISTA(0 5 0 0 0 0 0 0 0): Alice saß neben ihrer großen Schwester im Gras und langweilte sich. Ein paarmal hatte sie in das Buch geschaut, das ihre Schwester las, aber es waren keine Bilder drin, und die Leute unterhielten sich nicht.
4. USER:
5. VISTA(10 4.1 0 0 0 0 0 0 0): Alice fragte sich "Wozu macht man eigentlich Bücher ohne Bilder und in denen die Leute nicht miteinander reden?" Dann überlegte sie so gut es ging, denn es war ein heißer Tag, der schläfrig und duselig macht, ob sie vielleicht einen Kranz aus Gänseblümchen flechten sollte, doch dann hätte sie aufstehen und die Gänseblümchen pflücken müssen.
6. USER:
7. VISTA(18.2 3.4 0 0 0 0 0 0 0): Ein weißes Kaninchen mit roten Augen lief an ihr vorbei und sagte: "O Gott, o Gott! Ich komme zu spät!"
8. USER:
9. VISTA(14.9 2.7 10 0 0 0 0 0 0): Als Alice sah, daß das Kaninchen eine Uhr aus der Westentasche zog, draufsah und schnell weiterlief, war ihr klar: "Ein Kaninchen mit einer Weste und obendrein einer Uhr in der Tasche - das war etwas Besonderes!" Voller Neugier rannte sie ihm über die Wiese nach und sah gerade noch, wie es in einem großen Kaninchenloch unter der Hecke verschwand.

10. USER:
11. VISTA(0 0 0 0 10 0 0 0 0): Findest Du es nicht langweilig, Dir einfach etwas erzählen zu lassen. Sag auch mal etwas.
12. USER:
13. VISTA(12.2 2.3 8.2 0 0 0 0 0 0): Im nächsten Augenblick kroch Alice hinter ihm hinein, ohne lange zu überlegen, wie sie da je wieder herauskommen sollte. Zunächst führte der Gang wie ein Tunnel in die Erde.
14. USER:
15. VISTA(9.9 1.8 6.7 0 0 0 0 10 0): Doch dann ging es so plötzlich senkrecht nach unten, daß Alice, ehe sie denken oder bremsen konnte, hinunterpurzelte.
16. USER:
17. VISTA(8.2 1.5 5.5 0 10 0 0 8.2 0): Entweder war der Schacht sehr tief, oder sie fiel ganz langsam. Während sie nämlich tiefer und tiefer sank, konnte sie sich in aller Ruhe umsehen und überlegen, was nun noch passieren würde. Sie versuchte hinunterzuschauen, wohin sie eigentlich fiel, aber dort war alles dunkel.
18. USER:
19. VISTA(0 0 0 0 10 0 0 0 0): Wenn Du nicht mehr mit mir sprechen willst, erzähl ich Dir auch nichts mehr!

In den VISTA-Äußerungen 11 und 19 wurde der aktuelle Emotionszustand aufgrund des vermutlich desinteressierten Benutzers deaktiviert und durch den Emotionszustand (0 0 0 0 10 0 0 0 0) ersetzt. Das erste Mal wurde der Emotionszustand nach fünf Interaktionen aktiviert, beim zweiten Mal bereits nach vier. Die dem entsprechende Zustandsvariable wird solange um eins erniedrigt, bis der Benutzer wieder etwas antwortet. Wenn die Zustandsvariable den Wert 0 erreicht hat, werden keine Geschichtsepisoden mehr dargeboten. Spätestens dann sollte der Benutzer etwas geäußert haben.

Das Konzept der Sonderemotionszustände eignet sich, um dem Benutzer zu suggerieren, der virtuelle Geschichtenerzähler würde seine Abwesenheit bemerken. Ein Überraschungseffekt bleibt auch dann nicht aus, wenn der Benutzer zwar interessiert die Geschichte verfolgt, aber denoch nichts äußert.

- Sowohl das Analyseverfahren als auch die Generierung der Antworten und Anwortregeln sind vom Autor erweiterbar. Das VISTA-System ist nicht auf bestimmte Dialogsequenzen begrenzt. Auch die Sprache ist vom Autor frei wählbar.
- Die modulare Systemarchitektur hat sich insbesondere zur Erstellung des interaktiven Dialogskriptes als vorteilhaft erwiesen. Zum schnellen Austesten des Skriptes kann das Textmodul auch ohne Animationsmodul, Sprachsynthese und 3D-Player benutzt werden.
- Das Konzept des Emotionszustandes hat sich als sinnvoll und praktikabel erwiesen. Problematisch ist jedoch noch die Darstellung subtiler Veränderungen des Emotionszustandes. Hierzu ist eine regelbasierte Generierung des nonverbalen Verhaltens notwendig. In [Bau96] wurde ein Verfahren zur automatischen Generierung von Emotionensausdrücken in der Mimik entwickelt. Die Implementierung kann an das VISTA-System gekoppelt werden.

Das VISTA-System befindet sich insofern noch in der Experimentierphase, als mit der derzeitigen Implementierung der technologische und konzeptionelle Rahmen für einen kreativen Umgang im Hinblick auf weitere Anwendungsszenarien geschaffen wurde. Desweiteren ist eine Integration von Sprach-, Mimik- und Gestikerkennungssoft- und -hardware [Hoc] [HF96] geplant, um die nonverbale Sprache in die Analyse der Benutzeräußerung zu integrieren.

8.2 Reaktionszeit

Durch die Verwendung von Hashtabellen (Abschnitt 7.1.5) ist weder die Analyse der Benutzereingabe noch die Navigation im Hypertextsystem zeitkritisch. Auch die Generierung der Körperbewegungen ist durch die Verwendung von Animationsbibliotheken, die bei Systemstart eingelesen werden, nicht Ursache einer für den Benutzer bemerkbaren Verzögerung der Antwort des VISTA-Systems. Die größte Zeitspanne in der Generierung der Systemantwort benötigt das Sprachsynthesesystem *Hadifix* zur Erzeugung der Audiodaten. Die dadurch entstehende Wartezeit ist abhängig von der Länge des Antwortsatzes. Die Gesamtreaktionszeit des VISTA-Systems beträgt ein bis zwei Sekunden. Dies ist für ein Dialogszenario akzeptabel. Eine Optimierung der Algorithmen ist jedoch insbesondere im Bereich der Sprachsynthese wünschenswert. [1]

Die durch die Sprachsynthese hervorgerufenen Verzögerungen treten bei der Verwendung vorproduzierter Audiodateien nicht auf. Hier entstehen jedoch die in Abschnitt 8.5 diskutierten Probleme der Synchronisation der Mundbewegungen. Eine exakte Ermittlung des Laufzeitverhaltens der im Rahmen dieser Arbeit entwickelten und implementierten Algorithmen ist nicht Bestandteil der vorliegenden Arbeit, dessen Schwerpunkt auf der Konzeption und Implementierung des Gesamtsystems liegt.

8.3 Synchronisation von verbalem und nonverbalem Verhalten

Das Synchronisationsmodell (Abschnitt 5.2.1) konnte in der zur Verfügung stehenden Zeit nur teilweise implementiert werden. Aus diesem Grund wirken unpassende Bewegungen noch störend. Auf expressive Gestik und Mimik ist in der Animationsbibliothek der derzeitigen Implementierung deshalb verzichtet worden. Die Synchronisationsregeln sind auf zwei Ebenen realisiert:

1. phonembasierte Synchronisation der Mundbewegungen und
2. Synchronisation des emotionalen Ausdrucks mit dem Erzählinhalt.

[1] Die in diesem Abschnitt dargestellten Erfahrungen wurden auf der UNIX-Workstation *Silicon Graphics Indigo 2 - Impact* gesammelt.

8.4 Vorproduzierte Animationssequenzen

Vorproduzierte Animationssequenzen zerstören die Illusion eines autonom agierenden 3D-Charakters, wenn sie wiederholt dargeboten und vom Benutzer wiedererkannt werden. Für zukünftige Entwicklung können Gestik und Mimik bei der Vorproduktion nach Bewegungseffektoren aufgeteilt werden. Während der Interaktion werden nach phonologischen, syntaktischen, semantischen und pragmatischen Kriterien die einzelnen Bewegungssequenzen zu einer Animation zusammengesetzt.

8.5 Sprachsynthese

Alternativ zur Synthetisierung des Textes arbeitet das System auch in einem Modus, welcher vorproduzierte Audiodateien mit menschlichen Sprachaufnahmen verwendet. Dieses Verfahren verbessert die Sprechqualität in bezug auf Betonung und Satzmelodie. Vorproduzierte Audiodateien erhöhen durch die qualitativ bessere Sprachwiedergabe die Glaubwürdigkeit der Figur und somit die Benutzerakzeptanz. Die Lippensynchronisation ist jedoch erheblich aufwendiger als unter Verwendung der Sprachsynthese: In [FHT] ist ein Verfahren beschrieben, welches ein Sprachsignal mit einem neuronalen Netz analysiert und die Zeitpunkte der charakteristischen Phoneme (Vokale, Um- und Schließlaute) liefert. Dieses System kann zur exakten Synchronisation der Mundbewegungen mit dem VISTA-System verbunden werden.

Durch die Sprachsynthese ist das System flexibel. Es kann z.B. Satzteile des Eingabestrings in der Antwort verwenden und so individuell auf den Benutzer oder die aktuelle Eingabe eingehen. Die Flexibilität der Sprachsynthese hat langfristig entscheidende Vorteile gegenüber der Verwendung vorproduzierter Audiodateien:

- Die Synchronisation der Mundbewegungen läßt sich über die von der Sprachsynthese erstellte Phonemliste erreichen.
- Für die Steuerungsparameter der Sprachsynthese (z.B. Lautstärke, Tonhöhe und Sprechgeschwindigkeit) lassen sich Betonungs- und Ausdrucksregeln implementieren. Dadurch kann z.B. der Klang der Stimme dem emotionalen Zustand der Figur angepaßt werden. Bei den vorproduzierten Audiosequenzen sind diese Parameter durch den Sprecher vorgegeben und nicht veränderbar.
- Die Produktionskosten der Sprachaufnahmen entfallen bei der Verwendung der Sprachsynthese. Die Zeitersparnis der durch die Sprachsynthese automatisierten Sprachaufnahmen wirkt sich vor allem bei der Produktion komplexer Geschichtsskripte aus.

Es kann folgende Schlußfolgerung gezogen werden: Mit vorproduzierten Audiodaten kann zwar kurzfristig eine bessere Qualität in der vokalen Ausdrucksfähigkeit des virtuellen Darstellers erzeugt werden. Langfristig sollte die Forschungs- und Entwicklungsarbeit jedoch die Vorteile der Sprachsynthese testen und in das VISTA-System integrieren.

8.6 Agentenkriterien

In diesem Abschnitt wird das VISTA-System anhand der in Abschnitt 5.3 angeführten Anforderungen an Agentensoftware bewertet.

1. *Autonomie*:
   Das Verhalten des VISTA-Systems ist dadurch, daß es durch die nichtlineare Erzählstruktur eingeschränkt ist, semi-autonom. Das System ist jedoch ausreichend komplex und die Anzahl der Freiheitsgrade hinreichend groß, daß ein Benutzer, der nichts von den Skripten weiß, für einen gewissen Zeitraum den Eindruck einer autonom handelnden Figur bekommt.

2. *Personalisierbarkeit*:
   Eine Lernkomponente, durch die das System sich auf einzelne Benutzer einstellen kann, ist nicht implementiert.

3. *Kommunikationsfähigkeit*:
   Durch verbale und nonverbale Sprache kann sich die Figur ausdrücken. In der Synchronisation von Gestik und Mimik mit der verbalen Sprache liegen die größten Schwächen. Verbal ist der Benutzer in einem ständigen Dialog mit dem VISTA-System.

4. *Risiko und Vertrauen*:
   Da die Software in den Unterhaltungsbereich einzuordnen ist, ist die Frage des Risikos nicht kritisch.

5. *Aufgabenbereich*:
   Die Aufgabe der Figur ist es, den Benutzer durch Erzählen einer Geschichte zu unterhalten. Kosten im Fall eines möglichen Versagens des Agenten sind dementsprechend nicht vorhanden.

6. *Sanfter Leistungsabfall*:
   Durch ein mehrschichtiges Analyseverfahren (Abschnitt 7.1.6), in dem die jeweils niedrigere Schicht in Aktion tritt, wenn auf der darüberliegenden Schicht keine Antwort erzeugt werden konnte, ist der Fall ausgeschlossen, daß die Figur keine Antwort gibt. Auf der untersten Schicht entschuldigt sich die Figur und bittet den Benutzer, seine Äußerung anders zu formulieren.

7. *Kooperation mit dem Benutzer*:
   Durch die Dialogform vollzieht sich die Kooperation mit dem Benutzer während der gesamten Interaktion.

8. *Anthropomorphismus*:
   Die Figur des sprechenden Kaktus in der Rolle als Geschichtenerzähler weist zwangsläufig menschenähnliche Züge auf.

9. *Erwartungshaltung des Benutzers*:
   Eine sprechende, semi-autonome, virtuelle Figur wirkt auf einen Benutzer zunächst überraschend. Für eine über einen längeren Zeitraum interessante Unterhaltung muß einerseits das Geschichtsskript, andererseits aber vor allem das Steuerungsmodul für Gestik und Mimik weiterentwickelt werden.

Aufgrund des obigen Bewertungsschemas kann die Implementierung des virtuellen Geschichtenerzählers VISTA als Agent bezeichnet werden. Die Aufgabe des virtuellen Geschichtenerzählers ist es, den Benutzer im Dialog zu unterhalten. Diese Anforderung wird in der derzeitigen Implementierung jedoch dadurch eingeschränkt, daß die nonverbale Ausdrucksfähigkeit des virtuellen Geschichtenerzählers auf Emotionsausdrücke beschränkt ist. Weitere Regeln für Gestik und Mimik konnten in der zur Verfügung stehenden Zeit nicht implementiert werden. Da weder für den Agentenbegriff noch für die hier angeführten Bewertungskriterien objektive Maßstäbe und Definitionen vorliegen, muß der Leser jedoch selbst entscheiden, ob der virtuelle Geschichtenerzähler als Agent bezeichnet werden kann.

8.7 Graphisches Dialogskriptmodellierungstool

Für das Erstellen von Dialogskripten mit komplex vernetzten Erzählsegmenten ist eine graphische Benutzeroberfläche zu entwickeln, die die Hypertextstruktur der Erzählung graphisch darstellt. Der Autor erstellt mit Hilfe eines solchen graphischen Dialogskriptmodellierungstools die Hypertextstruktur durch Auswahl und Manipulation der in einer Bibliothek zur Verfügung gestellten Elemente (Erzählknoten, Verbindung, Emotionsliste etc.). Das Dialogskriptmodellierungstool erstellt nach dem Modellierungsprozeß das Dialogskript im VISTA-Skriptformat. Die Erzählsegmente sollten auf Mausklick den Erzählinhalt anzeigen. Der Autor sollte in der Lage sein, bestehende Erzählsegmente in dieser Anzeige zu editieren und neue Erzählknoten zu erstellen. Mit Hilfe einer *Drag-and-Drop*-Funktionalität sollten Verbindungen zwischen Segmenten erstellt und durch Linien angezeigt werden. Bestehende Verbindungen sollten per *Drag-and-Drop* editiert werden können. Ein Gruppierungsmechanismus sollte mehrere vom Autor ausgewählte Erzählsegmente zu einer Episode zusammenfassen können. Die interne Struktur einer Episode ist nur auf Wunsch des Autors graphisch darzustellen. In der Standardeinstellung sollte ein *Icon* die gesamte Episode repräsentieren. Auf Mausklick könnten zur Darstellung der internen Struktur die Eintritts- und Austrittsverbindungen der Episode auf die entsprechenden Erzählsegmente aufgelöst werden.

Mit Hilfe der hier beschriebenen Gruppierungsfunktionalität könnten Standardgesprächssituationen modelliert und zur späteren Wiederverwendung in anderen Dialogskripten als Objekte gespeichert werden.

Durch Integration von Groupwarekonzepten in das hier beschriebene graphische Dialogskriptmodellierungstool könnte das Szenario einer verteilten Autorenschaft unterstützt werden.

8.8 Verteilte Autorenschaft

Interaktivskripte sind komplex. Für die Serienproduktion eines computergenerierten Geschichtenerzählers bietet sich das Szenario einer verteilten Autorenschaft an, wie es z.B. bei der Produktion der täglich ausgestrahlten *Soaps* und *Sitcoms* verwendet wird. Ein Team von Skriptautoren arbeitet an einem Drehbuch. Für die Skripte der nichtlinearen VISTA-Geschichten würde sich anbieten, zunächst die Makrostruktur aufzustellen und jeden Makroknoten einem Autor zuzuteilen. Der einzelne Autor schreibt dann die Mikrostruktur (Abschnitt 7.1.5).

8.9 Vormodellierte Standardstrukturen

Mit der im Rahmen dieser Arbeit entwickelten VISTA-Skriptsprache wird der Autor in die Lage versetzt, jede beliebige Hypertextstruktur aufzubauen. Ein graphisches Dialogskriptmodellierungstool sollte darüber hinaus die in Abschnitt 3.2 dargestellte Grammatiken für Strukturen höherer Verarbeitungsebenen in der Bibliothek von Grundelementen berücksichtigen. So könnten Standardstrukturen von Geschichten oder Episoden bereits vormodelliert sein. Die Aufgabe des Autors wäre es, diese Standardstrukturen mit Inhalt zu füllen.

8.10 Linguistische Textanalyse und -generierung

Die derzeitige Implementierung des VISTA-Systems basiert auf einer rudimentären Textanalyse und -generierung. Im Rahmen der vorliegenden Arbeit wurde das in [Kön95] beschriebene *LexGram*-System zur Entwicklung eines syntaktischen und semantischen Parsers getestet. Das *LexGram*-System eignet sich sowohl zur linguistischen Analyse der Benutzereingabe als auch zur Generierung der natürlichsprachlichen Systemantwort aus einer semantischen Repräsentationsform. Voraussetzung für die Entwicklung eines Parsers mit dem *LexGram*-System ist der Entwurf einer anwendungsspezifischen semantischen Repräsentationsprache. Diese ist im Rahmen der vorliegenden Diplomarbeit nicht implementiert, für Weiterentwicklungen des Systems jedoch empfehlenswert. Ein mit *LexGram* entwickelter Formalismus kann das in dieser Arbeit verwendete Verfahren der Codierung des invarianten Teils einer Menge von Äußerungen gleicher Bedeutung ersetzen. Der *LexGram*-Parser kann die vom Autor erwarteten Benutzerreaktionen in eine semantische Repräsentation übersetzen. Die Hypertextverbindungen würden dann auf dieser semantischen Ebene aktiviert. Der Parser analysiert sowohl die vom Autor antizipierte Benutzeräußerung als auch die während der Interaktion auftretende Äußerung des Benutzers und übersetzt beide in eine semantische Repräsentationssprache. Dadurch können beide auch dann als gleichbedeutend identifiziert werden, wenn keine gemeinsamen Wortmuster vorhanden sind.

Eine linguistische Analyse und Generierung der Dialogtexte ist auch für das in Abschnitt 5.2.1 entwickelte Synchronisationsmodell sinnvoll. Eine vollständige Implementierung dieses Modells benötigt u.a. die syntaktischen und semantischen Informationen des Textes, der von dem 3D-Charakter dargeboten werden soll.

Kapitel 9

Zusammenfassung

In der vorliegenden Arbeit wurden linguistische und psychologische Grundlagen der Mensch-Maschine-Kommunikation zusammengetragen. Hierauf aufbauend wurde ein Synchronisationsmodell für verbales und nonverbales Verhalten autonomer 3D-Charaktere erstellt. Für einen Steuerungsmechanismus zur Umsetzung von verbaler Sprache auf die begleitende Gestik und Mimik wurde ein theoretisches Modell erarbeitet. Hierbei determinieren die linguistischen Basiseinheiten eines zu animierenden Textes die Bewegungseffektoren des nonverbalen Verhaltensrepertoires eines virtuellen Akteurs.

Als Anwendungszenarien für dialogfähige 3D-Charaktere wurden der virtuelle Pädagoge und der virtuelle Geschichtenerzähler entwickelt. Der virtuelle Geschichtenerzähler wurde konzeptioniert und implementiert. Diese Konzeption verbindet das Synchronisationsmodell auf der pragmatischen Ebene mit dem Ansatz des nichtlinearen Geschichtenerzählens.

In dem VISTA-System sind die im Rahmen dieser Arbeit aufgestellten Konzepte *interaktives Dialogskript*, *Emotionszustand*, *Wechsel zwischen Erzähl- und Plaudermodus* und *Modifikation vorproduzierter Animationssequenzen* implementiert. Das interaktive Dialogskript dient einem Autor, eine nichtlineare Geschichtsstruktur zu gestalten und die Erzählinhalte dem Rezipienten in Dialogform zu vermitteln. Bei der Darbietung der Erzählinhalte durch den 3D-Charakter wird der Emotionszustand zur Synchronisation des nonverbalen Verhaltens mit dem Geschichtsverlauf auf der pragmatischen Ebene genutzt. Auf phonologischer Ebene werden die Mundbewegungen mit dem darzubietenden Text durch eine Phonemliste synchronisiert. Die Phonemliste wird von dem in der Konzeption des VISTA-Systems integrierten Sprachsynthesesystems *Hadifix* erzeugt. Der Wechsel zwischen Erzähl- und Plaudermodus verbindet die Hypertextnavigation der Erzählebene mit den Prinzipien des durch Eliza initiierten Genres der *Chatterbots*. Insbesondere werden für die Analyse der Benutzereingabe priorisierte Schichten verwendet, in denen nach einer passenden Systemantwort gesucht wird.

Das Konzept der Modifikation vorproduzierter Animationssequenzen verbindet die automatische Generierung von Bewegungsdaten mit der Nutzung von Animationssequenzen aus Bibliotheken. Zur Darstellung des Emotionszustandes im nonverbalen Verhalten des 3D-Charakters werden vorproduzierte Animationsdaten verwendet.

Anhang A

Behaviorismus

Der *logische Behaviorismus* geht von der Annahme aus, daß Aussagen über mentale Zustände anderer nur über das beobachtbare Verhalten verifiziert werden können. Der geistige Zustand selbst ist nur der betroffenen Person selbst zugänglich. Hierbei kommt es nicht darauf an, mentale Zustände empirisch über beobachtbare Verhaltensdispositionen zu identifizieren und eine Korrelation zwischen geistigen Zuständen und dem beobachtbarem Verhalten zu erstellen. Der logische Behaviorismus behauptet, daß Aussagen über mentale Zustände bedeutungsgleich sind mit Aussagen über Verhaltensdispositionen. Der *methodologische Behaviorismus* klammert dagegen geistige Zustände - weil sie nicht beobachtbar sind - aus den Untersuchungen aus und versucht, allgemeine Gesetzmäßigkeiten des Verhaltens ausschließlich aufgrund beobachtbarer Daten zu gewinnen. (vgl. [Hel91], S. 8)

Abbildungsverzeichnis

Literaturverzeichnis

[Bau96] S. Bauckmann. *Emotionale Steuerung virtueller Charaktere.* Diplomarbeit, Kunsthochschule für Medien Köln, Universität Dortmund, 1996.

[BHRM95] J. Bates, B. Hayes-Roth und P. Maes. *Interactive Storytelling Systems: Plot and Character*, 1995. AAAI Working Notes Spring Symposium.

[Bin91] J.-L. Binot. Natural language processing and logic. *From Natural Language Processing to Logic for Expert Systems*, S. 49–117. John Wiley & Sons, Chichester, 1991.

[Bla94] P. Blair. *Cartoon Animation.* Walter Foster Publishing, Tustin, California, 1994.

[BLH91] C. Beardon, D. Lumsden und G. Holmes. *Natural Language and Computational Linguistics: an Introduction.* Ellis Horwood Limited, 1991.

[BLK+97] G. Ball, D. Ling, D. Kurlander, J. Miller, D. Pugh, T. Skelly, A. Stankosky, D. Thiel, M. van Dantzich und T. Wax. Lifelike Computer Characters: the Persona project at Microsoft Research. *Software Agents*, MIT Press, 1997.

[BLR91] J. Bates, A.B. Loyall und W.S. Reilly. Broad Agents. *Proceedings of AAAI Spring Symposium on Integrated Intelligent Architectures*, Stanford, CA, March 1991.

[BO96] G. Bente und I. Otto. Virtuelle Realität und parasoziale Interaktion: Zur Analyse sozio-emotionaler Wirkungen computer-simulierten nonverbalen Kommunikationsverhaltens. *Zeitschrift für Individual- und Massenkommunikation*, S. 217–242, Opladen, September 1996.

[Bol91] J.D. Bolter. *Writing Space: the Computer, Hypertext, and the History of Writing.* Lawrence Erlbaum Associates, New Jersey, 1991.

[Bri] D. Brittan. *Talking Hands.* http://web.mit.edu/afs/athena/org/t/techreview/www/articles/apr96/Reporter.html am 20.05.1997.

[Bun96] O. Bunsen. *Animationsorientierte Optimierung von Polygonnetzen.* Diplomarbeit, Kunsthochschule für Medien Köln, Universität Dortmund, 1996.

[Car42] R. Carnap. *Introduction to Semantics.* Havard University Press, 1942.

[Car89] L. Carroll. *Alice im Wunderland*. Cecilie Dressler Verlag, Hamburg, 1989. Englische Originalausgabe, *Alice's Adventures in Wunderland*, 1865.

[Cho57] N. Chomsky. *Syntactic structures*. Mouton, 1957.

[Cho65] N. Chomsky. *Aspects of the theory of syntax*. MIT press, 1965.

[CHS94] M.e Courant, B. Hirsbrunner und K. Stoffel. Managing entities for an autonomous behavior. *Artificial Life and Virtual Reality*, John Wiley & Sons Ltd., 1994.

[CO71] W.S. Condon und W.D. Ogston. Speech and Body Motion Synchrony of the Speaker-Hearer. *The Perception of Language*, S. 150–184. Merrill, Columbus, Ohio, 1971.

[Col75] K. Colby. *Artificial Paranoia: A Computer Simulation of Paranoid Process*. Pergamon Press, New York, 1975.

[CPB+94] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Bechet, B. Douville, S. Prevost und M. Stone. Animated Conversation: Rule–Based Generation of Facial Expression, Gesture and Spoken Intonation for Multiple Conversational Agents. *Computer Graphics Proceedings, Annual Conference Series*, S. 413–420. ACM Press, July 1994.

[EF75] P. Ekman und W. Friesen. Unmasking the Human Face. *Consulting Psychologist*. Palo Alto, California, 1975.

[EF77] P. Ekman und W. Friesen. Manual for the facial action coding system. *Consulting Psychologist*. 1977.

[EF78] P. Ekman und W.V. Friesen. The Facial Action Coding System. *Consulting Psychologist Press*, Palo Alto, CA, 1978.

[Ekm88] P. Ekman. Gesichtsausdruck und Gefühl: 20 Jahre Forschung von Paul Ekman. *Reihe innovative Psychotherapie und Humanwissenschaft*. Vol. 38. Jungfermann, Paderborn, 1988.

[FHPD81] S. Frey, H.-P. Hirsbrunner, J. Pool und W. Daw. Das Berner System zur Untersuchung nonverbaler Kommunikation: I. Die Erhebung des Rohdatenprotokolls. *Methoden der Analyse von Face-to-Face Situationen*, S. 203–236, Stuttgart, 1981. Metzler.

[FHT] T. Frank, M. Hoch und G. Trogemann. Automated Lip-Sync for 3D-Character Animation. To be published on: 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, August 24-29, Berlin/Germany, 1997.

[FOM96] G. Fleischmann, R. Opalla und A. Mähler. Real-Time Animation of 3D Characters. *Proceedings of the 96 European SMPTE Conference on Imaging Media*, Cologne, 1996.

[Fon93] L.N. Foner. What's An Agent Anyway? A Sociological Case Study. Agents Group, MIT Media Lab, White Paper, May 1993.

[Gre86] J. Greene. *Language Understanding: A Cognitive Approach.* Open University Press, Philadelphia, 1986.

[GWF90] N. Gilbert, R. Wooffitt und N. Fraser. Organising Computer Talk. *Computers and Conversation.* Academic Press Limited, London, 1990.

[Hay] C. Hayden. Eliza Test. http://www.monmouth.com/ chayden/eliza/Eliza.html am 20.05.1997.

[Hel91] G. Helm. *Symbolische und konnektionistische Modelle der menschlichen Informationsverarbeitung: Eine kritische Gegenüberstellung.* Springer-Verlag, Berlin, 1991.

[HF96] M. Hoch und G. Fleischmann. Social Environment: towards an intuitive user interface. *Proceedings: 3D Image Analysis and Synthesis '96*, S. 155–161. Universität Erlangen-Nürnberg, November 1996.

[Hip95] M. St. Hippolyte. A Plot Beyond A Line: New Ways to Be Nonlinear. http://www.users.interport.net/ mash/nonlin.html am 20.05.1997. Last modified 10/16/95.

[Hoc] M. Hoch. Intuitive Schnittstelle. *Lab 3, Das Magazin der Kunsthochschule für Medien Köln.* To be published in July 97.

[HP96] B. Heuft und T. Portele. Synthesizing Prosody: A Prominence-Based Approach. *Proceedings ICSLP'96, S. 1361–1364, Philadelphia*, 1996.

[HRvGH96] B. Hayes-Roth, R. van Gent und D. Huber. Acting in character. *Proceedings of AAAI Workshop on AI and Entertainment*, 1996.

[IL79] H.-J. Ipfling und U. Lorenz. *Freude an der Schule.* München, 1979.

[Ipf74] H.-J. Ipfling. *Die emotionale Dimension in Unterricht und Erziehung.* München, 1974.

[Isa86] S. Isard. Levels of representation in computer speech and recognition. *Artificial Intelligence: principles and applications*, S. 111–122. Chapman and Hall Computing, London, 1986.

[Iza91] C.E. Izard. *The Psychology of Emotions.* Plenum Press, New York, 1991.

[Jav96] Java 1.0.2 API Documentation. http://java.sun.com:80/docs/api_ documentation.html am 20.12.1996.

[Joy87] M. Joyce. *Afternoon, a Story.* Eastgate, 1987.

[Ken80] A. Kendon. Gesticulation and speech: Two aspects of the process of utterance. *The relation between verbal and nonverbal communication*, S. 207–227. Mouton, 1980.

[KL95] D. Kurlander und D.T. Ling. Planning Based Control of Interface Animation. *Proceedings of CHI*, S. 472–479, 1995.

[Kön95] E. König. LexGram - A Practical Categorial Grammar Formalism. *Proceedings of the Workshop on Computational Logic for Natural Language Processing*, Edinburgh, Scotland, April 1995.

[Leh82] W. G. Lehnert. Plot Units: A Narrative Summarization Strategy. *Strategies for Natural Language Understanding*, S. 375–415. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1982.

[LW86] W. Lenders und G. Willée. *Linguistische Datenverarbeitung.* Westdeutscher Verlag, 1986.

[Mau94] M.L. Mauldin. Chatterbots, Tinymuds, and the Turing Test: Entering The Loebner Prize Competition. *Presented at AAAI-94*, 1994.

[McN92] D. McNeill. Hand and Mind: What Gestures Reveal about Thought. University of Chicago, 1992.

[Min75] M. Minsky. A framework for representing knowledge. *The Psychology of Computer Vision*, S. 211–277, New York, 1975. McGraw-Hill.

[Mor38] C. W. Morris. Foundation of the Theory of Signs. *International Encyclopedia of Unified Science*, Vol. 1, S. 77–137, University of Chicago Press, Chicago, 1938.

[Mül82] W. Müller. *Duden Fremdwörterbuch.* Bibliographisches Institut, Mannheim, 1982.

[NT94a] K. Nagao und A. Takeuchi. Social Interaction: Multimodal Conversation with Social Agents. *12th National Conference on Artificial Intelligence (AAAI)*, S. 22–28, 1994.

[NT94b] K. Nagao und A. Takeuchi. Speech Dialogue with Facial Displays: Multimodal Human-Computer Conversation. *32nd Annual Meeting of the Association for Computational Linguistics*, S. 102–109, 1994.

[OW75] R. Oerter und E. Weber. *Der Aspekt des Emotionalen in Unterricht und Erziehung.* Donauwörth, 1975.

[PHH94] T. Portele, F. Höfer und W. Hess. A mixed inventory structure for German concatenative synthesis. *Proceedings of the 2nd ESCA/IEEE Workshop on Speech Synthesis*, New York, 1994.

[PKS95] T. Portele, J. Krämer und D. Stock. Symbolverarbeitung im Sprachsynthesesystem HADIFIX. *Proc. 6. Konferenz Elektronische Sprachsignalverarbeitung*, S. 97–104, Wolfenbüttel, 1995.

[Pla95] C. Platt. Interactive Entertainment. *Wired Magazine*, 1995. http://www.hotwired.net/wired/3.09/features/interactive.html am 20.05.1997.

[SA77] R.C. Schank und R.P. Abelson. *Scripts, plans, goals, and understanding*. Lawrence Erlbaum Associates, Hillsdale, N.J., 1977.

[SAM97] SAMPA: Computer Readable Phonetic Alphabet. Department of Phonetics and Linguistics, University College London. http://www.phon.ucl.ac.uk/home/sampa/home.htm am 20.05.1997.

[Sch80] K. R. Scherer. The functions of nonverbal signs in conversation. *The Social and Phsysiological Contexts of Language*, S. 225–243. Lawrence Erlbaum Associates, 1980.

[SWZ90] N. Seibert, H. Wittmann und H. Zöpfl. Humor und Freude in der Schule. Donauwörth, 1990.

[Tho77] P.W. Thorndyke. Cognitive Studies in Comprehension and Memory of Narrative Discourse. *Cognitive Psychology*, Vol. 9, S. 77–100, 1977.

[Tru96] J. Truby. Was hält eine lang laufende Serie am Leben? *Seminar "Soaps & Sitcoms*. Hamburg, Juni 1996.

[Tur50] A. M. Turing. Computing Machinery and Intelligence. *Mind*, Vol. 54, S. 433–460, October 1950.

[Wal82] D. L. Waltz. The State of the Art in Natural Language Understanding. *Strategies for Natural Language Understanding*, S. 3–37. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1982.

[WBJ96] P. Watzlawik, J. Beavin und D. D. Jackson. *Menschliche Kommunikation*. Bern, 1996. Titel der Originalausgabe: Pragmatics of Human Communication, W.W.Norton & Company, New York, 1967.

[Wei66] J. Weizenbaum. ELIZA - A Computer Program for the Study of Natural Language Communication Between Man and Machine. *Communications of ACM*, 9(1):36–45, 1966.

[Wha96] T. Whalen. Computational Behaviourism applied to Natural Language, April 1996. http://debra.dgbt.doc.ca/chat/chat.theory.html am 20.05.1997.

[Win90] T. Winograd. Software für Sprachverarbeitung. *Computer-Anwendungen*, S. 86–100. Heidelberg, 1990.