

# Kognitive Modellierung von Kooperationsfähigkeiten für einen künstlichen Agenten

Dissertation zur Erlangung des Grades eines  
Doktors der Naturwissenschaften (Dr. rer. nat.)

**Nadine Pfeiffer-Leßmann**

# **Kognitive Modellierung von Kooperationsfähigkeiten für einen künstlichen Agenten**

Nadine Pfeiffer-Leßmann  
AG Wissensbasierte Systeme - Künstliche Intelligenz  
Technische Fakultät  
Universität Bielefeld  
Postfach 10 01 31  
D-33501 Bielefeld  
Germany  
E-Mail: nlessman@techfak.uni-bielefeld.de

Von der Technischen Fakultät der Universität Bielefeld zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Dissertation.

Dekan der Fakultät: Prof. Dr. Jens Stoye  
Erster Gutachter: Prof. Dr. Ipke Wachsmuth  
Zweiter Gutachter: Prof. Dr. Bernhard Jung

Einreichung der Arbeit: 15. Dezember 2010  
Tag der Disputation: 11. April 2011

Die offizielle Druckversion wurde auf älterungsbeständigem Papier gemäß DIN-ISO 9706 gedruckt.

## **Danksagung**

Für ihre Unterstützung während meiner Doktorarbeit möchte ich vielen Menschen einen herzlichen Dank aussprechen. Ich möchte mich bei Ipke Wachsmuth sehr für die gute lange Zusammenarbeit und Betreuung meiner Doktorarbeit bedanken. Bernhard Jung danke ich herzlich für die Begutachtung der Arbeit. Auch möchte ich mich bei meinen Kollegen Stefan Kopp, Alfred Kranstedt, Peter Biermann, Marc Latoschik, Christian Becker-Asano, Ian Voss sowie der jetzigen AG-WBS für die gute Zusammenarbeit und ihre konstruktive Kritik bedanken. Anika Tauchen danke ich für die Durchsicht meiner Arbeit und vor allem für ihr offenes Ohr und ihre freundschaftliche Unterstützung.

Mein größter Dank gilt meiner Familie, die immer an mich geglaubt hat und jederzeit für mich da war. Dabei möchte ich neben meinen Geschwistern ganz besonders meinen Eltern danken für ihre Geduld und Ermutigungen, die Durchsicht der Arbeit, liebevolle Kinderbetreuung, sowie ihre Unterstützung in jeglicher Art. Mein Dank richtet sich zudem in besonderer Weise an meinen Mann Thies für die ununterbrochene liebevolle Unterstützung in allen Bereichen sowie an Lucy und Miro, die tapfer zu mir gehalten haben, auch wenn ich einmal keine Zeit für sie hatte.



*Für meine Familie*



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielsetzung und Anforderungen . . . . .	2
1.3	Struktureller Aufbau der Arbeit . . . . .	4
<b>I</b>	<b>Kooperationsmodellierung</b>	<b>7</b>
<b>2</b>	<b>Theoretische Analyse</b>	<b>9</b>
2.1	Modelle der Kooperation . . . . .	12
2.1.1	Gemeinsame Kooperative Aktivität (Bratman) . . . . .	12
2.1.2	Teamarbeit (Cohen und Levesque) . . . . .	14
2.1.3	Gemeinsame Pläne (Grosz und Hunsberger) . . . . .	16
2.1.4	Gemeinsame Intentionalität (Tomasello et al.) . . . . .	18
2.1.5	Ideale Kooperation (Allwood) . . . . .	19
2.1.6	Zusammenfassung und Modellbildung . . . . .	21
2.2	Mechanismen der Kooperation . . . . .	24
2.2.1	Kommunikation . . . . .	25
2.2.2	Turn-Taking . . . . .	30
2.2.3	Initiative . . . . .	35
2.2.4	Verzahnung von Sprache und Handlung . . . . .	38
2.2.5	Verpflichtungen – Erwartungen – Obligationen . . . . .	41
2.2.6	Verbundenheit ( <i>Engagement</i> ) . . . . .	42
2.2.7	Gemeinsame Aufmerksamkeit ( <i>Joint Attention</i> ) . . . . .	44
2.2.8	Zusammenfassung . . . . .	45
2.3	Existierende Systeme . . . . .	46
2.4	Zusammenfassung . . . . .	53
<b>3</b>	<b>BCTC - Ein interaktives Kooperationsmodell</b>	<b>57</b>
3.1	Schema der Interaktionsbeiträge ( <i>Interaction-Move</i> ) . . . . .	58
3.2	Verhaltensebene ( <i>Behavior Layer</i> ) . . . . .	62

3.2.1	Emotionen . . . . .	63
3.2.2	Aufmerksamkeitsfokus - Blickbewegungen . . . . .	64
3.2.3	Gemeinsame Aufmerksamkeit ( <i>Joint Attention</i> ) . . . . .	64
3.3	Konversationsebene ( <i>Conversation Layer</i> ) . . . . .	67
3.3.1	Turn-Management . . . . .	67
3.3.2	Kontaktmanagement . . . . .	70
3.3.3	Diskursstrukturierung - Initiative . . . . .	72
3.3.4	Obligationen . . . . .	76
3.4	Aufgabenebene ( <i>Task Layer</i> ) . . . . .	78
3.4.1	Planen . . . . .	78
3.4.2	Dynamisch situiertes Problemlösen . . . . .	79
3.5	Kooperationsebene ( <i>Collaboration Layer</i> ) . . . . .	80
3.5.1	Gemeinsame Pläne . . . . .	81
3.5.2	Koordination - Verschachtelung von Planen und Handeln . . . . .	82
3.5.3	Repräsentation des Kooperationspartners . . . . .	82
3.6	Zusammenspiel der Ebenen . . . . .	83
3.7	Zusammenfassung - Architektur Anforderungen . . . . .	85

## **II Kognitive Architektur 89**

<b>4</b>	<b>Theoretische Grundlagen 91</b>
4.1	Kognition . . . . . 92
4.1.1	Kognitive Kontrolle . . . . . 92
4.1.2	Kognitive Architektur und kognitive Modellierung . . . . . 93
4.1.3	Arbeitsgedächtnis . . . . . 94
4.1.4	Emotionen . . . . . 107
4.1.5	Aufmerksamkeit . . . . . 108
4.1.6	Zusammenfassung der <i>bottom-up</i> Basisprozesse . . . . . 109
4.2	Mentale Zustände . . . . . 110
4.2.1	Belief-Desire-Intention-Modell ( <i>BDI</i> ) . . . . . 111
4.2.2	Relevanz des <i>BDI</i> -Modells . . . . . 114
4.2.3	Beliefs . . . . . 115
4.2.4	Ziele ( <i>Desires</i> ) . . . . . 116
4.2.5	Pläne . . . . . 117
4.2.6	Intentionen . . . . . 118
4.2.7	Zusammenfassung der <i>top-down</i> Prozesse . . . . . 126
4.3	Existierende Architekturen . . . . . 128
4.3.1	Kognitive Architekturen . . . . . 128
4.3.2	<i>BDI</i> -Systeme . . . . . 136
4.4	Diskussion . . . . . 140

<b>5</b>	<b>CASEC-Architektur</b>	<b>145</b>
5.1	Dynamisches Gedächtnismodell	148
5.1.1	Vorbewusste Zwischenspeicher	150
5.1.2	Episodischer Speicher ( <i>Episodic Buffer</i> ) - Beliefs	152
5.1.3	Zugriffstruktur	152
5.1.4	Repräsentation	155
5.1.5	Aktivierungsmechanismen	165
5.1.6	Verfall der Aktivierung ( <i>Decay</i> )	169
5.1.7	Zugriff auf Gedächtnisinhalte ( <i>Retrieval - Unification</i> )	171
5.1.8	Episodisches Gedächtnis	178
5.1.9	Semantisches Langzeitgedächtnis	178
5.2	Kognitive Schleife	179
5.2.1	Ziele in CASEC	180
5.2.2	Obligationen in CASEC	189
5.2.3	Deliberation	190
5.2.4	Handlungsoptionsgenerierung ( <i>Means-Ends-Reasoning</i> )	194
5.2.5	Pläne in CASEC	195
5.2.6	Intentionen in CASEC	201
5.2.7	Interaktionsverarbeitung	204
5.2.8	Commitmentverhalten	208
5.2.9	Einflussnahme von Emotionen	209
5.2.10	Aufmerksamkeitsfokus - <i>Joint Attention</i>	211
5.3	Zusammenfassung CASEC-Architektur	213
<b>III</b>	<b>Realisierung eines künstlichen Kooperationspartners</b>	<b>219</b>
<b>6</b>	<b><i>cMax</i> - Ein virtueller Kooperationspartner</b>	<b>221</b>
6.1	<i>Behavior Layer</i>	223
6.1.1	Aufmerksamkeitsmechanismen	223
6.1.2	Joint Attention	226
6.2	<i>Conversation Layer</i>	233
6.2.1	Kontaktmanagement	233
6.2.2	Turn-Taking	236
6.2.3	Dialogkompetenz	248
6.3	<i>Task Layer</i>	254
6.3.1	Einsatz externer Planer	254
6.3.2	Dynamisch situiertes Problemlösen	254
6.4	<i>Collaboration Layer</i>	260
6.4.1	Gemeinsame Pläne - ( <i>SharedPlans</i> )	260
6.4.2	Repräsentation des Kooperationspartners	266

6.5	Zusammenfassung . . . . .	267
<b>7</b>	<b>Resümee</b>	<b>271</b>
7.1	Diskussion und Ergebnisse . . . . .	271
7.2	Zusammenfassung . . . . .	276
<b>A</b>	<b>Formale Spezifikationen</b>	<b>305</b>
A.1	Modallogik . . . . .	305
A.2	Algorithmen . . . . .	308
A.2.1	Unifikationsalgorithmus . . . . .	308
A.2.2	Match-Algorithmus . . . . .	309
A.2.3	Applicable-Plan-List Algorithmus . . . . .	310

# Abbildungsverzeichnis

1.1	Interaktionen zwischen Roboter - ECA - Mensch . . . . .	2
2.1	Kooperationseinflüsse . . . . .	10
2.2	Abstraktes Modell kooperativer Einflüsse . . . . .	11
2.3	Gemeinsame kooperative Aktivität - Bratman . . . . .	14
2.4	Teamarbeit - Cohen und Levesque . . . . .	16
2.5	Gemeinsame Pläne - Grosz und Hunsberger . . . . .	18
2.6	Gemeinsame Intentionalität - Tomasello et al. . . . .	19
2.7	Ideale Kooperation - Allwood . . . . .	21
2.8	Einflussschwerpunkte der Kooperationsmodelle . . . . .	22
2.9	Modell der interaktiven Kooperation . . . . .	24
2.10	Interaktionsmanagementsfunktionen . . . . .	29
3.1	Ebenen des Kooperationsengagements . . . . .	59
3.2	Interaction-Move . . . . .	60
3.3	Joint Attention-Modell . . . . .	65
3.4	Regeln der Initiativmodellierung . . . . .	75
3.5	BCTC-Modell . . . . .	84
4.1	Mehrkomponentenmodell nach Baddeley (2000) . . . . .	99
4.2	Embedded Processes Model nach Cowan (1999) . . . . .	100
4.3	Arbeitsgedächtnis nach Oberauer (2002) . . . . .	102
4.4	Controlled Attention Framework nach Kane et al. (2007) . . . . .	103
4.5	Informationsverarbeitung in Act-R nach (Anderson et al., 2004) . . . . .	130
4.6	JAM-Architektur (Huber, 1999) . . . . .	138
5.1	Basismodell der CASEC-Architektur . . . . .	147
5.2	Gedächtnismodell der CASEC-Architektur . . . . .	149
5.3	Beispiel Detektorennetz: Turn-Taking-Detektor . . . . .	151
5.4	3 Teilung Gedächtnisstruktur . . . . .	153
5.5	Verschachtelter Belief in CASEC . . . . .	158
5.6	Mit Junktoren verknüpfte Belief-Formeln . . . . .	159

5.7	Berücksichtigung zeitlicher Aspekte bei <i>Belief-Chunks</i> . . . . .	162
5.8	Erweiterung der Beliefs um Aktivierungswerte . . . . .	162
5.9	Darstellung der Aktivierungswerte der Terme . . . . .	163
5.10	Vernetzte Repräsentation von <i>Belief-Chunks</i> . . . . .	165
5.11	Liste von Relationen - Netz von Relationen . . . . .	166
5.12	Dynamisches Arbeitsgedächtnismodell . . . . .	166
5.13	Aktivierung im Arbeitsgedächtnis . . . . .	170
5.14	Impulse, Aktivierungsfunktion und <i>Decay</i> . . . . .	172
5.15	Arbeitsgedächtnis - <i>Retrieval</i> . . . . .	175
5.16	<i>Goal</i> -Repräsentation . . . . .	181
5.17	Einflüsse der Zielaktivierung . . . . .	182
5.18	Modellierung der aktuellen Ziele . . . . .	183
5.19	Zielinteraktion . . . . .	188
5.20	Bedingung für den Einbezug des <i>Desirability</i> -Werts . . . . .	192
5.21	Entscheidungsgrundlagen <i>Deliberation</i> . . . . .	193
5.22	Intentions-Stack . . . . .	195
5.23	<i>Effort</i> -Berechnung . . . . .	199
5.24	Entscheidungsgrundlage Pläne . . . . .	200
5.25	Einfügen eines <i>Interrupt-Goals</i> in den Intentionsstack . . . . .	205
5.26	BDI und Aufmerksamkeitsfokus . . . . .	212
5.27	<i>Joint Attention</i> aus der Perspektive des Agenten <i>i</i> . . . . .	214
6.1	Detaillierte Übersicht der CASEC-Architektur . . . . .	222
6.2	Verarbeitungsmechanismen des <i>Cue-Processor</i> . . . . .	224
6.3	Beispiel: Blickfokussierung während der Referenzauflösung . . . . .	225
6.4	Aufmerksamkeitsdetektion . . . . .	227
6.5	Beispiel: <i>Joint Attention</i> -Interaktion . . . . .	227
6.6	Ablaufskizze <i>Joint Attention</i> . . . . .	230
6.7	Beispiel Kontaktmanagement: Begrüßung . . . . .	234
6.8	Beispiel Kontaktmanagement: Beenden einer Konversation . . . . .	236
6.9	<i>Taking-Turn</i> -Detektor . . . . .	238
6.10	<i>Giving-Turn</i> -Detektor . . . . .	239
6.11	Entscheidungsgrundlage für <i>TakingTurn</i> -Signale . . . . .	241
6.12	Entscheidungsgrundlage für <i>WantingTurn</i> -Signale . . . . .	242
6.13	<i>Taking-Turn</i> -Verarbeitung: Zeitlicher Ablauf . . . . .	245
6.14	Interlokutor unterbricht <i>cMax</i> . . . . .	246
6.15	Konversationaler Zustand <i>Gap</i> : Aushandeln des Turns . . . . .	247
6.16	<i>cMax</i> zeigt <i>WantingTurn</i> und <i>TakingTurn</i> Geste . . . . .	248
6.17	<i>cMax</i> zeigt <i>GivingTurn</i> Geste Stufe 1 und 2 . . . . .	249
6.18	Äußerungsgenerator . . . . .	253
6.19	Einschränkungsrepräsentation: Konstruktionsschritt . . . . .	255

6.20	Einschränkungsrepräsentation: Komplexer Konstruktionsschritt . .	256
6.21	Beispiel: Kooperation im Baufix-Szenario . . . . .	258
6.22	Individuelles Ziel, ein gemeinsames Ziel zu erreichen . . . . .	260
6.23	Entstehung gemeinsamer <i>Beliefs</i> . . . . .	262
6.24	Handlungsoptionen in Bezug auf eine referentielle Frage . . . . .	264
6.25	Intention der Planvervollständigung . . . . .	265
6.26	Modell der interaktiven Kooperation . . . . .	267
6.27	Beiträge der vorliegenden Arbeit für <i>cMax</i> . . . . .	267
7.1	Veranschaulichung der Struktur der vorliegenden Arbeit . . . . .	272



# Abkürzungsverzeichnis

<b>ECA</b>	<i>Embodied Conversational Agent</i> . . . . .	1
<b>BCTC</b>	<i>Behavior Conversation Task Collaboration</i> (siehe auch Kapitel 3) .	4
<b>CASEC</b>	<i>Cognitive Architecture for Situated Embodied Communicators</i> (siehe auch Kapitel 5) . . . . .	4
<b>BDI</b>	<i>Belief-Desire-Intention</i> (siehe auch Abschnitt 4.2.1) . . . . .	5
<b>CCR</b>	<i>Coordinated Cultivation Requirement</i> . . . . .	17
<b>IS</b>	<i>Information State</i> . . . . .	26
<b>DIT</b>	<i>Dynamic Interpretation Theory</i> . . . . .	28
<b>YTTM</b>	<i>Ymir-Turn-Taking-Modell</i> . . . . .	31
<b>FMTB</b>	<i>Function-Modalities-Time-Behavior</i> . . . . .	33
<b>CPS</b>	<i>Collaborative Problem Solving</i> . . . . .	40
<b>PFC</b>	<i>Präfrontale Cortex</i> . . . . .	92
<b>LTM</b>	<i>Long-term memory</i> . . . . .	98
<b>PRS</b>	<i>Procedural Reasoning System</i> . . . . .	114
<b>R-Graph</b>	<i>Rezept-Graph</i> . . . . .	125
<b>CSP</b>	<i>Constraint Satisfaction Problem</i> . . . . .	159
<b>cMax</b>	<i>cooperative Max</i> . . . . .	6
<b>COAR</b>	<i>Concepts for Objects, Assemblies, and Roles</i> . . . . .	254



# Kapitel 1

## Einleitung

### 1.1 Motivation

Während in früheren Dialogsystemen und Robotikanwendungen künstliche Systeme die Aufgabe eines Werkzeugs übernahmen und dem menschlichen Benutzer als untergeordnetes Hilfsmittel dienten, haben aktuelle Forschungsarbeiten die Sichtweise von künstlichen Systeme als *Partner* (Hoffman & Breazeal, 2004). Solche Systeme treten als autonome bzw. teilautonome Agenten auf, die Seite an Seite mit dem Menschen arbeiten. Sie reagieren nicht nur auf direkte Anweisungen, sondern versuchen auch selbstständig, sich an Lösungsprozessen zu beteiligen, um den Menschen zu unterstützen. Das früher zumeist vorhandene *Master-Slave*-Verhältnis der Interaktion wird damit aufgehoben.

Der Einsatz künstlicher Interaktionspartner ist nicht nur auf die reale Welt beschränkt. Auch im Bereich der *Virtuellen Realität* lassen sich Interaktionen mit künstlichen Agenten untersuchen. Beide Interaktionspartner sind dabei in einer gemeinsamen Umwelt situiert. Die Agenten können ihre Umwelt manipulieren, während die Umwelt auch selbst dynamisch ihren Zustand verändern kann. Um in dieser Umwelt kooperativ eine Aufgabe zu lösen, müssen sich die Partner aufeinander einstellen, zielgerichtetes Verhalten aufweisen und in der Lage sein, das Verhalten ihres Gegenübers einzuordnen sowie die gemeinsame Interaktionshistorie zu berücksichtigen.

Ein Beispiel solcher künstlicher Interaktionspartner stellen *Embodied Conversational Agents* (ECAs) dar (Cassell, Bickmore, Campbell et al., 2000a). Diese Agenten spielen insbesondere in Szenarien der *Virtuellen Realität* eine wichtige Rolle. Sie erlauben die Untersuchung von Aspekten der Kommunikation und Kooperation in „gleichberechtigten“ Dialogen, in welchen das System genau wie der menschliche Interaktionspartner die Initiative ergreifen kann. Interaktionen mit einer wechselnden Initiative (*mixed-initiative dialog*) die von An-

gesicht zu Angesicht stattfinden können eine direkte, natürliche Interaktion zwischen dem künstlichen System und dem Menschen fördern. Das Interaktionspotential der Agenten wird dabei in dem Grad erhöht, mit dem sie wechselseitig auf ihren jeweiligen Interaktionspartner eingehen können und zwar sowohl auf inhaltlicher als auch auf sozialer Ebene. Ansätze und Arbeiten, welche im Rahmen der Entwicklung von ECAs entstehen, lassen sich in großen Teilen auch auf Partner in Form von Robotern übertragen.



Abbildung 1.1: Interaktionen zwischen Roboter - ECA - Mensch <sup>1</sup>

Während sich die Forschung interaktiver Systeme bisher vornehmlich entweder auf Dialoge als eine Form des Austausches von Informationen oder auf das Planen und das zielgerichtete Ausführen von Instruktionen konzentrierte, beschäftigen sich bisher wenige Arbeiten mit der komplexeren Aufgabe der Integration von aufgabenorientierten Dialogen, dem Aushandeln von Plänen und der gleichzeitigen gemeinsamen Ausführung der Pläne. Bei solchen Systemen steht die Verzahnung von Sprache und Handlung im Vordergrund. Dabei wird deutlich, dass auch manipulative Handlungen kommunikative Funktionen übernehmen können.

## 1.2 Zielsetzung und Anforderungen

Das Ziel der vorliegenden Arbeit ist es, eine kognitive Modellierung von Kooperationsfähigkeiten für einen künstlichen Agenten vorzunehmen. Dafür gilt es eine kognitive Architektur zu entwickeln, die es ECAs erlaubt mit menschlichen Interaktionspartnern gleichberechtigt eine natürliche Kooperation einzugehen. Natürliche Interaktion wird verstanden als eine menschenähnliche, intuitive Interaktion, in der möglichst viele Modalitäten auf ähnliche Weise wie in Mensch-zu-Mensch Interaktionen eingesetzt werden können. Eine Architektur stellt in der Künstlichen Intelligenz ein abstraktes Modell dar, in welchem die statischen und dynamischen Anteile des Systems und die damit intendierten Funktionalitäten erfaßt werden;

<sup>1</sup>Bilder nach: (Breazeal et al., 2004), (Rickel & Johnson, 2000), (Cassell, 2011), (Biron, 2011)

eine kognitive Architektur dient dazu, kognitive Prozesse, die zum Kernbereich der Kognition zählen, zu modellieren (Strube et al., 1996).

Angesichts der Komplexität dieser Aufgabe wird von begrenzten Basis-Szenarien ausgegangen: Im ersten Szenario im Rahmen des SFB 360 *Situierte Künstliche Kommunikatoren* müssen ein menschlicher und ein künstlicher Interaktionspartner kooperieren, um eine Montageaufgabe - den Zusammenbau eines Flugzeugmodells aus Teilen eines Baufix-Konstruktionsbaukastens - gemeinsam zu lösen (Rickheit & Wachsmuth, 1996). Dadurch können Erkenntnisse in einem zwar domänenbezogen einfachen aber aufgabenbezogen anspruchsvollen Bereich gewonnen werden, die sich später auf andere Anwendungsbereiche übertragen lassen.

Die Übertragbarkeit und Tragfähigkeit des Modells sowie des daraus entwickelten Systems wird darüber hinaus in einem weiteren Szenario des SFB 673 *Alignment in Communication* im Rahmen des Projekts A1 *Modelling Partners* demonstriert. Unter Alignment werden Adaptations- und Koordinationsprozesse verstanden, welche während einer Kommunikation auftreten und den Kommunikationspartnern helfen, sich aufeinander einzustellen. Partnermodelle umfassen dabei Mechanismen und Repräsentationen, welche zentral für die Adaptationsprozesse im situationalen Kontext sind; dafür werden sowohl explizite Annahmen und Inferenzen berücksichtigt als auch implizite Mechanismen z.B. Aktivierungsmechanismen. Das Szenario des A1 Projekts umfasst eine Interaktion mit einem künstlichen Interaktionspartner im Rahmen eines Städte-Planungsspiel.

Damit ein künstlicher Kooperationspartner in die Lage versetzt wird, gleichberechtigt und natürlich mit einem Menschen zu interagieren, wird ein Kooperationsmodell menschlichen Verhaltens benötigt. Dieses muss neben festen Kooperationsregeln und Konventionen auch soziale Verhaltensweisen natürlichen Aufeinandereingehens umfassen. Neben Ansätzen der künstlichen Intelligenz werden daher philosophische, psychologische und kognitionswissenschaftliche Ansätze berücksichtigt. Damit zeichnet sich eine der Herausforderungen der Arbeit ab: die Integration von Modellen verschiedener Disziplinen. Zur Bewältigung der Aufgabe werden Ansätze aus den verschiedenen Bereichen kombiniert, aufeinander zurückgeführt und Brücken zwischen den verschiedenen Forschungsgebieten geschlagen.

Der Forschungsbereich der ganzheitlich konzipierten Agenten stellt einen besonders spannenden, vielversprechenden Ansatz dar, da gerade die Zusammenführung einzelner Fähigkeiten die Kompetenz eines Kooperationspartners ausmacht und so ein deutliches Mehr als nur die Summe der Einzelfähigkeiten entsteht. So stellt die Konzeption eines künstlichen, dynamisch operierenden Arbeitsgedächtnisses nicht nur einen Erkenntnisgewinn für sich dar, sondern das Arbeitsgedächtnis findet auch bei Verstehensprozessen von Äußerungen (z.B. der Referenzauflösung), bei der Äußerungsgenerierung und bei Planungsaufgaben Einsatz.

Als Lösungsansatz für das in der Zielsetzung formulierte Problem wird in der vorliegenden Arbeit als erstes ein *interaktives Kooperationsmodell* entwickelt. Das Modell beinhaltet Charakteristika der natürlichen, kooperativen Interaktion sowie deren Zusammenspiel und dient als Basis für die Entwicklung eines künstlichen Kooperationspartners. Das Modell konzentriert sich auf kooperative Prozesse verschiedener Interaktionsebenen, den interaktiven Austausch von Signalen zwischen den Interaktanten und die daraus resultierenden Verbindungen.

Um eine Basis für die Umsetzung der kooperativen Fähigkeiten zu schaffen, wird eine *kognitive Architektur* entwickelt, welche die Anforderungen des interaktiven Kooperationsmodells einlöst. Für die Entwicklung der kognitiven Architektur wird auf Arbeiten der Psychologie, Philosophie sowie der Kognitionsforschung zurückgegriffen.

Die kognitive Architektur dient abschließend als Grundlage für die *Realisierung* konkreter Kooperationsfähigkeiten im Rahmen eines virtuellen Kooperationspartners. Dieser ganzheitliche Ansatz ermöglicht die Konzeption eines natürlich agierenden, gleichberechtigten Kooperationspartners, welcher sich aktiv in Lösungsprozessen engagiert und dabei sowohl durch kommunikative Akte den Planungsprozess vorantreibt, als auch sich kooperativ durch lösungsorientierte, manipulative Handlungen an der Bewältigung der gemeinsamen Aufgabe beteiligt.

### 1.3 Struktureller Aufbau der Arbeit

Die Arbeit gliedert sich in drei Abschnitte. Im ersten Abschnitt „*Kooperationsmodellierung*“, werden theoretische Grundlagen beleuchtet und das eigene interaktive Kooperationsmodell *Behavior Conversation Task Collaboration (BCTC)* entwickelt. Aufbauend auf den aus dem Kooperationsmodell resultierenden Anforderungen an kooperative Prozesse wendet sich die Arbeit im zweiten Abschnitt „*Kognitive Architektur*“ der Modellierung kognitiver Basisprozesse sowie mentaler Zustände zu. Dieser Abschnitt schließt mit der Entwicklung der eigenen kognitiven Architektur *Cognitive Architecture for Situated Embodied Communicators (CASEC)*. Im dritten und letzten Abschnitt, Realisierung, wird gezeigt, wie basierend auf der kognitiven Architektur CASEC die kooperativen Prozesse des BCTC-Modells in einem konkreten künstlichen Kooperationspartner umgesetzt werden.

Der Abschnitt über Kooperationsmodellierung beginnt mit einer **theoretischen Analyse** (Kapitel 2). Es werden verschiedene Modelle kooperativer Aktivität diskutiert, welche Anforderungen für zunehmend komplexer werdende Formen der Kooperation beschreiben. Die Modelle werden dabei unter Zuhilfenahme eines *abstrakten Modells kooperativer Einflüsse* untersucht. Die Evaluation und Diskussion der Kooperationsmodelle führt zur Bildung eines *Modells der inter-*

*aktiven Kooperation*. Die in diesem Modell geforderten **Mechanismen der Kooperation** werden in dem darauf folgenden Abschnitt 2.2 untersucht. Die Mechanismen beziehen sich auf die Bereiche *Kommunikation, Verzahnung von Sprache und Handlung*, die Modellierung von *Obligationen* sowie soziales *Engagement*. Für die Betrachtung kommunikativer Mechanismen werden Ansätze aus dem Bereich der Dialogtheorien und der Kommunikationsforschung analysiert und Theorien des *Turn-Takings* und der Initiative Modellierung diskutiert. Nach den theoretischen Modellen werden **konkrete Systeme** unter den zuvor gesammelten Gesichtspunkten der interaktiven Kooperation vorgestellt und evaluiert. Die Analyse theoretischer Grundlagen der Kooperation unter Berücksichtigung des aktuellen Forschungsstands schließt mit einer Zusammenfassung.

Kapitel 3 präsentiert das im Rahmen der Arbeit entwickelte **Interaktive Kooperationsmodell BCTC**, welches definiert, auf welchen Ebenen und aufgrund welcher interaktiven Dynamik eine Kooperation ausgetragen wird. Dabei operieren auf der *Behavior*-Ebene Prozesse der Aufmerksamkeitssteuerung und des emotionalen Feedbacks. Die *Conversation*-Ebene hingegen ist für Fähigkeiten der Dialogmodellierung verantwortlich. Der *Task*-Ebene werden Mechanismen des dynamisch situierten Problemlösens zugeordnet. Die *Collaboration*-Ebene verarbeitet Prozesse der gemeinsamen Pläne. Es wird das Zusammenspiel der Ebenen beleuchtet und ein einheitliches Format für die Verarbeitung von Sprache und Handlung präsentiert - der Interaktionsschritt (**Interaction-Move**). Aus der Diskussion der Ebenen werden **Architektur Anforderungen** abgeleitet, welche bei der Konzeption eines künstlichen Kooperationspartners berücksichtigt werden müssen.

Der zweite Hauptabschnitt der Arbeit „*Kognitive Architektur*“ widmet sich der Modellierung eines künstlichen Kooperationspartners. In Kapitel 4 werden zunächst theoretische Grundlagen diskutiert. Einerseits werden dafür **kognitive Basisprozesse** aus einer kognitionspsychologischen Perspektive beleuchtet. In diesem Rahmen werden Modelle des **Arbeitsgedächtnisses** und der Einfluss von **Emotionen** diskutiert. Andererseits werden *Modelle mentaler Zustände*, wie sie in der Philosophie des Geistes vorkommen, hinterfragt und vorgestellt. Dabei spielt insbesondere das *Belief-Desire-Intention* (BDI)-Modell eine zentrale Rolle. Nach der Diskussion theoretischer Grundlagen werden *konkrete kognitive Architekturen* untersucht.

In Kapitel 5 wird die selbst entwickelte **CASEC-Architektur** präsentiert. Neben Wahrnehmungs- und Handlungsprozessen wird das dynamische Gedächtnismodell der CASEC-Architektur vorgestellt. Darauf aufbauend werden zielgerichtete kognitive Prozesse und Zustände der Architektur diskutiert. Abschließend wird gezeigt, wie emotionale Basisprozesse die kognitive Verarbeitung in der Architektur beeinflussen können.

Die **Realisierung eines künstlichen Kooperationspartners** *cooperative Max* (cMax) wird im dritten und letzten Hauptabschnitt der vorliegenden Arbeit dargestellt. Im Kapitel 6 werden die Realisierungen der Kooperationsfähigkeiten der BCTC-Ebenen beschrieben und jeweils durch konkrete Interaktionsbeispiele der Anwendungsszenarien verdeutlicht. Auf der *Behavior*-Ebene werden Arbeiten zur gemeinsamen Aufmerksamkeit diskutiert. Auf der *Conversation*-Ebene werden die Fähigkeiten der Dialogkompetenz und des *Turn-Takings* thematisiert. Problemlösen und die Verzahnung von Sprache und Handlung wird auf der *Task*-Ebene berücksichtigt. Zur *Collaboration*-Ebene werden Arbeiten in Bezug auf gemeinsame Pläne und gemeinsame Intentionalität vorgestellt.

Die Arbeit schließt in einem **Resümee** mit einer Diskussion und Zusammenfassung (Kapitel 7).

Teile der Arbeit sind bereits in verschiedenen Beiträgen vorveröffentlicht: (Wachsmuth & Leßmann, 2002; Leßmann & Wachsmuth, 2003; Fink et al., 2003; Kopp et al., 2003; Leßmann et al., 2004; Leßmann & Kopp, 2005; Leßmann et al., 2006; Becker et al., 2006; Pfeiffer-Leßmann & Wachsmuth, 2008, 2009).

**Teil I**

**Kooperationsmodellierung**



# Kapitel 2

## Theoretische Analyse

*Kooperation [lat.]: jede Zusammenarbeit zur Erreichung eines gemeinsamen Zwecks, wobei das Verhalten der Beteiligten durch gemeinsame Willensbildung aufeinander abgestimmt wird (dtv Lexikon, 1997).*

Kooperation stellt ein weitverzweigtes interdisziplinäres Forschungsfeld dar und wird unter verschiedenen Schwerpunktsetzungen analysiert. Während sich die Sozialwissenschaften auf die Auswirkung von Kooperationsstrategien auf die Gesellschaft (Axelrod, 1984) sowie Gesellschaftsstrukturen konzentrieren, untersuchen Psychologen und Verhaltensforscher die Ontogenese von Kooperationsverhaltensweisen (Tomasello et al., 2005). Im Bereich der Philosophie werden Anforderungen und Auswirkungen auf die mentalen Zustände der Kooperationspartner untersucht (Grice, 1969; Bratman, 1992; Searle, 1990; Tuomela, 2000). Die Informatik macht sich diese Erkenntnisse zu Nutze und baut auf den verschiedenen Kooperationsmodellen auf. Dabei haben sich verschiedene Unterdisziplinen herauskristallisiert. Die Bereiche der Multiagentensysteme und der verteilten Intelligenz konzentrieren sich auf die Kooperation zwischen künstlichen Agenten (Wooldridge & Jennings, 1994, 1999). Dabei werden Kooperationsmechanismen eingesetzt, um große Mengen von Agenten zu koordinieren und zu einem verteilten Problemlösen heranzuziehen (Jennings, 1993; Tambe, 1997; E. C. Oliveira & Mouta, 1993).

Im hier betrachteten Bereich der Mensch-Maschine-Interaktion (*Human-Computer-Interaction*) spielt Kooperation zwischen einem künstlichen System und dem Menschen die entscheidende Rolle. Es geht darum, die Interaktion zwischen Mensch und Maschine möglichst reibungslos zu gestalten. Hierbei wird oft ein zum Menschen komplementär ausgerichteter Ansatz verfolgt (Terveen, 1995). Sowohl das künstliche System als auch der Mensch werden dabei als einzigartiges System mit besonderen Fähigkeiten und Stärken gesehen. Dies führt zu einer Asymmetrie, bei welcher dem künstlichen System und dem Menschen unterschiedliche Rollen übertragen werden, sodass es keine Gleichberechtigung oder

Partnerschaft zwischen ihnen gibt. In der Künstlichen Intelligenz wird dagegen oft der Emulations-Ansatz verfolgt (Terveen, 1995). Die Modellierung menschenähnlicher Fähigkeiten und Verhaltensweisen steht dabei im Forschungszentrum. Es wird davon ausgegangen, dass hierdurch eine natürliche Interaktion gefördert wird und so eine Kooperation auf einer gemeinsamen Ebene stattfinden kann (Kinny et al., 1994).

Im Rahmen einer Kooperation kann zwischen dem Teilen von Aufgaben und dem Teilen von Ergebnissen unterschieden werden (Smith & Davis, 1988). Dabei muss modelliert werden, wie Aufgaben an andere Agenten delegiert und zugewiesen werden können und ob Informationen pro-aktiv oder reaktiv übermittelt werden. Im Falle der Aufgabenverteilung besteht der Prozess darin, das Problem in Teilprobleme zu zerlegen, individuelle Lösungen für die Unterprobleme zu finden und danach die Lösungen wieder zu einer Gesamtlösung zu integrieren.

Unter einer *Intention* wird in der vorliegenden Arbeit die Einheit eines Ziels mit einem Handlungsplan und einer damit verbundenen Handlungsabsicht verstanden. Bei der Auswahl konkreter Intentionen im Rahmen einer Kooperation müssen Rahmenbedingungen (Kontext der Umwelt, Konventionen und Verpflichtungen) und Handlungsoptionen (Koordinationsmechanismen und Kommunikation) berücksichtigt werden (Abbildung 2.1). Durch Koordination werden die

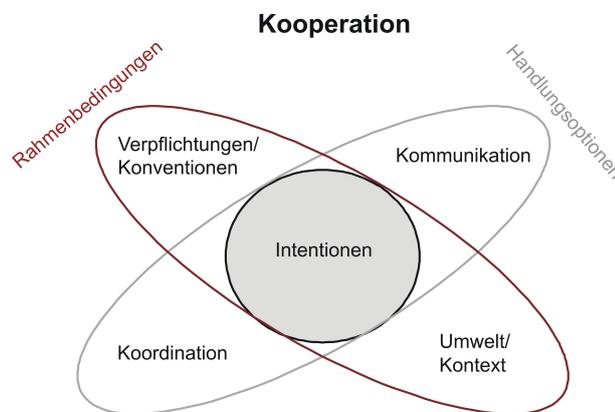


Abbildung 2.1: Kooperationseinflüsse

Abhängigkeiten und Wechselwirkungen zwischen den Aktivitäten der Agenten durch Mechanismen aufeinander abgestimmt.

Im Rahmen einer Kooperation gilt es, verschiedene Rahmenbedingungen zu berücksichtigen. Zunächst muss ein Agent die Option einer Kooperation wahrnehmen und mit einem potentiellen Kooperationspartner Kontakt aufnehmen. Der Begriff Agent beinhaltet im Folgenden sowohl künstliche Agenten als auch menschliche Interaktionspartner. Die Agenten müssen sich dann auf ein gemeinsames

Ziel einigen. Dazu zählt ferner, dass sie gemeinsam planen und festlegen, welcher Agent welche Aufgabe übernimmt und wie die Koordination stattfinden soll. In diesen Planungsprozess müssen auch der aktuelle Kontext und Umweltzustand einbezogen werden. Während der Handlungsphase der Kooperation müssen die Agenten den Fortschritt des gemeinsamen Plans sowie den Zustand des gemeinsamen Ziels verfolgen und berücksichtigen (*gemeinsamer Kontext*).

Kommunikation dient dabei sowohl dem Definieren und Aushandeln von Zielen als auch der Propagierung von Informationen bezüglich des Fortschritts des gemeinsamen Ziels. Adaptation an den Kooperationspartner spielt dabei eine entscheidende Rolle. Kooperation ist eng mit der Interaktionshistorie der Kooperationspartner verknüpft, wobei sowohl die aktuelle Interaktion als auch vergangene Zusammenkünfte von Bedeutung sind (Terveen, 1995).

Die Intentionen, welche die kooperierenden Agenten während der Planungs- und Handlungsphase bilden unterliegen zusätzlich den Einflussprozessen der Konventionen. Intentionen sorgen bei der Modellierung von Kooperationsprozessen für Stabilität und Vorhersehbarkeit, welche für soziale Interaktion entscheidend sind (Wooldridge, 2002). Insbesondere gilt es dabei auch, die Intentionen des Kooperationspartners zu berücksichtigen.

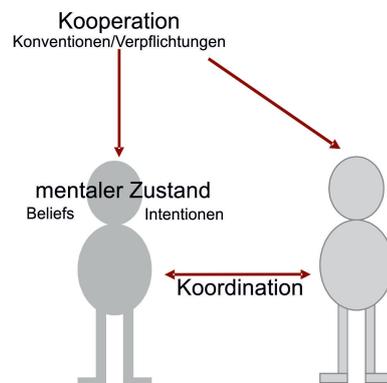


Abbildung 2.2: Abstraktes Modell kooperativer Einflüsse

Eine Kooperation entsteht durch verschiedene Einflüsse und Prozesse. Dabei lassen sich mentale Prozesse, welche in jedem Kooperationspartner nach Kooperationsregeln wirken, von seinem externen Kooperationsverhalten unterscheiden. Das offenkundige Verhalten der Kooperationspartner prägt und charakterisiert die Interaktion während einer Kooperation. Jedoch lässt sich eine Kooperation nicht vollständig durch das gezeigte Verhalten im Sinne einer Blackbox definieren. Für ein umfassendes Verständnis einer Kooperation sind die mentalen Zustände der Kooperationspartner mit einzubeziehen.

Für das in dieser Arbeit entwickelte Kooperationsmodell werden zwei Aspekte der Kooperation unterschieden, welche sich durch ihren Bezugspunkt auszeichnen. Den ersten Aspekt stellen die internen Prozesse und **mentalen Zustände** des einzelnen Agenten dar. Zu den mentalen Zuständen zählen Annahmen und Überzeugungen (*Beliefs*) des Agenten sowie seine Intentionen und Ziele. Diese werden auch durch Konventionen und Verpflichtungen, welche der Agent im Rahmen einer Kooperation eingeht, geprägt. So kann ein Agent beispielsweise Ziele verfolgen, die nicht aus seinen eigenen Wünschen entstanden sind, sondern vielmehr Verpflichtungen (*Commitments*) im Rahmen der Interaktion darstellen.

Den zweiten Aspekt bilden **Koordinierungsmechanismen**, welche die Interaktion zwischen den Agenten regeln. Dabei besteht der Bezugspunkt aus dem jeweiligen Kooperationspartner sowie den Handlungsmustern der Agenten. Während die *Commitments* abstrakte Regeln vorgeben, stellen die Koordinierungsmechanismen konkrete Verhaltensweisen der Agenten dar, die sie ausüben, um auf ihr jeweiliges Gegenüber einzugehen.

Diese Aspekte werden aus agentenzentrierter Sicht in Abbildung 2.2 als abstraktes Modell kooperativer Einflüsse zusammengefasst. Dieses Modell dient im Folgenden als Grundlage für die Diskussion verschiedener Kooperationsmodelle. Mit seiner Hilfe können die Modelle besser miteinander verglichen sowie Gemeinsamkeiten und Differenzen aufgezeigt werden.

## 2.1 Modelle der Kooperation

Im folgenden werden fünf Modelle der Kooperation vorgestellt, welche im Rahmen verschiedener Disziplinen entwickelt wurde. Das Modell der *Gemeinsamen Kooperativen Aktivität* von Bratman (siehe Abschnitt 2.1.1) kommt aus der Philosophie. Der wissenschaftliche Hintergrund von Cohen und Levesque, die ein Modell der *Teamarbeit* (siehe Abschnitt 2.1.2) vorstellen, besteht aus der Informatik insbesondere der künstlichen Intelligenz. Grosz und Hunsberger, die Entwickler des Modells der *Gemeinsamen Pläne* (siehe Abschnitt 2.1.3) arbeiten ebenfalls im Bereich der Informatik. Tomasello und Kollegen hingegen kommt aus dem Bereich der Verhaltensforschung und präsentieren ein Modell der *Gemeinsamen Intentionalität* (siehe Abschnitt 2.1.4). Der Entwickler des Modells der *Idealen Kooperation* (siehe Abschnitt 2.1.5), Allwood, ist Linguist. Die verschiedenen Modelle helfen Kooperation aus verschiedenen Blickwinkeln zu beleuchten.

### 2.1.1 Gemeinsame Kooperative Aktivität (Bratman)

Ein prozedural orientiertes Modell der Kooperation stammt von Bratman (1992) und bezieht sich auf den Prozess einer **gemeinsamen kooperativen Aktivität**

(*shared cooperative activity*). Dabei legt Bratman einen besonderen Schwerpunkt auf die Ausführungsphase der Kooperation. Er unterscheidet den Prozess gemeinsamer kooperativer Aktivität von einer im Vorhinein ausgehandelten Kooperation, bei welcher zwar ein kooperativer Plan gemeinsam aufgestellt wird, während der Ausführung des Plans jedoch nicht im Sinne eines gegenseitigen Aufeinandergehens zusammen gearbeitet wird. Konkret stellt Bratman drei Charakteristika einer gemeinsamen kooperativen Aktivität auf (Bratman, 1992):

- gegenseitiges Aufeinandergehen (*mutual responsiveness*)
- Verpflichtung zur gemeinsamen Aktivität (*commitment to joint activity*)
- Verpflichtung zur gegenseitigen Unterstützung (*commitment to mutual support*)

Das **gegenseitige Aufeinandergehen** bezieht sich dabei sowohl auf Intentionen als auch auf Handlungen des Interaktionspartners. Es geht darum, die eigenen Handlungen mit Blick auf das Verhalten des Anderen auszurichten. Zusätzlich herrscht gemeinsames Wissen darüber, dass der Interaktionspartner sich auf die gleiche Art und Weise verhält. Das gegenseitige Aufeinandergehen stellt ein Streben im Dienste des gemeinsamen Ziels dar. Es dient als Koordinierungsmechanismus und sichert das kooperative Zusammenspiel.

Die **Verpflichtung zu einer gemeinsamen Aktivität** beinhaltet die Existenz einer wohlfundierten Intention in Bezug auf die gemeinsame Aktivität. Dabei kann der Grund der Intention bei den einzelnen Agenten differieren. Es muss nur die Vorgabe erfüllt sein, dass alle beteiligten Agenten die gemeinsame Aktivität anstreben. Die Verpflichtung zur **gegenseitigen Unterstützung** während der Ausführung beinhaltet, dass die Interaktionspartner ihrem Partner helfen, damit er seinen Teil der Aktivität erfüllen kann. Dazu müssen sie sich laut Bratman auch in die Rolle des Kooperationspartners eindenken können.

Eine weitere wichtige Eigenschaft besteht laut Bratman aus dem **Ineinandergreifen und Zusammenführen von Teilplänen** (*meshing of subplans*) nach geltenden Kompatibilitätsbedingungen. Die einzelnen Teilpläne der gemeinsamen Aktivität der Agenten müssen nicht komplett miteinander übereinstimmen, es dürfen jedoch keine unlösbaren Widersprüche zwischen ihnen bestehen.

Bratman argumentiert, dass ein **ineinander greifendes Netz von eigenen Beliefs sowie gemeinsamer Beliefs und gewöhnlichen Intentionen** (Bratman, 1999) ausreichend für die Modellierung kollektiver Intentionalität sei, vorausgesetzt die Gruppe besitzt eine gemeinsame Intention in Übereinstimmung mit und aufgrund ihrer individuellen Intentionen unter Berücksichtigung der ineinander greifenden Teilpläne.

Die Einordnung der Charakteristika nach Bratman in das abstrakte Modell kooperativer Einflüsse (siehe Abbildung 2.2) zeigt Abbildung 2.3. Das Modell

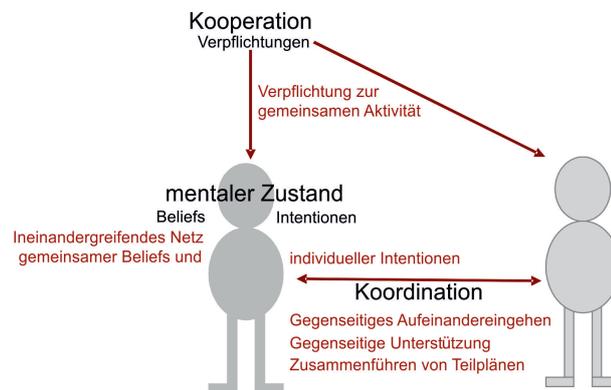


Abbildung 2.3: Gemeinsame kooperative Aktivität - Bratman

Bratmans spielt bei der Betrachtung von Kooperationsmodellen eine besondere Rolle, da viele neuere Modelle auf die Arbeiten Bratmans aufbauen. Insbesondere das Verknüpfen von Teilplänen und das unmittelbare Aufeinandereingehen stellen zentrale Punkte für ein Kooperationsmodell dar, welche von anderen Modellen nicht in diesem Detaillierungsgrad berücksichtigt werden.

### 2.1.2 Teamarbeit (Cohen und Levesque)

Im Gegensatz zu Bratman, der sein Hauptaugenmerk auf das Konzept einer gemeinsamen Aktivität ausrichtet, entwickeln P. Cohen & Levesque (1991) ein idealisiertes Konzept der *Teamarbeit* für die direkte Umsetzung in einem künstlichen Agenten. Sie konzentrieren sich darauf, Anforderungen an den mentalen Zustand eines kooperierenden Agenten zu definieren, wobei sie ein Team von Agenten als eine Art aggregierten Agenten ansehen (P. Cohen et al., 1998).

Die Definition eines gemeinsamen, **persistenten Ziels** mit abgeschwächter Erreichbarkeitsbedingung stellt einen zentralen Punkt ihres Modells dar. Nach P. Cohen & Levesque (1991) hat ein Team von Agenten dann ein *gemeinsames, persistentes Ziel* relativ zu  $q$ , um  $p$  zu erreichen, wenn gilt:

- alle glauben gemeinsam, dass  $p$  aktuell nicht gilt
- alle wissen gemeinsam, dass alle wünschen, dass  $p$  letztendlich wahr wird
- alle glauben so lange, dass sie  $p$  als Ziel mit abgeschwächter Erreichungsbedingung relativ zu  $q$  haben, bis sie gemeinsam glauben, dass entweder
  - $p$  gilt oder
  - $p$  niemals wahr sein wird oder

- $p$  irrelevant ist ( $q$  ist falsch)

Besonders zu beachten sind dabei die eingebetteten impliziten Verpflichtungen, den Zustand des gemeinsamen Ziels mitzuteilen (**Mitteilungsverpflichtung**), sowie die Möglichkeit der Spezifikation einer **Abbruchbedingung**. Diese gibt an, unter welchen Umständen das gemeinsame Ziel aufgegeben werden darf. Die Verpflichtung, **gemeinsames Wissen** zu erlangen, kann als zusätzlicher Teilaspekt des gemeinsamen Ziels gesehen werden. Diese Verpflichtung prognostiziert, dass Kommunikation zwischen den Agenten stattfinden wird, da dies der allgemeine Weg ist, Wissen zu propagieren. **Kommunikation** zwischen Teammitgliedern ist sowohl hilfreich als auch notwendig, da die Mitglieder eines Teams oft nur partielle Information zur Lösung des Problems besitzen und über unterschiedliche Fähigkeiten bei der Ausführung verfügen. Auf der Definition des gemeinsamen, persistenten Ziels aufbauend charakterisieren Cohen und Levesque das *gemeinsame Intendieren* der Agenten:

**Definition** Ein Team von Agenten *intendiert gemeinsam* eine Aktion  $p$  relativ zu einer Abbruchbedingung  $q$ , wenn alle Mitglieder des Teams ein gemeinsames, persistentes Ziel relativ zu der Bedingung  $q$  haben, die Aktion gemeinsam ausführen und dabei auch während der gesamten Aktion gemeinsam glauben, dass sie dies gemeinsam tun.

P. Cohen & Levesque (1990) führen die Definition der **gemeinsamen Intentionen** und das zugehörige *Commitmentverhalten* nicht wie Bratman auf die Definition individueller Intentionen zurück. Stattdessen definieren sie beide Intentionenformen mit Hilfe derselben Primitive (Modaloperatoren *BELIEF*, *GOAL*, *HAPPENS*, *DONE* siehe auch Anhang A.1), wobei die individuellen Intentionen aus der Definition der gemeinsamen Intentionen resultieren. Cohen und Levesque sehen gemeinsame Intentionen als eine Eigenschaft einer Gruppe von Agenten. In einer gemeinsamen Handlung teilen alle Teammitglieder sowohl das gemeinsame Ziel als auch einen gemeinsamen Plan für die Ausführung. Nach Cohen und Levesque kann ein Agent nicht eine eigene Intention formen, dass ein anderer Agent eine bestimmte Aktion ausführt, er kann jedoch in Bezug auf die Aktionen Anderer Verpflichtungen eingehen.

Cohens und Levesques Definition stellt konkrete Anforderungen an den mentalen Zustand der kooperierenden Agenten (siehe Abbildung 2.4). Dabei liegt ein besonderer Schwerpunkt auf der Modellierung gemeinsamer Intentionen und gemeinsamer Verpflichtungen.

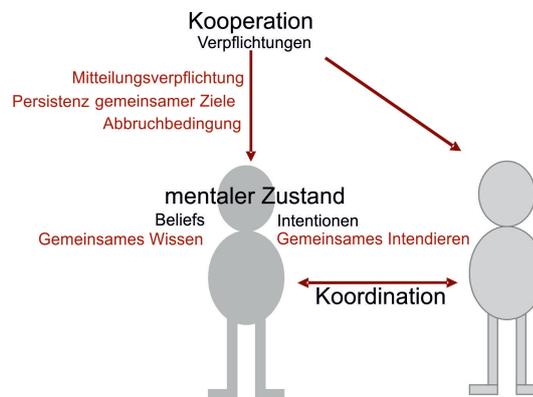


Abbildung 2.4: Teamarbeit - Cohen und Levesque

### 2.1.3 Gemeinsame Pläne (Grosz und Hunsberger)

Die *SharedPlans*-Theorie stellt eine Formalisierung der mentalen Zustände kooperierender Agenten während der Erstellung eines gemeinsamen Plans dar (B. Grosz & Kraus, 1993; B. J. Grosz & Kraus, 1998). In neueren Arbeiten wird ein Schwerpunkt auf die **Dynamik der Intentionen** während eines Kooperationsprozesses gelegt (B. Grosz & Hunsberger, 2006). Damit nähern sich Grosz und Hunsberger dem Phänomen der Kooperation und den damit verbundenen Prozessen von einer weiteren Seite. In ihren Arbeiten führen sie den *SharedPlans*-Ansatz mit Arbeiten der Intensionsdynamik zusammen. Abstrakt gesprochen postulieren B. Grosz & Hunsberger (2006), dass folgende *Beliefs* und *Intentionen* einen *SharedPlan* ausmachen:

#### Definition 1 *SharedPlans*

- (SP 1) *individuelle Intention, dass die Gruppe Aktivität a ausführt*
- (SP 2) *gemeinsamer Belief eines (eventuell unvollständigen) Rezepts für a*
- (SP 3) *individuelle oder Gruppenpläne für die Aktionen dieses Rezepts*
- (SP 4) *Intentionen, dass ausgewählte Agenten oder Untergruppen erfolgreich die ihnen zugewiesenen Aktionen ausführen*
- (SP 5) *im Falle eines unvollständigen SharedPlans individuelle Intentionen, dass die Gruppe den Plan vollendet*

Der *SharedPlans*-Formalismus spezifiziert die Anforderungen an die mentalen Zustände der beteiligten Interaktanten einer kooperativen Gruppenaktivität.

Eine konkrete Formalisierung beinhaltet dabei zwei unterschiedliche Repräsentationen intentionaler Einstellungen und die damit verbundenen Verpflichtungen, welche die Beteiligten im Rahmen der gemeinsamen Aktivität eingehen. Die eine Form besteht aus der Intention, eine Aktion auszuführen (*intention-to*). Die zweite Form verlagert den Schwerpunkt vom prozeduralen Aspekt auf einen zustands-orientierten Ansatz, nämlich auf die Intention, dass eine Proposition gilt (*intention-that*). Genauso wie die prozessorientierten Intentionen mit der Mittel-Ziel-Analyse (*Means-Ends-Reasoning*) assoziiert sind, stehen die zustandsorientierten Intentionen in einem engen Bezug zu einem Bildungs- bzw. Kultivierungsprozess (B. J. Grosz & Kraus, 1998).

Im Kontext einer kooperativen Aktivität zeichnen sich Gruppenentscheidungen durch Obligationen aus, welche allen Beteiligten auferlegt werden. Es lassen sich dabei verschiedene Formen der Gruppenentscheidungen unterscheiden: die Auswahl eines Rezepts, die Zuteilung einer Aufgabe und das Festlegen eines bestimmten Parameterwertes. Alle diese Entscheidungen müssen im Kontext der existierenden Gruppenintention getroffen und verarbeitet werden. Die Interaktion zwischen Rezeptauswahl und Aufgabenverteilung führt typischerweise zu einer komplexen Hierarchie von untergeordneten Intentionen, welche zusammen mit den gemeinsamen Beliefs den *SharedPlan* ausmachen. Die Bedingung der koordinierten Intensionskultivierung **Coordinated Cultivation Requirement (CCR)** besagt, dass die Mitwirkenden einer kooperativen Gruppenaktivität die gemeinsame Gruppenintention nur in Abstimmung mit der Gruppe aktualisieren dürfen.

Wie auch Bratman (2.1.1) betonen Grosz und Kollegen, dass keine neue Art der Intention benötigt wird, um kollektive Aktionen und gemeinsame Intentionen zu charakterisieren. Im Gegensatz zu anderen Forschern (Searle, 1990; Tuomela, 2000) ist ihr Modell darauf ausgerichtet, die Intentionen so in die Agentenarchitektur einzubetten, dass der Inhalt gemeinsamer Intentionen in gleicher Form repräsentiert werden kann wie der Inhalt individueller Intentionen, jedoch zusätzliche Mechanismen und Obligationen dafür sorgen, den Gruppenaspekt umzusetzen. Die Trennung des Inhalts einer Intention von den Einschränkungen, wie der Inhalt aktualisiert werden kann, ermöglicht den Einsatz des gleichen *Means-Ends-Reasoners*, welcher auch bei individuellen Intentionen eingesetzt werden kann.

Durch die Explizitmachung des dynamischen Zusammenspiels von Gruppenentscheidungsprozessen und den Regeln bezüglich der Aktualisierung agenteninterner Intentionen füllt das Modell der *Coordinated Cultivation of SharedPlans* eine wichtige Lücke in bestehenden Ansätzen der kooperativen Intentionalität. Während der *SharedPlans*-Ansatz im Bereich des kooperativen Planens einen wichtigen Beitrag leistet, ist er nicht ausgelegt für den Einsatz der Modellierung der Kooperation, die während der Ausführung des Plans entsteht (Blaylock, 2005). Er konzentriert sich auf das Modellieren von Einschränkungen, die während der Erstellung eines gemeinsamen Plans gelten. Wie ein Team gebildet wird und wel-

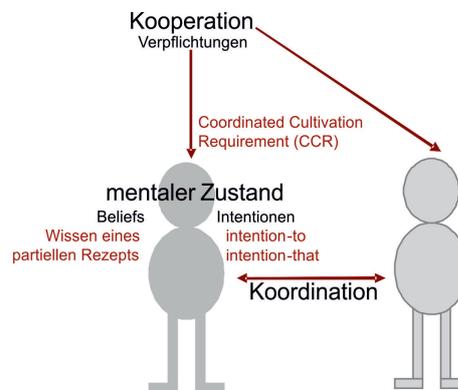


Abbildung 2.5: Gemeinsame Pläne - Grosz und Hunsberger

che Anforderungen und Mechanismen während der Ausführung eines gemeinsamen Plans gelten, wird in dem Ansatz nicht berücksichtigt (Nguyen & Wobcke, 2005).

#### 2.1.4 Gemeinsame Intentionalität (Tomasello et al.)

Von Tomasello et al. (2005) stammt ein Ansatz der Kooperation, der sich auf die Annahme einer *gemeinsamen Intentionalität* bezieht. Danach stellt gemeinsame Intentionalität die sozial-kognitiven sowie sozial-motivationalen Fähigkeiten einer kooperativer Interaktion dar. Zusätzlich gilt, dass im Rahmen dieser Interaktion die Beteiligten psychologische Zustände mit den anderen Teilnehmern teilen (Gilbert, 1989; Tuomela, 1995; Searle, 1995). Tomasello und Kollegen bauen auf den Arbeiten von Bratman (siehe Abschnitt 2.1.1) auf, setzen den Schwerpunkt der Kooperationsbetrachtungen aber auf die Beschreibung des **mentalen Zustands** der Interaktanten.

Gemeinsame Intentionalität stellt ein wahres inter-subjektives Teilen dar. Die damit verbundene gemeinsame Aufmerksamkeit erzeugt eine gemeinsame psychologische Basis, den **psychologischen Common Ground**, welcher kooperative Aktivitäten zu gemeinsamen Zielen sowie kooperative Kommunikation ermöglicht (Tomasello & Carpenter, 2007). Dabei bezieht sich **gemeinsame Intentionalität** auf kooperative Interaktionen, in welchen die Beteiligten ein gemeinsames Ziel teilen und koordinierte Handlungspläne verfolgen, um das Ziel gemeinsam zu erreichen. Die Aktivitäten der Kooperation selbst können dabei sowohl einfach als auch komplex ausfallen. Entscheidend ist, dass die Beteiligten der Interaktion auf eine bestimmte Art und Weise miteinander verbunden sind und sich in die Kooperation einbringen. In diesem Kontext fordern Tomasello et al. (2005), dass die Ziele und Intentionen der Interaktanten jeweils **Repräsentation**

**tionen der Ziele und Intentionen** des Kooperationspartners enthalten müssen.

Ferner stellen sie die Bedingung auf, dass die Beteiligten der Kooperation gemeinsame Pläne **holistisch** repräsentieren, indem sämtliche Rollen der Kooperationspartner in dem Plan enthalten und jedem Beteiligten bekannt sind. Dies ermöglicht gegenseitige Unterstützung und vereinfacht auch gegebenenfalls einen Rollentausch während der Kooperation (Tomasello et al., 2005). Abbildung 2.6 zeigt eine Einordnung der Anforderungen in das abstrakte Modell kooperativer Einflüsse.

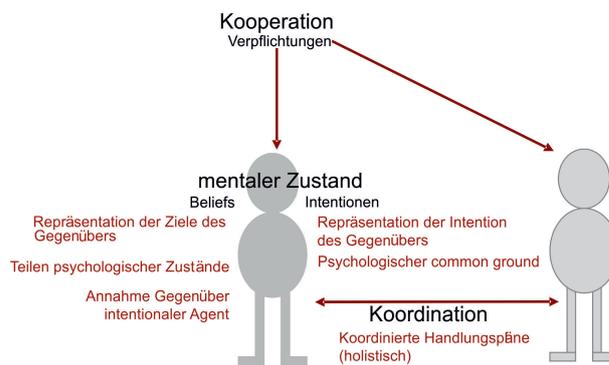


Abbildung 2.6: Gemeinsame Intentionalität - Tomasello et al.

Tomasello & Carpenter (2007) trennen klar zwischen koordinierten, sozialen Aktivitäten und gemeinsamer Intentionalität. Sie betonen, dass *gemeinsame Intentionalität* zwar auf den ersten Blick nur einen kleinen psychologischen Unterschied zu koordinierten sozialen Aktivitäten darstellt, dass gemeinsame Intentionalität jedoch ein entscheidender Fortschritt in der menschlichen Evolution ist. Durch das Hinzukommen gemeinsamer Intentionalität verwandelt sich Blickverfolgung in gemeinsame Aufmerksamkeit, soziales Lernen in instruiertes Lernen, soziale Manipulation in kooperative Kommunikation sowie Gruppenaktivitäten in Kooperation. Tomasello und Carpenter sehen in der Fähigkeit des Menschen, sich in kooperativen Aktivitäten mit gemeinsamen Zielen sowie sozial koordinierten Handlungsplänen zu engagieren, den gravierendsten Unterschied zwischen menschlicher Kognition und der Kognition anderer Lebewesen.

### 2.1.5 Ideale Kooperation (Allwood)

Allwood nähert sich dem Phänomen der Kooperation über die Analyse natürlicher Dialoge. Ausgehend von einer allgemeinen Definition idealer Kooperation (Allwood, 1976) nennt er die folgenden Charakteristika (Allwood, 2001). Zwei oder

mehr Agenten (im Folgenden A und B) können in dem Grad als kooperierend bezeichnet werden, in dem sie

- (1) sich gegenseitig in der Interaktion kognitiv in Betracht ziehen und **aufeinander eingehen**,
- (2) ein **gemeinsames Ziel** besitzen,
- (3) sich während der Interaktion gegenseitig **ethisch** und **moralisch berücksichtigen** und

sich gegenseitig vertrauen, nach (1)-(3) zu handeln.

Eine *ideale Kooperation* liegt vor, wenn alle vier Anforderungen erfüllt werden. Allwood betont jedoch, dass Kooperation kein Alles-oder-Nichts-Phänomen ist, sondern graduell vorliegen kann. Die erste Anforderung stellt eine Basisanforderung der Kooperation dar und bedeutet, dass A versuchen sollte, das wahrzunehmen und zu verstehen, was B macht, und seine Beiträge zu der Kooperation so zu gestalten, dass sie in Bezug darauf eine Relevanz besitzen. Die zweite Anforderung bezieht sich darauf, dass sowohl globale als auch lokale Zielsetzungen gemeinsam verfolgt werden. Die dritte Anforderung legt fest, dass A versuchen sollte, es B zu ermöglichen, die Interaktion als rational motivierter Agent fortzusetzen. Das heißt, B sollte die richtigen Information haben und es sollte B erlaubt sein, so frei wie möglich zu agieren. Dies kann erreicht werden, indem die Interessen des Anderen berücksichtigt werden und ihm beispielsweise unterstützendes Feedback gegeben wird. Die vierte Anforderung impliziert, dass A nicht nur versucht, die gemeinsame Zielsetzung zu verfolgen, während A sich an Anforderung 1 und 3 hält, sondern zusätzlich auch noch glaubt, dass B das Gleiche tut, und darauf vertraut, dass Bs Verhalten in seinem Interesse ist.

Allwood hebt zwei zentrale Konzepte hervor, welche in enger Beziehung mit Kooperationsbeziehungen stehen: **Flexibilität** und **Konfliktvermeidung**. In beiden Fällen spielen kognitive und ethische Berücksichtigungen eine Rolle. Er stellt eine Liste von Wegen vor, welche die Beteiligten einer Kooperation beschreiten können, um zu der Flexibilität und Konfliktvermeidung während einer Kooperation beizutragen. Zu diesen zählen:

- gegenseitiges Wohlwollen,
- Feedback, welches gegenseitige Unterstützung und Übereinkunft ausdrückt,
- das Bekunden von Berücksichtigung und Interesse,

- und das Hervorrufen von gemeinsamer Aufmerksamkeit und gemeinsamen Überzeugungen

Allwood betont, dass in Dialogen die Teilnehmer durch ihre Beiträge kooperieren und diese in direkter sprachlicher Interaktion multimodal seien, d.h. Beiträge werden beispielsweise durch ein Nicken ergänzt oder sogar ersetzt, insbesondere im Bereich des Feedbacks (Kopp et al., 2008). Auch werden Emotionen durch Prosodie und Gesten oft den verbalen Nachrichten hinzugefügt, wodurch zusätzlich eine Übermittlung von affektiven Informationen gewährleistet wird.

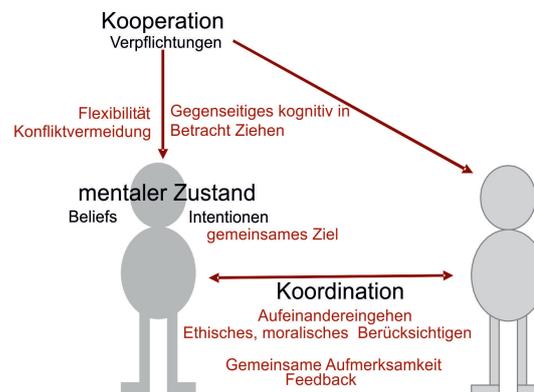


Abbildung 2.7: Ideale Kooperation - Allwood

Ein Schwerpunkt in Allwoods Modell liegt darauf, Kooperationen möglichst effizient und angenehm zu gestalten. In Abbildung 2.7 sind die zentralen Mechanismen und Bedingungen einer idealen Kooperation nach Allwood hervorgehoben. Allwood betrachtet die Anforderungen an das Kooperationsverhalten eines Agenten im Rahmen eines Dialogs mit Blick auf die Modellierung kooperativer Dialogsysteme und Avatare.

### 2.1.6 Zusammenfassung und Modellbildung

Die diskutierten Modelle kooperativer Aktivität stellen Anforderungen an eine immer komplexer werdende Form der Kooperation. Im Folgenden wird ein Kooperationsmodell gebildet, das die wichtigsten Anforderungen in einem gemeinsamen Rahmen repräsentiert sowie ihr Zusammenspiel aufzeigt.

Die einzelnen Kooperationsmodelle unterscheiden sich in ihrer Herangehensweise und den Schwerpunkten, die sie in ihren Formalisierungen setzen. Während das Modell von **Bratman** einen Schwerpunkt auf die Prozesse legt, welche während der Ausführung der Kooperation stattfinden, und damit die Koordination der kooperativen Handlungen in den Mittelpunkt rückt, hebt der Ansatz von

**Cohen und Levesque** in besonderem Maße die Verpflichtungen hervor, welche die Kooperationspartner eingehen. Insbesondere folgen aus den von **Cohen und Levesque** definierten Verpflichtungen kommunikative Handlungen. Wie solche kommunikativen Handlungen umgesetzt werden können, eventuell sogar multi-modal, beschreibt **Allwood** in seinem Modell der Kooperation. Der Ansatz der gemeinsamen Pläne von **Grosz und Kollegen** sowie die Überlegungen zur gemeinsamen Intentionalität von **Tomasello und Kollegen** konzentrieren sich in einem stärkeren Maße auf den mentalen Zustand der Kooperationspartner. Dabei dient der *SharedPlans*-Ansatz schwerpunktmäßig der Modellierung dynamischer Intentionen. **Tomasello und Kollegen** wenden sich in besonderer Weise den Überzeugungen der Interaktanten zu und fordern, dass in ihren Überzeugungen auch Inhalte über die Überzeugungen und Intentionen der Kooperationspartner sowie die Rolle der Interaktionspartner in gemeinsamen Plänen repräsentiert sein müssen. Abbildung 2.8 stellt die Einfluss Schwerpunkte der vorgestellten Kooperationsmodelle dar.

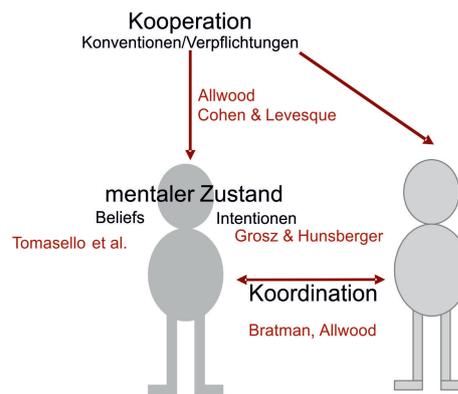


Abbildung 2.8: Einfluss Schwerpunkte der Kooperationsmodelle

Die Analyse der einzelnen Anforderungen und Definitionen führt zur Bildung eines **Modells der interaktiven Kooperation** (Abbildung 2.9). Die Graphik veranschaulicht das Modell. Sie zeigt die Prozesse der Interaktion, die beim kooperativen Verhalten zweier Agenten ablaufen, sowie die Verpflichtungen und Anforderungen an den mentalen Zustand der Agenten. Die Agenten müssen sich an **Kooperationsregeln** und damit verbundene Verpflichtungen halten und ihr Verhalten dem Kooperationsverlauf entsprechend ausrichten.

Die fett gekennzeichneten Begriffe in der mittleren Spalte der Graphik (*gemeinsame Intentionalität, gegenseitige Unterstützung, situierete Kommunikation, koordiniertes Planen und Handeln, gegenseitiges Aufeinandereingehen*) beschreiben die Prozesse, welche zwischen den Interaktionspartnern im Rahmen einer

Kooperation stattfinden. Sie stellen die **fünf zentralen Mechanismen** einer Kooperation dar. Umgesetzt werden sie unter Zuhilfenahme der groß hervorgehobenen Konzepte und Prozesse (soziale Faktoren insbesondere daraus resultierende Verpflichtungen, Kommunikation und Engagement). Finden kooperative Mechanismen im Rahmen aller spezifizierten Konzepte Einsatz, so wird die Kooperation zwischen den Agenten in diesem Modell als *echte Kooperation* klassifiziert. Diese zeichnet sich durch eine ganzheitliche Kooperationsbereitschaft und Handlungsweise aus. Zum Eingehen einer echten Kooperation gelten demnach folgende Maxime:

**Definition 2 *Echte Kooperation***

- (k1) *gegenseitiges Aufeinandereingehen (mutual responsiveness)*
- (k2) *situierte Kommunikation (situated communication)*
- (k3) *koordiniertes Planen und Handeln (coordinated planning and acting)*
- (k4) *gegenseitige Unterstützung (mutual support)*
- (k5) *gemeinsame Intentionalität (shared intentionality)*

Das gegenseitige Aufeinandereingehen schafft eine Basis für die Kooperation; es bezieht sich auf Koordinierungsprozesse und das Teilen von Verhaltensweisen. Situierte Kommunikation ist für das Aushandeln von Aufgaben und die Kommunikation von Konflikten zentral. Durch koordiniertes Planen und Handeln können Aufgaben durch eine verschachtelte Form von Planungs- und Handlungsprozessen angegangen werden. Die Versicherung der gegenseitigen Unterstützung impliziert, dass sich die Agenten helfen, um gemeinsam Ziele zu erreichen. In Bezug auf gemeinsame Intentionalität gilt es, die eigenen Ziele sowie die Ziele des Partners holistisch zu repräsentieren.

In der Kommunikation zwischen den Agenten werden die spezifischen Informationen für die Kooperation vermittelt. Ohne Kommunikation kann eine Kooperation nicht ausgehandelt, geplant und koordiniert werden. Dabei kann es zwischen den Kooperationspartnern durchaus Kompetenzunterschiede geben. Zu einer echten Kooperation gehört, dass jeder beteiligte Agent jederzeit nach seinem Vermögen zu einer gemeinsamen Lösungsfindung beitragen und dynamisch eingreifen kann, falls etwas nicht richtig läuft. Eine solche Kooperation zeichnet sich durch Gleichberechtigung zwischen den Interaktionspartnern und einer verteilten Initiative aus (*mixed-initiative*). Deshalb werden im Folgenden Dialogmodelle

(Abschnitt 2.2.1), Diskursstrukturen (Abschnitt 2.2.3) sowie Initiativ- (Abschnitt 2.2.3) und Turn-Taking-Modellierung (Abschnitt 2.2.2) diskutiert.

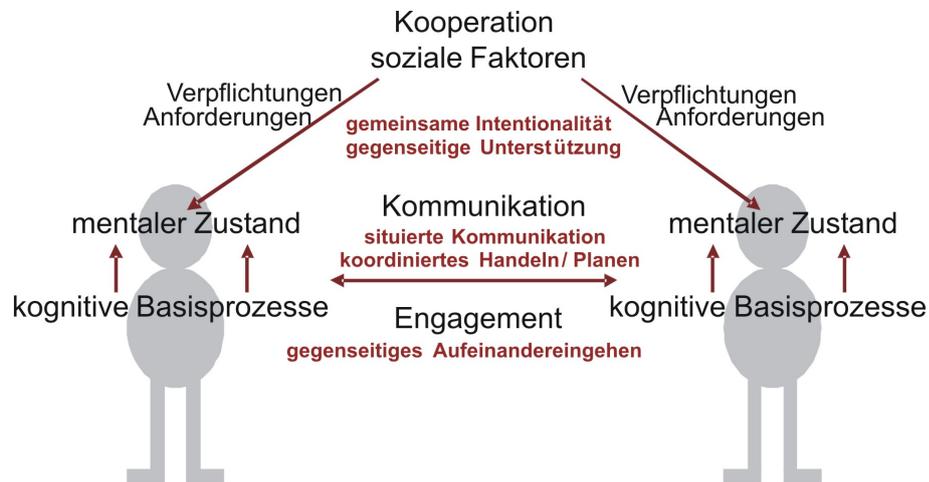


Abbildung 2.9: Modell der interaktiven Kooperation

In Bezug auf soziale Faktoren, die es im Rahmen einer Kooperation zu berücksichtigen gilt, spielen Verpflichtungen eine entscheidende Rolle. Obligationen als Teil des mentalen Zustands eines Agenten stellen explizite Mechanismen dar, ein Maß an Stabilität und Verlässlichkeit in eine Interaktion zu bringen, während sie den einzelnen Agenten dennoch Entscheidungsfreiheiten und Flexibilität zubilligen (siehe Abschnitt 2.2.5).

Im Rahmen einer Kooperation kommt es auf verschiedenen Ebenen zu Interaktionsprozessen zwischen den Kooperationspartnern. Die Kooperationsanforderungen und Verpflichtungen regeln die grundlegende Struktur, während *Engagement* die Ausdrucksformen und konkreten Verhaltensweisen der Kooperationspartner kennzeichnet (siehe Abschnitt 2.2.6). Es ist eng mit den Konzepten des gegenseitigen Aufeinandereingehens, der kognitiven und ethischen Berücksichtigung der Kooperationspartners sowie der Aufrechterhaltung der Flexibilität verwandt.

## 2.2 Mechanismen der Kooperation

Im Folgenden werden die zuvor aufgestellten Mechanismen der Kooperation genauer betrachtet und die relevante Forschungsliteratur diskutiert. In Anschluss daran werden konkrete Systeme unter den aus dieser Diskussion gewonnenen Erkenntnissen analysiert.

### 2.2.1 Kommunikation

Kommunikation stellt ein zentrales Mittel der Kooperation dar (siehe Cohen und Levesque (2.1.2), Grosz und Hunsberger (2.1.3), Allwood (2.1.5)). Damit ein Dialog in geordneten Bahnen verläuft, muss eine kooperative Aktivität zwischen den Agenten bestehen (McTear, 2002). Diese bezieht sich auf Prozesse, welche die Konversationspartner einsetzen, um den Dialog aufrecht zu erhalten und möglichst effizient abzuwickeln. Zu diesen Prozessen zählen auf der interaktionalen Ebene *Turn-Taking* sowie auf der inhaltlichen Ebene *konversationales Grounding*. McTear (2002) definiert einen Dialog als eine gemeinsame, kooperative Handlung zwischen zwei oder mehr Beteiligten. Damit sind Kooperation und Dialog eng miteinander verknüpft.

In der Literatur werden verschiedene Dialogmodelle diskutiert. Traditionelle Ansätze der Dialogmodellierung lassen sich in Zustands-basierte (*state-based*) und Rahmen-basierte (*frame-based*) Ansätze einteilen (McTear, 2002). Die zustands-basierten Ansätze bauen auf explizit repräsentierten Dialogzuständen auf und definieren, wie durch eine Dialoggrammatik und einen Dialogschritt (*dialogue move*) ein Zustand in einen anderen Zustand übergeführt werden kann. Sie werden in Szenarien eingesetzt, in welchen mögliche Dialogverläufe im Vorhinein bestimmt werden können. Rahmen-basierte Ansätze werden ebenfalls in Szenarien eingesetzt, in welchen im Vorhinein feststeht, welche Informationen behandelt werden müssen; sie zeichnen sich aber durch eine größere Flexibilität aus. Zustands-basierte Ansätze werden zumeist für einfache geskriptete Dialoge eingesetzt.

Systeme, in welchen der Dialogverlauf dynamisch aus der Interaktion durch Entscheidungsprozesse der Konversationspartner entsteht, werden oft als *Agenten-basierte* Ansätze konzipiert. Agenten-basierte Ansätze sehen Dialog als kooperativen Prozess zwischen einem oder mehreren Agenten. Um das Dialogverhalten der Agenten zu modellieren, werden den Agenten mentale Einstellungen wie *Beliefs*, Ziele, Pläne, sowie *Commitments* und Obligationen zugeschrieben (Traum, 1996). In Plan-basierten Ansätzen werden Äußerungen in Form von Sprechakten genauso wie Handlungen in einem Planungssystem behandelt. Plan-basierten Ansätze werden jedoch oft dafür kritisiert, undurchsichtig zu sein und keine Semantik für die Planoperatoren aufzuweisen.

#### Information State Approach

Der *Information State Approach* kombiniert struktur-basierte und plan-basierte Ansätze und vereinigt ihre Vorteile (Traum & Larsson, 2003). Dabei enthält der *Information State Approach* Aspekte eines Dialogzustands, erlaubt aber auch die Berücksichtigung mentaler Aspekte (z.B. Intentionen der Interlokutoren). Der An-

satz führt zu einer deklarativen Repräsentation des Dialogs, erscheint durch seine klare Struktur aber transparenter als prozedurale Ansätze. Der *Information State Approach* konzentriert sich darauf, Informationen, welche den Zustand und die Modellierung des Dialogsverlaufs betreffen, in einer strukturierten Weise zu repräsentieren. Er beinhaltet dabei die folgenden Komponenten (Larsson & Traum, 2000), (Traum & Larsson, 2003):

- (1) Informationskomponenten: Beschreibung der Elemente, die im *Information State* (IS) repräsentiert werden
- (2) formale Repräsentation der Informationen
- (3) Menge von *dialogue moves*, welche die Aktualisierung des IS anstoßen
- (4) Menge von Regeln, welche spezifizieren, wie der IS aktualisiert wird
- (5) Kontrollstrategie

Die Informationskomponenten umfassen beispielsweise das Wissen und die Annahmen der Konversationsteilnehmer, insbesondere Informationen darüber, welches Wissen die Konversationsteilnehmer teilen und darüber hinaus ihre relevanten Intentionen, Ziele und Obligationen. Es wird dabei angenommen, dass jeder Agent ISs repräsentiert, welche im Entscheidungsprozess darüber, welche Handlung als nächstes vorgenommen werden soll, eingesetzt werden. Die ISs werden unter Berücksichtigung der Dialogakte beider Interaktanten schritthaltend aktualisiert. Der IS wird als allgemeiner Bezugspunkt für die Interpretation, die Generierung und die Aktualisierung des Dialogzustands eingesetzt. ISs können sowohl die Information eines als auch den Wissensstand mehrerer Konversationsteilnehmer umfassen; sie können sogar eingesetzt werden, um eine externe Sicht auf den Verlauf der Konversation zu repräsentieren.

Der *Information State Approach* formalisiert keine konkrete Interaktionstheorie, sondern stellt ein Rahmenwerk zur Verfügung, welches Strukturen vorgibt, mit deren Hilfe Dialogmodelle gebildet werden können. Durch den gemeinsamen Rahmen können Modelle, die dem *Information State Approach* folgen, gut miteinander verglichen und evaluiert werden. Mehrere Systeme beruhen auf dem Framework der *Information States* und wenden sich dabei unterschiedlichen Aspekten des Dialogs zu z.B., das GODIS System (*Gothenburg Dialogue System*) (Larsson et al., 2000) oder das WITAS System (*Wallenberg laboratory for research on Information Technology and Autonomous Systems*) (Lemon et al., 2001). Auch das MRE System (*Mission Rehearsal Experience*) setzt den *Information State Approach* ein und erweitert und konkretisiert ihn dabei (siehe auch Abschnitt 2.3).

**Dialogmodell im Rahmen des *Mission Rehearsal Exercise* (MRE)**

Während der *Information State Approach* eine allgemeine Strukturierung vorgibt, stellt das MRE-Modell von Traum & Rickel (2002) eine Konkretisierung und Weiterentwicklung des Ansatzes dar. Das Modell ist für multimodale Dialoge in immersiven virtuellen Welten konzipiert und besteht aus fünf Ebenen, welche die folgenden Aspekte modellieren:

- Kontaktebene (*contact*): Ansprechbarkeit, Erreichbarkeit möglicher Konversationspartner
- Aufmerksamkeitsfokus (*attention*): Objekte bzw. Handlungen, auf welche der Agent sich konzentriert
- Konversationsebene (*conversation*)
  - Teilnehmer (*participants*): aktive und passive Interaktanten
  - Turn (*turn*): Recht für die Ausführung der nächsten Äußerung
  - Initiative (*initiative*): Lenkung für den Verlauf der Konversation
  - Vorkenntnisse (*grounding*): Information (*Common Ground Unit*), welche der gemeinsamen Basis angehört, beinhaltet Effekte auf den sozialen Zustand, Obligationen, Verpflichtungen
  - Thema (*topic*): beeinflusst die Relevanz von Elementen
  - Rhetorik (*rhetorical*): rhetorische Verbindungen zwischen einzelnen Beiträgen
- Ebene der sozialen Verpflichtungen (*social commitments*): sowohl Obligationen zum Handeln als auch Einschränkungen auf mögliche Handlungsoptionen
- Verhandlungsebene (*negotiation*): Aushandeln von Verpflichtungen

Das Modell trennt abstrakte Dialogakte in Bezug auf die Aspekte der jeweiligen Ebenen auf, und modelliert damit die aus dem Akt resultierenden Effekte auf den Dialogzustand. Dabei werden sowohl verbale Äußerungen als auch non-verbale Signale berücksichtigt. Umgesetzt wird dies durch eine modulare Implementation, welche leicht um weitere neue kommunikative Modalitäten erweitert werden kann. Entscheidungen, wie ein Agenten in einem Dialog auftritt, sind dabei eng mit Entscheidungsprozessen der Handlungsauswahl des Agenten verbunden (Swartout et al., 2006).

### Dynamic Interpretation Theory

Die *Dynamic Interpretation Theory* (DIT) (Bunt, 1996a) stellt ein Dialogmodell dar, in welchem Dialogakte und ein komplexes Kontextmodell eine besondere Betonung erfahren. In dem DIT-Modell werden die Kontextfaktoren sehr genau strukturiert und stellen damit eine geeignete Basis für die Entwicklung eines künstlichen Systems dar.

DIT ist aus der Untersuchung von Informationsdialogen, welche von Mensch zu Mensch geführt wurden, hervorgegangen und versucht, fundamentale Prinzipien des Dialogs offen zu legen (Bunt, 1996b). Ein Grundgedanke besteht dabei darin, kommunikative Akte als Operationen anzusehen, welche den Kontext des Dialogs verändern. Der Ansatz, die Bedeutung einer Äußerung als eine Kontextveränderung zu modellieren, bietet eine solide Basis, um komplexe Dialogmechanismen zu untersuchen. Bunt definiert *Dialog-Akte* als funktionale Einheiten, welche vom Sprecher dazu benutzt werden, den Kontext zu verändern. Dabei unterscheidet er fünf verschiedene Kategorien der Kontextfaktoren:

- physikalischer/perzeptueller Kontext (Präsenz des Konversationspartners, Aufmerksamkeit)
- sozialer Kontext (Obligationen, kommunikative Rechte)
- linguistischer Kontext (Dialoghistorie, linguistisches Rohmaterial)
- semantischer Kontext (Zustand der zugrunde liegenden Aufgaben)
- kognitiver Kontext (Zustand der Verarbeitungsmodelle, Modell des Konversationspartners)

Bunt betont drei *Grundmotivationen* für kommunikative Aktivitäten. Als erstes nennt er die Motivation, rational seine eigenen Ziele zu verfolgen. Als zweites kommt die Disposition zum Tragen, kooperativ zu sein und dem Konversationspartner beim Erreichen seiner Ziele zu helfen. Und drittens kann eine Motivation aus dem Wunsch bestehen, sich an soziale Normen und Obligationen zu halten und die Kommunikation so angenehm wie möglich zu gestalten. Diese Motivationen führen zu der Ausführung von Dialogakten.

Dialogakte lassen sich in aufgabenorientierte und dialogorientierte Akte klassifizieren. Ein Dialogakt enthält jeweils einen semantischen Inhalt und eine kommunikative Funktion. Funktionale Einheiten stimmen nicht immer mit natürlich-sprachlichen Äußerungen, Sätzen oder anderen linguistisch strukturierten Einheiten überein, da Äußerungen oft multi-funktional sind. *Aufgabenorientierte* Akte bestehen aus Akten der *Informationsbeschaffung* und der *Präsentation* von Informationen. Zu diesen zählen Fragen, Antworten, Überprüfungen und Bestätigungen, welche alle durch die zugrunde liegende Aufgabe motiviert werden.

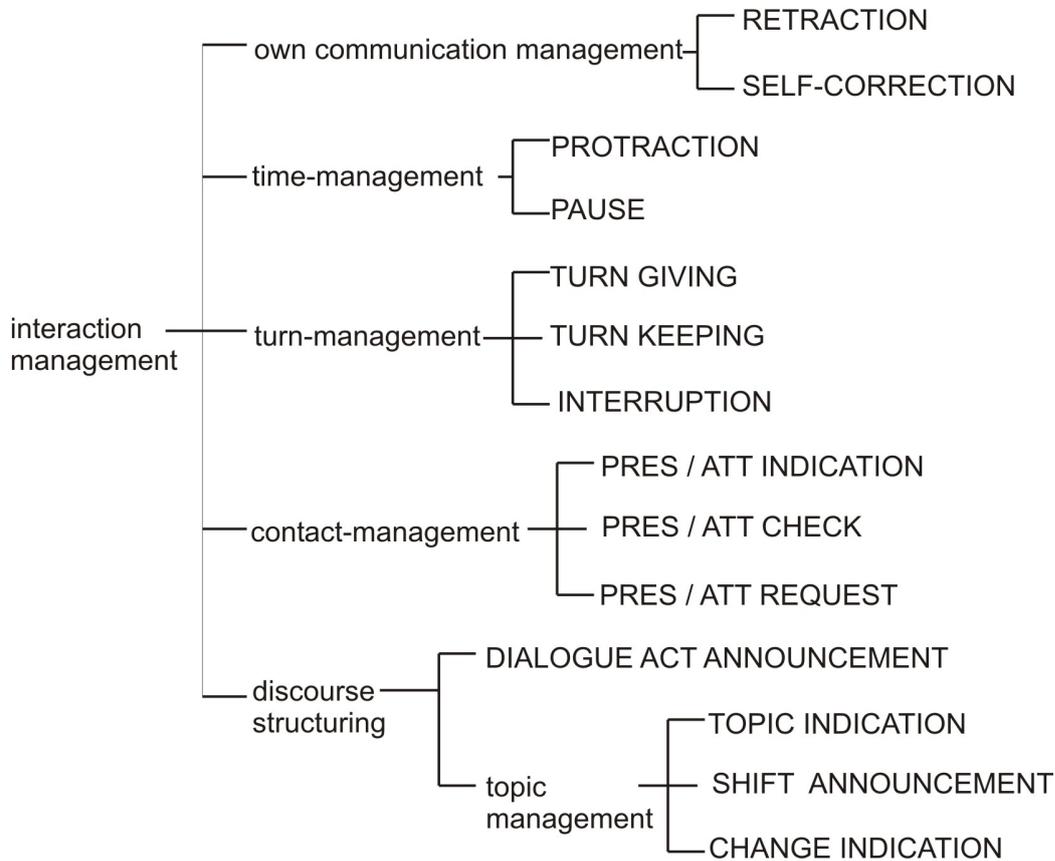


Abbildung 2.10: Interaktionsmanagementsfunktionen nach (Bunt 1996)

Eine Untergruppe der dialogorientierten Akte stellen Handlungen in Bezug auf die Kontrolle des Dialogs (*Dialogue Control Acts*) dar. Diese sind auf die Interaktion selbst gerichtet. Sie dienen dazu, die Bedingungen für eine erfolgreiche, reibungslose Kommunikation zu schaffen und aufrecht zu erhalten. Zu ihren Untergruppen zählen: Feedback, Interaktionsmanagement und das Management von sozialen Obligationen, welches Verhaltensweisen wie Begrüßen, Danken, Verabschieden oder Entschuldigen betrifft. Interaktionsmanagement-Akte konzentrieren sich darauf, die Interaktion selbst zu lenken. Die Interaktionsmanagement-Funktionen lassen sich in fünf Untergruppen kategorisieren (siehe Abbildung 2.10). Dazu zählen *own communication management*-Funktionen, welche sich auf Selbstverbesserungen und die Zurücknahme von Informationen beziehen, sowie *time management*-Funktionen, welche durch Planungsaktivitäten für Äußerungen

des Sprechers entstehen können und Pausen sowie Verzögerungsäußerungen wie „Äh“ umfassen.

Ein weiterer Teil der Interaktionsmanagement-Funktionen lässt sich dem *Turn-Management* zuordnen; hierzu zählen *turn-giving*, *turn-keeping* und Unterbrechungsaktivitäten. Auf der Ebene des *Kontaktmanagements* existieren die Funktionen, Präsenz bzw. Aufmerksamkeit eines Konversationspartners zu überprüfen, einzufordern oder ihm anzuzeigen. Auf der Ebene der Diskursstrukturierung lassen sich schließlich Funktionen der Ankündigung eines Dialogaktes sowie Funktionen der Themenstrukturierung finden.

Die Interaktionsmanagementfunktion des *Turn-Takings* soll im Folgenden genauer betrachtet werden.

### 2.2.2 Turn-Taking

Ein zentraler Punkt, der während eines Dialogs geklärt und ausgehandelt werden muss, besteht darin, wer zum jeweiligen Zeitpunkt die Sprecherrolle einnehmen darf. *Turn-Taking* stellt einen interaktiven Basismechanismus dar, um die Ablaufplanung des Rederechts während einer Konversation zu koordinieren.

#### Grundlagen

Die Konversationsanalyse geht davon aus, dass ein lokaler Mechanismus die Vergabe des Rederechts reguliert. An einer übergaberelevanten Stelle eines Gesprächsbeitrags (*Transition Relevant Place* (siehe Sacks et al., 1974)), die u. a. durch intonatorische und semantische Mittel angezeigt wird und den Abschluss einer Einheit markiert, kann ein Sprecherwechsel nach bestimmten Regeln erfolgen. Sacks und Kollegen sind Verfechter des kontextfreien, regelbasierten *Turn-Takings* (Sacks et al., 1974). Sie sehen *Turn-Taking* als ein Phänomen, das durch die Einhaltung der Regeln durch die Beteiligten als emergentes Muster aus den Interaktionsregeln hervorgeht. Der Ansatz von Sacks und Kollegen erscheint sehr sinnvoll, um fundamentale Einheiten des Turnaustausches zu beschreiben, der Ansatz geht jedoch nicht darauf ein, wie die Konversationsteilnehmer diese Einheiten erkennen können, und bezieht zudem den internen Zustand kognitiver Prozesse des Agenten nicht in die Regeln ein. Sacks und Kollegen behandeln in ihrem Modell ausschließlich gesprochene Beiträge der Konversation.

Während die Konversationsanalyse den kontextfreien, regelbasierten Charakter des *Turn-Takings* betont, haben Duncan (1972) und auf den Arbeiten aufbauende Wissenschaftler empirische Untersuchungen durchgeführt, welche dokumentieren, dass eine Aushandlung der Sprecherrolle durch interaktive Signale geleitet wird. Duncan beschreibt die Existenz von Signalen, welche Interaktanten als

Hinweise in Bezug auf die Turn-Koordination aussenden. Dabei untersucht Duncans *signalbasierter* Ansatz, welche linguistischen Oberflächenphänomene als Signal für das Ende eines *Turns* gelten können. Sowohl der regelbasierte als auch der signalbasierte Ansatz werden von den modernen Dialogtheorien (Clark, 1996; Goodwin, 1981a) aufgegriffen.

Goodwin führt dabei in seinen Arbeiten den Ansatz von Sacks und Kollegen fort. Er hält einen signalbasierten Ansatz allein nicht für ausreichend, um die Phänomene des *Turn-Takings* zu erklären, sieht die Signale allerdings als ein wichtiges Mittel der Verständigung. Nach Goodwin (1981a) beschreibt die Definition des *Turns* als eine statische Einheit mit festen Grenzen die Struktur des *Turns* nicht in akkurater Weise, vielmehr sollte der *Turn* als zeitgebundener Prozess konzeptualisiert werden. Er sieht das Phänomen des *Turn-Takings* in einem größeren sozialen Kontext. Die Aushandlung eines *Turns* erfordere einen *kooperativen Prozess* zwischen Sprecher und Hörer und stelle damit eine elementare Instanz des Erlangens einer sozialen Ordnung dar (Goodwin, 1981b). Dieser umfassende Ansatz macht Goodwin zu einem noch immer aktuellen und oft zitierten Autor in seinem Gebiet.

Auch Clark (1996) liefert einen Beitrag zum Thema des Dialogmanagements. Basierend auf seiner Dialogtheorie lassen sich verschiedene Ebenen des Dialogmanagements unterscheiden, wobei jede einen spezifischen Informationzustand beinhaltet. Eine Menge von kommunikativen Akten kann einen Einfluss auf diese Zustände ausüben und sie verändern. Auf der *Turn*-Ebene werden fünf verschiedene *Turn-Taking*-Aktionen unterschieden: Ergreifen des *Turns* (*take-turn*), Anfordern des *Turns* (*request-turn*), Aufgabe des *Turns* (*release-turn*), Halten des *Turns* (*hold-turn*) und Zuweisen des *Turns* (*assign-turn*).

### ***Ymir-Turn-Taking-Modell***

Aufbauend auf den Arbeiten von Goodwin und Duncan stellt Thorisson das *Ymir-Turn-Taking-Modell* (YTTM) auf. Die *Ymir*-Architektur von Thorisson (1997) spielt eine fundamentale Rolle für die Entwicklung rechnergeeigneter *Turn-Taking* Modelle der Mensch-Maschine-Interaktion. Nach Thorisson (2002) geht es bei der Modellierung natürlichen *Turn-Takings* nicht allein darum, kontextsensitive Regeln aufzustellen, sondern vielmehr müssen die Mechanismen des *Turn-Takings* bei der Entwicklung der Agentenarchitektur berücksichtigt und in dieser verankert werden. Thorisson stellt ein Schichtenmodell vor, welches den kompletten Wahrnehmungs-Handlungsgenerierungs-Prozess modelliert. Dieser besteht aus den folgenden Schritten:

- (1) multimodale Wahrnehmung
- (2) Wissensrepräsentation

## (3) Entscheidungsfindung

## (4) Handlungsgenerierung für Blick, Gesten, Mimik, Sprachplanung- und ausführung

Thorisson sieht die größte Herausforderung der Modellierung natürlichen *Turn-Taking*-Verhaltens in der Umsetzung der Wahrnehmungskomponenten. Aufbauend auf dem Modularisierungsansatz der verhaltensbasierten Künstlichen Intelligenz schlägt Thorisson ein paralleles System von Prozessen mit unterschiedlicher Aktualisierungsfrequenz und Wahrnehmungs-Handlungsschleifen vor, um Ereignisse auf verschiedenen Zeitskalen koordinieren zu können. Um ein kohärentes Auftreten zu generieren, reagieren die verteilten modularen Prozesse in verschiedenen Wahrnehmungs-Handlungsschleifen auf die unterschiedlichen kontextuellen Hinweise. Die Komplexität des *Turn-Takings* entsteht dabei durch die Bandbreite der Kontexteinflüsse, welche ein emergentes Gesamtverhalten hervorrufen.

Thorisson unterscheidet in seinem YTTM-Modell drei Ebenen. Zu unterst liegt die *reaktive Ebene*. Diese ist im Fall des Sprechers (nach Thorisson *content presenter*) zuständig für eine grobe Analyse der Funktion. Auf Seite des Zuhörers (nach Thorisson *content interpreter*) operieren auf der reaktiven Ebene ebenfalls Prozesse der groben Funktionsanalyse, die Analyse des Inhalts wird auf den weiteren Ebenen fortgesetzt und so verfeinert. Zusätzlich sind der reaktiven Ebene auch Feedbackmechanismen zugeordnet. Auf der *Prozess-Kontroll-Ebene* operieren im Falle des Sprechers Mechanismen, welche die Ausgabeprozesse des Zuhörers analysieren, sowie weitere Kontrollprozesse. Die Prozesse des Zuhörers beziehen sich auf dieser Ebene auf die Interpretation des Inhalts und das Signalisieren des Status der Inhaltsanalyse. Auf höchster Ebene, der *Inhalts-Ebene*, analysiert der Sprecher die Reaktion des Zuhörers, während er den Inhalt präsentiert. Der Zuhörer interpretiert den Inhalt und generiert inhaltsbezogene Feedbackinformationen. Dabei operieren die Interpretations- und Generierungsprozesse parallel.

Im Rahmen von Interaktionen, welche von Angesicht zu Angesicht stattfinden, spielen Prozesse auf sehr unterschiedlichen Zeitskalen eine Rolle. Deshalb sind laut Thorisson für eine geeignete Architektur zur Modellierung des *Turn-Takings* folgende Echtzeitanforderungen zu berücksichtigen:

- Reaktivität (Fähigkeit der Dialogteilnehmer, in Bereitschaft zu stehen und angemessen auf eingehende Dialoginformationen zu antworten)
- Verschiedene Zeitachsen (Fähigkeit mit zeitlichen Fristen umzugehen und diese einzuhalten)
- dynamische Anpassung (Fähigkeit, Prioritäten unter Berücksichtigung von Ressourcen, Fristen und aktueller Auslastung flexibel anzupassen)

- Geschwindigkeit (Echtzeitfähigkeit in der realen Welt)

Um im Rahmen des *Turn-Takings* in angemessenen Zeitfenstern antworten zu können, müssen die Gelegenheiten des Turnwechsels im Voraus erkannt werden. Dazu spielen Mechanismen der antizipatorischen Wahrnehmungsverarbeitung eine entscheidende Rolle. Im Gegensatz zu B. Grosz & Sidner (1986) und Clark & Schaefer (1989) bezieht Thorisson allgemeines Wissen über den Gesprächsgegenstand und den Zusammenhang des Inhalts in seinem Modell nicht mit ein und schränkt es auf einen einzelnen Gesprächsgegenstand ein. In diesem Rahmen leistet sein System ein korrektes Wahrnehmen und Generieren von akzeptablen Turn-Übergängen und produziert das erforderliche Turn-Taking-Verhalten in Echtzeit. Dies führt zu Dialogmustern, wie sie auch in Mensch zu Mensch Konversation beobachtet werden. Eingesetzt wurde das YTTM-Modell in den Systemen *Gandalf* (Thorisson, 1997) und *Puff the magic LEGO dragon* (Bryson & Thorisson, 2000).

### ***Function-Modalities-Time-Behavior (FMTB)***

Aufbauend auf der Verallgemeinerung der Sprechakttheorie auf multimodale Äußerungen (*Communicative Act Theory* von Poggi & Pelachaud (2000)) schlagen Cassell, Bickmore, Campbell et al. (2000b) in dem FMTB-Modell die Abstraktion konversationaler Funktionen und die getrennte Verarbeitung propositionaler und interaktionaler Aspekte vor. *FMTB* steht dabei für *Function-Modalities-Time-Behavior* und verdeutlicht die wichtigsten Prinzipien des Modells. Es wird eine Unterscheidung zwischen propositionalen und interaktionalen Funktionen vorgenommen, welche durch unterschiedliche Mechanismen verarbeitet werden. Propositionales Verhalten steht für bedeutungstragende sprachliche Äußerungen sowie Gesten und Intonation, welche den Inhalt der Äußerung komplettieren und vervollkommen. Interaktionale Verhaltensweisen bestehen hingegen aus Signalen, welche die Konversation regulieren und den konversationalen Zustand beeinflussen. Diese Prozesse beinhalten eine Menge an non-verbale Verhaltensweisen (z.B. kurzes Kopfnicken, um anzudeuten, dass man der Konversation folgt) sowie regulative Sprachäußerungen wie „hä?“, „aha“.

Cassell und Kollegen betonen zudem, dass man im Rahmen einer Interaktion von Angesicht zu Angesicht die *Körperlichkeit* der Agenten berücksichtigen muss. Multimodale Äußerungen spielen dabei eine entscheidende Rolle. Durch den Einsatz unterschiedlicher Modalitäten können parallel auf verschiedenen Kanälen Informationen vermittelt werden. Dies deutet auf einen weiteren Aspekt des Modells hin, den der parallelen Verarbeitung. Der Agent muss in der Lage sein, Äußerungen zu produzieren und gleichzeitig die Feedbacksignale des Interaktionspartners wahrzunehmen. Außerdem muss er die parallele

Generierung von Signalen in unterschiedlichen Modalitäten beherrschen. Dabei spielt der Aspekt der Synchronität eine entscheidende Rolle. Cassell nimmt ferner eine Unterscheidung zwischen *konversationalen Verhaltensweisen* und deren *konversationalen Funktionen* vor. Die Form eines kommunikativen Akts kann unter verschiedenen Bedingungen und Kontexten sehr unterschiedlich ausfallen und dennoch das Gleiche bedeuten.

Um einen künstlichen Agenten mit den fundamentalen konversationalen Fähigkeiten auszustatten, muss dies bereits bei der Konzeption der Architektur des Agenten berücksichtigt werden. Die Design-Richtlinien der Architektur umfassen dabei (Cassell et al., 1999; Cassell, Bickmore, Campbell et al., 2000b):

- Verständnis und Synthese sowohl propositionaler als auch interaktionaler Information, dafür Aufbau eines User-Model (Bedürfnisse, Wissens des Interaktionspartners)
- Planungsmodul für Mehrsatzausgaben, Einhaltung der Reihenfolge bei der Präsentation abhängiger Informationen,
- Modellieren von sowohl statischem als auch dynamischem Wissen, Modellierung des konversationalen Zustands
- Verarbeitung multimodaler Eingaben und Ausgaben
- Timing (Feedback, Behandlung von Turnanfragen, paralleles Verfolgen verschiedener Stränge einer Konversation)
- Modell der konversationaler Funktionen

Cassell und Kollegen gehen von einer symmetrischen Architektur in Bezug auf Eingabe- und Ausgabeverarbeitung des Systems und seines Interaktionspartners aus. Sowohl die Generierung als auch die Interpretation propositionalen Inhalts basiert auf einem dynamischen Modell des Diskurses, welches die bereits erörterten Informationen sowie die dahinter liegenden Gründe beinhaltet. Es findet eine sequentielle Verarbeitung der Eingaben des Interaktionspartners statt, welche je nach konversationaler Funktion entweder zu einer fest verdrahteten, instantanen Reaktion führen können oder in dem deliberativen Modul des Agenten genauer analysiert und verarbeitet werden. Typische *Diskursfunktionen* bestehen aus einer Einladung zur Konversation, *Turn-Taking*, Feedback, Kontrast und Betonung sowie der Beendigung der Interaktion. Das deliberative Modul baut auf dem *Aufmerksamkeitsfokus* des Konversationspartners, einem *Diskursmodell*, welches auch Entitäten beinhaltet, vorangegangenen Äußerungen und direkten kommunikativen Zielen auf und ist verantwortlich für die Handlungsauswahl des Agenten.

Das Einnehmen der Sprecherrolle wird explizit durch den konversationalen Zustand (*conversational state*) modelliert und mögliche Änderungen des konversationalen Zustands werden durch einen endlichen Automaten erfasst. Dieser Ansatz wurde in eine allgemeine Architektur für verkörperte, künstliche Agenten integriert und durch den ECA Rea demonstriert (siehe Abschnitt 2.3).

### 2.2.3 Initiative

Während die Mechanismen des *Turn-Takings* auf der interaktionalen Ebene operieren, lässt sich auf der inhaltlichen Ebene ebenfalls ein Abwechseln zwischen den Interaktionspartnern ausmachen. *Initiativwechsel* treten oft zeitgleich mit der Übergabe oder Übernahme des Turns auf, allerdings unterscheiden sie sich in ihrem Bezugspunkt. Betrachtet man die Interaktionen genauer, so fällt auf, dass sich die Interaktanten auf verschiedenen Ebenen einbringen und es unterschiedliche Strukturen und Aktivitätsmuster der Initiativwechsel gibt, welche von *master-slave*-Beziehungen bis hin zu dynamischen *mixed-initiative*-Interaktionen reichen. *Mixed-initiative*-Interaktionen gehen über normale Interaktionen hinaus und beinhalten ein breites Spektrum des kooperativen Problemlösens, das sich durch verschachtelte Beiträge der Interaktanten und *asynchrone* Initiativwechsel auszeichnet (Horvitz, 2007).

Die Initiative bezieht sich auf ein Abwechseln auf inhaltlicher Ebene. Für die Modellierung des inhaltlichen Verlaufs einer Interaktion existieren Modelle der Diskursstruktur.

#### Diskursstruktur

Genauso wie ein Satz aus Phrasen besteht, besteht ein Diskurs aus Diskurssegmenten (B. Grosz & Sidner, 1986). Die einzelnen Beiträge zu einem Dialog bilden eine Diskursstruktur. Ein Diskurs besteht aus einzelnen Diskurssegmenten, welche in einer spezifischen Beziehung zueinander stehen. Möchte man den neben dem interaktionalen Verlauf einer Konversation auch den inhaltlichen Verlauf einer Interaktion erfassen, stellt die Diskursstruktur ein geeignetes Modell dar.

Die Diskursanalyse besteht aus den folgenden drei Prozessen:

- Erkennen, wie eine Äußerung des Diskurses sich in Segmente einteilen lässt
- Erkennen, worin die Intentionen der einzelnen Segmente bestehen
- Erkennen, in welcher Beziehung die erkannten Intentionen zu den bereits bekannten Intentionen stehen

Eine Analyse der Diskursstruktur bietet die Basis, um die Bedeutung eines Diskurses erkennen zu können. Das Modell dient insbesondere dazu, intentionale

Aspekte der Interaktion zu erfassen und zu modellieren. Die Dialogstruktur kann damit dafür eingesetzt werden, den Dialog auf eine *top-down*-Weise zu analysieren (Strayer et al., 2003).

Eine der prominentesten Theorien zur Diskursstruktur stellt die Theorie von B. Grosz & Sidner (1986) dar. Sie schlagen eine Dreiteilung der Diskursstruktur vor, welche aus einem Wechselspiel von *linguistischer Struktur*, *intentionaler Struktur* und dem *Aufmerksamkeitszustand* der Konversationspartner besteht. Die *linguistische Struktur* bezieht sich auf den Inhalt der Äußerungen. Sie besteht aus den Diskurssegmenten und den eingeschlossenen Beziehungen zwischen den einzelnen Segmenten. Die *intentionale Struktur* dagegen umfasst den Sinn und die Intention hinter den einzelnen Beiträgen, sowie deren Zusammenhänge. Dafür wird jedem Segment eine Diskurssegment-Absicht zugeschrieben (*Discourse Segment Purpose*), sodass jedes Segment zu der Absicht des übergeordneten Segment beiträgt und zwar im Sinne des *SharedPlans*-Ansatzes (Rich & Sidner, 1998) (siehe Abschnitt 2.1.3). Der *Aufmerksamkeitsfokus* dient als Speicher der Entitäten, welche eine hervorstechende Rolle im augenblicklichen Diskurs spielen. Eine der Hauptaufgaben des *Aufmerksamkeitsfokus* besteht darin, den Bereich der Diskurssegment-Absichten, zu denen eine neue Diskurssegment-Absicht in Beziehung gesetzt werden kann, einzuschränken.

Ein Agent muss erkennen können, ob eine Äußerung ein neues Segment beginnt, ein bestehendes vervollständigt oder zu dem aktuellen Segment beiträgt. Die intentionale Struktur eines Diskurses spielt eine Schlüsselrolle für den Verständnisprozess der Konversationspartner und lässt den übergeordneten Zusammenhang der einzelnen Beiträge erkennen.

Während das Modell die intentionale Grundstruktur genau beschreibt und zudem spezifiziert, wie sich die einzelnen Diskurssegmente zueinander verhalten, liefert das Modell jedoch keine Informationen zu der internen Struktur der Diskurssegmente in Bezug auf die *Initiativzuschreibung* innerhalb eines Segments. Ferner wird auch nicht spezifiziert, unter welchen Umständen die Interaktanten ein neues Segment beginnen oder das aktuelle beenden (Strayer et al., 2003). Um der Initiativmodellierung innerhalb eines Segments Rechnung zu tragen, werden im Weiteren verschiedene Ansätze der **Initiativmodellierung** diskutiert.

### **Initiativmodellierung**

In Bezug auf die Modellierung der Initiative während einer Interaktion gibt es verschiedene Ansätze. Einige Autoren definieren, dass die Initiative von dem Sprecher ausgeht, welcher an einem gegebenen Zeitpunkt die Konversation vorantreibt (Whittaker & Stenton, 1988; Walker & Whittaker, 1990; Novick & Sutton, 1997). Walker und Kollegen betonen dabei, dass die Initiativsegmente die Diskurssegmente aus der Theorie von B. Grosz & Sidner (1986) umfassen, wobei der Spre-

cher, welcher die Initiative hält, der Initiator des Segments ist und die Bestimmung des Diskurssegments festlegt. Walker & Whittaker (1990) stellen für die Beschreibung der Initiativverteilung ein Annotationsschema auf, bei welchem *assertions*, *questions*, *commands* und *prompts* unterschieden werden. Dabei hält der Initiator der jeweiligen Äußerung die Initiative, außer er antwortet auf eine Frage oder sein Beitrag besteht aus einem *prompt*.

Chu-Carroll & Brown (1997) gehen hingegen davon aus, dass das Konzept *einer Initiative* allein nicht ausreichend ist, um die Phänomene der Initiativverteilung einer Interaktion beschreiben und modellieren zu können. Sie unterscheiden deshalb zwischen *dialogue initiative* und *task initiative*. Die *task initiative* beschreibt, wer die Entwicklung eines Plans vorantreibt, und regelt wie die Aufgabe ausgeführt wird. Die *dialogue initiative* hingegen hält der Agent inne, der den Fokus des aktuellen Diskurses bestimmt. Zusammen bestimmen diese Formen der Initiative, zu welchem Zeitpunkt und auf welche Art ein Agent ein Thema ansprechen wird (Chu-Carroll & Brown, 1997). Das System berechnet basierend auf Hinweisen, die es während eines Turns wahrgenommen hat, sowohl Indizien für die *task initiative* als auch für die *dialogue initiative*. Die Indizien werden dann integriert und unter Zuhilfenahme der *Dempster-Shafer-Theory* (Shafer, 1976) werden die Halter der Initiativen bestimmt.

Haller & Fossum (1999) gehen in ihrem Modell noch einen Schritt weiter und unterscheiden drei Formen der Initiative. Sie definieren neben *task initiative* und *dialogue initiative* auch noch *planning initiative*. Dabei hält derjenige die *task initiative*, welcher die Aufgabe kennt und die Ausführung der Aufgabe koordiniert; die *dialogue initiative* hält derjenige, welcher die Konversation kontrolliert und beispielsweise gemeinsame Annahmen etabliert. Der Inhaber der *planning initiative* zeichnet sich dadurch aus, dass er Vorschläge zur Erreichung des Ziels macht. Für die interne Verarbeitung schlagen Haller und Kollegen den Einsatz von Protokollen vor.

Im Gegensatz zu den bereits beschriebenen Ansätzen, welche davon ausgehen, dass Initiative explizit modelliert und mitprotokolliert werden muss, gibt es auch Forscher, welche die Initiative als untergeordnetes Phänomen ansehen. Strayer & Heeman (2001) finden in ihren Untersuchungen Widersprüche zu den Theorien von Chu-Carroll & Brown (1997) und Walker & Whittaker (1990). In ihrem Modell gehen sie davon aus, dass die Initiative der Intensionsstruktur untergeordnet ist. Der Nicht-Initiator eines Diskurssegments kann Äußerungen machen, welche zu der Bestimmung des aktuellen Diskurssegments beitragen, beispielsweise *forward-acknowledgements* und *other-contributions*. Dabei bleibt in ihrem Modell die Initiative jedoch bei dem Segmentinitiator. Dies dient ihnen als Grund für die Hypothese, dass die Initiative nicht einzeln mitverfolgt werden muss, da sie noch immer bei dem Initiator des Diskurssegmentes liegt. Der Halter der Initiative folgt aus der Modellierung der Diskursstruktur. Lochbaum (1998); Lochbaum et

al. (2000) stellen ebenfalls heraus, dass die Initiative von dem Segment-Initiator gehalten wird, während der Nicht-Initiator nur zu dem aktuellen Segment beiträgt (indem er beispielsweise auf Fragen in einem Teildialog antwortet).

Auch Ferguson & Allen (2007) sehen die Initiative als ein emergentes Phänomen, welches keiner expliziten Modellierung bedarf. Im Hinblick auf die Kontrolle des Initiativwechsels und des proaktiven Verhaltens verfolgen sie den Ansatz, dass dies automatisch aus den Zielen und Intentionen des Agenten folgt.

#### 2.2.4 Verzahnung von Sprache und Handlung

Während sich ein Szenario der gemischten Initiative dadurch auszeichnet, dass die Diskursbeiträge mit verteilter Initiative einhergehen, lässt sich im Bereich der aufgabenorientierten Kooperation auch eine Verschachtelung von Planen und Handeln finden, womit eine Vermischung der Planungs- und Ausführungsebenen stattfindet. Bisher konzentrierte sich die Forschung interaktiver Systeme vornehmlich entweder auf Dialog als eine Form des Austausches von Informationen oder auf Planen und zielgerichtetes Ausführen von Instruktionen. Wenige Arbeiten beschäftigten sich mit der Integration von *aufgabenorientierten* Dialogen, dem Aushandeln von Plänen und der gleichzeitigen gemeinsamen Ausführung der Pläne. Blaylock & Allen (2005) zeigen Defizite im Bereich der integrierten Systeme auf, welche sowohl Kommunikation als auch Planung sowie die Ausführung von Plänen in verschachtelter Form umfassen.

Auch B. Grosz (1996) betont, dass man sich nicht allein auf Planungssysteme verlassen kann, welche zunächst einen Gesamtplan aufstellen und diesen dann in der vorliegenden Version auszuführen versuchen. Planen lässt sich nicht komplett von der Ausführung der Pläne trennen. Kooperative Pläne entwickeln sich während einer Interaktion. Deshalb müssen Systeme in der Lage sein, Planen und Handeln ineinander zu verschachteln.

Zu den Beiträgen, die miteinander verbunden werden müssen, gehören die kommunikativen Akte während der Planerstellung. Diese bestehen zumeist aus Vorschlägen, Bestätigungen und Aushandlungen. Um kommunikative Akte genauer zu analysieren, wird auf den Ansatz der Sprechakte zurückgegriffen.

#### Kommunikative Akte

In der Sprachphilosophie besteht eine lange Tradition darin, Sprechen als eine Form des Handelns zu sehen (Austin, 1962; Searle, 1969). Äußerungen in einem Dialog werden produziert, um etwas zu erreichen. Austin betont, dass verschiedene „Kräfte“ in einer Äußerung mitschwingen und wirken: Der *lokutionäre* Akt der Äußerung der Wörter, der *perlokutionäre* Akt des Effekts auf den Adressaten und eventuell sogar eine Veränderung des Weltzustands. Für das perlokutionäre Ziel

löst eine Äußerung eine bestimmte Aktion aus, den *illokutionären* Akt. Basierend auf dieser Definition prägt Searle den Begriff des Sprechakts (*speech acts*), welcher sowohl einen propositionalen Inhalt umfasst als auch eine mentale Einstellung. Andere Forscher haben für verwandte Konzepte die folgenden Begriffe geprägt: *communicative act* (Allwood, 1976); (Poggi & Pelachaud, 2000), *conversational move* (Carletta et al., 1997) oder *dialogue move* (Cooper & Larsson, 1999).

Poggi und Kollegen definieren einen kommunikativen Akt (*communicative act*) als die kleinste Einheit der Kommunikation, welche aus einem Signal und einer Bedeutung besteht (Poggi & Pelachaud, 2000). Zur Bedeutung gehört dabei ein propositionaler Inhalt sowie ein *Performativ*, welches die Aktion repräsentiert, die der Akt ausführt. Man kann verschiedene Typen von Performativen unterscheiden. Poggi und Kollegen unterscheiden drei unterschiedliche Typen: *inform*, *query* und *request*. Auch wenn sich Performative einem der Grundtypen zuordnen lässt, so kann man sie auch noch feiner charakterisieren. Das Gesamt-Performativ ist zugeschnitten auf den Inhalt, auf welchem es operiert (z.B. ob etwas behauptet wird oder ob es sich um eine Forderung handelt). Ferner spielen kontextspezifische Faktoren (die aktuelle Situation, der Adressat, die Art der Begegnung, der Grad der Sicherheit) eine Rolle. Das Verfeinern eines Performativs kann sogar retrospektiv auftreten.

### **Kooperatives Problemlösen**

Blaylock & Allen (2005) bilden ein Modell des kooperativen Problemlösens. Neu dabei sind die Aspekte des verschachtelten Problemlösens, wodurch sowohl eine Kooperation während des Planungs- als auch während des Ausführungsprozesses ermöglicht wird. Blaylock et al. (2003) beschreiben das Problemlösen eines einzelnen Agenten als einen Dreiphasenprozess:

- Ziele beschließen und festlegen, welches der Ziele als nächstes verfolgt wird
- einen Plan bzw. ein Rezept für ein noch ausstehendes Ziel auswählen
- ein Rezept ausführen und die Ausführung zeitgleich überwachen

Der entscheidende Punkt ist, dass die Phasen in beliebiger Reihenfolge sowie beliebig verschachtelt auftreten können. Blaylock und Allen modellieren keine spezifische Planlösungsstrategie, sie bilden vielmehr ein Modell, das es Agenten ermöglicht, innerhalb einer gemeinsamen Kooperation erfolgreich zu interagieren, ohne dass sie dabei die gleiche Strategie verfolgen müssen. Dies lässt Kooperationen zu, in denen nur in der Planungsphase zusammengearbeitet wird, sowie Szenarien, in denen die Kooperation in allen drei Phasen auftritt.

In dem kooperativen Problemlösungsmodell *Collaborative Problem Solving* (CPS) von Blaylock & Allen (2005) werden sechs abstrakte Problemlösungsobjekte beschrieben:

- *Ziele* (wobei diese sich auf Aktionen und nicht auf Zustände beziehen)
- *Rezepte* (Annahmen, wie man ein Ziel erreichen kann)
- *Einschränkungen* (welche sich sowohl auf einzelne Objekte als auch auf Lösungen beziehen können)
- *Evaluationen* (Abschätzungen eines Objekts in einem bestimmten Kontext)
- *Situationen* (Zustände bzw. mögliche Zustände der Welt)
- *Ressourcen* (Domänen-bezogene Objekte der Welt sowie Konzepte)

Der kooperative Problemlösungsprozess kann als Entscheidungsfindungsprozess gesehen werden. Dabei geht es Blaylock und Allen darum, nicht nur die Entscheidung sondern auch den Entscheidungsprozess selbst zu modellieren. Im Rahmen des Entscheidungsprozesses existieren oft mehrere Optionen in Bezug auf ein Ziel oder eine Objektauswahl. Um die Grundlage der Entscheidung explizit zu machen, schlagen Blaylock und Allen vor, eine *slot and filler*-Ebene in das Modell zu integrieren. Diese dient dazu, konkret modellieren zu können, welche Optionen berücksichtigt werden, wie diese bewertet werden und welche Einschränkungen und Aspekte dann zu der endgültigen Entscheidung führen. Dabei können die *Slots* neben konkreten Werten die Attribute *identified* oder *adopted* annehmen. Zusätzlich können *Einschränkungen* für einen *Slot* spezifiziert werden. Die *Slots* werden sowohl für Objekte als auch für Ziele und Evaluationen eingesetzt, allerdings lassen die Autoren offen, wie man eine Evaluation repräsentieren könnte.

Die Agenten verfügen über einen *CPS*-Zustand, welcher durch die Ausführung von *CPS*-Aktionen verändert werden kann. Diese lassen sich in zwei Kategorien einteilen: zum einen gibt es Schlussfolgerungshandlungen (*reasoning acts*, z.B. *c-focus*, *c-defocus* *c-identify*) und zum anderen Handlungen, die sich auf das Eingehen von Verpflichtungen (*commitment acts*, *c-adopt*, *c-abandon*, *c-defer*, *c-release*, *c-select*) beziehen. Mit Hilfe dieser Handlungen kann beispielsweise der Fokus auf einen bestimmten Aspekt der Kooperation gelenkt werden oder die Agenten können Verpflichtungen eingehen. Der *CPS*-Zustand kann allerdings nur gemeinsam von den kooperierenden Agenten verändert werden. Daher gibt es eine Menge an Interaktionshandlungen (*interaction acts*), die die einzelnen Agenten ausführen können, um die Veränderungen auszuhandeln. Die Interaktionshandlungen umfassen *begin*, *continue*, *complete* und *reject*. *CPS*-Acts werden in diese Interaktionshandlungen eingebettet.

Die *CPS*-Acts können als feiner granuliert *Shared-Plans*-Operatoren (siehe Abschnitt 2.1.3) gesehen werden. Dabei lässt sich das Modell von Blaylock und Allen auch auf die gemeinsame Ausführung des Plans beziehen. Weitere Arbeiten sind darauf gerichtet, das Modell in einem Dialogmanager umzusetzen (Blaylock & Allen, 2005).

Nachdem aufgezeigt wurde, wie in einer Interaktion Planen, Handeln und Dialogaspekte eng miteinander verbunden auftreten und sich dynamisch aufeinander abstimmen, soll im folgenden Abschnitt diskutiert werden, wie interaktives Verhalten aufgrund von Regeln und Erwartungen der einzelnen Agenten entstehen kann.

### 2.2.5 Verpflichtungen – Erwartungen – Obligationen

Sowohl im Bereich der Dialogmodelle als auch im Bereich der Kooperationsmodelle spielen Obligationen eine entscheidende Rolle. Während Modelle wie das von Cohen und Levesques (siehe Abschnitt 2.1.2) oder der *SharedPlans*-Ansatz (siehe Abschnitt 2.1.3) einige Diskursphänomene und Kooperationsphänomene schlüssig erklären können, scheitern sie auf der Diskursebene daran, eine einfache Erklärung für Agentenverhalten zu liefern, das den Zielen der Agenten zuwider läuft. Sie setzen ein hohes Maß an Kooperationsbereitschaft voraus, welche aber nicht in jeder Interaktion vorhanden sein muss. Interaktionen, in welchen die Agenten nicht auf einer höheren Ebene gemeinsame Ziele verfolgen, zeigen jedoch zumeist eine gleich flüssige Abwicklung wie andere Interaktionen. Allein auf der Inhaltsebene lassen sich Differenzen erkennen. Die Kooperationsmodelle sind für eine höhere Ebene der Interaktion geeignet, jedoch für die Modellierung von flüssigen Interaktionen auf Diskursebene müssen andere Mechanismen greifen.

Der *Agenten-basierte* Ansatz der Dialogmodellierung erlaubt, zusätzlich zu den mentalen Kategorien der Beliefs, Ziele, und Pläne auch soziale Einstellungen für die Verhaltensmodellierung einzusetzen. So können Erwartungen, die ein Sender mit seiner kommunikativen Handlung verknüpft, oder Obligationen, die durch die Handlungen (z.B. Zeigen oder Übergeben einer Aufgabe) für den Rezipienten entstehen, die Verhaltensweisen der Interaktanten grundlegend beeinflussen. Obligationen lassen sich dabei jedoch nicht auf einfache Erwartungen reduzieren. Erwartungen können die Verhaltensinterpretation leiten und Plan-Erkennungsprozesse unterstützen (Carberry, 1990), sie können aber nicht selbst als eine genügende Motivation gelten, welche den Agenten dazu bringt, sich angemessen zu verhalten. Dazu können jedoch Obligationen dienen.

Obligationen erweitern das planbasierte Deliberations-Modell des Agenten. Sie leiten das Verhalten des Agenten, ohne dass eine Intentionserkennung erfolgen muss oder *SharedPlans* auf der Diskursebene eingesetzt werden müssten. All-

gemein lassen sich Obligationen in Bezug auf den Modaloperator *zulässig* (*permissible*) definieren (Traum & Allen, 1994):

*Eine Aktion ist obligatorisch, wenn es nicht zulässig ist, sie nicht auszuführen.  
Sie ist verboten, wenn sie nicht zulässig ist.*

Das Modell der Obligationen unterscheidet sich von den intentionsbasierten Ansätzen dadurch, dass Obligationen unabhängig von gemeinsamen Plänen und Intentionserkennung sind. Obligationen sind vielmehr das Ergebnis interner Regeln und sozialer Normen, an denen der Agent sein Verhalten ausrichtet.

Obligationen stellen explizite Mechanismen dar, welche Stabilität und Beständigkeit (Verlässlichkeit) in eine Interaktion bringen, während sie auch Flexibilität ermöglichen. Sie ermöglichen Entscheidungsfreiheit mit bekannten explizit modellierbaren Konsequenzen (Dignum et al., 2002). Obligationen können von den persönlichen Intentionen und Zielen eines Agenten differieren und lassen sich deshalb auch nicht auf sie reduzieren. Wenn das Interaktionsverhalten des Agenten lediglich als eine automatische Annahme von Zielen modelliert wird, verhindert dieses Schlussfolgerungs- und Entscheidungsprozesse, welche gegen Obligationen gerichtet sind. Ein solches Interaktionsverhalten nimmt dem Agenten die Möglichkeit, aufgrund von persönlichen Zielen gegen die allgemeinen Interaktionsregeln zu verstoßen. Umgekehrt kann es aber auch vorkommen, dass ein Agent einer Obligation folgt und eine Aktion vornimmt, welche konträr zu seinen Zielen verläuft.

Im Rahmen von Teamaktivitäten entstehen Obligationen durch die Interaktion zweier Agenten, welche sich gegenseitig Versprechen geben oder sich auf Verträge einigen. Obligationen können aber auch durch allgemeine soziale Regeln, welche während einer Interaktion gelten und durch explizite oder implizite Interaktionsprotokolle entstehen (Dignum et al., 2002).

Nach der Diskussion von Kommunikation, Verpflichtungen und Erwartungen als zentrale Aspekte einer kooperativen Interaktion wird im nächsten Abschnitt das Phänomen des Engagements näher beleuchtet. Engagement tritt auf verschiedenen Ebenen auf und stärkt den effizienten, reibungslosen Ablauf einer Interaktion.

### **2.2.6 Verbundenheit (*Engagement*)**

Interaktion ist ein Prozess der Verbundenheit (*Engagement*). Wenn Menschen für eine Interaktion eine Bindung zu ihrem Interaktionspartner aufbauen, den Interaktionsprozess gestalten und schließlich beenden, so tun sie dies unter Einsatz von Verbundenheitssignalen (*Engagementsignalen*), durch welche die Effizienz und Reibungslosigkeit einer Interaktion gefördert wird.

Die Forschung im Bereich der kooperativen Interaktion hat sich in den letzten Jahren vermehrt dem Phänomen des *Engagements* zugewandt (Nakano & Nishida, 2005a; Sidner et al., 2003b). Sidner und Kollegen definieren:

**Definition** *Engagement is the process by which two (or more) participants establish, maintain and end their perceived connection during interactions they jointly undertake* (Sidner et al., 2003a).

Engagementsignale steuern eine Kooperation (Sidner et al., 2003b). Sie sind zunächst dafür verantwortlich, dass es überhaupt zu einer Interaktion kommt. Ferner dienen sie zur Aufrechterhaltung des Interaktionsprozesses. Dabei operieren sie im Hintergrund und wirken für einen reibungslosen Ablauf der Interaktion. Sind die Ziele der Agenten erreicht oder gibt es andere gute Gründe dafür, dass die Interaktion nicht weiter fortgesetzt werden sollte, sorgen Engagementprozesse für eine angemessene Beendigung der Interaktion. Es ist wichtig, Engagementsignale verarbeiten und generieren zu können, da angenommen werden kann, dass Menschen diese konversationellen Fähigkeiten auch im Umgang mit künstlichen Agenten einsetzen (Sidner et al., 2003b).

In Interaktionen zwischen Menschen lassen sich verschiedene Grundmechanismen von Engagement Phänomenen feststellen. Engagement kann sowohl durch *verbale* als auch durch *non-verbale* Signale gefördert werden. Im Bereich der non-verbale Signale kann insbesondere der *Aufmerksamkeitsfokus*, repräsentiert durch die Blickfokussierung der Interaktionspartner, als zentrales Engagementsignal eingesetzt werden (Argyle & Cook, 1976; Clark & Schaefer, 1989; Duncan, 1972, 1974; Kendon, 1967).

Laut Clark & Schaefer (1989); Clark (1996); Nakano & Nishida (2005b) ist ein im Verlauf einer Interaktion während des Turns auf den Sprecher gerichteter Aufmerksamkeitsfokus eine grundlegende Form des Adressaten, um Verständnis zu signalisieren und zu zeigen, dass man in die Interaktion involviert ist. Das Richten der Aufmerksamkeit beispielsweise auf ein gemeinsames Referenzobjekt kann als ein Hinweis für Verständnis dienen (Nakano et al., 2003). Goodwin berichtet, dass Sprecher den Blick ihres Zuhörers benötigen, um mit einer Konversation beginnen zu können (Goodwin, 1981a).

Das Zurücknehmen des Aufmerksamkeitsfokus im Rahmen der Interaktion spielt eine weitere wichtige Rolle. Nakano & Nishida (2005a) berichten, dass es bei Menschen als ein Signal des Nicht-Verstehens gewertet wird, wenn ein Zuhörer über die Äußerung des Sprechers hinaus zu lange seinen Blick auf sein Gegenüber fokussiert. Auch im Bereich der interaktionalen Gesten lassen sich Engagementsignale finden. So dienen *Turn-Taking*-Signale als Indikatoren für ein aktives Einbringen und Involviertsein in die Interaktion; denn allein die Entscheidung, sich aktiv an der Interaktion zu beteiligen, kann als ein Indikator für Engagement gewertet werden kann.

Die Analyse der *Ontogenese* von Engagementprozessen liefert ein Schema verschiedener Komplexitätsstufen (Tomasello et al., 2005):

- Erste Stufe: *Dyadisches Engagement* bezieht sich auf das Teilen von Verhaltensweisen und Emotionen. Dabei gehen die Interaktionspartner wechselseitig durch den Ausdruck von Emotionen und durch *Turn-Taking* Signale auf ihr jeweiliges Gegenüber ein.
- Zweite Stufe: *Triadisches Engagement* zeichnet sich durch ein gemeinsames Ziel (bzw. externe Objekte) aus. Die Interaktionspartner agieren dabei ausgerichtet auf ein gemeinsames Ziel. Während sie dies tun, beobachten sie jeweils das zielgerichtete Verhalten und die Wahrnehmung des Interaktionspartners.
- Dritte Stufe: *Kooperatives Engagement* beinhaltet koordinierte Pläne und gemeinsame Intentionen für das Erreichen eines gemeinsamen Ziels. Die Interaktionspartner handeln auf ein gemeinsames Ziel hin durch den Einsatz koordinierter Handlungspläne, welche sich als gemeinsame Intentionen, gemeinsame Aufmerksamkeit und geteiltes Wissen manifestieren.

### 2.2.7 Gemeinsame Aufmerksamkeit (*Joint Attention*)

Wie bereits diskutiert spielen Aufmerksamkeitsprozesse sowohl im Bereich des Kontaktmanagements (Abschnitt 2.2.1), der Analyse der Diskursstruktur (Abschnitt 2.2.3) als auch als Engagementsignal (Abschnitt 2.2.6) eine entscheidende Rolle im Rahmen einer Kooperation. Gemeinsame Aufmerksamkeit kann zudem eine Basis schaffen, welche kooperative Aktivitäten ermöglicht (siehe Abschnitt Gemeinsame Intentionalität 2.1.4 und Abschnitt Ideale Kooperation 2.1.5).

Aufmerksamkeit wird oft als gesteigertes Bewusstsein in Bezug auf sowohl interne als auch externe Aspekte der Perzeption, Konzeption und der Verhaltensweisen verstanden. Dieses Bewusstsein kann sowohl durch unwillkürliche als auch deliberative Prozesse hervorgerufen werden (Brinck, 2003). Aufmerksamkeit wird daher nicht als ein einheitlicher Prozess gesehen, sondern als komplexes Phänomen. Aufmerksamkeit kann als intentional gerichtete Perzeption definiert werden (Tomasello et al., 2005). Die Aufgabe liegt dabei in der Förderung und Aufrechterhaltung von ziel-orientiertem Verhalten.

Während Interaktionen wird die Aufmerksamkeit bei Menschen durch die Wahrnehmung der Blickrichtung und Inferenzen, die daraus gezogen werden moduliert. Aktuelle Experimente deuten darauf hin, dass interagierende Agenten genauso viel Aufmerksamkeit auf die Intentionen ihres Interaktionspartners richten wie auf die wahrnehmbaren Handlungen desselben (Nuku & Bekkering, 2008).

Hobson unterstreicht zusätzlich, dass Agenten die Fähigkeit benötigen, sich miteinander durch das Teilen von Erfahrungen zu verbinden und darüber eine intersubjektive Verbindung eingehen. Um einen Zustand der Gemeinsamen Aufmerksamkeit (*Joint Attention*) zu erreichen, müssen die Agenten sich dabei des Aufmerksamkeitsfokus des Gegenübers wie auch des Prozesses des Teilens der Aufmerksamkeit selbst bewusst sein (Hobson, 2005).

Eines der umfangreichsten Modelle der *Joint Attention* stammt von Baron-Cohens (1994) Arbeit in Bezug auf Autismus. Er postuliert ein Modell welches vier Module beinhaltet; einen Intentionalitäts-Detektor (*intentionality detector*), einen Blickrichtungs-Detektor (*eye-direction detector*), ein Modul für gemeinsame Aufmerksamkeit (*shared attention module*) und ein *theory of mind module*. Dabei wird ein Schwerpunkt auf *Theory of mind* als Endresultat und Meta-Repräsentation als Prozess gesetzt, wodurch jedoch Schlüsselbeziehungen insbesondere zwischen Aufmerksamkeit und Intentionalität nicht beschrieben werden (Tomasello, 1995).

### 2.2.8 Zusammenfassung

Möchte man Kommunikation als zentrales Mittel einer Kooperation modellieren, so bieten Dialogmodelle, die sich mit dem Austausch von Dialogbeiträgen beschäftigen, eine gute Basis. Während das MRE-Modell als Weiterentwicklung des *Information State Approach* für die Modellierung virtueller Menschen entwickelt wurde und dadurch ein Schwerpunkt auf die technischen Realisierbarkeit liegt, ist die *Dynamic Interpretation Theory* (DIT) aus der Untersuchung der Informationsdialoge von Mensch zu Mensch hervorgegangen. Die DIT bietet ein komplexes Modell in Bezug auf eine feine Kategorisierung der Dialogbeiträge und hilft wichtige Kontextfaktoren eines Dialogs detailliert aufzuzeigen. Das MRE-Modell hingegen bietet eine gute Ausgangsbasis dafür, welche Ebenen es für die Umsetzung eines virtuellen Agenten zu berücksichtigen gilt. In beiden Modellen wird insbesondere eine Modellierung der *Aufmerksamkeit*, des *Turn-Takings* sowie der sozialen *Obligationen* gefordert. Im Gegensatz zu der DIT-Theorie sieht das MRE-Modell neben einer Modellebene des *Turn-Takings* auch eine extra Modellebene für die Initiative vor.

In der DIT wird der perzeptuelle Kontext und insbesondere der Aufmerksamkeitsfokus als zentraler Kontextfaktor eines Dialogs gesehen. Während die Modellierung der Aufmerksamkeit sich in dem MRE-Modell auf Objekte und Handlungen bezieht, auf die sich der Agent konzentriert, werden in der DIT konkrete *Contact-Management Acts* vorgestellt, um die Aufmerksamkeit des Gegenüber abzufragen oder zu beeinflussen (*indication/check/request*). Blicke in Form von Engagementsignalen können im Rahmen einer Interaktion die Effizienz und Reibungslosigkeit der Interaktion fördern.

MRE und DIT sehen beide Turn-Management-Funktionen vor, liefern jedoch keine konkreten Anhaltspunkte für eine Umsetzung der Fähigkeiten in einem Gesamtsystem. Die Modelle des *Turn-Takings* hingegen stellen konkrete Anforderungen an eine Architektur auf. So werden Reaktivität und Echtzeitfähigkeit, Umgang mit zeitlichen Fristen, sowie eine dynamische Anpassung an die Interaktion gefordert (siehe *Ymir*). Das FMTB-Modell betont, dass sowohl eine schnelle fest verdrahtete Reaktion auf *Turn-Taking*-Signale vorgesehen werden muss als auch, dass Entscheidungen aufgrund einer deliberativen Verarbeitung getroffen werden. Insbesondere wird auch die Möglichkeit einer parallelen Verarbeitung verschiedener Konversationsstränge gefordert. Für eine einfachere Verarbeitung werden im FMTB-Modell *Turn-Taking*-Signale auf Funktionen abgebildet.

Im Bereich der Initiativmodellierung besteht eine große Bandbreite an Initiativdefinitionen. Es wird zwischen der Initiative des Dialogs und der Initiative in Bezug auf eine Aufgabe unterschieden. Es existieren jedoch auch prominente Ansätze, die die Initiative als emergentes Phänomen aus der Intensionsstruktur ableiten.

Kooperatives Problemlösen und das damit verbundene verschachtelte Auftreten von Planungs- und Handlungsphasen stellen eine große Herausforderung dar. Hilfreich ist es in diesem Kontext, Entscheidungen explizit zu machen, indem protokolliert wird, welche Optionen im Rahmen einer Entscheidung berücksichtigt wurden und welche ausgewählt wurde. Auch der Ansatz der *SharedPlans* hilft bei der Modellierung kooperativen Problemlösens.

## 2.3 Existierende Systeme

Zum Abschluss der theoretischen Analyse von Kooperation werden konkrete Systeme vorgestellt, die kooperative Interaktionen mit Menschen realisieren. Die Systeme bilden unterschiedliche Schwerpunkte, besitzen jedoch alle eine Relevanz für die Untersuchung von kooperativem Verhalten. Es werden acht Systeme vorgestellt, die teilweise aufeinander aufbauend verschiedene Anwendungsbereiche abdecken. Tabelle (2.1) stellt die interaktiven Aspekte der einzelnen Systeme dar und verdeutlicht ihre Schwerpunkte.

### IPOC

Nakano & Nishida (2005a) präsentieren ein System, das sich auf die *Engagement*-Signale der Blickfokussierung konzentriert. Im Rahmen der immersiven Kommunikationsumgebung IPOC (*Immersive Public Opinion Channel*) wurde ein Prototyp eines konversationalen Agenten implementiert, welcher den menschlichen Benutzer durch ein Szenario des Geschichtenerzählens führt. Um seiner eigenen

	Engagement	Dialog	Aufgabe	Kooperation
<b>IPOC</b> (Nakano & Nishida, 2005)	x	x		
<b>Mel</b> (Sidner & Lee, 2003)	x	x		
<b>Rea</b> (Cassell et al., 2000)		x		
<b>MRE</b> (Swartout et al., 2006)		x	x	
<b>Steve</b> (Rickel & Johnson, 2000)		x	x	
<b>SASO-ST</b> (Traum et al., 2005)		x	x	x
<b>Leonardo</b> (Breazeal et al., 2004)	x		x	x
<b>TRIPS</b> (Allen et al., 2002)		x	x	x

Tabelle 2.1: Übersicht Systeme

Involviertheit in die Interaktion Nachdruck zu verleihen, setzt der konversationale Agent ein *Engagement-Behavior-Generation* Module ein. Dieses Modul erkennt das Engagementverhalten des Konversationspartners und seinen Aufmerksamkeitsfokus und generiert seinerseits ein angemessenes Antwortverhalten.

Für die Ermittlung des Konversationszustands identifiziert das System sowohl wer gerade spricht als auch, auf welchen Inhalt derjenige sich bezieht, und berechnet die Position der fokussierten Objekte. Anhand dieser Information beurteilt das System, ob das Gegenüber sich in der Konversation engagiert. Der *Blickfokus* des Gegenübers wird dafür in vier Klassen eingeteilt: Aufmerksamkeit auf den Agenten, auf eine gemeinsame Referenz, auf andere Hintergrundobjekte oder abgewandte Aufmerksamkeit. Die genauen Regeln stellen Nakano & Nishida (2005a) wie folgt dar: Ist der Aufmerksamkeitsfokus vor Beginn einer Interaktion auf den Agenten gerichtet, so wird dies als Engagementssignal im Hinblick auf den Wunsch einer Interaktion gewertet. Darauf eingegangen wird mit einer Begrüßung. Während einer Interaktion signalisiert der Agent durch wiederholte kurze Blicke, dass er den Zustand des Gegenübers im Blick behält. Richtet der Mensch nach einer Interaktion weiter seinen Blick auf den Agenten, versucht dieser neue Themenobjekte einzuführen. Wenn der Benutzer des Systems seine Aufmerksamkeit auf ein Objekt richtet, versucht der Agent dasselbe Objekt zu fokussieren, um *Joint Attention* zu erreichen. Fokussiert der Mensch ein anderes

Objekt als der Agent, versucht der Agent durch deutlichere Gesten die Aufmerksamkeit seines Gegenübers zurückzugewinnen. Wenn ein Konversationsziel abgearbeitet ist, wird das neue Thema auf das Objekt gelenkt, welches der Mensch gerade fokussiert.

### Mel - Collagen

Mel ist ein weiteres System, das sich auf plausibles Engagementverhalten konzentriert (Sidner & Lee, 2003); es wird an den *Mitsubishi Electric Research Labs (MERL)* entwickelt. Basierend auf der Analyse von Angesicht-zu-Angesicht-Konversationen arbeiten Sidner et al. (2003a) an einem humanoiden Roboter, welcher menschliches konversationales Blickverhalten in kooperativen Unterhaltungen imitiert. Für die konversationalen und kollaborativen Fähigkeiten des Agenten wird *Collagen* (Rich et al., 2001) eingesetzt. Collagen ist ein Framework für intelligente, interaktive Systeme, welche auf das dreigeteilte Diskursmodell von B. Grosz & Sidner (1986) aufsetzen (siehe Abschnitt 2.2.3). Collagen implementiert ferner Teile des *SharedPlans*-Ansatzes (siehe Abschnitt 2.1.3). Das System besteht aus einer *mixed-initiative*-Architektur. Sidner et al. (2005) betonen jedoch, dass in ihrem System die Konversation zum Großteil von Mel kontrolliert wird, da im Falle einer *mixed-initiative*-Interaktion die Varianz der Äußerungen in solchem Maße steigt, dass die Spracherkennung zu unzuverlässig wird.

Nach Analyse von Mensch-Mensch-Interaktionen und unter Berücksichtigung linguistischer und psycholinguistischer Literatur schlagen Sidner & Lee (2003) folgende *Engagement*-Regeln vor:

**Begrüßung** Um eine Interaktion zu beginnen, sollte der Agent den Menschen anschauen, um ihn zu einer Interaktion einzuladen, gefolgt von einer konversationalen Begrüßung.

**Blickverhalten** Während einer Interaktion gilt in Bezug auf das Blickverhalten des Agenten, dass er immer den Menschen angucken sollte, wenn dieser den Turn ergreift. Während eines Turns sollte er auch, so vorhanden, weitere Teilnehmer der Konversation anschauen.

**Beenden einer Interaktion** Wenn der Mensch die Interaktion abbrechen möchte, den Turn nicht ergreift, wenn er dies müsste, oder seinen Kopf aus dem visuellen Blickfeld des Agenten bewegt, sollte der Agent nicht sofort die Interaktion beenden, aber dies explizit festzustellen. Er sollte sich dann mittels Dialog versichern, ob der Mensch wirklich die Interaktion beenden möchte. Ein kurzes „*goodbye*“ und ein langen Blick von dem Interaktanten weg können für die Beendigung einer Interaktion eingesetzt werden. Wenn

dem Agenten die Themen für die Konversation ausgehen, sollte er die Konversation mit einem normalem Abschluss beenden.

Mel stellt ein komplexes Robotersystem dar, welches in Demonstrationsdialogen mit einem menschlichen Konversationspartner ähnliche soziale Engagementsignale einsetzt wie die Agenten in dem IPOC-System.

### **Rea**

Das Rea-System wurde in der *Gesture and Narrative Language Group* am MIT Media Lab (Cassell, Bickmore, Vilhjálmsón & Yan, 2000) entwickelt und nutzt die FXPAL-Architektur (Cassell, Bickmore, Campbell et al., 2000a). Rea ist ein ECA, welcher sich in multimodalen von Angesicht-zu-Angesicht-Konversationen in Echtzeit engagiert.

Das Rea-System ist in der Lage, eine Unterhaltung zu führen, in welcher es sowohl Aspekte der Aufgabendomäne vermittelt, als auch auf die verbalen sowie non-verbalen Eingaben des Interaktionspartners eingeht. Für das System wurde ein Diskursplaner entwickelt (Bickmore & Cassell, 2005), welcher auf einem relationalen Dialogmodell (Cassell & Bickmore, 2003) aufbaut und *Smalltalk* sowie aufgabenbezogene Äußerungen ineinander verschachtelt und so zu einer natürlichen, sozialen Interaktion beiträgt. Das Planen des Diskurses auf Grundlage einer statischen Welt wurde durch einen Ansatz ersetzt, welcher auf den *activation networks* (Maes, 1989) aufsetzt. Das hat den Vorteil, dass durch das einfache Justieren von Gewichten das Rea-System unterschiedlich kohärent, höflich, aufgabenorientiert oder deliberativ in Bezug auf seine Äußerungen wirken kann. Die Auswahl des nächsten kommunikativen Akts wird von einer nicht-diskreten Funktion ermittelt, welche die folgenden Faktoren miteinbezieht: *Vertrautheit, Thema, Relevanz, aufgabenbezogene Ziele und logische Vorbedingungen*.

Das Rea-System setzt das FMTB-Modell (siehe Abschnitt 2.2.2) um. Rea stellt einen sehr kompetenten Konversationspartner dar, welcher multimodale und non-verbale Signale einsetzt sowie über komplexe *Turn-Taking*-Kompetenzen verfügt.

### **Steve**

Steve (*Soar Training Expert for Virtual Environments*) wurde in virtuellen 3D-Welten eingesetzt und behandelt immersive Aspekte des Dialogs in der Rolle eines Tutors (Rickel & Johnson, 2000). Steve besitzt domänenunabhängige Fähigkeiten, aufgabenorientierte Dialoge in dreidimensionalen virtuellen Welten zu führen. Sein Kognitionsmodul ist in drei Schichten organisiert, von denen die unterste Schicht auf *Soar* (siehe Abschnitt 4.3.1) basiert. Die darüber liegende Schicht ist für domänenunabhängige Aspekte wie für die Fähigkeit zur aufgabenorientierten

Kooperation zuständig. In der dritten Schicht ist das domänenspezifische Wissen modelliert. Steve ist in der Lage, seine Pläne während der Interaktion dynamisch anzupassen und zu revidieren, wenn sich seine Umwelt verändert, und kann damit auch auf die Aktionen des Gegenübers eingehen.

Die Interaktion selbst folgt einer hierarchischen Initiativverteilung (*master-slave*). Der Interlokutor kann den Agenten zwar unterbrechen sowie Zwischenfragen stellen und Handlungen vornehmen, es findet jedoch kein Aushandeln einer gemeinsamen Aufgabe statt. Steve setzt hierarchisch organisierte Pläne ein, welche durch kausale Verbindungen und einschränkende Bedingungen (*constraints*) zusammengehalten werden. Für die Planung selbst setzt er einen Planungsalgorithmus der partiellen Ordnung ein.

Um den Anforderungen der Interaktion gerecht zu werden, modelliert Steve sowohl den Aufgabenkontext als auch den Kontext des Dialogs. Im Bereich der Interaktion ist das Verhalten des Agenten relativ unflexibel und wenig dynamisch. In den Interaktionsverlauf eingreifen kann sein Gegenüber durch Zwischenfragen („*what next*“, „*why*“) oder durch die Aufforderung an Steve, die Aufgabe zu Ende zu stellen oder abubrechen. Steve lässt sich dann zwar unterbrechen, aber nur, nachdem er seinen gesamten Sprechakt beendet hat. Auch Feedbacksignale kann der Agent erst dann aussenden, wenn er eine komplette Äußerung gehört und geparkt hat.

Steve setzt seine Körperorientierung als Signal seines Aufmerksamkeitsfokus ein, sein Blickverhalten nutzt er, um die Aufmerksamkeit auf ein Objekt zu lenken, oder seine Bereitschaft zum Zuhören zu bekunden.

## MRE und SASO-ST

Aufbauend auf der Steve-Architektur wurde das System der *Mission Rehearsal Experience (MRE)* (Rickel et al., 2002; Swartout et al., 2006) entwickelt. Das MRE-System wird in einer Installation der Virtuellen Realität eingesetzt, in welcher der Benutzer interaktiv in eine Geschichte eintauchen kann. Dabei besteht das System aus einem Hybridkomplex aus Geschichtenerzählen, Interaktivität und agentenbasierter Technologie. Ziele des Systems sind, Benutzer auf reale Situationen in militärischen Bereichen vorzubereiten.

Für die intelligenten Hauptakteure wird eine komplexe Architektur eingesetzt. Die Agenten basieren auf *Soar* (siehe Abschnitt 4.3.1) und setzen auf dem Steve-System auf. Zusätzlich werden folgende Komponenten in die Architektur integriert:

- *Dialogmodell* mit Unterstützung von Dialogebenen (siehe Abschnitt 2.2.1).
- *Sprachmodul* für die Generierung natürlichsprachlicher Äußerungen.

- *Kontrollmodul für die Bewegungen des Körpers*: Es steuert die Position des Körpers, den Blickfokus und die Gestenproduktion entsprechend der Interaktion, der Überwachung der Aufgabe und der Involviertheit in die Konversation.
- *Aufgabenmodell*: Es besteht aus Wissen über Ereignisse, welche in der Welt eintreten können, und aus Sequenzen von Plänen für die Erreichung bestimmter Zielzustände.
- *Emotionsmodell*: Es beinhaltet emotionale Basiszustände, Einschätzungen von emotionalen Zuständen und Bewältigungsstrategien sowie die Modulation des Verhaltens (mentale Aktionen, Annehmen von Zielen) (Marsella & Gratch, 2001; Gratch & Marsella, 2004).

Im Rahmen des MRE-Systems gibt es auch Arbeiten, die soziale Beziehungen zwischen Interaktionspartnern einbeziehen. Dabei spielt die Integration eines Emotionssystems basierend auf der *appraisal*-Theorie eine entscheidende Rolle (Gratch & Marsella, 2004). Für das daraus entwickelte SASO-ST System (Traum et al., 2005) wird ein Aufgabenmodell eingesetzt, welches auf hierarchisch aufgebauten Aufgaben-Netzwerken basiert. Zusätzlich werden entscheidungstheoretische Konzepte der Nützlichkeit und Wahrscheinlichkeit für die Modellierung nicht-deterministischer Entscheidungen berücksichtigt. Die Aufgabe des Benutzers in dem SASO-ST-Szenario besteht darin, einen Arzt davon zu überzeugen, dass seine Klinik an einen anderen Ort verlegt werden muss. Die Bildung und der Einsatz von Aushandlungs- und Verhandlungskompetenzen stehen dabei im Vordergrund. Die Arbeiten des SASO-ST-Modells orientieren sich an der Definition Allwoods (siehe Abschnitt 2.1.5) zur idealen Kooperation.

### **Leonardo**

Bei Leonardo liegt der Schwerpunkt der Arbeiten darauf, einen Roboter zu entwickeln, der als Partner für den Menschen auftritt und über soziale Kompetenzen der Kooperation verfügt (Breazeal et al., 2004). Leonardo tritt in diesem Kontext als Teamkollege auf und soll zunächst eine Aufgabenbewältigung interaktiv lernen, um sie dann kooperativ mit dem menschlichen Gegenüber ausführen zu können. Im Rahmen des Szenarios muss der Roboter über *Theory of mind* Kompetenzen verfügen. Zu diesen Fähigkeiten zählen:

- Gemeinsame Aufmerksamkeit auf externe Objekte lenken
- Repräsentation der eigenen Annahmen sowie der Annahmen des Gegenübers

- Einnahme der visuellen Perspektive des Gegenübers
- Abstimmung der mentalen Zustände der Interaktanten durch kommunikativer Akte

Leonardo verfügt über keine Sprachgenerierungsfähigkeiten. Er ist aber in der Lage, seinen internen Zustand durch Einsatz von Blicken, Gesten und sozialen Signalen zu vermitteln. So schaut er beispielsweise kurz dorthin, wo sich etwas Signifikantes an dem Weltzustand geändert hat, z.B. wenn der Mensch ein Teilziel erfüllt. Stimmt er mit der Aktion überein, zeigt er dies durch ein kurzes Kopfnicken an, um anzudeuten, dass gemeinsame Einigung darüber herrscht, was bisher erreicht wurde und was noch getan werden muss. Dabei finden verschiedene Formen von *Engagementsignalen* Einsatz.

Für die Modellierung der Kooperationsfähigkeiten greifen Hoffman & Breazeal (2004) auf die Ansätze von Bratman (siehe Abschnitt 2.1.1) und Cohen und Levesque (siehe Abschnitt 2.1.2) zurück. Sie benutzen dabei das Zusammenführen von Teilplänen. Bevor Leonardo einen Teil der Aufgabe ausführt, handelt er mit seinem Gegenüber aus, wer diesen Aufgabenteil übernehmen soll. Aufgaben repräsentiert Leonardo in hierarchischen Aktionsstrukturen mit den jeweiligen Teilzielen. Ziele können sich sowohl auf Aktivitäten als auch auf Zustände der Welt beziehen, sie stellen den *Common Ground* zwischen den Interaktanten her.

*Turn-Taking* wurde mit Hilfe von Gesichtsmimik und Körperhaltung umgesetzt. Um seinen Turn abzugeben, nimmt Leonardo Blickkontakt mit seinem Gegenüber auf, hebt seine Augenbrauen und bringt seine Arme in eine entspannte untere Position. Wenn der Interaktionspartner spricht, guckt der Roboter zu dem Sprecher und hebt seine Ohren, um Aufmerksamkeit zu signalisieren und um zu verdeutlichen, dass er zuhört. Wenn der Interaktionspartner seine Äußerung beendet hat, hebt Leonardo seine Arme, um die Übernahme der Initiative zu bekunden und bricht den Blickkontakt ab. Oft guckt er dann zu dem Objekt, auf welches sein Gegenüber sich als letztes in der Äußerung bezogen hat. Leonardo gibt auch *back-channel*-Signale. Wenn er nicht in der Lage ist, eine Äußerung zu verstehen, guckt er verwirrt, um sein Verständnisproblem zu verdeutlichen. Ein kurzes zustimmendes Nicken deutet an, dass der Roboter eine Äußerung verstanden hat. Leonardo baut sich ein Modell seines Gegenübers und von dessen Aufmerksamkeitsfokus auf und unterstützt sein Gegenüber bei der Ausführung der gemeinsamen Aufgabe. Aufgrund der fehlenden Sprachkompetenz können jedoch keine längerfristigen Pläne ausgehandelt werden.

## TRIPS

Aufbauend auf TRAINS (Traum, 1996) wird TRIPS (*The Rochester Interactive Planning System*) an der Rochester Universität als Prototyp eines domänenun-

abhängigen Dialogsystems entwickelt (Allen et al., 2001), welches schon in verschiedenen Szenarien Einsatz gefunden hat (*TRIPS Pacifica Emergency evacuation*, *TRIPS Monroe Emergency 911 operation*, *TRIPS Medication Advisor*). Ein Schwerpunkt liegt dabei auf den kooperativen Fähigkeiten des Systems sowie der Interaktion mit wechselnder Initiative. Eine Schlüsselrolle spielt die Berücksichtigung verschiedener Kooperationsebenen (siehe Abschnitt 2.2.4). Die Komponenten des Systems lassen sich in drei Funktionseinheiten unterteilen: Generation, Interpretation, und Verhalten. Die Module operieren kontinuierlich, asynchron und parallel. Die Struktur des kooperativen Agenten basiert auf dem *Belief-Desire-Intention* Modell (siehe Abschnitt 4.2.1). Im Rahmen der Arbeit an TRIPS entwickeln Allen (1999) ein allgemeines Modell des Planungsprozesses und dem damit verbundenen Koordinationssprozessen. Schwerpunkte liegen auf der inkrementellen Entwicklung von Plänen, der Planerkennung und der Entwicklung der Fähigkeit, effektiv die Struktur und die Implikationen von Plänen vermitteln zu können.

In Bezug auf die Kooperationsmodelle lehnen sich ihre Arbeiten den Arbeiten von Cohen und Levesque (Abschnitt 2.1.2) an. Es gibt aber auch Verknüpfungspunkte zu dem *SharedPlans*-Ansatz (siehe Abschnitt 2.1.3). Blaylock et al. (2003) betonen, dass ihr Ansatz mit dem Modell der *SharedPlans* kompatibel ist und die beschriebenen *CPS-Acts* als Spezifizierung der *SharedPlans*-Operatoren auf der Äußerungsebene gelten können. TRIPS versucht vier komplexe Anforderungen an ein kooperatives konversationales System zu erfüllen: das Parsen von natürlicher Sprache, die Integration von Dialog und aufgabenbezogenen Akten, Intentionserkennung und Interaktion mit wechselnder Initiative.

Für TRIPS wird deshalb besonders intensiv an der Modellierung von Sprechakten, von Planungsprozessen und von Aushandlungsinteraktionen gearbeitet. Der Austausch von sozialen Signalen sowie von *Engagement*-Signalen auf unterer Ebene spielt in dem System bisher keine Rolle. Da kein verkörperter Agent simuliert wird, werden auch Aspekte des visuellen Aufmerksamkeitsfokus nicht modelliert.

## 2.4 Zusammenfassung

Jedes der vorgestellten Systeme hat seinen Schwerpunkt und spezifische Stärken. Während einige Systeme eine möglichst natürliche Interaktion gewährleisten, indem menschenähnliches Dialog- und Interaktionsverhalten modelliert wird, setzen andere Systeme den Schwerpunkt auf den inhaltlichen Verlauf der Kooperation. Darüber hinaus gibt es eine große Varianz hinsichtlich der Art des gewählten Mediums. So wurden sowohl ein Dialogsystem als auch verschiedene Roboterarten sowie ECAs vorgestellt. Zwar sind nicht alle der Systeme als Kooperations-

partner konzipiert, aber sie setzen dennoch in ihrer Domäne verschiedene Aspekte der **zentralen Mechanismen (k1)-(k5)** (siehe Abschnitt 2.1.6) einer echten Kooperation um.

Das **gegenseitige Aufeinandereingehen (k1)** kann auf unterster Ebene durch *Engagementsignale* umgesetzt werden (siehe Abschnitt 2.2.6). Dazu bieten IPOC sowie Mel solide Ansätze. In beiden Systemen werden konkrete Regeln aufgestellt, wie Aufmerksamkeitssignale wahrgenommen, ausgedrückt und insbesondere erwidert werden können. Damit werden Bereiche des dyadischen Engagements abgedeckt.

Im Rahmen der **situierten Kommunikation (k2)** gilt es, den Konversationskontext sowie die Umwelt in die kommunikative Verarbeitung und Verhaltensproduktion einzubeziehen. Rea leistet auf diesem Gebiet einen wichtigen Beitrag durch die komplexe FMTB-Architektur und die Umsetzung Kontext-sensitiven *Turn-Takings* (siehe Abschnitt 2.2.2). Der Ansatz, durch das Justieren von Gewichten die Bezugsebene des Dialogs von sozialem Smalltalk auf aufgabenorientierte Diskussion verschieben zu können, unterscheidet Rea von den anderen Systemen. Damit kann sich Rea dem situationalen Kontext anpassen. Der Einbezug von Smalltalk kann zudem als eine weitere Form von Engagement in die Interaktion gelten. Das MRE-System unterstützt durch das eingesetzte Dialogmodell aufbauend auf dem *Information State Approach* (siehe Abschnitt 2.2.1) eine besonders komplexe Modellierung situierter Kommunikation. Durch die Berücksichtigung der Umwelt, des konversationalen Zustands, des *Grounding* sowie verschiedener Obligationen werden eine Vielzahl bedeutsamer Kontextfaktoren berücksichtigt.

**Koordiniertes Planen und Handeln (k3)** ist zentrales Konzept im TRIPS-System; es wird ein abstraktes Problemlösungsmodell präsentiert (siehe Abschnitt 2.2.4). Planungsphasen und Handlungsphasen können in beliebiger Reihenfolge auftreten, und die Initiative kann zwischen den Interaktanten wechseln. In den Systemen IPOC, Mel und Rea gibt es keine kooperative Aufgabe zu lösen. IPOC und Mel bestehen jeweils aus interaktiven Demonstrator-Systemen, und auch in Rea liegt eine der Hauptaufgaben des Systems darin, Informationen zu präsentieren. Eine Initiativmodellierung auf Aufgabenebene existiert damit nur in MRE, Leonardo und TRIPS. Das Leonardo-System beherrscht das Zusammenführen und Anpassen von Plänen während einer Interaktion sowie das dynamische Lösen einer Aufgabe. Jedoch ist Leonardo aufgrund der fehlenden Sprachkompetenz nicht in der Lage, eine komplexe Kooperation mit den dazugehörigen Aushandlungsprozessen einzugehen. Das SASO-ST-System versteht sich auf Prozesse mit Verhandlungskompetenz, aber es gibt in dem Szenario keine Verzahnung von Plänen und Handeln.

Hinsichtlich des Aspekts der **gegenseitigen Unterstützung (k4)** bietet das Steve-System eine besondere Form der Hilfestellung. Allerdings fällt die Un-

terstützung im Rahmen des Tutor-Szenarios einseitig aus. Dennoch können die Kompetenzen für Feedback und für die Modellierung partieller, dynamischer Pläne als Grundlage für die Modellierung von Unterstützungskompetenzen im Rahmen einer Kooperation dienen. Sowohl in dem Leonardo-System als auch im TRIPS-System bestehen Ansätze der gegenseitigen Unterstützung.

Betrachtet man den Einsatz der zuvor vorgestellten Kooperationsmodelle, so orientiert sich das SASO-ST Modell an Allwoods Ansatz idealer Kooperation (siehe Abschnitt 2.1.5). Das Leonardo-System orientiert sich an Arbeiten von Bratman (siehe Abschnitt 2.1.1) sowie Cohen und Levesque (siehe Abschnitt 2.1.2). Das TRIPS-System greift den Ansatz der SharedPlans von Grosz und Kollegen auf (siehe Abschnitt 2.1.3).

Einen ersten Schritt in Richtung **gemeinsamer Intentionalität (k5)** geht das Leonardo-System. Leonardo baut ein Modell auf, das den Aufmerksamkeitsfokus und Annahmen seines Gegenübers repräsentiert.

Im folgenden Kapitel wird ein komplexes **interaktives Kooperationsmodell** entwickelt. Die diskutierten Ansätze mit ihren Anforderungen und Lösungsvorschlägen dienen als Basis für die Bildung des aus eigenen Ideen entwickelten interaktiven Kooperationsmodells BCTC. Für die einzelnen Ebenen des Modells werden konkrete Regeln formuliert und Verfahren angegeben, die aufzeigen, wie kooperatives Verhalten auf den jeweiligen Ebenen generiert und verarbeitet werden kann.



# Kapitel 3

## ***BCTC* - Ein interaktives Kooperationsmodell**

Die theoretische Analyse der Kooperationsmechanismen in Kapitel 2 lässt erkennen, dass es eine große Bandbreite an Signalen gibt, die zwischen den Kooperationspartnern während einer Interaktion ausgetauscht werden. Im Rahmen einer Kooperation müssen die Kooperationspartner auf die Signale und Informationen des Gegenübers eingehen, diese in ihre Verhaltensgenerierung miteinbeziehen und sich an die Verhaltensmuster und Regeln einer Kooperation halten.

Das *interaktive Kooperationsmodell BCTC* (*Behavior Conversation Task Collaboration*) erfasst die Charakteristika der natürlichen kooperativen Interaktion sowie deren Zusammenspiel und liefert im Weiteren die Basis für die Entwicklung eines künstlichen Kooperationspartners.

Das BCTC-Modell baut auf dem bereits präsentierten Modell der interaktiven Kooperation (siehe Abbildung 2.9) auf und erweitert dieses durch seine Strukturierung in vier hierarchisch angeordnete Ebenen. Dieser Modellstruktur liegt die Hypothese zugrunde, dass es verschiedene Ebenen der Interaktion gibt, auf denen sich die Interaktanten einbringen und engagieren (Leßmann & Kopp, 2005). Im Modell wird beschrieben, auf welchen Ebenen und aufgrund welcher interaktiven Dynamik eine Kooperation ausgetragen wird. Dabei werden funktionale, inhaltsbasierte Beiträge von ihren Ausdrucksformen unterschieden. Während *Engagement* signalisiert, dass sich ein Agent in eine Interaktion einbringen möchte und Bereitschaft in Form entsprechender Signale demonstriert, stellt die Kooperation die dahinter liegende Motivation und Organisation dar und bezieht sich auf eine zielorientierte Ebene. Wie in den verschiedenen Kooperationsmodellen aufgezeigt, lässt sich Kooperation durch ihre Verpflichtungen sowie durch die mentalen Zustände der Interaktanten definieren (siehe Abschnitt 2.1). *Engagement* drückt sich hingegen in den overten Verhaltensweisen der Kooperationspartner aus. Es regelt das konkrete Verhalten des Agenten im Dienste der Kooperation.

Ein Kooperationsmodell, das sich auf natürliche und gleichberechtigte Interaktion bezieht, muss beide Aspekte modellieren. Für die Entwicklung des interaktiven Kooperationsmodells spielen daher neben den Kooperationsmodellen die Arbeiten Tomasellos zur Ontogenese von Engagementverhalten sowie die Engagement-Definition von Sidner und Kollegen (siehe Abschnitt 2.2.6) eine entscheidende Rolle.

Um die Interaktionsphänomene zwischen den Kooperationspartnern besser fassen zu können, ist es hilfreich, die Mechanismen und den Austausch von Signalen in Bezug auf verschiedene Interaktionsebenen zu strukturieren (Leßmann & Kopp, 2005). Für die Strukturierung in vier Ebenen dient neben der Analyse kooperativen Verhaltens die Ontogenese des menschlichen Engagementverhaltens als Inspiration. Auch die verschiedenen Initiativebenen von Haller & Fossum (1999) (siehe Abschnitt 2.2.3) spiegeln sich in dem Modell wider.

Den fünf zentralen Mechanismen einer echten Kooperation (siehe Definition 2, Abschnitt 2.1.6) *Gegenseitiges Aufeinandereingehen, Situierete Kommunikation, Koordiniertes Planen und Handeln, Gegenseitige Unterstützung, Gemeinsame Intentionalität* lassen sich vier Ebenen zuordnen, auf denen Engagement- und Kooperationsprozesse ablaufen (siehe auch Abbildung 3.1):

- **Behavior** (Ebene der Verhaltensformen)
- **Conversation** (Konversationsebene, Formen der Verständigung)
- **Task** (Aufgabenorientiertes Handeln)
- **Collaboration** (Kooperation, Formen der Zusammenarbeit)

Der Name des Modells leitet sich aus den englischen Begriffen der Ebenen ab. Die Ebenen werden im Folgenden beschrieben, nachdem die besonders wichtige Begriffsbildung des *Interaction-Moves* in ihrer Bedeutung dargestellt wurde.

### 3.1 Schema der Interaktionsbeiträge (*Interaction-Move*)

Ausgehend von dem *Information State Approach* (siehe Abschnitt 2.2.1) wird in der vorliegenden Arbeit eine Form der Interaktionsmodellierung vorgenommen, in welcher sowohl manipulative Handlungen als auch Sprechakte als Interaktionsbeiträge gewertet werden (Leßmann et al., 2006). Dabei wird die Interaktion aus der Sicht des Agenten modelliert, und die internen Prozesse des Agenten sowie

### 3.1. SCHEMA DER INTERAKTIONSBEITRÄGE (INTERACTION-MOVE) 59



Abbildung 3.1: Ebenen des Kooperationsengagements

der wechselseitige Umgang mit seinem Interaktionspartner finden Berücksichtigung.

Im Abschnitt 2.2.4 wurde bereits betont, dass im Rahmen einer Kooperation sowohl Sprache als auch manipulative Handlungen eine entscheidende Rolle spielen. Während die Sprechakttheorie postuliert, dass Sprechakte auch eine Form der Handlung darstellen (Searle, 1969), wird in der vorliegenden Arbeit die folgende These aufgestellt.

**These (Interaktionsbeiträge)** *Auch manipulative Handlungen können als kommunikative Akte eingesetzt werden. Der gemeinsame Nenner besteht dabei in der Intention des Agenten, zu der Interaktion bzw. Kooperation beitragen zu wollen und diese voranzutreiben.*

Kommunikative und manipulative Akte werden daher im Rahmen des BCTC-Modells in einem gemeinsamen Format repräsentiert - dem *Interaction-Move*. Der *Interaction-Move* spannt einen Rahmen auf, welcher alle informationalen Aspekte umfasst, die in einer Interaktion in Bezug auf eine Handlung bzw. ein Ereignis als relevant erscheinen. Der *Interaction-Move* beinhaltet dabei sieben Dimensionen (siehe Abbildung 3.2), auf welchen er unabhängig von den konkreten kommunikativen bzw. manipulativen Handlungen der Interaktanten funktional wirken kann. Die dafür genutzten einzelnen Felder sind weder unabhängig voneinander noch autark, da für die Modellierung von Entscheidungs- und Interpretationsfunktionen oft mehrere Felder berücksichtigt werden müssen. Die einzelnen Felder dienen dazu, signifikante Aspekte der Interaktion explizit zu machen (Leßmann et al., 2006). Sie geben eine Struktur vor, welche inkrementell gefüllt werden kann und zwar sowohl bei der Interpretation als auch bei der Generierung einer

Handlung. Einzelne Felder können dabei auch leer bleiben. Wenn der Agent beispielsweise die Hand hebt, um den Turn anzufordern, so wird nur das Feld mit der *Turn-Taking-Funktion* gefüllt.

Die einheitliche Struktur vereinfacht die Handhabung der Interaktionsbeiträge. Die einzelnen Felder können dem Agenten einen Hinweis darauf vermitteln, wie er auf einen Interaktionsbeitrag seines Gegenübers reagieren und antworten sollte; es wird in der Regel eine Antwort auf gleicher Ebene erwartet (siehe Abbildung 3.5).

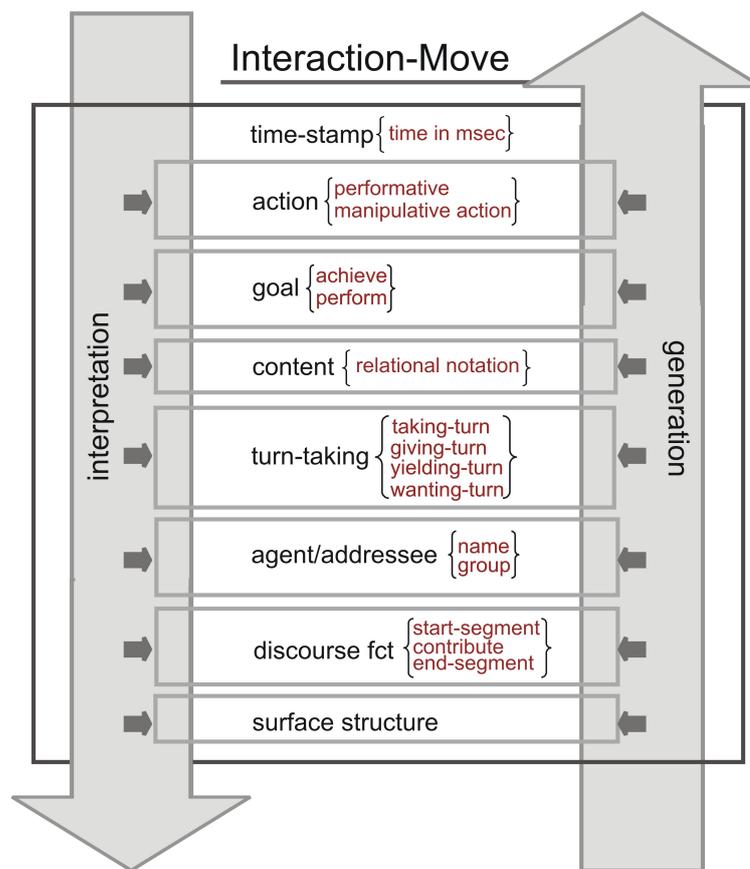


Abbildung 3.2: Interaction-Move

Die Felder des *Interaction-Move* orientieren sich an den konversationellen Ebenen des MRE-Modells (2.2.1), gehen jedoch darüber hinaus. Wurde bereits ein Kontakt aufgebaut, so enthält das Feld **agent/addressee** den einen oder gegebenenfalls auch mehrere Adressaten des Interaktionsbeitrags. Handelt es sich um den ersten Interaktionsbeitrag, der an den Adressaten gerichtet wird, muss der Kontakt erst noch explizit hergestellt werden. Während das **turn-taking**-Feld da-

### 3.1. SCHEMA DER INTERAKTIONSBEITRÄGE (*INTERACTION-MOVE*) 61

zu dient, eine konversationale Funktion zu spezifizieren, repräsentiert das **action**-Feld den illokutionären Anteil des *Interaction-Moves*. Dabei wird davon ausgegangen, dass ein manipulativer Akt auch eine kommunikative Funktion ausführen kann. Im diesem Fall enthält das *action*-Feld die vorzunehmende Handlung. Im Fall einer sprachlichen Äußerung kodiert das Performativ den Typ des kommunikativen Akts. Diese Information ist für die Berechnung der **Diskursfunktion** (*discourse function*) von entscheidender Bedeutung. Die Diskursfunktion selbst spielt für die Modellierung der Initiative eine wichtige Rolle. Die möglichen Werte der Diskursfunktion beziehen sich auf den intentionalen Kontext, dem sie zugeordnet sind, und umfassen *start-segment*, *contribute*, *end-segment* (siehe auch (Lochbaum, 1995)).

Während Cassell (siehe 2.2.2) betont, dass eine Architektur auf der Ebene von Funktionen agieren muss, fordert das BTC-Modell eine zusätzliche Berücksichtigung der Ziele. Um eine Kooperation modellieren zu können, reicht es nicht aus, nur die Funktionen zu berücksichtigen. Auf den unteren Ebenen genügt es, reaktiv auf die Funktionen eines *Interaction-Moves* zu reagieren, auf höheren Ebenen jedoch ist es unabdingbar, auch die dahinter liegenden Ziele zu berücksichtigen. Deshalb fügen wir zusätzlich zu den Ebenen, welche in dem MRE-Modell postuliert werden, noch eine Ebene der Intentionen ein, welche durch die Repräsentation des zugrunde liegenden Ziels umgesetzt wird. Dies wird von der *Sprache-als-Handlung*-Tradition unterstützt, nach welcher laut Pickering und Garrod Äußerungen unter Berücksichtigung des Kontextes sowie der Ziele und Intentionen der Interaktanten interpretiert werden (Pickering & Garrod, 2004).

Das **goal**-Feld behandelt die *perlokutionäre* Funktion des *Interaction-Moves*. Dabei kann das Ziel, wie auch im *SharedPlans*-Modell (siehe 2.1.3) sowohl zustandsorientiert (*intention-that*) als auch handlungsorientiert (*intention-to*) ausfallen. Die Ziel-Ebene repräsentiert die Formation, die Aufrechterhaltung und das Überlappen von Zielen zwischen den Interaktanten. Ziele repräsentieren den Grund hinter einer intentionalen Handlung und abstrahieren von der konkreten Umsetzung in der Situation. Das **content**-Feld umfasst den propositionalen Gehalt des *Interaction-Moves*, welcher zusammen mit dem illokutionären und perlokutionären Anteil den *Interaction-Move* komplettiert. Der Inhalt wird als propositionale Information in einer Interaktionshistorie gespeichert und in einer logikbasierten Notation vorgehalten.

Die Bedeutung eines *Interaction-Move* wird durch die Effekte definiert, die er in Bezug auf die Aspekte des Interaktionsmodells hervorruft. Diese Effekte lassen sich oft nicht in einer genau definierten Weise festlegen, da sie von vielen verschiedenen Facetten des *Interaction-Move* und des Kontextes abhängen, in dem er ausgeführt wurde. Beispielsweise führt ein einfacher Akt, der die einfache *Turn-Taking*-Funktion *wanting-turn* erfüllt, nur dann automatisch zu einem neuen konversationalen Zustand, wenn er in dem Zustand *gap* ausgeführt wird. Aus

diesem Grund unterhalten IS-basierende Systeme typischerweise eine große Anzahl an Aktualisierungsregeln, um die kontextsensitiven Effekte von bestimmten Akten zu modellieren. Im BCTC-Modell werden die Regeln auf den jeweiligen Ebenen des Modells realisiert.

Die Felder des *Interaction-Move* lassen sich den verschiedenen Ebenen des BCTC-Modells zuordnen. Das Ziel-Feld spielt eine besonders wichtige Rolle im Rahmen der *Collaboration*-Ebene. Auf der *Task*-Ebene wird insbesondere auf den Feldern der Aktion und des Inhalts operiert, während für die *Conversation*-Ebene die Felder des Performativs, der Diskursfunktion, des Adressaten sowie des *Turn-Takings* ausgewertet werden. Signale der *Behavior*-Ebene werden in der Regel in einem kürzeren Interaktionszyklus ausgetauscht als Signale der anderen Ebenen. Daher werden Signale der *Behavior*-Ebene nicht als eigenes Feld im Rahmen des *Interaction-Move* repräsentiert. Sie werden parallel zu den Signalen der anderen Ebenen zunächst reaktiv verarbeitet. Werden die Signale der *Behavior*-Ebene jedoch auf höherer Ebene integriert, so können sie Funktionen der *Conversation*-Ebene repräsentieren und in diesem Kontext Eingang in den *Interaction-Move* finden.

Der *Interaction-Move* dient dazu, ein situiertes Interaktionsmanagement zu modellieren, in welchem Prozesse eng mit dem externen Weltzustand und der Interaktionshistorie gekoppelt werden und damit Anforderungen an eine effiziente natürliche Interaktion erfüllen. Der *Interaction-Move* stellt dabei eine komplexe Informationsstruktur dar, welche als Format für die Generierung, Interpretation auch für die Speicherung in einer Interaktionshistorie dient. Jeder *Interaction-Move* ist mit einem Zeitstempel versehen, sodass eine Nachfolgeordnung gegeben ist und eine Historie über den zeitlichen Verlauf der einzelnen *Interaction-Moves* repräsentiert werden kann.

Im Folgenden werden die einzelnen Ebenen des Kooperationsmodells (siehe Abbildung 3.1) vorgestellt.

## 3.2 Verhaltensebene (*Behavior Layer*)

Basale soziale Verhaltensweisen, welche gerade auch für das gegenseitige Aufeinandereingehen verantwortlich sind, werden auf der *Behavior*-Ebene modelliert. Das Engagement auf dieser Ebene ist *dyadisches Engagement* zwischen zwei Interaktanten. Kennzeichnend für diese Ebene sind das Teilen von Verhaltensweisen und Emotionen sowie das Senden von sozialen Signalen. Zu diesen Signalen zählen: Blick- und Blickfokussierungsverhalten, der Gesichtsausdruck, Nicken, Gesten, die Körperhaltung und die räumliche Beziehung zwischen den Interaktanten. Für ein soziales, effizientes miteinander Agieren ist ferner das Senden von Feedbackinformationen auf dieser Ebene von entscheidender Bedeutung. Feed-

backsignale stellen dabei eine Möglichkeit dar, Aufmerksamkeit und Involviertheit zu signalisieren.

Dies sind Prozesse, von denen beim Menschen anzunehmen ist, dass sie zu meist unbewusst ablaufen und in sehr kleinen Zeitfenstern operieren (Argyle, 1973). Das bedeutet, dass ein Agent reaktive Verhaltensweisen benötigt, welche schnell auf Engagementsignale dieser Ebene eingehen und überzeugende Feedbacksignale produzieren. Mit Blick auf das verhaltensbasierte Engagement spielt der Aufmerksamkeitsfokus des Gegenübers eine besonders wichtige Rolle. Das Teilen eines Aufmerksamkeitsfokus gilt als Zeichen für erfolgreiches Engagement und steuert durch die Bezugnahme auf ein Objekt in Richtung *triadische* Interaktion. Auf der *Behavior*-Ebene findet zwar ein abwechselndes Senden der Signale zwischen den Interaktanten statt, jedoch ist dies von dem klassischen *Turn-Taking* zu unterscheiden. Die Wahrnehmung und Generierung der Engagementsignale dieser Ebene operieren in sehr viel kleineren Zyklen innerhalb eines Turns und müssen damit nebenläufig und parallel zu dem aktuellen Turn verarbeitet und produziert werden.

Kennzeichnend für die Behavior-Ebene ist ein *gegenseitiges Aufeinandereingehen* der Interaktanten, wie von Bratman gefordert (siehe Abschnitt 2.1.1). Dazu zählen: der Austausch emotionaler Signale, der Austausch von Feedbacksignalen, gegenseitiges Anschauen und das Fokussieren von referenzierten und relevanten Objekten sowie soziale Signale. Die *Behavior*-Ebene leistet einen wichtigen Beitrag für das von Allwood geforderte flexible, kooperative Miteinanderumgehen (siehe Abschnitt 2.1.5). Sowohl Feedback, welches gegenseitige Unterstützung ausdrückt, als auch das Hervorrufen von *gemeinsamer Aufmerksamkeit* werden damit auf der *Behavior*-Ebene angeregt.

### 3.2.1 Emotionen

Die ontogenetische Entwicklung des Engagements beginnt mit dyadischen Interaktionen (siehe Abschnitt 2.2.6). Diese zeichnen sich durch ein direktes Engagement aus, bei welchem externe Dinge (z.B. Objekte, Umgebungsgerausche) keine Rolle spielen. Es steht vielmehr die Interaktion selbst und der Interaktionspartner im Aufmerksamkeitsfokus. Eine Form des dyadischen Engagements ist das gegenseitige Anschauen (*mutual gaze*). Dabei blicken sich die Interaktanten direkt in die Augen und es können emotionale Ausdrücke ausgetauscht und die Verhaltensweisen des Gegenübers widerspiegelt werden. Durch das Austauschen von Emotionen entsteht eine Verbindung zwischen den Agenten. Das gegenseitige Aufeinandereingehen und Widerspiegeln der Emotionen des Anderen findet in einer turn-basierten Weise statt (Tomasello et al., 2005).

### 3.2.2 Aufmerksamkeitsfokus - Blickbewegungen

Im Rahmen einer Kooperation ist es von entscheidender Bedeutung, den Aufmerksamkeitsfokus des Gegenübers wahrzunehmen: Der Kooperationspartner schaut ein Objekt an oder er schaut seinem Kooperationspartner in die Augen. Das Blickverhalten erlaubt Rückschlüsse über den mentalen Zustand des Gegenüber und kann für eine kontextspezifische Referenzauflösung eingesetzt werden (siehe Abschnitt 6.1.2).

Das Aufmerksamkeitsverhalten und die damit verbundenen Blickbewegungen eines Agenten haben zwei grundlegende Ursachen. Zum einen tritt eine Verschiebung des Aufmerksamkeitsfokus im Rahmen interner Prozesse des Agenten auf und gewährt damit einen Einblick in interne Verarbeitungsprozesse, z.B. kann ein Fokussieren von Objekten darauf hindeuten, dass eine Referenz aufgelöst wurde oder dass die Objekte als Kandidaten für einen Plan angesehen werden. Zum anderen passt der Agent seinen Aufmerksamkeitsfokus an, während er auf sein Gegenüber eingeht.

### 3.2.3 Gemeinsame Aufmerksamkeit (*Joint Attention*)

**Anforderungen** In Anlehnung an Kaplan & Hafner (2006) sehen wir *Joint Attention* als einen bilateralen aktiven Prozess, welcher das Alternieren der Aufmerksamkeit beinhaltet und nur in seiner Komplexität verstanden werden kann, wenn man davon ausgeht, dass er von intentionalen Agenten realisiert wird. Um *Joint Attention* zu erreichen, müssen die Agenten sich der Koordinationsmechanismen bewusst sein, die sich auf die Analyse, das Überwachen und das Ausrichten der Intentionen beziehen, die dem Verhalten des Gegenübers zugrunde liegen. Sie müssen sich des Aufmerksamkeitsfokus des Gegenübers ebenso bewusst sein wie des Prozesses des Teilens der Aufmerksamkeit selbst (Hobson, 2005).

Hierzu muss der Agent in der Lage sein,

- (r1) das Aufmerksamkeitsverhalten des Gegenübers durch Blickverfolgung zu überwachen,
- (r2) die Menge der Objektkandidaten abzuleiten, auf welche der Interaktionspartner seinen Fokus ausrichtet,
- (r3) zu erkennen, ob die Aufmerksamkeitssignale des Gegenübers intentional ausgesandt werden oder nicht,
- (r4) augenblicklich auf sein Gegenüber zu reagieren und einzugehen, da Simultaneität eine entscheidende Rolle im Rahmen der *Joint Attention* spielt,

(r5) ein ädequates, offenkundiges Verhalten zu zeigen, welches sein Gegenüber wahrnehmen und erkennen kann.

Um das Aufmerksamkeitsverhalten des Interlokutors beeinflussen zu können, sollte der Agent proto-deklaratives Zeigen beherrschen, die Fähigkeit, durch Zeigen einen Aspekt der Welt zu kommentieren oder eine Person auf etwas hinzuweisen. Dieses Verhalten kann eingesetzt werden, wenn Blickverhalten allein nicht ausreicht. Es können auch verbale Referenzen, Lächeln, Nicken und Kopfgesten als aufmerksamkeitslenkend (*initiate-act*) oder als Feedback (*respond-act*) eingesetzt werden (Tasker & Schmidt, 2008).

Wenn Interaktionspartner das gleiche Objekt fokussieren, entsteht zunächst eine verhaltensbasierte Koordination. Sie lässt sich als *gemeinsame Wahrnehmung* (*Shared Attention*) beschreiben. Gemeinsame Aufmerksamkeit (*Joint Attention*) geht darüber hinaus und basiert auf einer *intendierten Koordination* der Aufmerksamkeit. Es muss das Ziel sein, die Ausrichtung der Aufmerksamkeit gemeinsam zu koordinieren. Eine Definition der *gemeinsamen Aufmerksamkeit* wird von Brinck (2003) aufgestellt: Gemeinsame Aufmerksamkeit tritt demnach dann auf, wenn zwei oder mehrere Agenten als Konsequenz der Beobachtung des Aufmerksamkeitszustands des Gegenübers simultan ihre Wahrnehmung auf einen gemeinsamen Stimulus ausrichten.

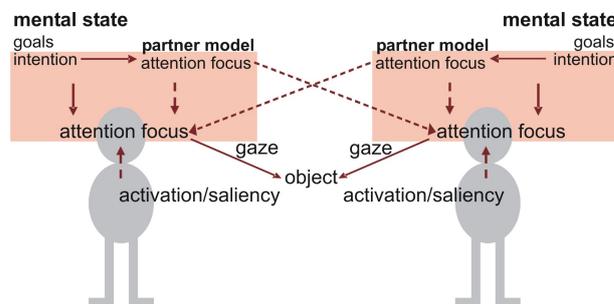


Abbildung 3.3: Joint Attention-Modell

Im Rahmen des BCTC-Modells wird gemeinsame Aufmerksamkeit wie in Abbildung 3.3 modelliert (Pfeiffer-Leßmann & Wachsmuth, 2008). Um *Joint Attention* zu erreichen, müssen drei Einflussfaktoren berücksichtigt werden: der mentale Zustand, ein Partnermodell und Aktivierungsprozesse. Der **mentale Zustand** des Agenten wirkt auf Aufmerksamkeitsprozesse und ermöglicht intentional gesteuertes Verhalten. Das für die Modellierung von *Joint Attention* benötigte **Partnermodell** beinhaltet den Aufmerksamkeitsfokus des Interlokutors. Dieser muss aus dem overt Verhalten des Gegenübers abgeleitet werden. Im Rahmen der Situiertheit des Agenten müssen zusätzlich Umweltfaktoren berücksichtigt werden. Dies wird

durch *bottom-up* **Aktivierungsprozesse** erreicht, welche relevante Objekte als salient markieren. Um den Prozess des Erreichens gemeinsamer Aufmerksamkeit zu unterstützen, muss der Agent seinerseits seinen eigenen Aufmerksamkeitsfokus durch overt Verhalten so verdeutlichen, dass sein Interlokutor ihn inferieren kann. Dabei können auch Zeigegesten und verbale Ausdrücke eingesetzt werden.

In dem BCTC-Modell werden die folgenden Heuristiken zur Steuerung des Aufmerksamkeitsfokus aufgestellt (formale Umsetzung der Regeln siehe Abschnitt 5.2.10):

**(Att1)** Wenn der Agent in keine Interaktion involviert ist und keine konkreten Ziele verfolgt, welche sein Blickverhalten aktiv beeinflussen, fokussiert er (als Basismechanismus) die Objekte, welche für ihn die größte Signifikanz besitzen.

**(Att2)** Im Falle einer Interaktionssituation greift ein deliberativer Mechanismus, welcher für die Dauer der Interaktion angestoßen wird und Intentionen in Bezug auf das Blickverhalten des Agenten modelliert. Ist der Interaktionspartner im Besitz des Turns, verfolgt der Agent eine Form des *aktiven Zuhörens*. Dies äußert sich darin, dass er den Sprecher während seines Turns anschaut und dessen Aufmerksamkeitsfokus zu verfolgen sucht.

**(Att3)** Fokussiert der Interaktant ein Objekt in einem Zeitfenster mehrfach und fokussiert zwischendurch auch immer wieder den Agenten, so wird dies als Versuch interpretiert, *Joint Attention* herzustellen. Der Agent erwidert das Verhalten, sodass es zu einer *Joint Attention*-Situation kommt. Neben der Blickfokussierung können auch weitere non-verbale Signale wie Zeigegesten, aber auch verbale Referenzen zur Manipulation des Aufmerksamkeitsfokus eingesetzt werden.

**(Att4)** Befindet sich der Agent in einer Kooperationssituation mit seinem Gegenüber, so lenkt der Agent seinen Aufmerksamkeitsfokus auch bei signifikanten Ereignissen auf die Handlungen seines Gegenübers, selbst wenn gerade keine Konversation zwischen den beiden Interaktanten stattfindet.

**(Att5)** Um einen Einblick in seine internen Verarbeitungen und Prozesse zu gewähren, zeigt der Agent mittels Blickverhalten, welche Objekte gerade in seinem Aufmerksamkeitsfokus liegen.

### 3.3 Konversationsebene (*Conversation Layer*)

Auf der *Conversation*-Ebene wirken Mechanismen für die strukturelle Organisation einer Interaktion und die Dialogsteuerung der Interaktanten. Diese Mechanismen können bis zu einem bestimmten Grad ohne Berücksichtigung des inhaltlichen Beitrags operieren. Sie bilden eine *interaktionale Grundkompetenz*. Sie bewirken insbesondere das Einleiten einer Konversation und das Beenden derselben. Um während einer Interaktion ein geordnetes, effizientes Zusammenspiel der Beiträge der Interaktanten zu gewährleisten, muss der Agent ein Modell des Konversationszustands (*conversational state*) verwalten. Das Modell ist zum einen Grundlage für die Klassifizierung der Verhaltensweisen des Gegenübers, zum anderen berücksichtigt es auch die eigene Verhaltensgenerierung. Zu dem Repertoire der Konversationsebene zählen auch soziale Regeln. Sie beinhalten die konversationalen Obligationen des Agenten und seine Erwartungen an das Verhalten seines Gegenübers.

Die *Conversation*-Ebene regelt Einhalten von Diskursregeln und sozialen Obligationen. Aufmerksamkeitssignale sowie explizite *Turn-Taking*-Signale veranschaulichen, dass Konversationspartner eine Interaktion durch explizite Beiträge steuern können.

Die Regeln, die für Interaktionen auf der *Conversation*-Ebene gelten, resultieren unter anderem aus den von Bunt (1996a) postulierten *dialogue control acts* der DIT (siehe Abschnitt 2.2.1). Insbesondere die **Interaktionsmanagementfunktionen** des **Kontaktmanagements**, des **Turn-Managements** und der **Diskursstrukturierung** werden auf der *Conversation*-Ebene modelliert.

#### 3.3.1 Turn-Management

Turn-Management bezieht sich auf das Aushandeln des Rederechts. Das BCTC-Modell folgt dem Ansatz von Goodwin (1981b), die Aushandlung des Turns als einen kooperativen Prozess zu sehen. Dabei wird jedoch in Anlehnung an Sacks et al. (1974) davon ausgegangen, dass im Rahmen einer Konversation kontextfreie Regeln bestehen, die dem Agenten seine Handlungsspielräume aufzeigen. Diese Regeln werden in dem BCTC-Modell als konversationale Obligationen konzeptualisiert. Der Agent wird sich im Normalfall an die Regeln halten, er kann aber auch aus eigener Intention gegen sie verstoßen. Die Handlungsoptionen, die durch die kontextfreien Regeln spezifiziert sind, lassen Freiräume, in welchen der Turn durch interaktive Signale ausgehandelt werden kann.

Eingehende Signale des Gegenübers werden auf *Turn-Taking*-Funktionen abgebildet, sodass die Entscheidungsgrundlage für angemessene Reaktionen des Agenten nicht aus dem Signal selbst besteht, sondern die Funktion der Signale

(gegebenenfalls zusammen mit einem Maß für die Stärke des Signals) als Berechnungsgrundlage dient. Folgende *Turn-Taking*-Funktionen werden unterschieden:

- *taking-turn*
- *wanting-turn*
- *yielding-turn*
- *giving-turn*

Wie von Cassell, Sullivan et al. (2000) gefordert, wird dabei von multimodalen *Turn-Taking*-Signalen ausgegangen, die sowohl aus verbalen Äußerungen oder aus Gesten (z.B. Blick, Mimik, Postur) als auch aus Kombinationen derselben bestehen können. Dabei ist die Detektion in beiden konversationalen Kanälen kontextsensitiv.

In dem BCTC-Modell werden die *konversationalen Rollen* der Interlokutoren getrennt von dem *konversationalen Zustand* modelliert. Die aktuell zu vergebenden Rollen bestehen aus *speaker* und *listener*, das Modell ließe sich aber auch um weitere Rollen erweitern, um eine Konversation zwischen mehreren Personen zu modellieren. Der konversationale Zustand wird aus der Sicht des Agenten modelliert und kann neben den Werten eines Sprechers und eines Zuhörers auch zusätzlich die Werte *gap* und *overlap* annehmen. Diese werden in den meisten klassischen *Turn-Taking*-Systemen als Fehlerzustand bezeichnet; in dem BCTC-Modell werden sie als Übergangszustände klassifiziert (Leßmann et al., 2004). Auch wenn die Interlokutoren versuchen, diese Zustände zu vermeiden, so sind sie dennoch ein häufig zu beobachtenden Phänomen in natürlichen Konversationen, in welchen der Turn ausgehandelt wird. Die Verarbeitung dieser Zustände stellt einen wichtigen Bestandteil des *Turn-Takings* dar. Deshalb werden im BCTC-Modell die folgenden konversationalen Zustände unterschieden:

\* *no-conversation*

Sobald der Agent eine Konversation mit seinem Gegenüber eingeht,

- \* *my-turn*
- \* *others-turn* (oder der Name des Agenten)
- \* *gap*
- \* *overlap*

Die Regeln, welche die Handlungsoptionen des Agenten in Abhängigkeit von dem konversationalen Zustand sowie der erkannten konversationalen Funktion

Conversational State	Detected Conversational Function	Alternative Reactions
MyTurn	WantingTurn TakingTurn	GivingTurn, YieldingTurn, HoldingTurn GivingTurn, HoldingTurn
OthersTurn	GivingTurn YieldingTurn HoldingTurn	TakingTurn TakingTurn No reaction, WantingTurn
Gap	WantingTurn TakingTurn GivingTurn YieldingTurn	GivingTurn, WantingTurn No reaction TakingTurn TakingTurn, GivingTurn
Overlap	GivingTurn YieldingTurn HoldingTurn	No reaction No reaction, GivingTurn GivingTurn

Tabelle 3.1: Mögliche Reaktionen auf *Turn-Taking*-Signale in Abhängigkeit des konversationalen Zustands und der detektierten konversationalen Funktion

spezifizieren, sind in Tabelle 3.1 (nach Kranstedt (Leßmann et al., 2004)) aufgelistet. Die letztendliche Entscheidung des Agenten unterliegt jedoch einer Bandbreite von Kontexteinflüssen, aus denen sein Gesamtverhalten resultiert.

Die Regeln werden dabei, wie von Cassell, Bickmore, Campbell et al. (2000a) gefordert, symmetrisch eingesetzt. Einerseits dienen sie dazu, die Handlungsspielräume des Agenten in Form von Obligationen festzulegen. Andererseits können sie auch als Grundlage für die Erwartungen des Agenten eingesetzt werden, der in den jeweiligen konversationalen Zuständen nach den entsprechenden konversationalen Funktionen suchen wird (z. B. wenn der konversationale Zustand aus *MyTurn* besteht, wird der Agent sensitiv auf die konversationalen Funktionen *TakingTurn* und *WantingTurn* reagieren).

Der Agent muss jedoch nicht nur die *Turn-Taking*-Signale seines Gegenübers erkennen, sondern er muss auch selbst aktiv Signale produzieren. Dabei propagieren die Signale des Agenten seine Intentionen nach außen und spiegeln beispielsweise die Dringlichkeit seines Anliegens wider. Hat der Agent mit den bisher ausgesandten Signalen keinen Erfolg, so kann er deutlichere Signale produzieren. Für die Generierung von *Turn-Taking*-Signalen muss der Agent zudem die Echtzeitanforderungen und die Einhaltung von Fristen (siehe Abschnitt 2.2.2) berücksichtigen.

Durch die Unterbrechung eines Turns wird es möglich, Zwischenfragen zu stellen, Einwände zu erheben und sich schneller und effektiver in eine Interaktion einzubringen, als das bei einer strikten festgelegten Einhaltung des Rede-rechts geschieht. Für eine angemessene Verarbeitung solcher Beiträge muss der Agent verschiedene Konversationsbeiträge und deren zugrunde liegende intentionale Struktur repräsentieren und vorhalten können.

### 3.3.2 Kontaktmanagement

Um mit seinem Gegenüber in eine Interaktion treten zu können, benötigt der Agent Interaktionskompetenzen. Zu diesen zählt das Beachten sozialer Regeln der Interaktion. Aus den Untersuchungen zwischenmenschlicher Interaktion lassen sich einige grundlegende Regeln des Kontaktmanagements gewinnen (siehe auch Arbeiten von Nakano & Nishida (2005a) und Sidner & Lee (2003)). Für eine Beschreibung der Handlungen und Prädikate siehe Tabelle A.1 und A.2 im Anhand A.

**(k1) Einleiten eines Kontakts:** Wenn ein potentieller Konversationspartner  $j$  wahrgenommen wird, diesen kurz demonstrativ anschauen und begrüßen. Führt der Agent gerade keine Konversation, den potentiellen Konversationspartner  $j$  zu einer Konversation einladen. Wird der Agent  $i$  von einem anderen Agenten  $j$  begrüßt, zurückgrüßen, Aufmerksamkeitsfokus auf Agent  $j$  ausrichten und Bereitschaft zur Interaktion signalisieren.

**(contact-rule 1a)**  $(DONE \{P(j)\}_i \wedge (is\_a\ j, agent)) \wedge \neg(BEL_i \{facing\ j\}) \Rightarrow \{Greet(j)\}_i$

**(contact-rule 1b)**  $(DONE \{P(j)\}_i \wedge (is\_a\ j, agent)) \wedge \neg(BEL_i(facing\ j)) \wedge BEL_i(convState\ noConv) \Rightarrow \{BeginConv(j)\}_i$

**(contact-rule 1c)**  $(DONE \{Greet(i)\}_j) \Rightarrow \{Greet(j)\}_i \wedge \{P(j)\}_i$

**(contact-rule 1d)**  $(GOAL_i(\{Speechact(j\ \varphi)\}) \wedge BEL_i(convState\ noConv)) \Rightarrow \{BeginConv(j)\}_i$

**(k2) Während eines Kontakts:** Sobald der Agent in eine Interaktion involviert ist, den Interlokutor immer anschauen, wenn er den Turn ergreift. Während des Turns hauptsächlich den Interlokutor anschauen, welcher Konversationspartner ist (siehe *Turn-Taking* Abschnitt 3.3.1). Während des Kontakts Aufmerksamkeitsfokus auch auf Handlungen des Interlokutors ausrichten.

**(contact-rule 2a)**  $BEL_i(conversationWith\ j) \wedge BEL_i(convState\ othersTurn) \wedge DONE(\{TakingTurn\}_j) \Rightarrow \{T(j)\}_i$

**(contact-rule 2b)**  $BEL_i(\text{conversationWith } j) \wedge (DONE \{TakingTurn\}_i \wedge DONE(\{Speechact(j, \varphi)\}_i) \Rightarrow \{T(\theta)\}_i \wedge \theta \neq j$

**(contact-rule 2c)**  $BEL_i(\text{conversationWith } j) \wedge (DONE \{\alpha(\theta)\}_j \Rightarrow \{T(\theta)\}_i \alpha \in ACT$

**(k3) Beenden eines Kontakts:** Feststellen ob der Interlokutor entweder den Turn nicht ergreift, wenn er dies müsste, oder sich aus dem visuellen Blickfeld des Agenten bewegt. Sich mittels Dialog versichern, ob der Interlokutor wirklich die Interaktion beenden möchte. Die Interaktion beenden, wenn der Interlokutor die Interaktion abrechnen möchte. Die Konversation beenden, wenn der Agent seine konversationalen Ziele erreicht hat.

**(contact-rule 3a)**  $BEL_i(\text{conversationWith } j) \wedge (\neg(\exists t_1 \mid \{P(j)\}_i)_{[t_1]} \wedge (t_1 - t_{now} \leq 10s)) \Rightarrow BEL_i(POSSIBLY(GOAL_j(\text{convState noConv})))$

**(contact-rule 3b)**  $BEL_i(\text{conversationWith } j) \wedge DONE(\{GiveTurn(j)\}_i)_{[t_1]} \wedge \neg(\exists t_2 \mid DONE(\{TakeTurn\}_j)_{[t_2]} \wedge ((t_2 - t_1) \leq 6s)) \Rightarrow BEL_i(POSSIBLY(GOAL_j(\text{convState noConv})))$

**(contact-rule 3c)**  $BEL_i(\text{conversationWith } j) \wedge BEL_i(POSSIBLY(GOAL_j(\text{convState noConv}))) \Rightarrow \{VERIFY(BEL_i(GOAL_j(\text{convState noConv})))\}_i$

**(contact-rule 3d)**  $BEL_i(\text{conversationWith } j) \wedge BEL_i(GOAL_j(\text{convState noConv})) \Rightarrow \{EndConv(j)\}_i$

**(contact-rule 3e)**  $BEL_i(\text{conversationWith } j) \wedge (\neg GOAL_i(\text{conversationWith } j) \vee \neg(\exists g \in GOAL_i \mid g(\varphi) \wedge j \in t\_set(\varphi))) \Rightarrow \{EndConv(j)\}_i$

Das Befolgen solcher Regeln setzt Wahrnehmungs- und Interpretationsfähigkeiten des Agenten voraus. Die Regeln müssen in die Generierung der Handlungsoptionen des Agenten eingeflochten werden. Einige der Regeln stellen dabei Vorbedingungen für Interaktionshandlungen dar (beispielsweise nur dann eine Frage an ein Gegenüber zu richten, wenn bereits eine Interaktion eingeleitet wurde). Dabei sollten solche Anforderungen jedoch nicht „fest verdrahtet“ eingesetzt werden, sondern nur als Richtlinien und Obligationen vorliegen, die der Agent notfalls auch verletzen könnte. Andere Regeln implizieren Mechanismen, welche den Aufmerksamkeitsfokus des Agenten immer wieder auf bestimmte Aspekte und Signale richten und überwachen, ob bestimmte Regeln vom Konversationspartner eingehalten werden. Damit der Agent diese Aspekte auch während seiner Äußerungen im Blick behalten kann, ist ein paralleles Operieren der Wahrnehmung und Verhaltensgenerierung erforderlich.

### 3.3.3 Diskursstrukturierung - Initiative

Während das Turn-Management allein das Rederecht umfasst, bezieht sich die Initiative auf die Inhalte des Interaktionsgeschehens. Der Halter der Initiative besteht aus dem Initiator einer Interaktion bzw. aus dem Agenten, der den inhaltlichen Verlauf der Interaktion bestimmt, z.B. besitzt ein Anrufer eines Auskunftssystems die Initiative, wohingegen das Auskunftssystem, selbst wenn es das Rederecht (den Turn) ergreift, um eine klärende Frage zu stellen, dadurch nicht die Initiative erhält.

Die Initiativmodellierung orientiert sich im BCTC-Modell an den Arbeiten von Strayer & Heeman (2001) und Ferguson & Allen (2007). Es wird davon ausgegangen, dass die Initiative emergent aus der Intentions- und Diskursstruktur hervorgeht. Dafür ist jedoch eine genaue Modellierung der Diskursstruktur von entscheidender Bedeutung. Um die Diskursstruktur zu erfassen, sind insbesondere die Intentionen des Gegenübers zu beachten. Sie müssen aus seinen Handlungen abgeleitet werden. Ein erster Schritt besteht darin, das *Performativ* der Beiträge des Gegenübers zu erkennen. Ausgehend von der Klassifizierung des Performativs werden in dem BCTC-Modell Regeln aufgestellt, welche helfen, die Diskursstruktur zu erkennen und darauf aufbauend die Initiative zu modellieren.

**Performativ** Die Kodierung des Typs eines kommunikativen Akts wird im Falle einer sprachlichen Äußerung in der vorliegenden Arbeit explizit durch entsprechende Performative durchgeführt (siehe *Interaction-Moves* Abschnitt 3.2). Dafür wird die Klassifikation von Poggi & Pelachaud (2000), welche *inform*, *query* und *request* unterscheiden, herangezogen. Diese Klassifikation reicht aber für die komplexe Kooperationsmodellierung des BCTC-Modells nicht aus; sie erfordert eine wesentliche Erweiterung. Diese Erweiterung besteht in der Ergänzung der Klassifikation um einen vierten Performativtyp, dem *propose*-Performativ. Das *propose*-Performativ spielt eine zentrale Rolle in Kooperationssituationen, in denen der Lösungsprozess mit seinem Handlungsablauf ausgehandelt wird. Die Performative werden in der vorliegenden Arbeit auf folgende Weise klassifiziert:

Performative:

- *inform*: Übermittlung von Fakten, **Ziel**: Annahmen (*Beliefs*) des Adressaten ändern
- *query*: Abfrage von Informationen, **Ziel**: ein neues *Belief* zu erhalten oder ein bestehendes zu verifizieren
- *request*: Aufforderung zu einer (zumeist manipulativen) Handlung, **Ziel**: Ausführung der Handlung

- *propose*: Vorschlag einer Handlung, geprägt durch ein Maß an Kooperationsbereitschaft oder Unsicherheit, **Ziel**: Aushandeln mit Interaktionspartner

Die Basisstruktur ist jetzt durch die vier so ergänzten Performative gegeben. Oft lassen sich die Performative jedoch durch anschließende Spezifikation während des Generierungs- und des Interpretationsprozesses noch feiner charakterisieren (Poggi & Pelachaud, 2000). Das fein spezifizierte Performativ kodiert detailliert den Inhalt (z. B. ob etwas behauptet wird (*inform.assert*), oder ob eine Forderung (*request.order*), eine Bitte (*request.beg*) oder Meinung (z.B. *inform.compliment*) vorliegt) und kontextspezifische Faktoren.

Die Tabelle 3.2 listet Verfeinerungen der *Performative* in Form einer kompositionellen Notation (z. B. *inform.agree* siehe auch (Leßmann et al., 2006)). Die Tabelle erhebt keinen Anspruch auf Vollständigkeit, präsentiert aber die Auswahl der wesentlichen Verfeinerungen, die im Rahmen der hier vorliegenden Arbeit ein natürliches Interaktionsverhalten besonders gut erfassen und abbilden können.

Für die Analyse der von einem Agenten empfangenen Sprechakte werden die folgenden Heuristiken eingesetzt: Für die Klassifizierung eines *query*-Performativs wird eine Äußerung nach Fragewörtern untersucht sowie der Satzbau analysiert. *Propose*-Performative lassen sich meist durch den Einsatz modaler Hilfsverben (können, wollen, sollen) sowie des Konjunktivs erkennen. *Request*-Performative zeichnen sich durch eine imperative Struktur aus.

Um den Halter der Initiative identifizieren zu können, werden sowohl das Performativ als auch die zugehörige Diskursfunktion des Aktes benötigt. Dabei wird davon ausgegangen, dass die Initiative aus der Intensionsstruktur abgeleitet werden kann (siehe Abschnitt 2.2.3). Wie in Lochbaum (1998) wird in dem BCTC-Modell davon ausgegangen, dass die Initiative in Falle von *inform*- und *query*-Performatives von dem Diskurssegment-Initiator gehalten wird, während der Nicht-Initiator nur zu dem aktuellen Segment beiträgt (z. B. indem er auf Fragen in einem Teildialog antwortet). Mit einem Befehl (*request*) jedoch ergreift ein Agent die Initiative und beginnt ein neues Diskurssegment. Die zusätzliche Einbeziehung des *propose*-Performativs in die Intitativmodellierung führt im Rahmen der in der vorliegenden Arbeit vorgenommenen BCTC-Modellbildung zu folgenden Regeln (siehe Abbildung 3.4):

#### **Initiativregeln:**

- (i1) Wenn das Performativ als *inform*-Performativ auftritt, hat der Sprecher die Initiative, vorausgesetzt, es beginnt ein neues Diskurssegment.
- (i2) Im Falle eines *query*-Performativs hat der Sprecher die Initiative, außer wenn

<b>Inform</b>		
inform.assert	proposition	assert a fact
inform.reassert	proposition	assert a fact again
inform.confirm	proposition	confirm/disconfirm
inform.disconfirm		the truth value of a fact
inform.ref	referential exp	describe a reference
inform.understood	proposition/	signals (no) understanding
inform.not_understood	action exp	optionally a reason can be given
inform.compliment	proposition/	signals affective state
inform.criticize	action exp	towards action or attribute
inform.commit	action exp/	signals disposition to perform an action optionally a condition or reason can be given
inform.refuse	proposition	
inform.agree	action exp/	signals agreement optionally
inform.disagree	proposition	a condition or reason can be given
<b>Query</b>		
query.if	action exp/ proposition	question that can be answered by yes/no
query.ref	referential exp	question about a reference
query.how	action descr/ proposition	question about an action/ proposition
query.who	actor descr	question about an actor/action
<b>Request</b>		
request.order	action exp	own interest, dominant
request.advice	action exp	other's interest
request.beg	action exp	own interest, polite/ subordinate
request.indirect	action exp	indirect form of a request/propose
<b>Propose</b>		
propose.proposition	proposition	uncertain, expressing an opinion
propose.action	action exp	uncertain, proposing an action
propose.action.offer	action exp	uncertain, offer to perform an action
propose.ref	referential exp	uncertain, reference

Tabelle 3.2: Kompositionelle Notation der Performative

es sich um eine Klärungsfrage handelt, welche zu einem Diskurssegment beiträgt.

- (i3) Das *request*-Performativ überträgt die Initiative immer an den Ausführenden des Performativs.
- (i4) Im Falle eines *propose*-Performativs ist es nicht von vornherein klar, ob die Initiative zu dem Sprecher wechselt oder nicht. Dies ist abhängig von der Reaktion des Gegenübers. Wenn der Sprecher versucht, ein neues Segment zu beginnen und der Interlokutor den Vorschlag akzeptiert, so wird die Initiative auf den Sprecher übertragen. Wird der Vorschlag jedoch nicht anerkannt, so bleibt der Zustand der Initiativzuordnung bestehen.

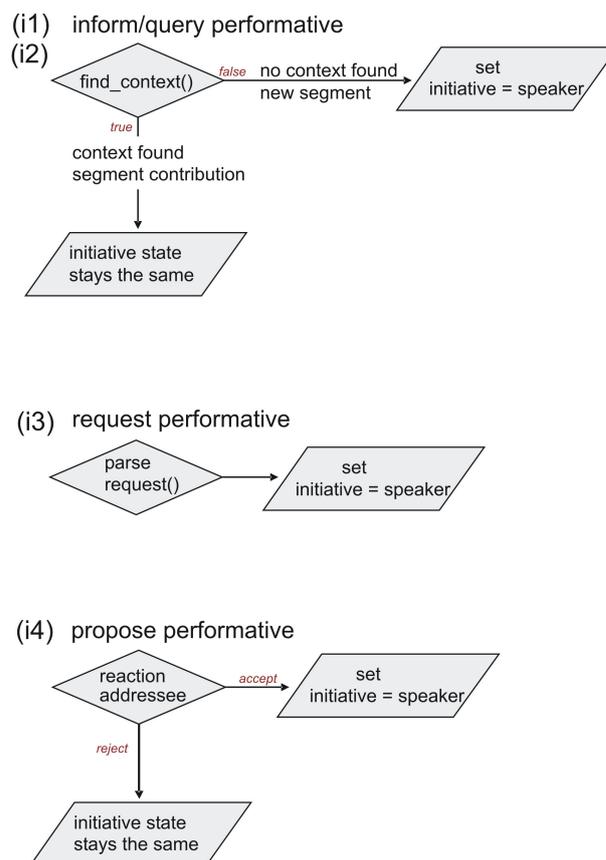


Abbildung 3.4: Regeln der Initiativmodellierung

### 3.3.4 Obligationen

Obligationen stellen konkrete Verpflichtungen dar, denen die Teilnehmer einer Interaktion unterliegen. Die Beeinflussung des Agenten durch den Einsatz von Obligationen ermöglicht es einerseits, das Verhalten des Agenten zu steuern und nach sozialen Normen auszurichten. Andererseits besteht jedoch durch den agentenbasierten, kognitiven ganzheitlichen Ansatz die Möglichkeit, gegen Obligationen zu verstoßen. Es wird kein festes Regelsystem aufgestellt, sondern vielmehr werden die Handlungsoptionen aufgezeigt und gewichtet.

Intentionen des Agenten und Obligationen können in einem Konflikt zueinander stehen. Als Entscheidungsgrundlage müssen soziale Konventionen sowie die soziale Beziehung zwischen den Agenten (z.B. *master-slave*, gleichberechtigt) berücksichtigt werden. Da der Agent in dieser Arbeit als kooperativer Interaktionspartner modelliert werden soll, wird die Modellierung von Obligationen direkt vorgenommen.

Der Agent wird dazu verpflichtet, auf alle Anweisungen, Fragen und Vorschläge zu reagieren. Es ist dem Agenten jedoch das Wie seiner Reaktion freigestellt, also ob er dabei die Fragen beantwortet, Anweisungen ausführt oder seine Weigerung bekannt gibt, auf das Gesagte einzugehen. Er kann Rückfragen stellen, wenn er nicht alles verstanden hat. Andererseits setzt der Agent auch Erwartungen in das Verhalten seines Interaktionspartners. Er erwartet, dass auch dieser die Diskursregeln und die damit verbundenen Obligationen einhält. Die daraus entstehenden Erwartungen lassen sich folgendermaßen beschreiben.

Sei  $\phi$  eine Zustandsformel,  $Inform, Inform.disconfirm \in ACT$ ,  $involved \in PREDICATES$ ,

es gelten folgende Erwartungen:

- (e1) Der Agent sieht die grundlegende Semantik eines *inform*-Performativs darin, den Interlokutor mit Informationen zu versorgen. Wenn der Agent ein solches Performativ benutzt, erwartet er nicht unbedingt eine Antwort. Wenn der Adressat nicht antwortet, aber dennoch in der Interaktion engagiert auftritt, geht der Agent davon aus, dass sein Adressat die Information verstanden hat und in einem gewissen Maß damit übereinstimmt. Damit ist für den Agenten sein Ziel allein durch das Ausführen des kommunikativen Aktes erreicht, vorausgesetzt sein Gegenüber ist noch in die Interaktion involviert (*maintain interaction*) und widerspricht der Information nicht.

$$DONE(\{Inform(j, \phi)\}_i) \wedge (involved(j)) \wedge \neg(DONE(\{Inform.disconfirm(i, \phi)\}_j)) \Rightarrow BEL_i(BEL_j(\phi))$$

- (e2) Im Gegensatz zu einem *inform*-Performativ erwartet der Agent eine Reaktion auf eine Frage (*query*-Performativ). Der Adressat kann dem Agenten mitteilen, dass er die Antwort auf die Frage nicht kennt oder nicht antworten

möchte, aber er unterliegt der Obligation, explizit auf die Frage zu antworten. Wenn der Adressat in keiner Weise antwortet, ist das Ziel des Agenten nicht erreicht, und er wird versuchen, sein Gegenüber zum Antworten zu bewegen.

$$\begin{aligned} DONE(\{Query(j, \phi)\}_i) \Rightarrow \\ & WAIT_i(DONE \{Inform(i, \phi)\}_j) \\ & \vee DONE \{Inform.not\_known(i, \phi)\}_j \\ & \vee DONE \{Inform.refuse(i, \phi)\}_j \end{aligned}$$

$$\begin{aligned} DONE(\{QUERY.if(j, \phi)\}_i) \Rightarrow \\ & WAIT_i(DONE \{Inform.confirm(i, \phi)\}_j) \\ & \vee DONE \{Inform.disconfirm(i, \phi)\}_j \\ & \vee DONE \{Inform.not\_known(i, \phi)\}_j \\ & \vee DONE \{Inform.refuse(i, \phi)\}_j \end{aligned}$$

- (e3) Das Gleiche gilt für eine Aufforderung bzw. Anweisung (*request*-Performativ). Allerdings unterscheiden sich in diesem Fall die Erwartungen des Agenten insofern, als hier nicht ein kommunikativer Akt allein, sondern auch die geforderte Handlung das Ziel des Agenten erfüllt. Der Agent wird deshalb warten, bis sein Gegenüber entweder die Aktion ausführt oder ablehnt.

$$\begin{aligned} DONE(\{Request(j, \phi)\}_i) \Rightarrow \\ & WAIT_i(DONE \{Inform.confirm(i, \phi)\}_j) \\ & \vee DONE \{Inform.refuse(i, \phi)\}_j \\ & \vee DONE \{\phi\}_j \end{aligned}$$

- (e4) Das *propose*-Performativ liegt in seiner Wirkung in Bezug auf die Erwartungen des Agenten zwischen den Wirkungen der anderen Performative. Der Agent erwartet eine Reaktion, aber diese kann sehr unterschiedlich ausfallen. Sie kann von der Ausführung der Aktion bis hin zu einer Zustimmung oder einem Ablehnung eines Vorschlags reichen.

$$\begin{aligned} DONE(\{Propose(j, \phi)\}_i) \Rightarrow \\ & WAIT_i(DONE \{Inform.agree(i, \phi)\}_j) \\ & \vee DONE \{Inform.disagree(i, \phi)\}_j \\ & \vee DONE \{Inform.commit(i, \phi)\}_j \\ & \vee DONE \{Inform.refuse(i, \phi)\}_j \\ & \vee DONE \{\phi\}_j \end{aligned}$$

### 3.4 Aufgabenebene (*Task Layer*)

Der Schwerpunkt der *Task-Ebene* liegt auf der Bewältigung einer Aufgabe bzw. dem Erreichen eines Zielzustands. Der Prozess, der zu diesem Zustand führt, ist dabei sekundär. Die vorgesehenen Mechanismen können sowohl bei der Bearbeitung einer gemeinsamen Aufgabe Einsatz finden als auch für eine Einzelaufgabe genutzt werden. Im Falle einer Zusammenarbeit kann die Aufgabe zwischen den Agenten aufgeteilt werden und jeder der Agenten ist dann für einen Teil der Aufgabe verantwortlich. *Master-Slave*-Interaktionen sowie die Delegation von Aufgaben finden auf der *Task-Ebene* statt. Die Prozesse auf dieser Ebene bilden eine Vorstufe zur echten Kooperation. Die Agenten lösen eine Aufgabe und verpflichten sich dabei zur gegenseitigen Unterstützung. Die Unterstützung umfasst jedoch nicht, wie von Bratman (siehe Abschnitt 2.1.1) gefordert, dass ein Agent sich in die Rolle des anderen hinein versetzen kann und gegebenenfalls dessen vorgesehene Rolle bei der Bewältigung der Aufgabe übernimmt; es geht nur darum, dass beide Agenten versuchen, dieselbe Aufgabe zu lösen. Dabei muss nicht automatisch eine koordinierte Zusammenarbeit auf der Behavior-Ebene stattfinden. Die Agenten können sich komplett auf die eigenen Intentionen und Aktionen konzentrieren, solange sie nicht wissentlich den Anderen bei der Ausführung seiner Handlung behindern.

Auch auf der *Task-Ebene* gibt es Obligationen, die der Agent eingehen kann. Zu diesen zählen inhaltliche Abkommen in Bezug auf die gemeinsame Aufgabe. Der Agent muss sich an Versprechen halten, bestimmte Aktionen auszuführen. Um in Verhandlungen über das gemeinsame Ziel treten zu können, ist es für den Agenten hilfreich, über eine dynamische, kontext-sensitive Sprachgenerierung zu verfügen. Engagement auf der *Task-Ebene* erfordert nicht, dass die Interaktanten konkrete, gemeinsame Pläne aufstellen und ihre Handlungen im Detail koordinieren. Diese Verhaltensweisen sind vielmehr der nächst höheren Ebene (*Collaboration-Ebene*) zuzuordnen. Das Engagement auf der *Task-Ebene* besteht allerdings bereits aus einem triadischen Engagement, in dem Sinne, dass zusätzlich zu dem wechselseitigen Bezug aufeinander ein konkretes Ziel zu berücksichtigen ist (beispielsweise eine Konstruktion). Die Interaktanten müssen dabei jeweils in der Lage sein, zu planen sowie Pläne dynamisch situiert auszuführen. Die Repräsentation von Plänen betrifft nur abstrakte *intention-that* und eigene *intention-to* Pläne (siehe Abschnitt 2.1.3).

#### 3.4.1 Planen

Im Rahmen von Planungs- und Problemlösungsprozessen stellt *Means-Ends-Reasoning* ein mächtiges Werkzeug dar. Abstrakte hierarchische Pläne für das Erreichen des gewünschten Weltzustands bilden die Basis des erfolgreichen Pro-

blemlösens. Unter Berücksichtigung von Domänenwissen sowie der Abschätzung eigener Kompetenzen und Fähigkeiten können Handlungsoptionen erkannt und evaluiert werden. Dafür stellen explizit repräsentierte Wissensbasen eine wichtige Grundlage dar. Sowohl prozedurale Fähigkeiten des Agenten als auch domänen-spezifische Konzeptstrukturen lassen sich diesen Wissensbasen zuordnen. Aufbauend auf Wissensbasen umfassen Bewertungskriterien für Handlungsoptionen die folgenden Aspekte:

- Ressourcen, Aufwand und Kosten
- Konflikte mit anderen Zielen (negative Ziel-Interaktion)
- positive Seiteneffekte (positive Ziel-Interaktion)

Dabei können je nach Planungsbereich neben Ausschlusskriterien der einzelnen Handlungsoptionen aufgrund massiver Konflikte mit weiteren Zielen oder Plänen auch Effizienzeffekte berücksichtigt werden. Auf der *Task*-Ebene wird die Bewertung der Handlungsoptionen jedoch nur in Bezug auf eigene Pläne vorgenommen. Der Agent repräsentiert nicht die konkreten Pläne seines Kooperationspartners mit. Er muss so auch nicht Inkonsistenzen oder Konflikte im oder mit dem Aufgabenbereich seines Gegenübers erkennen; dies ist der *Collaboration*-Ebene zuzuordnen.

Für das dynamische Anpassen der Pläne werden Mechanismen zum Spezifizieren von Einschränkungen benötigt. Einschränkungen können von allgemein spezifizierten Attributen bis hin zu kontextsensitiven Bedingungen reichen, welche spezifizieren, dass beispielsweise ein bestimmtes Objekt nicht für die Aufgabe eingesetzt werden darf. Die zu berücksichtigten Einschränkungen von Planaspekten (Handlungsoptionen, beteiligte Objekte bzw. Agenten) lassen sich wie folgt charakterisieren:

- Ausschlusskriterien
- Attributeinschränkungen
- Belegungseinschränkungen
- zeitliche Einschränkungen

### 3.4.2 Dynamisch situiertes Problemlösen

Auf der *Task*-Ebene wird eine gemeinsame Aufgabe durch konkrete Aufgabenteilung sowie dynamische Interaktion bearbeitet. Dabei verfolgt jeder Agent ein Ziel und seinen damit verbundenen Plan, beobachtet seinen Interaktionspartner und

passt seinen Plan dynamisch an die Handlungen des Gegenübers an, das heißt: Es werden nicht im voraus ein gemeinsamer Plan aufgestellt und die Rollen der Ausführung im Detail verteilt, sondern Konflikte werden in dem Moment koordiniert, in dem sie auftreten (*dynamisch situiertes Problemlösen*). Der Agent muss dabei sowohl Sprechakte als auch manipulative Handlungen des Gegenübers im Hinblick auf den gemeinsamen Problemlösungsprozess bewerten. Der Agent profitiert von einer dynamischen Planrepräsentation, da er diese kontextsensitiv an die Umwelt und die Aktionen seines Interaktionspartners anpassen kann. Der Plan wird so offen wie möglich spezifiziert, um das Spektrum der Interaktionsmöglichkeiten flexibel zu gestalten. Die Aufgaben des Agenten sollten dafür in Form von explizit repräsentierten Zielen und Plänen verarbeitet werden.

### 3.5 Kooperationsebene (*Collaboration Layer*)

Die vierte Ebene bezieht sich auf das gemeinsame Lösen einer Aufgabe. Sie geht über die *Task*-Ebene insofern hinaus, als das auf dieser Ebene koordinierte Pläne und gemeinsame Intentionen der Interaktanten modelliert werden. Während sich die *Task*-Ebene nur auf eine konkrete Lösung eines Problems bzw. einer Aufgabe bezieht, wird auf der *Collaboration*-Ebene auch die Kooperation als solche thematisiert. Die Koordination der Handlungen und der Prozess selbst, welcher zu der Lösung der Aufgabe führt, stehen im Zentrum der Betrachtung. Neben der Berücksichtigung der Aufgabenstellung und des gemeinsamen Ziels, beachten die Interaktanten auf dieser Ebene auch den Zustand, die Fähigkeiten und Eigenschaften ihres Gegenübers. Die Kooperation beinhaltet auf dieser Ebene koordiniertes Handeln, bei welchem die Rolle des Interaktionspartners mitrepräsentiert wird, sodass die Rollen im Falle gleicher Kompetenzen getauscht werden könnten (*holistische* Repräsentation, siehe 2.1.4). Dies ermöglicht eine besondere Form der gegenseitigen Unterstützung und der gemeinsamen Intentionalität. Die internen Repräsentationen der Interaktanten müssen dabei auch die Beliefs und Intentionen des jeweils Anderen beinhalten (siehe Abschnitt 2.1.4). *Echte Kooperation* (siehe Definition 2, Abschnitt 2.1.6) besteht aus einem gemeinsamen Engagement der Beteiligten, welches während eines koordinierten Bestrebens auftritt, ein Problem zu lösen.

Auf der *Collaboration-Ebene* kann nicht mehr nur das Verhalten eines einzelnen Agenten betrachtet werden, es geht um das Zusammenspiel mehrerer, kooperierender Agenten. Wenn sich Agenten im Rahmen einer Kooperation auf ein gemeinsames Ziel festlegen und sich diesem verpflichten, so gehen damit informationelle Obligationen einher. Die Agenten müssen sich über Gruppenentscheidungen informieren (und diese gegebenenfalls aushandeln) und sie sind im Rahmen einer *echten Kooperation* auch dazu verpflichtet, sich über den Zustand gemein-

samer Ziele und Pläne zu informieren. Sie müssen aushandeln, wer zu dem Team gehört, das die Aufgabe bearbeitet, und welche Rezepte für die Erreichung des Ziels dienen sollen. Des Weiteren muss kommuniziert werden, wenn sich der Zustand eines gemeinsamen Ziels signifikant ändert (das Ziel plötzlich gilt, niemals wahr werden kann oder irrelevant wird (siehe Abschnitt 2.1.2)).

### 3.5.1 Gemeinsame Pläne

In der Kooperation spielen *gemeinsame Pläne* eine zentrale Rolle. Der *Shared-Plans*-Formalismus stellt eine konkrete Spezifikation der Anforderungen an die mentalen Zustände der Beteiligten einer Gruppenaktivität dar (siehe Abschnitt 2.1.3). Zusätzlich zu den vorher beschriebenen *Beliefs* und Intentionen müssen auf der *Collaboration*-Ebene auch ein gemeinsamer Plan bzw. das Rezept eines Problemlösungsprozesses und die damit verbundenen Intentionen repräsentiert werden; darin ist die Regelung der Rollenverteilung bei der Vergabe der Aufgaben enthalten (B. Grosz & Hunsberger, 2006). Wie von Pollack (1990) gefordert, wird zwischen Plänen in Form mentaler Einstellungen und Plänen in Form von Rezepten unterschieden.

Ein Rezept kann zunächst unvollständig vorliegen. Im Verlauf einer Interaktion wird es dann sukzessive verfeinert und ausgearbeitet. Dabei können im Rahmen der Kooperation Aufgaben an Kooperationspartner delegiert werden. Der Agent muss dennoch deren Ausführung und Erfolg überwachen, da sie zum gemeinsamen Gesamtplan gehören. Im Gegensatz zur *Task*-Ebene, auf welcher nur eigene bzw. abstrakte *intention-that*-Pläne und eigene *intention-to*-Pläne repräsentiert werden, können im Rahmen der Kooperation auch fremd auszuführende *intention-that*-Pläne auftreten. D.h. der Agent repräsentiert für jeden Teilplan eines seiner Kooperationspartner den gewünschten Weltzustand oder die gewünschte auszuübende Handlung und berücksichtigt sie in seinen Planungs- und Schlussfolgerungsprozessen. Damit finden Repräsentationen des intentionalen Zustands des Kooperationspartners Eingang in den mentalen Zustand des Agenten.

Diese „fremden“ Intentionen (Intentionen, die der Kooperationspartner ausführen soll) werden im Rahmen der Verhaltensgenerierung des Agenten gemäß ihrer Priorität genauso berücksichtigt wie persönliche Intentionen des Agenten. Für eine „fremde“ Intention gelten jedoch alternative Handlungsoptionen, welche nicht die intendierte Handlung direkt ausführen, sondern vielmehr spezifizieren, wie der Agent es erreichen kann, dass sein Gegenüber einen Teilplan ausführt. Zu den Handlungsoptionen in Bezug auf eine „fremde“ Intention zählen im BCTC-Modell:

- Abwarten
- Anweisung

- Vorschlag
- Lenkung der Aufmerksamkeit

Für eine echte Kooperation (siehe Definition 2, Abschnitt 2.1.6) müssen alle Gruppenentscheidungen unter den Kooperationspartnern abgestimmt werden.

### 3.5.2 Koordination - Verschachtelung von Planen und Handeln

Aufgrund des expliziten Aushandelns gemeinsamer Pläne kann es im Rahmen einer Kooperation zu einer Verschachtelung von Planen und Handeln kommen. Der Agent muss die Teilpläne seines Kooperationspartners in Betracht ziehen. Eine Zusammenführung der individuellen Pläne der Kooperationspartner (siehe Abschnitt 2.1.1) kann durch explizites Aushandeln sowie durch dynamische Anpassung während der Ausführung der Pläne erfolgen. Die Pläne müssen nicht miteinander übereinstimmen, aber die jeweilig spezifizierten Einschränkungen und Effekte dürfen nicht zu einem unlösbaren Widerspruch führen.

### 3.5.3 Repräsentation des Kooperationspartners

Die Agenten verpflichten sich im Rahmen einer echten Kooperation zur *gegenseitigen Unterstützung*. Die Verpflichtung entsteht im Rahmen einer Kooperation durch die Bitte eines Interaktionspartners um Hilfe. Diese Obligation beinhaltet, die Lage des Gegenübers abschätzen und sich ein Bild seines mentalen Zustands machen zu können (siehe Abschnitt 2.1.1). Nur wenn der Agent die Annahmen und Intentionen und Ziele seines Gegenübers berücksichtigt, kann er ihn wirklich unterstützen. Der Agent verfügt jedoch auch über die Option, die Kooperation abzubrechen. Kooperative Obligationen regeln zwar den allgemeinen Fall während einer Kooperation, sie stellen dabei jedoch lediglich Richtlinien für Verhaltensweisen dar. Der Agent sollte immer auch in der Lage sein, auf Kosten des eventuellen Abbruchs einer kooperativen Interaktion eigene Ziele zu verfolgen. Im Rahmen einer *echten Kooperation* (siehe Definition 2, Abschnitt 2.1.6) spielen damit ToM-Kompetenzen (*Theory of Mind*) eine wichtige Rolle. Die Repräsentation des Interaktionspartners kann unterschiedlich detailliert ausfallen. Es kann beispielsweise eine Repräsentation sämtlicher angenommener Intentionen, Überzeugungen und Pläne vorgenommen werden oder aber auch nur die Repräsentation der Intentionen und Pläne in Bezug auf eine konkrete Aufgabe.

## 3.6 Zusammenspiel der Ebenen

Im Folgenden werden die Prozesse zwischen den Ebenen behandelt. Abbildung 3.5 präsentiert die Einflussprozesse und Feedbackschleifen zwischen den einzelnen Ebenen. Dabei sind zwischen der Perzeptions- und der Generierungsschicht jeweils die Modelle und Komponenten angegeben, welche als Berechnungsgrundlage dienen.

**Feedbackschleifen** Kooperationsprozesse werden als bi-direktionale Prozesse gesehen. Die Interaktanten müssen jeweils die Signale ihres Gegenübers wahrnehmen, interpretieren und in die Berechnung ihrer Verhaltensgenerierung einbeziehen. Durch diesen Prozess entsteht eine Form des gegenseitigen Aufeinander-eingehens, welche die Grundlage einer jeden Kooperation darstellt. Um erfolgreich in einer gemeinsamen Umwelt zu agieren, müssen Wahrnehmungsprozesse sowohl die Umwelt als auch den Interaktionspartner genau erfassen. Perzepte müssen kontextsensitiv interpretiert werden, um sowohl den situationalen Kontext als auch komplexe Referenzen auf Objekte oder Ereignisse und gemeinsame Handlungen aufzulösen. Die Skala der auf der Perzeptionsschicht operierenden Prozesse reicht von rein reaktiven Prozessen, die beispielsweise den Aufmerksamkeitsfokus des Gegenübers verfolgen, bis hin zu deliberativen Interpretations- und Planungsprozessen, die den Inhalt und die Intention der Äußerungen des Gegenübers analysieren und dem inhaltlichen Kooperationsaspekt zuzuordnen sind. Andererseits müssen Engagementsignale erkannt und produziert werden. Dadurch werden Interesse und Beteiligung an der Interaktion sowie dem Gegenüber ein Gefühl von Akzeptanz vermittelt.

**Interaktion zwischen den Ebenen** Prozesse, die auf einer Ebene ablaufen, beeinflussen und stoßen Prozesse auf den jeweiligen benachbarten Ebenen an. Auf der *Generierungsschicht* triggert jede Ebene jeweils die darunter liegende. Eine Interaktion auf einer bestimmten Ebene regt auch Prozesse auf der darunter liegenden Ebene an, welche die Interaktion zusätzlich unterstützen. Möchte der Agent beispielsweise einen gemeinsamen Plan vorschlagen, so stößt er zusätzlich Prozesse an, welche ihm den Turn sichern, und er teilt auf unterster Ebene eventuell die Begeisterung über den Plan mit seinem Gegenüber durch den Austausch von emotionalen Gesichtsausdrücken.

In der *Perzeptionsebene* ist die Richtung der Interaktion zwischen den Ebenen entgegengesetzt. Sie verläuft in Richtung von *Behavior* zu *Conversation* über *Task* bis hin zu *Collaboration*. Hat der Agent Signale auf einer Ebene wahrgenommen, so werden Interpretationsmechanismen der nächst höher liegenden Ebene angestoßen, um zu prüfen, ob das Verhalten im aktuellen Kontext auch eine Signalwir-

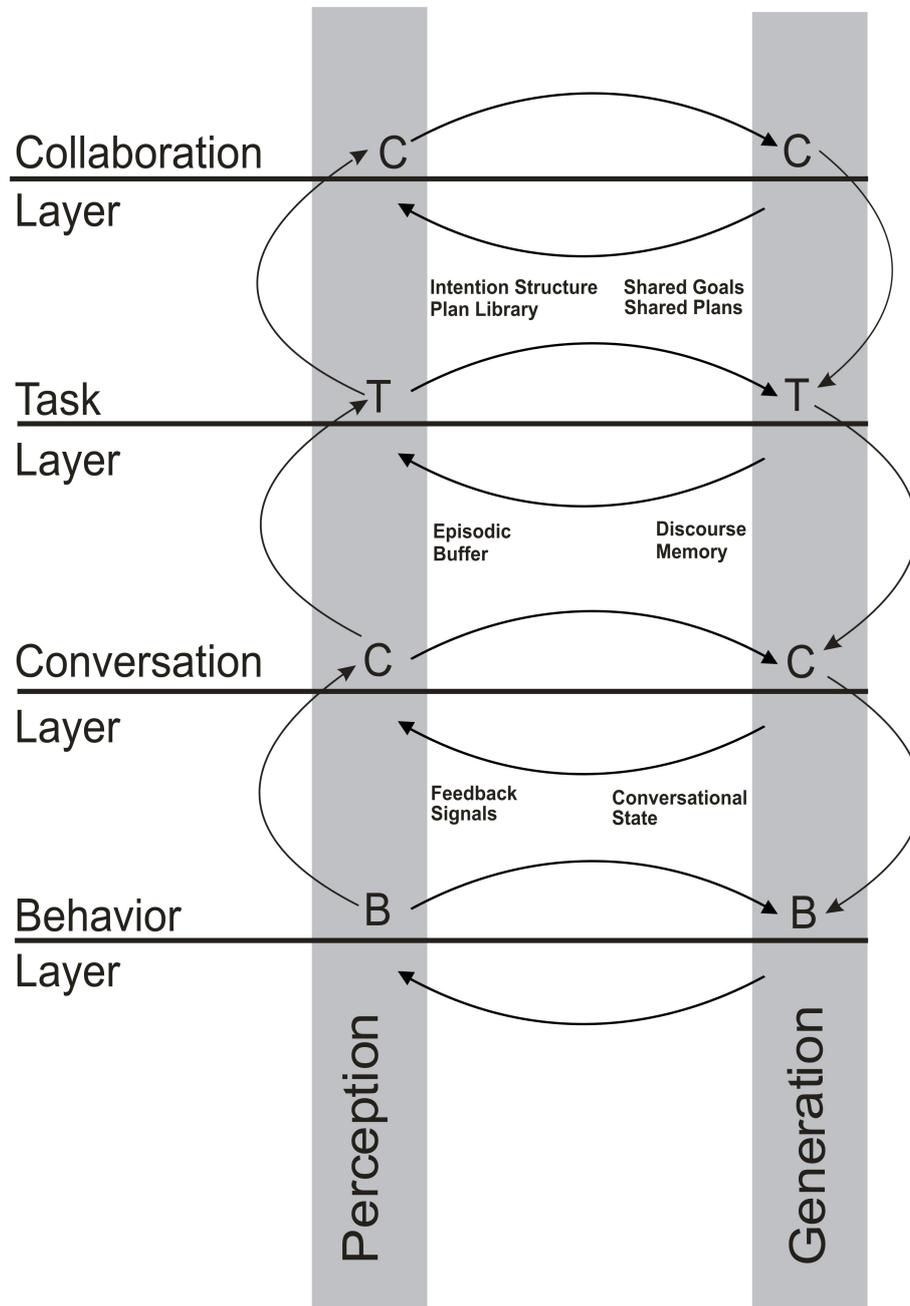


Abbildung 3.5: BCTC-Modell – Behavior (Verhaltensebene), Conversation (Konversationsebene), Task (Aufgabenebene), Collaboration (Kooperationsebene)

kung besitzt, die sich auf dieser höheren Ebene einordnen lässt.

### 3.7 Zusammenfassung - Architekturanforderungen

Das aufgestellte Kooperationsmodell BCTC basiert auf der Analyse menschlichen Kooperationsverhaltens und menschlichen kognitiven Verhaltens. Ein gleichberechtigter künstlicher Kooperationspartner muss in der Lage sein, die in dem interaktiven Kooperationsmodell BCTC beschriebene Rolle mit allen spezifizierten Anforderungen auszufüllen. In Anbetracht der Komplexität der Anforderungen und der Notwendigkeit, dass nur die Gesamtheit aller Mechanismen zu einem adäquaten Verhalten des Agenten führt, erscheint ein *kognitiver Ansatz* unumgänglich, der sich in seiner Struktur und seinen Mechanismen am Menschen orientiert. Durch eine Modellierung in Anlehnung an menschliche kognitive Prozesse wird es dem Agenten zudem ermöglicht, sich in sein Gegenüber hineinzuversetzen. Wenn overt Verhalten sich auf die auslösenden, kognitiven Mechanismen und damit auf die dahinter liegenden mentalen Zustände zurückführen lässt, werden entscheidende Erkenntnisse über den Kooperationspartner offenbart, die ein besseres Verständnis im Rahmen der Kooperation zur Folge haben. Nicht zuletzt können wichtige Erkenntnisse über den menschlichen Kooperationsprozess aufgedeckt werden, wenn man die beteiligten kognitiven Faktoren durch ihren Nachbau in ihrem Zusammenspiel erforscht.

Die Umsetzung der in diesem Kapitel spezifizierten Regeln und Verhaltensweisen in einen technischen Agenten führt zu einer Reihe von **Architekturanforderungen**. Damit das Verhalten des Agenten natürlich wirkt, muss eine Reaktion auf der *Behavior*-Ebene sehr schnell (z. B. im Millisekundenbereich) erfolgen. Dazu ist es erforderlich, dass Eingaben und Wahrnehmungen des Agenten, bevor sie einer längeren und komplexeren Analyse unterzogen werden, schnell verarbeitet werden, um eine **schnelle Reaktionszeit** des Agenten zu ermöglichen. Es muss demnach eine **reaktive Schicht** vorgesehen werden, welche **parallel** zur deliberativen Schicht in kleineren Zeitzyklen agiert. Die verarbeitenden Module müssen **inkrementell** arbeiten. Für die schnelle und situative Analyse von Referenzen muss die Menge potentieller Kandidaten kontextspezifisch eingeschränkt werden. Dafür wird ein **dynamisches Arbeitsgedächtnis** benötigt sowie der Zugriff auf eine **episodische Diskurshistorie**.

Im Rahmen der Umsetzung des Interaktionsmanagements der *Conversation*-Ebene bestehen weitere Anforderungen an die Architektur. Für die Fähigkeit einer parallelen Verfolgung unterschiedlicher Konversationsstränge ist eine **explizite Intentionsmodellierung** erforderlich. Mechanismen für die **Generierung von Handlungsoptionen** sowie für dynamische Entscheidungsprozesse werden benötigt, um das Einhalten der Interaktionsregeln zu gewährleisten. Es sind zudem

Mechanismen gefordert, welche parallel zum zielgerichteten Verhalten des Agenten agieren und bei wichtigen Ereignissen oder Regelüberschreitungen einen Einfluss auf den Aufmerksamkeitsfokus des Agenten durch die **Instantiierung kontextspezifischer Ziele** ausüben. Neben den individuellen Intentionen des Agenten wird die Repräsentation von **Obligationen** benötigt, welche dafür sorgen, dass der Agent sowohl bestehende Regeln berücksichtigt als auch Intentionen seines Gegenübers. Dabei sollte der Agent jedoch auch in der Lage sein, gegen Obligationen zu verstoßen.

Im Rahmen der *Task*-Ebene benötigt der Agent eine Repräsentation prozeduralen Wissens mit spezifizierten Anwendungsbedingungen. Für die Fähigkeit des dynamisch situierten Problemlösens ist der Einsatz **partieller Pläne mit Einschränkungsoptionen** erforderlich. Um komplexe Aufgaben lösen zu können, muss der Agent über die Fähigkeit der **Verarbeitung hierarchischer Pläne** verfügen. Es muss die Möglichkeit bestehen, **Einschränkungen** zwischen Teilzielen spezifizieren zu können. Der Agent benötigt Mechanismen des **Means-Ends-Reasoning**. Anforderungen der Effizienz und des situativen Problemlösens erfordern eine Anpassung der Menge der möglichen Handlungsoptionen in Bezug auf die aktuelle Relevanz. Der Agent muss zudem in der Lage sein, eigene Handlungen und auch Handlungen seines Gegenübers **abschätzen** und **evaluieren**, sowie **Konflikte erkennen** zu können.

Für eine echte Kooperation auf der *Collaboration*-Ebene muss der Agent in der Lage sein, **gemeinsame Pläne** zu repräsentieren. Ferner muss er sich auf eine **Verzahnung von Planen und Handeln** einzustellen können. Dafür ist internes Wissen darüber erforderlich, ob ein Plan **vollständig** ist und ob die involvierten Variablen ausreichend spezifiziert sind. Die Intentions- und Zielmodellierung im Rahmen einer echten Kooperation unterliegt besonderen Anforderungen. Einerseits muss der Agent eigene Intentionen mit Bezug auf Handlungen seines Kooperationspartners repräsentieren können. Andererseits muss er im Laufe der Kooperation Annahmen über die aktuellen Intentionen des Kooperationspartners bilden. Für das **Aushandeln** eines gemeinsamen Plans ist Wissen über involvierte **Ressourcen** und resultierenden **Aufwand** erforderlich. Die Anforderungen lassen sich folgendermaßen zusammenfassen:

#### **Anforderungen:**

- (a1) Verarbeitungsprinzip: parallel, inkrementell, modular, unterbrechbar
- (a2) Explizite Repräsentation mentaler Zustände: Überzeugungen, Ziele, Intentionen, (gemeinsame) Pläne, Obligationen, *Commitments*
- (a3) Situietheit: dynamische, kontextsensitive Verarbeitung (Umwelt, Interaktionshistorie)

- (a4)** Wissen über Anwendungsbedingungen und Effekte von Handlungen (explizite Repräsentation prozeduralen Wissens)
- (a5)** Effiziente, kontextsensitive Generierung von Handlungsoptionen
- (a6)** Deliberationsprozesse sowie reaktive Verhaltensweisen
- (a7)** Berücksichtigung Verzahnung von Sprache und Handeln (*Interaction-Move*)



**Teil II**

**Kognitive Architektur**



# Kapitel 4

## Theoretische Grundlagen

Für die kognitive Modellierung der Kooperationsfähigkeiten eines künstlichen Agenten ist eine Architektur zu entwickeln, welche die Architekturanforderungen aus Kapitel 3.7 erfüllt. Dazu sind zum einen Mechanismen nötig, welche den Agenten in die Lage versetzen, seine Entscheidungen aufgrund von **relevanter, kontextabhängiger Information** zu treffen (Anforderung **a3**), und zum anderen Mechanismen für die **zielorientierten Entscheidungsprozesse** selbst (Anforderung **a5** und **a6**). Für das Bereitstellen kontextrelevanter Information sind *bottom-up* Prozesse erforderlich, welche sowohl auf internen Wissensquellen als auch auf der Wahrnehmung der Umwelt und insbesondere des Interaktionspartners operieren (Anforderung **a3**). Um die Modellierung der Prozesse auf ein solides Fundament zu stellen, werden im nächsten Abschnitt Prozesse der Kognition, insbesondere der **kognitiven Kontrolle** betrachtet (siehe Abschnitt 4.1). Da die *bottom-up*-Prozesse im Rahmen eines Arbeitsgedächtnisses die mentalen Zustände beeinflussen und das **Arbeitsgedächtnis** dabei die Grundlage für die deliberativen Prozesse darstellt, werden im Weiteren verschiedene Arbeitsgedächtnismodelle diskutiert.

Um den Forderungen einer expliziten Repräsentation von Zielen und Intentionen sowie eines expliziten Commitmentverhaltens in einem technischen System gerecht zu werden (Anforderung **a2**), wird auf die Modellierung **mentaler Zustände** zurückgegriffen (siehe Abschnitt 4.2). Nachdem durch die Analyse von psychologischen und philosophischen Modellen wichtige Mechanismen und Prinzipien isoliert betrachtet wurden, werden im Anschluss komplexe **Architekturen** diskutiert, in denen diese Mechanismen und Prinzipien technisch umgesetzt sind (siehe Abschnitt 4.3). Damit werden praktische Ansätze vorgestellt, welche eine Einordnung der selbst entwickelten Architektur **CASEC** in die bestehende Forschungslandschaft der kognitiven Architekturen erlauben.

## 4.1 Kognition

Der Begriff der Kognition, von lateinisch *cognoscere* bzw. griechisch *gignoskein* (erkennen, wahrnehmen, wissen) abgeleitet (Strube et al., 1996), bezieht sich auf Prozesse, die aufgrund der Leistungsfähigkeit des Gehirns ablaufen. In der Psychologie wird Kognition als allgemeiner Begriff für die mentalen Prozesse eines Individuums verwendet, die sich auf Formen des Erkennens und Wissens beziehen. Zu diesen zählen Aufmerksamkeit, Erinnerungen, Urteile, Vorstellungen, Antizipation, Planen, Entscheiden, Problemlösen und das Mitteilen von Ideen (Zimbardo, 1995). Kognition umfasst zudem Prozesse der mentalen Repräsentation wie beispielsweise das Klassifizieren und Interpretieren. Zwischen Kognitionen und Emotionen besteht ein Wechselspiel; so können Kognitionen Emotionen beeinflussen aber auch ebenso durch sie beeinflusst werden.

### 4.1.1 Kognitive Kontrolle

Die Kernprozesse, welche die Kognition steuern, lassen sich unter dem Begriff der *kognitiven Kontrolle* vereinen. Braver & Cohen (2000) betonen, dass ein besonders wichtiger Aspekt kognitiver Kontrolle die Fähigkeit sei, auf angemessene Weise Kontextinformationen von Zielen aufrecht zu erhalten und diese Information dazu zu nutzen, die kognitive Verarbeitung eines Agenten über längere Zeitspannen zu koordinieren.

Laut E. Miller & Cohen (2001) rührt kognitive Kontrolle von der aktiven Aufrechterhaltung von Aktivitätsmustern im Präfrontalen Cortex her, welche Ziele und die dazu gehörigen Wege, sie zu erreichen, repräsentieren. Diese Aktivitätsmuster sorgen dafür, dass das Verhalten durch interne Zustände und Intentionen gesteuert wird. Der *Präfrontale Cortex* (PFC) sendet dabei Signale durch das restliche Gehirn, und beeinflusst nicht nur visuelle Prozesse und sensorische Modalitäten, sondern auch Gedächtniszugriffe, emotionale Einschätzungen und die Verhaltensgenerierung des Menschen. In Abhängigkeit von dem Ziel der Beeinflussung können die Repräsentationen im *PFC* folglich als

- Aufmerksamkeitsschablone (*attentional template*),
- Abfragehinweis (*retrieval cue*),
- Regel (*rule*) oder
- Ziel (*goal*)

beschrieben werden, je nachdem ob sie sich auf sensorische Prozesse, interne Prozesse, spezielle Handlungsoptionen oder die intendierten Ergebnisse einer Handlung beziehen.

Die vom *PFC* ausgehende Kontrolle wird von der Kontrollausübung des *Hippokampus* komplementiert. Dieser ist dafür verantwortlich, Informationen in einer spezifischen Episode zu binden und konzentriert sich auf tatsächlich ausgeführte Aktionen. Der Hippokampus setzt eine *Gewichts-basierte* Kontrolle ein, welche dazu führt, dass permanente assoziative Verbindungen zwischen Elementen entstehen, welche dann das Langzeitgedächtnis definieren (O'Reilly et al., 1999). Im Gegensatz dazu besteht die Kontrolle des *PFC* aus einer *Aktivitäts-basierten* Steuerung und dient dazu, Regularitäten und Ziele sowie Aufgaben bezogene Regeln zu erkennen und zu aktivieren. Dies entspricht dem Ansatz der Produktionssysteme, bei welchen die exekutive Kontrolle auf der Aktivierung der Repräsentationen der aktuellen Ziele und relevanten Regeln beruht (siehe auch 4.3.1 und 4.3.1 zu Act-R und Soar).

Das Zusammenspiel von *PFC* und Hippokampus kann als Basis für das Verstehen der vorausschauenden Kontrolle bzw. des Planens dienen. Alleiniges Aufrechterhalten der Aktivierung eines Ziels erscheint für längere Zeiträume nicht den relevanten Mechanismus darzustellen. Vielmehr kann vorausschauende Kontrolle als *Ziel* im *PFC* kodiert werden, welches zu einem geeigneten Zeitpunkt aktiviert wird. E. Miller & Cohen (2001) schlagen dazu vor, dass der Hippokampus Assoziationen zwischen dem gewünschten Zielzustand und den Umständen, unter welchen das Ziel erweckt werden soll, kodiert. Die kognitive Kontrolle besitzt die kritische Eigenschaft, dass sie Zugriff auf diverse Informationen benötigt, welche sowohl den internen mentalen Zustand als auch die externe Welt betreffen.

#### 4.1.2 Kognitive Architektur und kognitive Modellierung

Die Prozesse der kognitiven Kontrolle spielen sowohl bei der Entwicklung *Kognitiver Architekturen* und als auch bei der *Kognitiven Modellierung* eine entscheidende Rolle. Eine Architektur arbeitet auf verschiedenen Ebenen. Die Kognitionsebene besteht dabei aus Fähigkeiten und Mechanismen, die komplett losgelöst und unabhängig von der Anwendung und dem direkten Verhalten des Agenten sind; sie stellt die Voraussetzung für die Interaktion dar.

Der Begriff der kognitiven Architektur wurde von Newell (1990) geprägt. Mit dem Begriff wird eine Infrastruktur beschrieben, in der genau das enthalten ist, was bei allen erkennenden, intelligenten Systemen gleich und fest ist. Damit sind im Wesentlichen die Wissensrepräsentation des Systems, die Eigenschaften seines Speichers, und der Prozess, der auf dem Speicher arbeitet, gemeint; dagegen ist es unabhängig von der zugrunde liegenden Wissensbasis.

Eine kognitive Architektur beschreibt, wie künstliche Berechnungsprozesse als kognitive Systeme agieren können. Der Begriff Architektur betont dabei, dass nicht nur versucht wird, ein bestimmtes Verhalten zu erzeugen, sondern dass auch die zugrunde liegenden Strukturen bei der Modellierung berücksichtigt werden.

Die Spezifikation einer kognitiven Architektur besteht aus ihren repräsentationalen Annahmen, der Charakteristika und Struktur ihrer Gedächtnisse, und den Prozessen und Mechanismen, die auf diesen Gedächtnissen operieren.

Forschung in Bezug auf kognitive Architekturen findet zumeist in einem engen Zusammenhang mit der *kognitiver Modellierung* statt, wobei beide Disziplinen versuchen die Bandbreite menschlichen Verhaltens zu erklären und zumindest in Ansätzen die breiten Fähigkeiten der menschlichen Intelligenz zu modellieren (Byrne, 2003). Dabei bezieht sich kognitives Modellieren auf das Untersuchen und Nachbilden einzelner Module oder Phänomene, wohingegen die kognitiven Architekturen einen ganzheitlichen Ansatz verfolgen und integrativ ausgelegt sind. Choi & Langley (2006) zählen zu den Vertretern des ganzheitlichen Architekturansatz und fordern dazu auf, nicht mehr isolierte Komponenten zu betrachten sondern sich vielmehr auf intelligente, integrierte Systeme zu konzentrieren, da diese für Haushaltsroboter, Tutoren auf menschlicher Ebene, interaktives Entertainment sowie für glaubwürdige Charaktere eine wichtige Voraussetzung darstellen.

### 4.1.3 Arbeitsgedächtnis

Es lässt sich eine Vielzahl von Definitionen des Arbeitsgedächtnisses in der Literatur finden. Die vorliegende Arbeit schließt sich der Definition von Miyake & Shah (1999) an:

*„Working memory is those mechanisms or processes that are involved in the control, regulation, and active maintenance of task-relevant information in the service of complex cognition, including novel as well as familiar skilled tasks. It consists of a set of processes and mechanisms and is not a fixed „place“ or „box“ in the cognitive architecture“.*

Unter komplexer Kognition wird dabei die Bearbeitung einer Aufgabe verstanden, die in ihrer Komplexität mindestens eine der folgenden Anforderungen umfasst: eine kontrollierte Verarbeitung, mehrere Verarbeitungsschritten, Beteiligung multipler Komponenten des Gedächtnissystems, schneller Zugriff auf große Mengen von Informationen.

#### **Begriffsklärung**

In diesem Abschnitt werden Begriffe, die Rahmen der Diskussion des Arbeitsgedächtnisses auftreten, vorab erklärt.

*Chunking* ist ein Prozess des Rekodierens ursprünglich separater Informationseinheiten, bei dem auf der Grundlage allgemeiner Ordnungsprinzipien oder durch das Aktivieren von Vorwissen kompakte Einheiten höherer Ordnung gebildet werden (Strube et al., 1996). Die Speicherkapazität wird durch die Integration

einer Anzahl von disparaten Merkmalen zu einer Einheit erhöht (G. A. Miller, 1956). *Chunking* kann auf verschiedenen Ebenen operieren.

Der Begriff des *Primings* (auch Bahnung) bezieht sich auf Formen der Voraktivierung und wird oft inflationär und ohne klare Definition verwendet. Er bezeichnet dann alle möglichen Wirkungen einer Vorausinformation auf die Verarbeitung eines Reizes (Strube et al., 1996). Priming lässt sich als der Prozess definieren, bei dem durch Präsentation eines (*Priming*)-Stimulus ein oder mehrere bestehende Gedächtnisinhalte aktiviert werden. Diese Aktivierung wird nicht zwangsläufig bewusst erlebt, beeinflusst aber sowohl darauf folgende Wahrnehmungsprozesse als auch mentale Prozesse. Priming bildet quasi eine Verbindung zwischen implizitem und explizitem Gedächtnis und ermöglicht aktivierte Gedächtnisinhalte leichter abzurufen (P. Gray, 1999).

Im Bereich des Primings lassen sich *Perceptual Priming*, *Conceptual Priming* und *Associative Priming* (eine Form des indirekten Primings) unterscheiden. Beim *Perceptual Priming* geht es um die Beeinflussung der Fähigkeit, einen Teststimulus zu identifizieren. Auslöser des *Perceptual Priming* können beispielsweise visuelle, taktile oder auch akustische Signale sein. *Conceptual Priming* ist nicht an sensorische Modalitäten gebunden. Es beinhaltet eine Form des semantischen Lernens durch das Aktivieren relevanter Konzepte und Gedächtnisinhalte im Semantischen Gedächtnis (Tulving & Schacter, 1990). Es beeinflusst den Gedankenfluss und hilft, die Gedanken auf einen Themenbereich zu fokussieren. Beim *Associative Priming* wird von einem netzartig strukturierten Semantischen Gedächtnis ausgegangen. Wird ein Konzept aktiviert, breitet sich die Aktivierung entlang der Kanten auf die Nachbarknoten aus. In Bezug auf die Stärke der sich übertragenden Aktivierungsimpulse bestehen verschiedene Theorien.

Bei der Theorie der *Spreading Activation* (Aktivierungsausbreitung) wird davon ausgegangen, dass die Menge der Aktivierung, welche zwischen dem Prime-Stimulus und dem Ziel fließt, aus einer Funktion der Anzahl und der Stärke der Verbindungen zwischen dem Prime-Stimulus und dem Ziel besteht. McKoon & Ratcliff (1992) argumentieren hingegen, dass die Menge der Aktivierung von dem Bekanntheitsgrad zwischen Prime-Stimulus und Ziel als assoziative Einheit besteht.

### **Modellauswahl**

Für die vorliegende Arbeit wurden für die theoretischen Grundlagen vier Modelle aus den Bereichen der Psychologie und Neuro-Psychologie ausgewählt, um eine adäquate Bandbreite des Stands der Forschung zu gewährleisten. Bei der Diskussion konkreter Architekturen wird zudem auf zwei weitere Arbeitsgedächtnismodelle eingegangen (siehe Act-R und Soar). Aus Platzgründen und da diese aufgrund anderer Schwerpunktsetzungen eine geringere Relevanz für die vorlie-

gende Arbeit besitzen, werden Modelle wie die von EPIC (Kieras et al., 1999) und das Modell von O'Reilly et al. (1999) hier nicht ausführlich diskutiert.

Das **Multiple Component Model** von Baddeley und Kollegen (Baddeley & Logie, 1999; Baddeley, 2007) zählt zu den einflussreichsten und bekanntesten Arbeitsgedächtnismodellen der Kognitionspsychologie. Für die vorliegende Arbeit erscheinen die besondere Form der Strukturierung des Arbeitsgedächtnisses sowie das Konzept des episodischen Buffers von besonderer Relevanz.

Die Arbeiten von Cowan und Kollegen am **Embedded Processes Model** verlaufen komplementär zu dem *Multiple-Component Model*. Während im *Multiple-Component Model* ein Schwerpunkt darauf liegt, unterschiedliche Komponenten des Arbeitsgedächtnisses zu identifizieren und zu charakterisieren, betrachtet Cowan (1999) das Arbeitsgedächtnis unter funktionalen Gesichtspunkten. Seine Betrachtung der verschiedenen Zugriffsmöglichkeiten sowie funktionalen Mechanismen spielen für die Entwicklung und Konzeption der CASEC-Architektur eine entscheidende Rolle.

Das **Concentric Model of Working Memory** von Oberauer baut auf den Arbeiten des *Embedded Processes Model* auf, setzt das Modell jedoch zusätzlich in Beziehung zu Aspekten des *Chunkings* (Oberauer, 2002).

Engle und Kollegen lassen sich von den Arbeiten von Baddeley & Hitch (1974) in Bezug auf die Unterscheidung zwischen allgemeinen Aufmerksamkeitsprozessen und Domänen-spezifischen Speicherungsprozessen inspirieren. Ihre Arbeiten unterscheiden sich jedoch durch die stärkere Betonung von Funktionen und Prozessen. Kane et al. (2007) nehmen wie Cowan eher eine prozessorientierte Sichtweise ein. Das **Controlled-Attention Framework** von Engle und Kollegen bietet für die Modellierung des Aufmerksamkeitsfokus im Rahmen eines Arbeitsgedächtnisses wichtige Erkenntnisse. Des Weiteren bietet die Gedächtniskonzeption des *Controlled Attention Frameworks* interessante Erkenntnisse in Bezug auf die Verbindung des Arbeitsgedächtnisses mit emotionalen Prozessen (Engle et al., 1999).

### **Betrachtungsschwerpunkte**

In dieser Arbeit werden die Modelle im Hinblick auf die folgenden Schwerpunkte betrachtet: Zunächst wird die generelle *Herangehensweise*, der Ursprung sowie der Kontext, in welchem die Modelle entwickelt wurden, beleuchtet. Darüber hinaus wird diskutiert, welche *Aufgabe* die einzelnen Forscher und Forschergruppen dem Arbeitsgedächtnis zuschreiben. Danach wird konkret auf die *Struktur und Repräsentation* des Arbeitsgedächtnisses eingegangen. Im Rahmen der Beschaffenheit wird insbesondere die Beziehung zum Langzeitgedächtnis untersucht, da Arbeits- und Langzeitgedächtnis oft als konzeptuelle Einheit gesehen werden. Im Weiteren werden die *Basismechanismen* betrachtet, welche die *bottom-up-*

Prozesse des Arbeitsgedächtnis umfassen und für die Enkodierung und Bereitstellung von Informationen verantwortlich sind. Schließlich werden die ergänzenden *top-down*-Mechanismen der Kontrolle und Regulierung diskutiert und in Beziehung zu Prozessen der komplexen Kognition gesetzt.

### Herangehensweise der Modelle

Baddeley und Kollegen gehen in ihrem **Multiple-Component Model** von einer neuro-psychologischen Perspektive aus und versuchen, kognitive Psychologie praktisch anzuwenden. Sie setzen dabei zumeist Experimente ein, welche dem Paradigma der *dual-task* Interferenz folgen (Baddeley & Hitch, 1974; Della et al., 1995). Cowans Forschung im Rahmen des **Embedded-Processes Model** konzentriert sich auf Gedächtnis- und Aufmerksamkeitsprozesse und die Rolle, welche sie in der menschlichen Kognition spielen. Dabei wird in seinem Labor insbesondere die kindliche Entwicklung des Kurzzeit- und des Arbeitsgedächtnisses untersucht (Cowan et al., 2006). Cowan und Kollegen untersuchen schwerpunktmäßig die Beziehung zwischen Arbeitsgedächtnis und selektiver Aufmerksamkeit im Rahmen der Informationsverarbeitung (Cowan, 1995). Bei dem **Concentric Model Of Working Memory** liegen Oberauers Interessen auf der Erforschung der Arbeitsgedächtniskapazität durch faktoranalytische Studien in Bezug auf individuelle Unterschiede (Oberauer et al., 2000), kognitive Experimente und altersvergleichende Studien (Oberauer, 2001). Das **Controlled-Attention Framework** von Engle und Kollegen baut auf dem Ansatz auf, durch die Untersuchung der individuellen Unterschiede beim Menschen in Bezug auf die Kapazität des Arbeitsgedächtnisses ein theoretisches Modell aufstellen und verifizieren zu können (Engle et al. (1999).

### Aufgabe

In Bezug auf die Aufgabe des Arbeitsgedächtnisses besteht ein relativ breiter Konsens, auch wenn die einzelnen Forschergruppen unterschiedliche Aufgabenschwerpunkte setzen.

Die Aufgabe des **Multi-Component Model** wird von Baddeley und Kollegen in der Regulierung des Informationsflusses, der Vorbereitung und dem Einsatz verschiedener Verarbeitungsstrategien sowie der Kontrolle und Koordination untergeordneter Systeme gesehen. Diese Systeme dienen dazu, aktive Gedächtnisspuren der Wahrnehmungen sowie Elemente, welche durch Probehandlungen entstehen können, vorzuhalten. Die Gedächtnisspuren erlauben es dem Menschen, seine direkte Umwelt mental zu repräsentieren, Informationen bezüglich der unmittelbar vorangegangenen Erfahrungen bereitzustellen, die Akquisition von neuem Wissen vorzunehmen, Probleme zu lösen, Ziele zu formulieren sowie diese

zueinander in Beziehung zu setzen und entsprechend zu handeln (Baddeley & Logie, 1999).

Cowan hingegen definiert das **Embedded-Processes Model** als Einheit der mnemonischen Funktionen, welche Informationen temporär vorhalten, um kognitive Aufgaben lösen zu können. Er betont dabei insbesondere die Rolle der aufmerksamskeitsgesteuerten Kontrolle des Arbeitsgedächtnismodells (Cowan, 1995).

Oberauer sieht das **Concentric Model of Working Memory** als ein System, welches für das simultane Speichern und Verarbeiten von Informationen verantwortlich ist und den selektiven Zugriff auf Inhalte des Langzeitgedächtnisses im Dienste zielgerichteter Verarbeitung organisiert (Oberauer, 2003).

Kane et al. (2007) betonen im Rahmen ihres **Controlled-Attention Models** die Synergie der Aufmerksamkeits- und Gedächtnisprozesse des Arbeitsgedächtnisses durch Aufrechterhaltung, Bereitstellung und Blockierung von Informationen. Einerseits werden damit Langzeitgedächtnisrepräsentationen von Stimuli, Zielen und Handlungsplänen, welche über einem bestimmten Schwellwert liegen, dem Arbeitsgedächtnis zugeordnet. Andererseits zählen auch prozedurale Fertigkeiten des *Rehearsal*, der Stimuluskodierung sowie leitende Aufmerksamkeitsprozesse zum Arbeitsgedächtnis.

### Struktur und Repräsentation

Baddeley und Kollegen gehen in ihren Arbeiten von einem Mehrkomponentenmodell des Arbeitsgedächtnisses aus und postulieren, dass sich das Arbeitsgedächtnis aus mehreren verschiedenen Komponenten der Kognition zusammensetzt (Abbildung 4.1). Zu den wichtigsten Konzepten zählen dabei der *Phonological Loop* (phonologische Schleife), das *Visuospatial Sketchpad* (räumlich visueller Notizblock) und seit neuerer Zeit auch der *Episodic Buffer* (episodischer Puffer) (Baddeley, 2000; Baddeley et al., 2010). Dabei steht das Arbeitsgedächtnis zwar in enger Verbindung zur Perzeption und zum Langzeitgedächtnis (*Long-term memory* (LTM)), wird aber dennoch als eigenes System mit eigenem Speicher beschrieben.

Der *Phonological Loop* übernimmt die Aufgabe eines verbalen Arbeitsgedächtnisses und ist in seiner Struktur mit dem ursprünglichen Kurzzeitspeicher-Konzept verwandt. Sprachliches Material wird hier verarbeitet. Dabei kommen ein phonologischer Speicher sowie ein artikulatorischer Kontrollprozess zum Einsatz (Baddeley, 1997). Das *Visuospatial Sketchpad* übernimmt die Aufgabe der vorübergehenden Speicherung von visuellen sowie räumlichen Informationen. Zusätzlich wird die Komponente mit der Repräsentation und Planung von Bewegungen in Verbindung gebracht (Baddeley & Logie, 1999). Wie auch das verbale Äquivalent, ist das visuelle Arbeitsgedächtnis kapazitätsbeschränkt und zwar auf

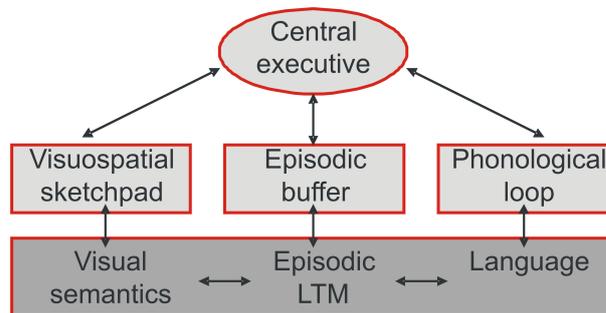


Abbildung 4.1: Mehrkomponentenmodell nach Baddeley (2000)

drei bis vier Objekte (Baddeley, 2003). Da die visuelle Welt jedoch in der Regel über eine gewisse Zeit bestehen bleibt und damit als kontinuierlicher Gedächtnisbeitrag zur Verfügung steht, erscheinen die visuellen Eindrücke oftmals redundant in Wahrnehmungen und Speicher. Insgesamt stellt die Untersuchung des *Visuospatial Sketchpads* eine offene, aktive Forschungsrichtung dar, deren Ergebnisse bisher erst rudimentär in die komplexen Mehrkomponentenmodelle eingebunden sind (Baddeley, 2003).

Die *Central Executive* dient dazu, die einzelnen Subkomponenten zu verwalten und zu steuern, und wurde selbst zunächst als ausschließliches Aufmerksamkeitssystem ohne eigenen Speicher angesehen. Um die Lücke des Arbeitsspeichers für die *Central Executive* zu schließen, wird seit einigen Jahren der *Episodic Buffer* von Baddeley (2000) postuliert. Während der initiale Ansatz der Modellierung des Arbeitsgedächtnisses nach Baddeley einen Schwerpunkt auf die Aufteilung und Unterteilung des Gedächtnisses legt, stellt der *Episodic Buffer* eine Integrationskomponente dar (Baddeley & Logie, 1999). Als diese lenkt er die Aufmerksamkeit auf die Konzepte der Integration und des *Chunkings*. Dem *Episodic Buffer* wird zugeschrieben, an dem komplexen Prozess des *Chunkings* beteiligt zu sein.

Der *Episodic Buffer* selbst wird als Speichersystem mit beschränkter Kapazität angesehen, in welchem temporär Information in multi-dimensionaler Repräsentation vorgehalten und manipuliert wird (Baddeley, 2000, 2001). Er dient damit als eine Art Modellierungsraum. Der Begriff *episodisch* reflektiert seine Fähigkeit, Episoden zu integrieren, welche sich sowohl durch die Zeit als auch durch den Raum erstrecken. Dabei bindet er die Informationen der einzelnen Komponenten zu neuen Einheiten. Der Begriff des *Buffers* verdeutlicht, dass multi-dimensionaler Code verschiedener Module des Arbeitsgedächtnisses integriert, mit dem Langzeitgedächtnis verknüpft und für eine gewisse Zeit vorgehalten wird. Die Kapazitätsbeschränkung lässt sich auf die rechenintensive Multi-

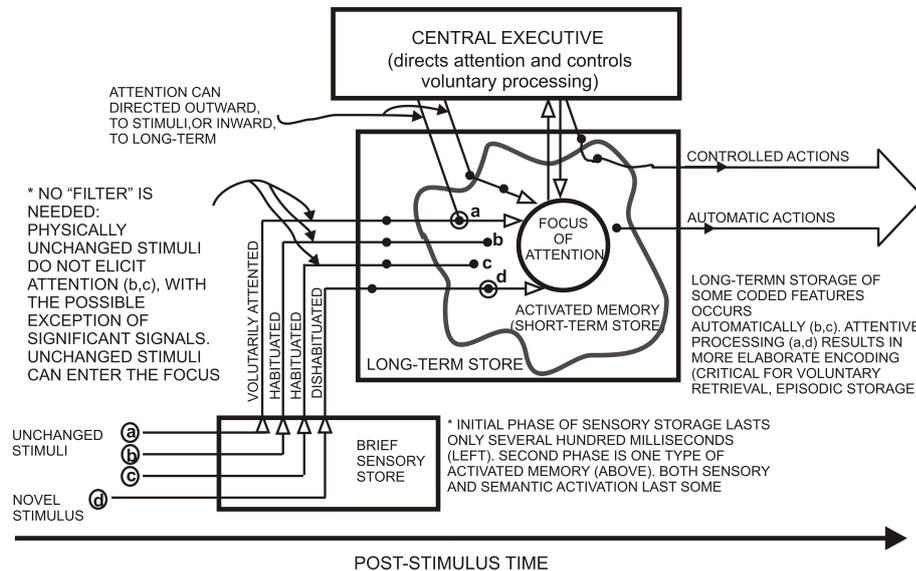


Abbildung 4.2: Embedded Processes Model nach Cowan (1999)

Dimensionalität des Buffers zurückführen (Hummel, 1999). Das Modell des *Episodic Buffers* unterscheidet sich von Tulvings Konzept des episodischen Gedächtnisses (1972) dadurch, dass dem Episodic Buffer eine temporäre Speicherfunktion zugeschrieben wird, wenngleich diese auch eng mit dem episodischen Langzeitgedächtnis verbunden ist. Ferner unterscheidet sich der *Episodic Buffer* von dem Modell des *Long-term Working Memory* von Ericsson & Kintsch (1995) insofern, als ein separates temporäres Speichersystem zusätzlich zu dem aktiviertem Langzeitgedächtnis postuliert wird. Der Buffer wird durch die *Central Executive* kontrolliert, wobei bewusste Aufmerksamkeit als Hauptabfragemechanismus dient (Baddeley, 1993; Baddeley & Andrade, 1998). Baddeley betont, dass der *Episodic Buffer* nicht nur als ein System dient, welches sich dem klassischen *Binding Problem* zuwendet und die Umwelt repräsentiert sowie zugreifbar für die bewusste Aufmerksamkeit macht, sondern dass durch den *Episodic Buffer* auch Erfahrungen nutzbar gemacht werden, um die Zukunft zu modellieren. Durch einfaches Aktivieren von Langzeitgedächtnisinhalten sei dieses Ziel nicht zu erreichen (Baddeley, 2002). Daher gehen Baddeley und Kollegen davon aus, dass das Arbeitsgedächtnis und das Langzeitgedächtnis zwei verschiedene, funktional trennbare kognitive Systeme darstellen.

Das **Embedded-Processes Model** (siehe Abbildung 4.2) von Cowan und Kollegen wird, im Gegensatz zu dem *Multiple-Component Model*, als ein in das Langzeitgedächtnis eingebettetes Arbeitsgedächtnismodell konzeptualisiert. Es besteht aus einem komplexen hierarchischen Konstrukt, welches den aktivierten, zugäng-

lichen Teil des Langzeitgedächtnisses umfasst. Dabei werden drei Bereiche des Arbeitsgedächtnisses anhand ihrer Zugriffsmöglichkeiten unterschieden (Cowan, 1999):

- Elemente im Aufmerksamkeitsfokus
- Elemente, welche sich nicht im Fokus befinden, aber temporär über eine hohe Aktivierung verfügen
- inaktive Elemente des Langzeitgedächtnisses, welche durch *Retrieval Cues* zugreifbar sind

Oberauer übernimmt in seinem **Concentric Model of Working Memory** die Unterscheidung hinsichtlich der Zugriffsmöglichkeiten von dem *Embedded-Processes Model*. Er konzeptualisiert dabei das Arbeitsgedächtnis als eine konzentrische Struktur, welche aus dem aktivierten Teil des Langzeitgedächtnisses, einer kapazitätsbeschränkten Region des direkten Zugriffs sowie dem Aufmerksamkeitsfokus selbst besteht (Abbildung 4.3).

- (1) Der aktivierte Teil des Langzeitgedächtnisses (*activated part of LTM*) hält Informationen über längere Zeiträume vor.
- (2) Die Region des direkten Zugriffs (*region of direkt access*) dient dazu eine limitierte Anzahl von Chunks für die aktuellen kognitiven Prozesse vorzuhalten.
- (3) Der Aufmerksamkeitsfokus (*focus of attention*) selbst hält zu jeder Zeit immer genau den Chunk vor, der für die nächste kognitive Operation ausgewählt wurde.

Oberauer sieht das Arbeitsgedächtnis damit als eine organisierte Menge von Repräsentationen, welche sich hinsichtlich ihrer Zugriffsmöglichkeiten und ihrer Zugriffsgeschwindigkeit für kognitive Prozesse klassifizieren lassen (Oberauer, 2003). Die Unterscheidung in drei Bereiche wird laut Oberauer zudem vorgenommen, um den Phänomenen der Interferenz und der Zugriffsgeschwindigkeit gerecht zu werden, welche sich allein durch das aktivierte Langzeitgedächtnis nur schwer erklären lassen. Oberauer betont dabei jedoch, dass er die drei Regionen nicht als strukturell oder gar anatomisch verschiedene Untersysteme ansieht, sodass Information von einer Region in eine andere transferiert werden müsste. Die drei Regionen stellen vielmehr funktional verschiedene Zustände der Repräsentation im Arbeitsgedächtnis dar. Auch wenn das konzentrische Modell sich nicht explizit auf eine strukturelle Unterscheidung im Sinne eines Mehrkomponentenmodells festlegt, so ist es laut Oberauer aber auch nicht inkompatibel mit Arbeitsgedächtnismodellen, welche eine solche Struktur postulieren. So sei das Modell

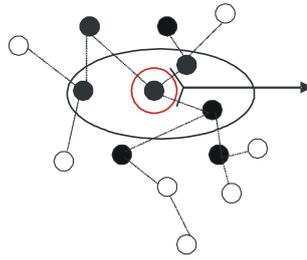


Abbildung 4.3: Arbeitsgedächtnis nach Oberauer (2002)

beispielsweise prinzipiell kompatibel mit dem Modell von Baddeley (Oberauer, 2002).

Setzt man das *Multiple-Component Model* in Beziehung zu dem *Embeddded-Processes Model* und dem *Concentric Model of Working Memory*, so korrespondieren Teile des aktivierten Gedächtnisses mit dem *Phonological Sketchpad* bzw. mit dem Speicher des *Visuospatial Sketchpad*.

Engle und Kollegen sehen in ihrem Arbeitsgedächtnismodell des **Controlled-Attention Frameworks** (siehe Abbildung 4.4) ähnlich wie Cowan und Oberauer ein System, bestehend aus

- (a) einem Speicher in Form von aktivierten Langzeitgedächtnisinhalten, welche über einem bestimmten Schwellwert liegen,
- (b) Prozessen, um die Aktivierung hervorzurufen und aufrecht zu erhalten,
- (c) einer beschränkten Kapazität der kontrollierten Aufmerksamkeit.

Unterschiede im Vergleich zu den Arbeiten Cowans und Oberauers bestehen in Bezug auf individuell unterschiedliche Arbeitsgedächtniskapazitäten. Während Cowan und Oberauer die unterschiedlichen Kapazitäten dem Aufmerksamkeitsfokus zusprechen, gehen Engle und Kollegen davon aus, dass eine bessere Verarbeitung außerhalb des Aufmerksamkeitsfokus für eine erhöhte Leistung sorgt. Sie sehen die individuellen Unterschiede der Kapazitäten, in der Fähigkeit, außerhalb des bewussten Aufmerksamkeitsfokus größere Mengen aufgabenrelevanter Informationen vorzuhalten, auf welche bei Bedarf schnell zugegriffen werden kann (Kane et al., 2007).

### Basismechanismen - *Bottom-up* Verarbeitung

Im **Multiple-Component Model** verlagert Baddeley einige Basisprozesse in die Untermodule des Arbeitsgedächtnisses. Dabei wird dem Phonological Loop

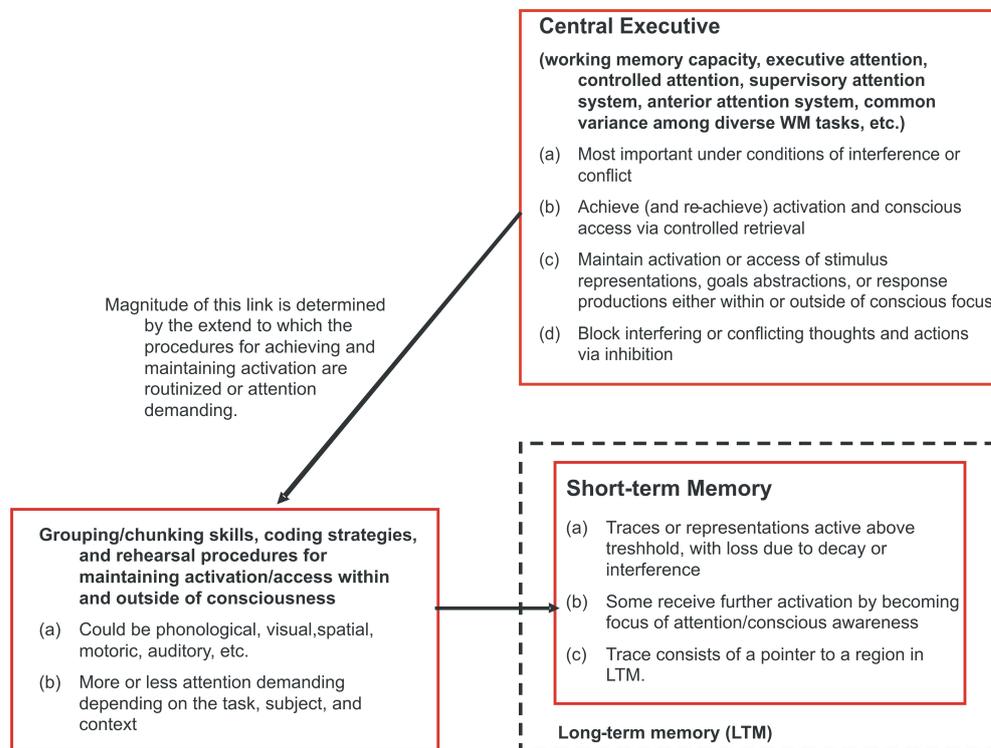


Abbildung 4.4: Controlled Attention Framework nach Kane et al. (2007)

zusätzlich zu seinem Speicher ein artikulatorischer Kontrollprozess (*Rehearsal*) zugeordnet (Baddeley, 1997). Dieser Prozess wird als eine Form der Subvokalisation des Sichselbstvorsprechens beschrieben. Die Zeitspanne der Vorhaltung ohne den *Rehearsal* Prozess umfasst nur wenige Sekunden (Baddeley, 2001). Innerhalb des *Visuospatial Sketchpads* bestehen laut Logie (1995) ebenfalls *Rehearsal* Prozesse. Er argumentiert, dass das Sketchpad kein wahrnehmungsbasierter Speicher ist, sondern erst entsteht, nachdem die visuelle Information durch das Langzeitgedächtnis verarbeitet wurde (Baddeley, 2003).

Auch in dem **Embedded-Processes Model** wird zwischen verschiedenen Abrufmechanismen unterschieden. Laut Cowan unterscheidet sich der Abruf (*Retrieval*) von Informationen aus dem aktiven Gedächtnis von dem Abruf von Informationen aus dem Langzeitgedächtnis auf signifikante Weise. Der Abruf aus dem Langzeitgedächtnis erfolgt aus einer umfassenderen Informationsstruktur und ist zeitlich dadurch begrenzt, dass die Abfrage nicht länger dauern sollte, als es die aktuelle Situation bzw. Interaktion erlaubt. Der Abruf aus dem aktivierten Gedächtnis unterliegt zusätzlich der Einschränkung, dass der Abruf aufgrund der zeitlich begrenzten Aktivierung der Elemente sehr schnell erfolgen muss, da die

Elemente nach etwa 10-20 Sekunden eventuell nicht mehr im aktivierten Teil des Arbeitsgedächtnisses vorliegen. Die Aufrechterhaltung (*Maintenance*) der Aktivität relevanter Elemente kann sowohl durch aktive, willentliche Prozesse der *Central Executive* erfolgen als auch durch Operationen der *Central Executive* als Begleiterscheinung auftreten. So sagt Cowan, dass der Prozess der Suche durch eine Menge von Elementen diese reaktivieren kann; durch das Kreisen des Aufmerksamkeitsfokus über den Elementen können diese im Aufmerksamkeitsfokus vorgehalten werden (Cowan, 1999).

Das **Concentric Model of Working Memory** von Oberauer spezifiziert die Beziehung zwischen dem Speichern und dem Verarbeiten innerhalb des Arbeitsgedächtnisses. Die beiden Prozesse können unabhängig voneinander so lange ablaufen, wie kein direkter Zugriff auf Gedächtnisinhalte erforderlich ist. Durch das Problem, mehrere verschiedene Elemente für den direkten Zugriff bereit zu halten, entsteht eine Kapazitätseinschränkung des Arbeitsgedächtnisses. Oberauer (2002) vermutet, dass die aktivierten Informationen des Langzeitgedächtnisses die Prozesse des Aufmerksamkeitsfokus beeinflussen können. Die Aktivierung von Elementen des Langzeitgedächtnisses ist zwar nicht kapazitätsbeschränkt, jedoch können dort die Aktivierungen durch Inferenzen und Abklingen der Aktivierungswerte verloren gehen.

Die Bereitstellung von Informationen geschieht auf drei hierarchisch aufgebauten Ebenen, welche mit drei Zuständen der Repräsentation korrespondieren, durch die die Elemente sukzessive in den Aufmerksamkeitsfokus überführt werden können. Auf erster Ebene werden Repräsentationen durch perzeptuellen Input oder assoziative Aktivierung des Langzeitgedächtnisses aktiviert. Auf der zweiten Ebene wird eine kleine Menge an *Chunks* (repräsentationale Elemente) in den Bereich des direkten Zugriffs gebracht. Hier können die Elemente temporär an Stellen eines mentalen Koordinatensystems gebunden werden (z.B. *spatial frame, time dimension, causal schema*) und in den laufenden kognitiven Prozessen genutzt werden. In diesem Bereich postuliert Oberauer auch das Phänomen des *Chunkings*. *Chunking* bedeutet, dass Verbindungen zwischen Entitäten im Langzeitgedächtnis aufgebaut oder bereits bestehende genutzt werden. Damit formen die Elemente zusammen einen *Chunk*, welcher im Langzeitgedächtnis repräsentiert wird. Nur ein Pointer, welcher mit dem *Chunk* assoziiert ist, wird in der Region des direkten Zugriffs bereitgehalten. Das Konzept des *Chunkings* bereichert das Konzept des aktivierten Langzeitgedächtnisses von Cowan, indem betont wird, dass die Elemente nicht nur aktiviert sondern auch miteinander verbunden sind (Oberauer, 2002). Auf der dritten und letzten Ebene wählt der Aufmerksamkeitsfokus ein Element für die nächste kognitive Operation aus und manipuliert dieses Objekt.

Nach Engle und Kollegen resultiert im **Controlled-Attention Framework** das Abrufen von Informationen aus automatischen Mechanismen der verteilten

Aktivierung (*Spreading Activation*), es kann aber durch kontrollierte aufwändige Suche beeinflusst werden. Neben dem *Neuheitswert*, mit welchem ein Element aktiviert wurde, und seiner Relevanz für das aktuelle Ziel spielt auch eine *emotionale Markierung* eine Rolle für die Zugreifbarkeit des Elements (Engle et al., 1999). Einige Elemente können durch die Auszeichnung mit einem *Salience*-Wert (Valenz-Markierung) unabhängig von ihrem Bezug zum aktuellen Ziel in den aktivierten Bereich rücken. Während die Prozesse des Abrufens und der Bereitstellung der Informationen ihrer Natur nach automatisch sind, ist eine kontrollierte Verarbeitung notwendig, um ihre Ergebnisse angemessen zu gestalten.

### **Kontrolle und Regulierung - Top-down Verarbeitung**

In allen vier Arbeitsgedächtnismodellen übernimmt eine *Central Executive* Kontroll- und Regulierungsprozesse im Dienste der Kognition. Mit dem Begriff der *Central Executive* werden Mechanismen zusammengefasst, von denen angenommen wird, dass sie der Koordination, Geordnetheit, Flexibilität und Adaptivität menschlicher Denk- und Handlungsabläufe zugrunde liegen (Kluwe, 2000). Komplexe kognitive Prozesse bestehen dabei beispielsweise aus Sprachverstehen, Problemlösen sowie Schlussfolgern.

Im **Multiple-Component Model** ist die *Central Executive* für die folgenden Kontroll- und Regulierungsfunktionen des Arbeitsgedächtnisses verantwortlich (Baddeley & Logie, 1999):

- Koordination der Bearbeitung verschiedener Aufgaben
- Anstoßen unterschiedlicher Abfrage- und Enkodierungsstrategien
- Selektives Richten der Aufmerksamkeit auf einen bestimmten Stimulus und dabei simultanes Unterdrücken anderer Stimuli
- Koordination der untergeordneten Systeme (siehe Abbildung 4.1)
- Aktivierung relevanter Informationen aus dem Langzeitgedächtnis
- Manipulation von Informationen, welche den verschiedenen temporären Speichern entspringen

Die genaue Beschaffenheit der Prozesse, welche der *Central Executive* zugeschrieben werden, sind empirisch abgeleitet. Wie sich diese Prozesse jedoch konkret organisieren lassen, bleibt eine offene Frage, die es noch weiter empirisch zu untersuchen gilt.

Im **Embedded-Processes Model** wird der Aufmerksamkeitsfokus sowohl durch willentliche, anstrengungsbehaftete Prozesse der *Central Executive* als auch

durch automatische Rekrutierung der Aufmerksamkeit durch physikalische Stimuli (Orientierungsreaktion) kontrolliert (Cowan, 1999). Beide Prozessgruppen operieren durch die Einflüsse der Aktivierung eng zusammen und überführen durch sie Elemente in den Aufmerksamkeitsfokus. Innerhalb des aktivierten Bereichs des LTM können neue Kombinationen von Informationen geformt werden.

Im **Concentric Model of Working Memory** charakterisiert Oberauer (2003) das Arbeitsgedächtnis durch drei zusammenspielende Arbeitsgedächtnisfunktionen:

- *Simultane Speicherung und Verarbeitung*
- *Steuerung und Kontrolle*
- *Koordination*

Dabei wird *Verarbeitung* als die Transformation von Informationen und das Ableiten neuer Informationen definiert, im Gegensatz zu kognitiven Prozessen, welche die Information in gleich bleibender Weise vorhalten. *Speicherung* wird als Vorhalten neuer Daten über einen gewissen Zeitraum verstanden, in dem sie nicht mehr wahrgenommen werden. *Steuerungs-* und *Kontrollprozesse* bestehen aus dem Überwachen von kognitiven Prozessen und Handlungen. *Koordination* meint Bildung neuer Relationen, welche in weitere Strukturen integriert werden können.

Im **Controlled-Attention Framework** beschreiben Kane et al. (2007) das Verhältnis und Zusammenspiel der Komponenten des Arbeitsgedächtnisses, wobei die *Central Executive* eine besonders wichtige Rolle unter Bedingungen des Konflikts und der Interferenz spielt. Sie übernimmt generell die folgenden Aufgaben:

- Erzielen von Aktivierung und bewusstem Zugriff durch kontrollierte Aufmerksamkeit
- Aufrechterhaltung der Aktivierung bzw. des Zugriffs auf die Stimuluspräsentation, die Zielabstraktionen oder die Antwortproduktionen sowohl innerhalb als auch außerhalb des bewussten Fokus
- Blockieren von interferierenden und konfligierenden Gedanken durch Inhibition

Dabei können die Prozeduren, welche für die Aktivierung und ihre Aufrechterhaltung sorgen, sowohl routiniert sein als auch in größerem Maße Aufmerksamkeit beanspruchen, je nachdem, um welche Aufgabe es sich handelt und wie diese bearbeitet wird. Die *Central Executive* steuert ferner *Chunking-* und *Gruppierungsmechanismen*, welche sich auf phonologische, visuelle, räumliche und motorische Inhalte beziehen können.

### Kognitionsansatz in der KI

Auch im Bereich der künstlichen Intelligenz haben sich Forscher mit der Zugreifbarkeit von Informationen beschäftigt. Das spiegelt sich beispielsweise in der Gleichung von Newell (1982) wider: „*representation = knowledge + access*“. Arbeiten im Kontext des LILOG Projekts beschäftigten sich mit der Strukturierung von Wissen und dessen Zugriffsmöglichkeiten (Wachsmuth, 1987; Wachsmuth & Gängler, 1991). Dabei wird zwischen den Zuständen *VISIBLE* und *REACHABLE* unterschieden. Basierend auf einer empirischen Untersuchung wurde erkannt, dass domänenspezifisches Wissen gruppiert gespeichert wird. Es werden Prinzipien der Strukturierung abgeleitet, welche auf Wissens-elementen (z.B. logische Sätze oder Produktionsregeln) operieren. In der Strukturierung wird zwischen verschiedenen Wissenspaketen unterschieden, welche jeweils weitere Pakete beinhalten können. Die Wissenspakete können alternative, konkurrierende Lösungsmethoden enthalten; es ist aber jeweils immer nur ein Wissenspaket zugreifbar (*ACCESSED*). Dynamische Zugriffsprozesse beschäftigen sich damit, wie das *ACCESSED*-Tag zwischen den verschiedenen Wissenspaketen wechseln kann. Die Unterscheidung zwischen sichtbaren Elementen, welche den Lösungsprozess beeinflussen können, und den zugreifbaren Elementen, welche als Kandidatenmenge dienen, ähnelt in dieser Hinsicht dem *Embedded Processes Model* von Cowan bzw. dem *Concentric Model of Working Memory* Oberauers.

#### 4.1.4 Emotionen

Emotionale Zustände spielen im Rahmen der kognitiven Kontrolle eine weitere regulative Rolle. Nach Miyake & Shah (1999) sollen bei der Modellierung des Arbeitsgedächtnisses auch Prozesse der emotionalen Regulierung berücksichtigt werden. So wie das Arbeitsgedächtnis als Interface zwischen Perzeption und Kognition dient, könnte es auch als Interface zwischen Emotion und Kognition vermitteln.

Engle et al. (1999) postulieren in ihrem Ansatz, eine *emotionalen Markierung* bei der Berechnung der Zugreifbarkeit eines Elements im Arbeitsgedächtnis zu berücksichtigen. Auch J. R. Gray & Braver (2002) sehen die Rolle der Emotionen darin, *bottom-up* Einflüsse auf den Präfrontalen Cortex (insbesondere auf den Bereich, der mit dem Arbeitsgedächtnis in Verbindung gebracht wird) auszuüben, und dadurch den Einfluss der aktiven Ziele auf das gewählte Verhalten zu erhöhen. An diesem Punkt der Verarbeitung löst sich die funktionale Spezialisierung auf, und Emotion und Kognition tragen gemeinschaftlich und gleichberechtigt zu der Kontrolle der Gedanken und des Verhaltens bei (J. R. Gray, 2004).

Auch Damasio und Kollegen plädieren dafür, dass der Entscheidungsprozess durch Markierungssignale der bio-regulatorischen Prozesse beeinflusst wird, und

dass zu diesen beeinflussenden Signalen auch Emotionen und Gefühle zählen (Bechara, 2004). In neurowissenschaftlichen Untersuchungen fanden Braver et al. (2002), dass induzierte Emotionen neuronale Aktivität innerhalb des Präfrontalen Cortex, der für kognitive Kontrolle bekannt ist, modulieren. Damit könnte eine Funktion der emotionalen Zustände darin bestehen, kognitive Prozesse situationsabhängig so zu modulieren, dass Verarbeitungsprioritäten zwischen konfligierenden Alternativen dynamisch gesetzt werden (J. R. Gray, 2004).

Diese Annahme ist in Einklang mit dem Modell der *somatischen Marker* von Damasio (1994). Er betont, dass Emotionen eine integrale Rolle bei der Kontrolle und Regulation kognitiver Aktivitäten spielen. Im Falle einer Situation der komplexen Entscheidungsfindung gibt es eine große Anzahl an Handlungsoptionen. Bei der Einschränkung des Entscheidungsraums können assoziierte Emotionen helfen. Durch sie können schnell und unbewusst Optionen, welche mit negativen Emotionen verknüpft sind, ausgeschlossen werden.

Emotionale Zustände könnten damit einen Weg darstellen, um diverse abstrakte Kontexthinweise in eine Gesamteinschätzung der Situation zu konvertieren und das gesamte kognitive System so zu einer möglichst effektiven Reaktion zu bringen.

Eine weitere Funktion des emotionalen Zustands könnte ferner darin bestehen, verschiedene Formen der Informationsverarbeitung zu unterstützen. Eine integrative Perspektive der Emotion und Kognition ist laut J. R. Gray (2004) in Einklang mit der allgemeinen Linie anderer theoretischer Perspektiven einschließlich der aktuellen Ansätze der emotionalen Intelligenz (Salovey et al., 2002) und der Selbstregulation (Luu & Tucker, 2004).

#### 4.1.5 Aufmerksamkeit

J. Cohen et al. (2004) folgen dem Ansatz von Posner & Petersen (1990) und gehen dabei von drei Aufmerksamkeitssystemen aus:

- (1) ein *Anterior Attentional System* welches sich auf kognitive Kontrolle und Handlungsselektion bezieht
- (2) ein *Posterior Attentional System* welches mit der Ausrichtung der perzeptuellen Aufmerksamkeit in Verbindung gebracht wird
- (3) und ein *Arousal System* welches sich auf Wachsamkeitsphänomene auf unterer Ebene bezieht.

In Anbetracht der deliberativen Aspekte von *Joint Attention* gilt es, den Aufmerksamkeitsfokus der kognitiven Kontrolle zu berücksichtigen. Kognitive Kontrolle und Aufmerksamkeit können als emergente Eigenschaften der Informationsrepräsentation des Arbeitsgedächtnis gesehen werden (Courtney, 2004).

In Einklang mit dieser Forschung betont Cowan (1999), dass der Aufmerksamkeitsfokus gemeinschaftlich durch vorsätzliche Prozesse (*central executive system* und unwillkürliche Prozesse (*attention orienting systems*) gesteuert wird (siehe auch Abschnitt Arbeitsgedächtnis 4.1.3).

#### 4.1.6 Zusammenfassung der *bottom-up* Basisprozesse

Die Arbeitsgedächtnismodelle unterscheiden sich durch die in ihnen wirkenden verschiedenen Mechanismen. Es sind zwei Richtungen der Verarbeitung zu erkennen, denen die Prozesse folgen. Zum einen finden *bottom-up* Prozesse statt, welche vornehmlich die Aufgabe übernehmen, den Agenten dynamisch und flexibel an seine Umwelt anzupassen. Sie modellieren den externen Einfluss auf interne Prozesse und erstellen einen Kontext relevanter Informationen. Von diesen sind *top-down* Mechanismen abzugrenzen, welche es dem Agenten ermöglichen, Ziele über einen längeren Zeitraum zu verfolgen. Aus den beiden Prozessgruppen der *bottom-up* und *top-down* Mechanismen emergiert die kognitive Kontrolle.

Zu den Mechanismen der *bottom-up* Prozessgruppe zählen Wahrnehmungsfähigkeiten, welche einen Einfluss auf die Kontextinformationen ausüben. Diese Prozesse sind für die *Situietheit* des Agenten verantwortlich. In diesem Zusammenhang sind die folgenden Mechanismen zu nennen, welche zu der (Re)aktivierung von Kandidaten (Objekten und Eigenschaften) führen bzw. die Strukturen der Arbeitsgedächtniselemente der Situation anpassen:

##### **Bottom-up Prozesse:**

- *Rehearsal* (siehe *Multiple-Component Model*)
- *Retrieval* und *Maintenance* (siehe *Embedded Processes Model*)
- *Chunking* (siehe *Concentric Model of Working Memory*)
- *Activation, Priming, Spreading Activation* (siehe *Controlled-Attention Frameworks*)
- *Valenz - Salience Markers* (siehe *Controlled-Attention Frameworks, Emotionen*)

Parallel zu der ungerichteten Aktivierung von Elementen, stellen *Valenzwerten*, eine weitere Quelle der Vorauswahl und Einordnung potentieller Kandidaten dar.

Das enge Zusammenspiel automatischer *bottom-up* Prozesse und intentionaler *top-down* Prozesse wird auch in der Kognitionspsychologie genannt. Hommel

(2000) sieht die Verbindung zwischen den Prozessgruppen darin, dass die intentionalen Prozesse nicht mit den automatischen Prozessen konkurrieren, sondern die Voraussetzungen für sie schaffen.

## 4.2 Mentale Zustände

In der Psychologie und Kognitionsforschung werden in erster Linie *bottom-up*-Basisprozesse untersucht; sie sind in eingeschränkten *Experimental-Settings* konkret zu messen. Auf höheren Ebenen operierende Prozesse sind durch Experimente schwerer sichtbar zu machen. Deshalb kann es in diesem Fall sehr hilfreich sein, auf abstrakte Modelle der Philosophie zurückzugreifen, diese jedoch einer Plausibilitätsprüfung durch den Vergleich mit den zugrunde liegenden kognitiven Basismechanismen zu unterziehen. Während im Bereich der Psychologie und Kognitionswissenschaft zumeist Phänomene auf einer *Prozess-orientierten Ebene* betrachtet werden und dabei einzelne Mechanismen und Prozesse beleuchtet werden, setzen die Ansätze im Bereich der Philosophie auf einer sehr hohen Abstraktionsebene an und sind im Gegensatz zu den kognitiven Basisprozessen eher *Zustands-orientiert* in der Beschreibung des jeweiligen mentalen Zustands. Entsprechende Arbeiten untersuchen schwerpunktmäßig, welche Einflussfaktoren im Entscheidungsprozess eine Rolle spielen und welche verschiedenen *Commitment*-Strategien existieren. Die Prozesse und Mechanismen, die für das Zustandekommen der Annahmen verantwortlich sind, werden nicht näher definiert. Es wird zwar spezifiziert, welche Faktoren es zu berücksichtigen gilt, und wie oft bestimmte Funktionen unter vorgegebenen Berechnungsaufwand ausgeführt und betrachtet werden dürfen, aber nicht, wie diese genau operieren.

Auf der Ebene biologischer Prozesse lassen sich keine abstrakten *Commitment*-Strategien definieren. Viele Konzepte zur Beschreibung des komplexen Verhaltens des Menschen machen erst ab einer bestimmten Abstraktionsstufe Sinn. Ohne Konzepte wie *Obligation*, *Norm* oder *Verpflichtung* lassen sich jedoch Verhaltensweisen im Rahmen einer Kooperation nur schwer motivieren und charakterisieren. Die Mechanismen, welche auf unterer Ebene zu diesen Phänomenen führen, können dabei sehr vielfältig sein und sich aus verschiedenen Prozessen zusammensetzen.

Forschungsansätze der Philosophie sehen das Verhältnis zwischen neuro- bzw. kognitionswissenschaftlichen Modellen und Ansätzen der mentalen Modelle kontrovers. Dabei liegen laut der *Encyclopedic Reference of Neuroscience* nach Schuette (2004) die Ansätze des *Intentionalen Realismus* und des *Eliminativen Materialismus* am jeweils gegenüberliegenden Ende eines Spektrums.

Die *Eliminativen Materialisten* wie beispielsweise Churchland (1981) behaupten, dass es prinzipiell neuro-wissenschaftliche Erklärungen für das mensch-

liche Verhalten gibt, welche übergeordnet und inkompatibel mit den intentionalen Erklärungen sind. Ein Verfechter der Gegenposition ist Fodor (1988). Er verteidigt den *Intentionalen Realismus*; *Beliefs* und *Desires* seien reale und kausal hervorgerufene mentale Zustände. Es gäbe keinen Grund zu bezweifeln, dass es möglich sei, mit Hilfe von wissenschaftlicher Psychologie die allgemeinen Konzepte von *Beliefs/Desires* zu verteidigen und zu rechtfertigen.

Zwischen den Extrempositionen gibt es auch viele Befürworter einer gemäßigten Zwischenposition wie beispielsweise Dennett (1987). Er betont, dass es nicht zielführend sei, intentionale Erklärungen komplett abzuschaffen, da es gewisse menschliche Verhaltensweisen gibt, welche nur mittels der Sichtweise der *intentionalen Einstellung (intentional stance)* entdeckt und beschrieben werden können. Diese und andere gemäßigte Zwischenpositionen versuchen zu zeigen, dass die semantischen Konzepte der intentionalen Zustände real genug sind, um die intentionalen Erklärungen zu untermauern, aber nicht so real, dass sie eine genaue physikalische Implementation verlangten. Dabei versuchen die Ansätze die *Mentalen Repräsentationen* im weiteren Sinne beizubehalten, ohne sich dabei der Existenz der mentalen Repräsentationen im engeren Sinne verpflichtet zu fühlen (Schuette, 2004). Searle (1987) hält geistige Zustände für genauso real wie alle anderen biologischen Phänomene hält. Er sieht sich dabei als *biologischer Naturalist*.

Diese Position steht in Einklang mit Positionen von Wissenschaftlern aus dem Bereich der künstlichen Intelligenz. Für Georgeff et al. (1999) sollte ein System, welches in dynamischen, unsicheren unvorhersehbaren Welten agieren soll, eine Repräsentation von *Beliefs*, *Desires* und *Intentions* enthalten. Elio (2002) geht es nicht darum, ein genaues Korrelat der mentalen Zustände zu implementieren, sondern vielmehr darum, eine Ebene zu finden, auf welcher man das Verhalten eines Agenten beschreiben kann. Sie argumentiert, dass jede Architektur, welche sich dem Ressourcen-beschränkten-Schlussfolgern widmet, in der einen oder anderen Form die allgemeinen Konzepte des *Belief-Desire-Intention* Modells umsetzen muss. Jede Architektur liefere eine Erklärung auf einer bestimmten Abstraktionsebene und die Herausforderung bestehe darin, diese Erklärungsebenen in einen umfassenderen Ansatz der Kognition zu überführen. Braubach et al. (2004) sehen einen generellen Konsens darin, dass die Konzepte *Belief*, *Desire* und *Intention* sehr gut dazu geeignet sind, den mentalen Zustand eines Agenten zu beschreiben. Es bestehe jedoch eine Diskrepanz zwischen den theoretischen bzw. logischen BDI-Modellen und den implementierten BDI-Systemen.

### 4.2.1 Belief-Desire-Intention-Modell (BDI)

*Intentionalität* stellt die Fähigkeit des Geistes dar, sich auf etwas zu beziehen, etwas zu repräsentieren bzw. für etwas zu stehen. Dabei bezieht sich das "etwas"

auf Dinge, Eigenschaften sowie Sachverhalte. Intentionalität ist ein philosophischer Begriff und leitet sich von dem lateinischen Wort *intentio* ab, das sich seinerseits von dem Verb *intendere* ableitet, welches die Bedeutung trägt „auf ein Ziel oder eine Sache ausgerichtet sein“. Wahrnehmungen, Annahmen und Überzeugungen (Beliefs), Wünsche und Ziele (Desires) sowie Intentionen und viele andere propositionale Einstellungen stellen mentale Zustände mit Intentionalität dar. Sie betreffen bzw. repräsentieren Objekte und Sachverhalte unter einem bestimmten psychologischen Modus. Sie illustrieren die grundlegende Dualität der Intentionalität des Mentalen: die Dualität zwischen *mind-to-world* und *world-to-mind* (*Stanford Encyclopedia of Philosophy*, 2009). Das BDI Modell baut auf zwei miteinander verwandten Theorien über das philosophische Konzept der *Intentionalität* auf.

- (i) *Intentional Stance* stellt eine Abstraktionsebene dar (Dennett, 1987), auf welcher Verhalten in Hinblick auf mentale Eigenschaften betrachtet wird. Dabei spielen beispielsweise propositionale Einstellungen wie Beliefs und Desires eine wichtige Rolle.
- (ii) *Practical Reasoning* Theorie (Bratman, 1987), beschäftigt sich mit der Entscheidungsfindung beim Menschen und stellt darauf aufbauend ein allgemeines psychologisches Rahmenwerk auf, um uns selbst und andere zu verstehen. Es basiert auf *Beliefs*, *Desires* und *Intentions*.

Diese beiden Ansätze der Intentionalität liefern die Werkzeuge, um Agenten mit einem solchen Abstraktionsgrad zu beschreiben, dass sie kompatibel mit der intentionalen Beschreibung sind und das *Practical Reasoning* unterstützen. Es wird allgemein akzeptiert, dass Agenten aus Gründen der Berechnungskomplexität nicht in der Lage sind, Optimierungen im Raum aller möglichen Handlungsoptionen vornehmenzukönnen. Bratmans Ansatz hilft, diesen Raum so einzuschränken, dass *Reasoning*-Prozesse darauf operieren können (Georgeff et al., 1999). Im Rahmen der *Reasoning*-Prozesse besteht die Möglichkeit *Meta-Pläne* einzusetzen, welche den Entscheidungsprozess des Agenten dynamisch in Abhängigkeit von *Meta-Beliefs* steuern und anpassen. Dadurch kann der *Reasoning*-Prozess sich selbst modifizieren.

Ein BDI-Modell besteht aus den folgenden Strukturen und Prozessen.

- Die *Beliefs* repräsentieren den deklarativen Wissensstand des Agenten. Dabei besteht die Faktenmenge aus Informationen über den Zustand der Welt, und kann zusätzlich durch Inferenzfakten des Agenten ergänzt werden. Auslöser neuer Beliefs sind die Wahrnehmungskomponente sowie ausgeführte Intentionen. Der Agent kann mit einer statischen Wissensbasis zu Beginn seiner Instantiierung ausgestattet werden, welche dann im Rahmen

seiner Interaktion mit der Umwelt und seinen internen Prozessen dynamisch ergänzt wird. Dabei können die Beliefs auch sogenannte *Meta-Beliefs* umfassen, welche sich auf Agenteninterna wie Entscheidungs- und Suchstrategien beziehen können und in den Deliberationprozess des Agenten einbezogen werden können.

- Die *Desires* bestehen aus den Zielen des Agenten. In den meisten BDI-Modellen besteht die Forderung, dass die Ziele untereinander konsistent sein müssen. Auf der Zielebene lassen sich sogenannte *Meta-Ziele* spezifizieren, welche Prozesse auf der Meta-Ebene steuern können.
- Prozedurales Wissen wird in dem BDI-Modell anhand von *Plänen* umgesetzt. Dabei verfügt ein Plan über eine *Triggering-Condition*, welche spezifiziert, für welche Aufgaben ein Plan eingesetzt werden kann. Des Weiteren kann über Einschränkungsbedingungen die Berücksichtigung des Anwendungskontextes erwirkt werden. Der Hauptteil eines Plans beschreibt die auszuführenden Aktionen und wickelt den prozeduralen Teil des Plans ab. Auch Pläne können sich auf die Meta-Ebene beziehen.
- *Intentionen* bestehen aus einer Verknüpfung von Zielen und Plänen. Der Agent verpflichtet sich, ein Ziel auf eine bestimmte prozedurale Weise zu verfolgen. Die Intensionsstruktur wird durch Intensionsstacks gebildet. Dabei wird zwischen *Top-Level-Goals*, welche einen eigenen Stack besitzen und Subgoals unterschieden, welche im Zuge des *Means-Ends-Reasoning* aufgeworfen und auf den jeweiligen Stack eines *Top-Level-Goals* geschoben werden.

---

**Algorithm 1** BDI interpreter
 

---

```

1: procedure BDI INTERPRETER CYCLUS
2:   initialize_states();
3:   loop
4:     options := options_generator(event_queue);
5:     select_options := deliberate(options);
6:     update_intentions(selected_options);
7:     execute;
8:     get_new_external_events();
9:     drop_successful_attitudes();
10:    drop_impossible_attitudes();
11:   end loop
12: end procedure

```

---

Der BDI-*Interpreter* operiert auf Ereignissen. Ereignisse können aufgrund von Veränderungen der *Beliefs* sowie durch die Erzeugung neuer Ziele entstehen (siehe *Algorithm 1* nach Rao & Georgeff (1995)). Für die Ereignisse wird durch den Prozess des *Means-Ends-Reasoning* eine Liste mit Handlungsoptionen generiert, welche beinhaltet, durch welche Mittel ein Ziel erreicht werden kann. Die Optionen werden auf Basis eines Abgleichs mit der *Triggering-Condition* der Pläne gewonnen, wobei die Einschränkungsbedingungen der Pläne berücksichtigt werden. Die gültigen Handlungspläne stellen die *Optionen* (siehe *Algorithm 1*, Zeile 4) des Agenten dar.

In einem zweiten Schritt der *Deliberation* (siehe *Algorithm 1*, Zeile 5) wägt der Agent die Optionen gegeneinander ab, wählt die nächste zu verfolgende Intention aus und aktualisiert seine Intensionsstruktur (siehe *Algorithm 1*, Zeile 6). Die Intention wird dann im nächsten Schritt ausgeführt (siehe *Algorithm 1*, Zeile 7). Abschließend wird der Zustand der Intentionen überprüft und erfolgreiche sowie obsolete bzw. unerreichbare Intentionen aus der Intensionsstruktur entfernt (siehe *Algorithm 1*, Zeile 9-10).

Das System führt *Means-Ends-Reasoning* und Deliberationsprozesse im Kontext bestehender Intentionen aus. Eine spezielle Charakteristik der BDI-Architektur besteht dabei darin, dass ein Agent, der sich einmal dazu verpflichtet hat, ein Ziel auf eine Weise zu verfolgen, nicht weiter nach besseren Lösungsmöglichkeiten sucht, sondern nur durch explizit angestoßene *reconsider*-Mechanismen seine Strategie noch einmal „überdenkt“. Durch dieses *Commitment-Verhalten* kann viel Zeit im Entscheidungsprozess eingespart werden. Es können je nach Szenario verschieden persistente *Commitment*-Strategien eingesetzt werden.

#### 4.2.2 Relevanz des BDI-Modells

Das BDI-Modell ist unter den Gesichtspunkten der Situiertheit, der Zielgerichtetheit, der Kompetenz reaktiver Verarbeitungsmechanismen sowie der sozialen Interaktion entworfen worden. Das bedeutet, dass ein BDI-Agent in der Lage ist, auf Änderungen zu reagieren, mit seiner Umwelt zu kommunizieren, und seine Ziele konsistent zu verfolgen.

Die Relevanz des *Belief-Desire-Intention*-Modells für rationale Agenten lässt sich in Bezug auf die folgenden Punkte erklären. Zum einen baut der Ansatz auf einer wohlfundierten philosophischen Basis der Intentionalität auf. Des Weiteren existieren elegante abstrakte logische Semantiken (Rao & Georgeff, 1998; Singh et al., 1999; Wooldridge, 2000), sowie eine Vielzahl verschiedener erfolgreicher Implementierungen wie beispielsweise IRMA (Bratman et al., 1991), *Procedural Reasoning System* (PRS) (Georgeff & Lansky, 1987) und die Familie der PRS-angelehnten Systeme wie dMARS (d’Inverno et al., 1998), UM-PRS (Lee et

al., 1994), JAM (Huber, 1999), Jack (Howden et al., 2001) und Jadex (Pokahr et al., 2005b). Diese Implementierungen führten zu einem breiten Spektrum an erfolgreichen Anwendungen und Applikationen: z.B. ein Diagnosesystem für Space Shuttle, ein Factory Process Control System sowie ein Einsatz im Business Process Management (George & Rao, 1996).

### 4.2.3 Beliefs

Als exemplarische Spezifikation der *Beliefs* dient das abstrakte Modell von *dMars*. *dMars* ist eine Implementation des *PRS*-Modells, welches das bekannteste Modell für BDI-Architekturen darstellt. d'Inverno et al. (1998) stellen damit eine Benchmark-Architektur auf, die als Grundlage dienen kann, BDI-Systeme miteinander zu vergleichen. Sie greifen für ihre Spezifikation auf die Z-Sprache zurück. Z ist eine modellorientierte, formale Spezifikationssprache, welche auf Mengentheorie und Prädikatenlogik der ersten Stufe basiert (ISO/IEC, 2002).

Für die weiteren Definitionen werden die folgenden Typen als Basistypen angenommen.

$$[Var, FunSym, PredSym] \quad (4.1)$$

Ein *Term* besteht entweder aus einer Variablen oder aus einem Funktionssymbol angewendet auf eine (möglicherweise leere) Sequenz von Termen:

$$Term ::= var \langle \langle Var \rangle \rangle \mid functor \langle \langle FunSym \times seqTerm \rangle \rangle \quad (4.2)$$

*Beliefs* werden als Literale der klassischen Prädikatenlogik erster Stufe definiert. Sie bestehen aus atomaren oder negativen atomaren Formeln, welche keine ungebundenen Variablen beinhalten dürfen.

#### Schema 1

<i>Atom</i> <i>head</i> : <i>PredSym</i> <i>terms</i> : <i>seqTerm</i>
--

$$BeliefFormula ::= pos \langle \langle Atom \rangle \rangle \mid not \langle \langle Atom \rangle \rangle \quad (4.3)$$

$$Belief == \{b : BeliefFormula \mid belvars b = \emptyset \bullet b\} \quad (4.4)$$

(*belvars* stellt eine Funktion dar, welche für eine gegebene Belief-Formel die Menge aller in der Formel enthaltenen Variablen zurück liefert.)

#### 4.2.4 Ziele (*Desires*)

d’Inverno et al. (1998) definieren Ziele formal durch eine einfache *Temporal Modal Language* mit zwei zusätzlichen unären Konnektiven *achieve* (!) und *query* (?). Ein Agent mit dem Ziel  $!\varphi$  hat das Ziel eine (möglicherweise leere) Sequenz von Aktionen auszuführen, sodass durch die Ausführung der Aktionen  $\varphi$  wahr sein wird. Für die Definition der zusätzlichen Konnektive werden *Situation-Formulas* definiert. Diese bestehen aus Ausdrücken, welche in Bezug auf eine Menge von Beliefs ausgewertet und auf Wahrheitswerte abgebildet werden können.

$$\begin{aligned}
 \textit{SituationFormula} & ::= \textit{belform} \langle\langle \textit{BeliefFormula} \rangle\rangle & (4.5) \\
 & | \textit{and} \langle\langle \textit{SituationFormula} \times \textit{SituationFormula} \rangle\rangle \\
 & | \textit{or} \langle\langle \textit{SituationFormula} \times \textit{SituationFormula} \rangle\rangle \\
 & | \textit{true} \\
 & | \textit{false}
 \end{aligned}$$

Eine *Goal-Formula* ist dann eine Belief-Formula mit einem vorangestellten *Achieve*-Operator oder eine Situation-Formula mit einem vorangestellten *Query*-Operator.

$$\textit{Goal} ::= \textit{achieve} \langle\langle \textit{BeliefFormula} \rangle\rangle | \textit{query} \langle\langle \textit{SituationFormula} \rangle\rangle \quad (4.6)$$

Die *Achieve*-Formula dient dazu, Zielzustände des Agenten zu spezifizieren, wohingegen der *Query*-Operator dazu eingesetzt wird, Informationen aus den Beliefs des Agenten abzufragen. Vor der Annahme eines *Achieve*-Goals sollte zunächst überprüft werden, ob der gewünschte Zustand nicht schon gilt. In diesem Fall sollte keine Aktion vorgenommen werden, und das Ziel gilt als erreicht.

Andere BDI-Modellierungen unterstützen neben dem *Achieve*- und *Query*-*Goal* noch weitere Zieltypen, z.B. (Huber, 1999; Pokahr et al., 2005a). Im Gegensatz zu einem *Achieve*-*Goal* wird ein *Perform*-*Goal* immer ausgeführt, sobald der Agent es einmal angenommen hat. Bei einem *Perform*-*Goal*, liegt der Fokus nicht auf der Erreichung eines Zustands, sondern es geht um die Ausführung der Handlung selbst. Diese Unterscheidung spiegelt die im *SharedPlans* geforderte Unterscheidung zwischen *intend-to* und *intend-that* wider und bildet damit eine zentrale Grundlage für die Umsetzung von Kooperationskompetenz durch den Ansatz der *SharedPlans* (siehe Abschnitt 2.1.3). Ein *Maintain*-Ziel dient dazu, einen bestimmten Weltzustand aufrecht zu erhalten. Wann immer dieser Zustand nicht mehr gilt, wird das Ziel aktiv.

### 4.2.5 Pläne

Pläne beziehen sich in den klassischen BDI-Systemen auf Sequenzen von Aktionen, die ein Agent ausführen kann, um eine oder mehrere seiner Intentionen zu erreichen. Damit bestehen Pläne aus einer Form deklarativen Wissens, welches beinhaltet, wie ein Ziel oder eine bestimmte Situation erreicht werden kann (Sardina et al., 2006). Pollack (1992) unterscheidet zwischen Plänen im Sinne eines Rezeptes, das es auszuführen gilt, und Plänen im Sinne einer mentalen Einstellung, welche aus einer strukturierten Kollektion von *Beliefs* und Intentionen besteht. Ein Plan in Form eines Rezeptes, welches in einer Plan-Bibliothek repräsentiert wird, besteht aus: einem *Header* (Name der Aktion, eindeutig zu identifizieren), einer *Precondition List* (enthält, was erfüllt sein muss, damit der Operator angewendet werden kann), *Effect List* (enthält, was nach Anwendung des Operators wahr sein wird) *Constraint List* (enthält, Einschränkungen für die legale Instantiierung des Operators), *Body* (enthält eine Menge von Teilzielen und Aktionen).

In *dMars* bestehen interne Aktionen aus dem Hinzufügen oder Löschen von *Beliefs*.

$$\text{IntAction} ::= \text{add} \langle \langle \text{BeliefFormula} \rangle \rangle \mid \text{remove} \langle \langle \text{BeliefFormula} \rangle \rangle \quad (4.7)$$

Externe Aktionen stellen eine Form eines externen Methodenaufrufs dar und bestehen aus einem Namenssymbol und einer Sequenz von Übergabewerten (Parametern).

#### Schema 2

$\begin{array}{l} \text{ExtAction} \\ \text{name} : \text{ActionSym} \\ \text{terms} : \text{seqTerm} \end{array}$
--

Daten-getriebene Pläne in *dMars* beinhalten ein *Trigger-Event* (auslösendes Ereignis), das als Auslöser für die Annahme eines Plans gilt. Die relevanten Ereignisse bestehen aus dem Hinzufügen bzw. Löschen eines *Beliefs*, dem Erhalt einer Nachricht oder der Annahme eines Ziels.

$$\begin{aligned} \text{TriggerEvent} := & \text{addbelevent} \langle \langle \text{Belief} \rangle \rangle \mid \text{rembelevent} \langle \langle \text{Belief} \rangle \rangle \mid \\ & \text{toldevent} \langle \langle \text{Atom} \rangle \rangle \mid \text{goalevent} \langle \langle \text{Goal} \rangle \rangle \end{aligned} \quad (4.8)$$

Der Teil eines Plans, mit den ausführbaren Handlungen wird *Body* genannt; er besteht aus internen bzw. externen Aktionen sowie möglichen Teilzielen. Sie sind in Form einer Baumstruktur aufgebaut.

$$\text{Branch} ::= \text{extaction} \langle \langle \text{ExtAction} \rangle \rangle \mid \text{intaction} \langle \langle \text{IntAction} \rangle \rangle \mid \text{subgoal} \langle \langle \text{Goal} \rangle \rangle$$

$$Body ::= End \langle\langle State \rangle\rangle \mid Fork \langle\langle \rho_1 (State \times Branch \times Body) \rangle\rangle \quad (4.9)$$

Ein Plan besteht demnach aus einem *Trigger-Event* und optional einer einschränkenden *Context*-Bedingung, die erfüllt sein muss, damit der Plan als Option berücksichtigt wird. Die *Maintenance*-Bedingung muss gültig sein, damit der Plan weiter ausgeführt werden darf. Es besteht die Möglichkeit, eine Menge interner Aktionen anzugeben, die ausgeführt werden, wenn der Plan erfolgreich beendet wird bzw. der Plan scheitert.

### Schema 3

<i>Plan</i>
<i>inv</i> : <i>TriggerEvent</i>
<i>context</i> : <i>optional</i> [ <i>SituationFormula</i> ]
<i>body</i> : <i>Body</i>
<i>maint</i> : <i>SituationFormula</i>
<i>succ</i> : <i>seq IntAction</i>
<i>fail</i> : <i>seq IntAction</i>

## 4.2.6 Intentionen

Eine Intention besteht aus einem Ziel und der Entscheidung, dieses auf eine bestimmte Art und Weise zu erreichen. In *dMars* wird eine Intention als eine Sequenz von *Plan-Instanzen* definiert.

$$Intention ::= seqPlanInstance \quad (4.10)$$

Eine Plan-Instanz besteht aus einem Plan zusammen mit einem *Binding* (Substitution) in welchem spezifiziert wird, mit welchen Werten Plan-Variablen belegt sind. Zusätzlich besitzt eine Plan-Instanz einen Zustand, der im Falle von *dMars* aus den Werten *active* oder *suspended* bestehen kann.

$$Status ::= active \mid suspended \quad (4.11)$$

(weitere formale Definition siehe (d'Inverno et al., 1998))

### Schema 4

<p><i>PlanInstance</i></p> <p><i>origplan</i> : <i>Plan</i></p> <p><i>env</i> : <i>Substitution</i></p> <p><i>state</i> : <i>State</i></p> <p><i>nextbranches</i> : <math>\wp</math> <i>Branch</i></p> <p><i>branch</i> : <i>optional</i>[<i>Branch</i>]</p> <p><i>status</i> : <i>Status IntAction</i></p> <p><i>id</i> : <i>PlanInstanceId</i></p> <hr/> <p><math>state \in PlansStates\ origplan</math></p> <p><math>state \in dom\ End \Rightarrow nextbranches = \emptyset</math></p> <p><math>nextbranches \subseteq PlanNextBranches\ origplan\ state</math></p> <p><math>branch \subseteq nextbranches</math></p>
---

Im Rahmen einer Intention geht ein Agent eine Verpflichtung ein, ein Ziel auf eine bestimmte Weise (Plan) zu erreichen. Der Grad, zu dem eine Verpflichtung beibehalten wird, kann in Abhängigkeit der Bereitschaft eines Agenten seine Intentionen zu überdenken variieren. Das Spektrum reicht dabei nach Rao & Georgeff (1991) von einem *blindly-committed* Agent, der so lange seine Intentionen verfolgt, bis sie erfolgreich ausgeführt sind, über *single-minded* Agenten, welche auch bereit sind eine Intention aufzugeben, wenn diese scheitert, bis hin zu *open-minded* Agenten, welche so lange an einer Intention festhalten, bis diese erfolgreich ausgeführt wurde oder das zugehörige Ziel aufgegeben wird. Thangarajah, Padgham & Harland (2002) schlagen einen Sonderfall des *Open-minded Commitments* vor, bei welchem Ziele aufgegeben werden dürfen, wenn ein konfligierendes Ziel mit einer höheren Priorität existiert. Sie sehen dies als eine Minimal-Anforderung an das Commitmentverhalten eines rationalen Agenten.

### Das klassische BDI-Modell - Basis für die Kooperationsmodellierung

Das BDI-Modell bietet eine gute Grundlage für die Modellierung mentaler Zustände (Rao & Georgeff, 1995). Für die Modellierung eines Kooperationspartners mit situierten Problemlösungskompetenzen müssen jedoch noch einige der mentalen Konzepte ausgebaut werden. Ein Defizit in vielen der BDI-Implementationen besteht in der impliziten Zielmodellierung. Die Vernachlässigung einer deklarativen Repräsentation geht dabei zu Lasten der Berücksichtigung einer **Zielinteraktion**, welche jedoch gerade in Kooperations Szenarien von entscheidender Bedeutung ist. Ohne eine explizite Repräsentation von Zielen und den damit verbundenen Intentionen können Ziele zudem nicht ausgehandelt werden und ein Kooperationspartner kann kein angemessenes Commitmentverhalten umsetzen. Gerade in Bezug auf die höchste Ebene der Kooperation, die Ebene der *Ge-*

*meinsamen Intentionalität* (siehe Abschnitt 2.1.4), ist es erforderlich, die eigenen Ziele sowie die Ziele des Gegenübers zu repräsentieren und in die Deliberationsprozesse einbeziehen zu können. Wichtig ist dabei die Berücksichtigung der Zielentstehung und die Modellierung von externen Verpflichtungen (**Obligationen**).

Die Komplexität der Planrepräsentation korreliert mit der Fähigkeit zur Kontextsensitivität in Bezug auf die Auswahl der Handlungsoptionen. Je mehr Informationen aus der Planrepräsentation gewonnen werden können, umso mehr Faktoren können in den *Means-Ends-Reasoning* und Deliberationsprozessen berücksichtigt werden. Das prozedurale Wissen bildet des Weiteren eine Basis für Prozesse der **Plan/Intentionserkennung**. Im Rahmen des koordinierten Planens und Handelns während einer Kooperation ist es unabdingbar, die Handlungen des Gegenübers erkennen und in den Kooperationskontext einordnen zu können.

### **Explizite Zielrepräsentation**

Beaudoin (1994) betont die Komplexität von Zielen. Er sieht Ziele als komplexe Kontrollzustände, die durch viele feine Verbindungen mit internen Prozessen verknüpft sind, welche wiederum Einfluss auf externe Aktionen nehmen. Demnach besteht die konzeptuelle Struktur eines Ziels aus den folgenden Aspekten: einer Zielbeschreibung, Annahmen und Überzeugungen, der Wichtigkeit, der Dringlichkeit, einer Insistenz, einem Grund, einem dynamischen Zustand, einem Verpflichtungszustand, einem Plan, einem Ablaufplan sowie einem Intensitätswert.

Eine explizite Repräsentation von Zielen ermöglicht es, Schemata als untereinander auswechselbar zu behandeln. Zudem lässt sich durch eine explizite Zielrepräsentation das Scheitern eines Plans von dem Scheitern eines Ziel trennen. Durch die explizite Repräsentation von Zielen kann über sie kommuniziert werden. Der Agent erhält einen direkten Zugriff auf seine Ziele und kann beispielsweise erkennen, ob ein Ziel schon erreicht ist. Er kann auch opportunistisch handeln, wenn er eine sehr günstige Gelegenheit ergreift, um auf ein Ziel hinzuarbeiten, welches gerade nicht höchste Priorität besitzt, aber durch aktuelle Umstände einen höheren Nützlichkeitswert – *Utility* – erreicht.

Laut Winikoff et al. (2002) sollten sowohl deklarative als auch prozedurale Aspekte von Zielen modelliert und explizit repräsentiert werden. Benötigt werden die deklarativen Bestandteile der Zielrepräsentation für die Fähigkeit, über die Eigenschaften von Zielen zu schlussfolgern. Prozedurale Repräsentationen werden dagegen eingesetzt, um ein Ziel effektiv in einer dynamischen Umwelt zu erreichen.

Winikoff et al. (2002) sehen eine charakteristische Eigenschaft von Zielen darin, dass sie durch eigenes Bemühen prinzipiell erreichbar sein müssen. Ein Agent mag einen bestimmten Weltzustand wünschen; hat er aber keine Möglichkeit, diesen Zustand zu beeinflussen, so kann er ihn auch nicht als Ziel annehmen. Um

zu gewährleisten, dass nur potentiell beeinflussbare Ziele angenommen werden, könnten nur solche Wünsche als Ziele zugelassen, für welche auch ein prozeduraler Plan vorliegt.

### Zielentstehung - Obligationen

Deliberative Konzepte spielen bei der Zielentscheidung keine ausschließliche Rolle. Verhalten wird sowohl durch interne Triebe, Impulse und reaktive Verhaltensweisen ausgelöst als auch durch höhere Emotionen, Verpflichtungen und soziale Konzepte (Obligationen, Normen). Es gibt Bestrebungen, das BDI-Modell auf theoretischer und praktischer Ebene um das Konzept der *Normen* und *Obligationen* zu erweitern. Obligationen werden dabei als explizite Mechanismen gesehen, welche das Verhalten des Agenten beeinflussen und für Stabilität und Zuverlässigkeit während der Interaktion sorgen, während sie auch flexibles Verhalten unterstützen.

Dastani et al. (2003) schlagen das Modell des *B-DOING*-Agenten vor. Bei der Erweiterung der BDI-Logik um Konzepte der *Desires*, *Obligations* und *Norms* besteht der zentrale Ansatzpunkt in dem *B-DOING*-Modell darin, alle motivationalen Einflüsse auf Präferenzen abzubilden, welche auf einer einheitlichen Skala miteinander verglichen werden können. Der Effekt motivationaler Einflüsse wird durch einen zweistufigen Prozess modelliert (Dignum et al., 2002). Im ersten Schritt der Deliberation ist ein Zielaufrechterhaltungs-Prozess dafür verantwortlich, *Desires*, *Obligationen* und *Normen* auszubalancieren. Dabei wird darauf geachtet, dass die Menge der angenommenen Ziele konsistent ist. In einem zweiten Schritt greifen dann die *Means-Ends-Reasoning*-Prozesse der BDI-Architektur. Durch diese Modellierung lassen sich auch verschiedene Grade einer Präferenz unterscheiden. Der Ansatz erinnert an die klassischen entscheidungstheoretischen Modelle der Utility-Funktionen (Thomason & Horty, 1996), (Boutilier & Brafman, 1997).

Auch Broersen et al. (2005) berücksichtigen Obligationen in ihrem Modell. Sie schlagen die *BOID*-Architektur vor, in welcher Feedback-Loops dafür Sorge tragen, dass alle Effekte einer Aktion berücksichtigt werden, bevor sich der Agent zu ihrer Ausführung verpflichtet. Einsatz finden dafür Mechanismen der Konfliktresolution, welche zwischen den vier Komponenten *Beliefs*, *Obligations*, *Intentions* und *Desires* vermitteln. Durch verschiedene Konfliktresolutions-Algorithmen, können unterschiedliche Typen sozialer Agenten umgesetzt werden. Die Resolution von Konflikten basiert dabei auch auf dem Prioritäten-Ansatz von Thomason (2000).

### Zielinteraktion

Ein vereinfachter Umgang mit Zielen besteht darin, jeweils nur das wichtigste Ziel als Intention zu übernehmen und kein paralleles Verfolgen weiterer Ziele zu erlauben. In diesem Fall werden die Ziele des Agenten sequentiell abgearbeitet und Konflikte werden weitestgehend vermieden. Die parallele (simultane) Ausführung von Zielen lassen einen Agenten jedoch deutlich effizienter und natürlicher agieren, vorausgesetzt, mögliche Interaktionen und Konflikte zwischen den verfolgten Zielen werden berücksichtigt. Unter *positiver Zielinteraktion* wird verstanden, dass ein Ziel zu der Erfüllung eines anderen Ziels beiträgt. *Negative Zielinteraktion* hingegen besteht, wenn sich mehrere Ziele in einer Konfliktsituation befinden und ein Ziel ein anderes behindert (Pokahr et al., 2005a).

Gerade im Bereich einer Kooperation spielt die parallele Ausführung und Bearbeitung von Zielen eine entscheidende Rolle. Der Agent muss in der Lage sein, neben dem aktuellen Ziel, das er gerade selbst ausführt, auch die Ziele und Handlungen seines Kooperationspartners im Auge zu behalten und mit in seine Überlegungen einzubeziehen. Konflikte sollten dabei frühzeitig erkannt und behoben werden können.

Das Thema der Ziel-Deliberation in einem einzelnen Agenten hat bisher in der BDI-Gemeinschaft noch nicht viel Aufmerksamkeit auf sich gezogen (Pokahr et al., 2005a). Während im Bereich der Planungs-Agenten ein beachtlicher Stand der Forschung zu dem Thema Scheduling besteht (Clement & Durfee, 1999), (Boutilier & Brafman, 1997), (Barber & Liu, 2000), welcher sich auf Konflikte während der Planentwicklung und dem Zusammenführen von Plänen bezieht, lassen sich diese Ansätze nicht ohne weiteres auf die Deliberation anwenden, da sie von vollständigen Plänen ausgehen und nicht auf Echtzeit-Entscheidungen in Bezug auf Pläne und Ziele ausgelegt sind (Thangarajah et al., 2003a).

Für die Entwicklung eines *Deliberationsalgorithmus* muss die innere Einstellung des Agenten in Form aktiver Ziele und aktiver Pläne berücksichtigt werden. Aber auch Meta-Informationen über diese Einstellungen sowie Ressourcen können relevante Faktoren darstellen (Thangarajah et al., 2003b,a; Thangarajah, Winikoff et al., 2002). Im Rahmen des Deliberationsalgorithmus muss festgelegt werden, wann und wie oft ein Agent über seine Ziele deliberieren soll (in bestimmten Intervallen (Zeit- oder Zyklus-basiert), nach Bedarf oder beides). In dynamischen Umwelten sind komplexe Deliberationsstrategien für Agenten zudem nur dann sinnvoll, wenn sie effizient operieren, sodass der Agent schnell genug auf Veränderungen seiner Umgebung eingehen kann.

Der **Ansatz Thangarajahs und Kollegen** bietet die Möglichkeit, Zielkonflikte zu erkennen und zu vermeiden, indem zusammenfassende Informationen über Ressourcen, Kontext- bzw. Vorbedingungen und Effekte von Zielen und Plänen explizit in *Informationsaufstellungen* repräsentiert werden (Thangarajah et al.,

2003b,a; Thangarajah, Winikoff et al., 2002). Der Algorithmus operiert auf den *Intentionen* des Agenten, indem sowohl Ziele als auch die mit ihnen verbundenen Pläne Berücksichtigung finden. Die Informationsaufstellungen werden eingesetzt, um sicherzustellen, dass die Effekte eines Ziels nicht mit den aktiven Kontextbedingungen und Abhängigkeitslinks anderer Ziele konfliktieren. Wenn der Agent entschieden hat, ein Ziel zu verfolgen, wird die Instanz einer *Interaktionsstruktur* erzeugt, welche während der Laufzeit aktualisiert und an die konkreten Ziele und Planinstanzen angepasst wird. Die Interaktionsstruktur besteht aus einem *Ziel-Plan-Baum*, an dessen Knoten jeweils die zusammenfassenden Informationsaufstellungen der Bedingungsanforderungen und resultierenden Effekte repräsentiert werden, zusammen mit den Plan-Knoten und den Abhängigkeitseinträgen des Plans. Die Interaktionsaufstellungen werden während der Kompilierung abgeleitet und während der Laufzeit aktualisiert. Es werden jedoch nur Abhängigkeiten zwischen Knoten berücksichtigt, die zum gleichen *Top-Level-Goal* zählen. Des Weiteren ist der Ansatz auf sogenannte *Achieve-Ziele* eingeschränkt. Der Ansatz unterscheidet zwischen *Plan-Links* und *Goal-Links*, welche *potenziell* oder *definit* sein können. Es wird zudem eine Rangfolge zwischen den verschiedenen Link-Arten aufgestellt, welche dazu eingesetzt werden kann, verschiedene Agententypen umzusetzen oder je nach Zeit und Komplexitätsanforderungen umsichtiger oder weniger umsichtig vorzugehen. Wenn eine potenzielle Interferenz aufgedeckt wird, dient diese Information dazu, entsprechende *Scheduling-Methoden* für einen reibungslosen Ablauf anzustoßen.

Konflikte können aufgrund von beschränkten Ressourcen auftreten. Dabei lassen sich notwendige (*necessary*) und mögliche (*possible*) Ressourcen unterscheiden. Die einen werden zwangsläufig für das Erreichen eines Ziel benötigt, die anderen nur bei der Wahl eines bestimmten Plans - es gäbe jedoch auch noch alternative Handlungsoptionen (Thangarajah et al., 2003a). Ressourcen lassen sich in *consumable* und *reusable* Ressourcen einteilen. Um zu entscheiden, ob es zu einem Konflikt in Bezug auf Ressourcen kommt, müssen die Ressourcen der jeweiligen Teilziele summiert werden und es muss berücksichtigt werden, ob die Ziele parallel oder sequentiell ausgeführt werden können. Thangarajah et al. (2003b) erläutern, dass ihre Ansätze der Zielinteraktionsberücksichtigung in Architekturen wie JAM, dMars und JACK implementiert werden könnten, aber sich auch erweitern ließen, um effektivere Kooperationen zwischen mehreren Agenten zu realisieren.

### **Erweiterungen Pläne**

Pollack (1992) unterscheidet zwei Arten von Plänen. Während *Enablement-Pläne* selbst keine feste Rolle für das Erreichen des Ziels spielen, sondern vielmehr dazu dienen, die richtigen Umstände zu schaffen und Wege zu ebnen, sodass *Genera-*

*tion*-Pläne zum Einsatz kommen können, spielen die *Generation*-Pläne eine konkrete Rolle für das Hauptziel und tragen essentiell zu dem Gelingen des Ziels bei. Diese Unterscheidung deckt sich mit Arbeiten aus der Psychologie von Cooper & Shallice (2006). In ihrem Modell definieren sie kritische Handlungen (*critical actions*) dafür, eine unabdingbare Rolle für das Erreichen eines Ziels zu spielen und direkt zu der Ausführung des Ziels zu gehören. Andererseits lassen sich davon ermöglichende Handlungen (*enabling actions*) abgrenzen, welche dazu dienen, dass ein relevanter Plan anwendbar wird, für sich allein jedoch nicht direkt zu der Erfüllung des Ziels beiträgt. Pläne dienen dazu, Handlungen zu leiten, das Schlussfolgern zu kontrollieren und die Koordination zwischen den Agenten zu fördern. Damit eine Handlung auch wirklich eine Rolle in einem Plan spielt, muss der Agent die Handlung als einen Weg intendieren, um den Plan zu erreichen. Dazu gehört nicht, dass der Agent jede Konsequenz des Plans intendieren muss, es kann auch unerwünschte Nebeneffekte geben.

Eine weitere wichtige Erkenntnis Pollacks besteht darin, dass Pläne *partiell* sein können. Zu dem Zeitpunkt der Bildung eines Plans werden oft noch nicht die genaueren Details der Ausführung des Plans festgelegt. Dies geschieht erst später in Einklang mit der dynamischen Umwelt, in welcher der Agent situiert ist. Auch das Abstraktionsniveau des Plans kann sehr unterschiedlich ausfallen. Zumeist wird nur ein grober Plan gefasst, wenn die Ausführung des Plans noch in ferner Zukunft liegt. Rückt der Termin der Ausführung jedoch immer näher, so werden Planvariablen genauer bestimmt und an Werte gebunden.

Das Plankonzept von Pollack wurde in dem Ansatz der *SharedPlans* aufgegriffen. Die Planspezifikation spielt nicht nur im Rahmen der Berechnung von Ziel- bzw. Intentioneninteraktion eine Rolle, sie stellt auch eine fundamentale Basis für die *Planerkennung* dar.

### **Plan/Intentionserkennung**

Planerkennung wird im Allgemeinen in zwei Bereiche unterteilt (Blaylock, 2005). Im Falle der „Schlüsselloch“-Erkennung ist sich der Agent nicht dessen bewusst, dass seine Handlungen beobachtet und interpretiert werden. Die *intendierte Erkennung* hingegen zeichnet sich dadurch aus, dass sich der Agent der Interaktion bewusst ist und gezielt sein Verhalten daran ausrichten kann, seinem Gegenüber seine Absichten zu vermitteln. Planerkennung wird oft als Mittel für die Verbesserung von Benutzerschnittstellen vorgeschlagen und für den Einsatz verbesserter Hilfeleistung. Wenn ein Agent in einer kooperativen Situation seinem Gegenüber assistieren möchte, muss der Agent über die Ziele und Intentionen seines Gegenüber informiert sein. Ein weiterer Vorteil besteht darin, dass durch die Planerkennung auch die Menge der kommunikativen Handlungen reduziert werden kann (Lesh et al., 1999). Durch Kompetenzen der Planerkennung muss während

der Interaktion nicht alles explizit geäußert werden.

*Intentions-* bzw. *Zielerkennung* stellt eine besondere Form der Planerkennung dar und besteht aus der allgemeinen Aufgabe, die Ziele und Pläne eines Agenten aufgrund seines overten Verhaltens zu inferieren (Blaylock, 2005). Obwohl Zielerkennung ein nicht so informatives Verfahren wie die komplette Planerkennung aufzeigt, stellt es eine aktive Forschungsrichtung dar. Ein Grund dafür besteht darin, dass viele Anwendungen nicht auf komplette Planerkennungsergebnisse angewiesen sind. Ein weiterer Teilbereich der Intentionserkennung besteht in den Ansätzen des *Dialogverstehens*. Während frühere Arbeiten von Datenstrukturen ausgingen und damit allein das „*was wurde gesagt*“ ermittelt wurde, rückt mit dem Plan-basierten Ansatz das „*warum wurde etwas gesagt*“ in den Vordergrund. Die Ansätze, welche auf einfachen Datenstrukturen aufbauen, leiden allerdings unter dem Mangel der Verallgemeinerung (Lochbaum, 1995). Lochbaum fordert zudem, dass ein Algorithmus für die Modellierung von Planerkennung in einem Diskurs in der Lage sein muss, über hypothetische und nur teilweise spezifizierte Handlungen zu schlussfolgern.

Während die klassische Definition der Planerkennung sich auf komplette und korrekte Pläne bezieht (Kautz & Allen, 1986), gibt es im Bereich von Kooperationszenarien auch partielle Ansätze. Einen Ansatz stellt *Collagen* (siehe auch Abschnitt 2.3) von Lesh et al. (1999) dar. Dieser nutzt drei Eigenschaften einer Kooperationssituation aus, um Planerkennung bewältigbar zu machen. Zum einen wird der *Aufmerksamkeitsfokus* des Agenten dazu eingesetzt, die Suche nach Plan-kandidaten einzuschränken. Zum anderen wird die Verschachtelung von Plänen und Handeln ausgenutzt, indem die während der Interaktion partiell aufgestellten, hierarchischen Pläne inkrementell erweitert werden. Es ist damit nicht notwendig, einen kompletten, komplexen Plan zu erkennen, es geht vielmehr darum, *inkrementell* die Handlungen im entsprechenden Kontext zu interpretieren und einzuordnen. Ein dritter Vorteil besteht darin, dass durch das gegebene Interaktionsszenario Rückfragen ermöglicht werden, sodass der Agent in Zweifelsfällen um Aufklärung bitten kann.

*Collagens* Definitionssprache für Rezepte unterstützt partiell geordnete Planschritte, Parameter, Einschränkungen, Vor- und Kontextbedingungen und alternative Zieldekompositionen. Die Implementierung des Planerkennungsalgorithmus selbst baut in dem Ansatz von Lochbaum auf der Struktur der *Rezept-Graphs* (R-Graphs) auf. *R-Graphs* bestehen aus der Komposition von Rezepten und repräsentieren die Beliefs des Agenten darüber, wie die einzelnen Beiträge der Interaktion zusammenhängen. Während ein Rezept nur eine Ebene der Handlungsdekomposition enthält, kann ein R-Graph mehrere Level enthalten und dabei Informationen in Bezug auf gebundene Parameter, Akteure sowie zeitliche Einschränkungen beinhalten. Während die Rezept-Einschränkungen in der Plandefinition explizit festgelegt werden, werden in den R-Graphs zusätzlich Kontex-

teinschränkungen berücksichtigt, welche sich aus den Kontextparametern ableiten lassen. R-Graphs beinhalten Statusinformationen, welche für jede Aktion innerhalb des R-Graphs die Annahmen des Agenten über den Status dieser Aktion in Bezug auf den Gesamtplan kodieren.

Die Arbeiten von Balkanski & Hurault-Plantet (2000) bauen auf den Arbeiten von Lochbaum auf. Sie gehen jedoch über den Ansatz hinaus, indem sie zusätzlich in ihrem System eine Verschachtelung von Planung und Ausführung zulassen. Ihr Modell beinhaltet einen *Interpretation-Algorithmus*, welcher den R-Graph gründlicher durchsucht als der *R-Graph Augmentation*-Ansatz Lochbaums und einen *Task-Advancement-Algorithmus*, welcher das System mit Planungs- und Generierungsfähigkeiten ausstattet, welche in der ursprünglichen Version Lochbaums fehlen.

#### 4.2.7 Zusammenfassung der *top-down* Prozesse

Im Bereich der Psychologie wird von explizit repräsentierten Zielen ausgegangen, welche den Kognitions- und Deliberationsprozess des Agenten steuern. Die aktivierten Ziele und insbesondere die aktuellen Intentionen regen die Aktivierungswerte der entsprechenden *Chunks* im Arbeitsgedächtnis an. Im Rahmen philosophischer Modelle dagegen wird eine logische Analyse der Handlungsoptionen postuliert. Diese berücksichtigt dabei die Effekte, Kontext- und Vorbedingungen von Handlungsplänen und die Interaktion der Pläne untereinander. Während logische Operationen zu einer „alles oder nichts Entscheidung“ führen, üben Aktivierungsprozesse graduelle Einflüsse auf die Ziel- und Handlungsauswahl aus.

Nachdem sich aufgrund der Kontrollprozesse eine Ordnung unter den Handlungsoptionen eingestellt hat, muss im Rahmen komplexer Kognition eine Entscheidung getroffen werden, welches Ziel der Agent annimmt und welche Operation er letztendlich ausführt. Diese Aufgabe wird zumeist der *Central Executive* zugeschrieben. Die Strategie, aufgrund derer die Entscheidung getroffen wird, kann durch *Meta-Reasoning* beeinflusst werden. Zu den *top-down* Mechanismen, welche Ziel-orientiert eine Kontrolle ausüben und die die Entscheidungsfindung durch Aktivierung bzw. Bewertung vorantreiben, zählen die folgenden:

##### ***Top-down* Prozesse:**

- Deliberationsprozesse (Zielauswahl, Zielpräferenzen, Zielgewichtung)
  - Zielinteraktion, (Einschränkungen, Ausschlusskriterien)
  - Aktivierungsprozesse, Inhibition
  - Obligationen

- *Means-Ends-Reasoning* (Handlungsplanselektion)
  - Vorselektion anhand von Vorbedingungen, Attributen sowie Aktivierungswerten
  - Planinteraktion
- Plan/Intentionserkennung
- *Meta-Reasoning*

Möchte man die *top-down* Kontrollprozesse in einer Architektur berücksichtigen, entstehen Anforderungen an die Repräsentationsstrukturen der Ziele, Pläne und Intentionen. Für die Deliberationsprozesse wird eine Repräsentation des Zieltyps (*Achieve/Perform/Maintain/Query*), eine Repräsentation der involvierten Objekte/Agenten bzw. deren Attribute sowie eine Repräsentation möglicher Einschränkungsbedingungen (*Vorbedingung/Kontextbedingung*) benötigt. Sollen Aktivierungsprozesse der Ziele unterstützt werden, so benötigen die Ziele Verbindungen zu den jeweiligen Chunks im Arbeitsgedächtnis.

Der *Means-Ends-Reasoning* Prozess kann von der Repräsentation der Plantypen (*Enablement/Generation*) und der benötigten *Ressourcen* profitieren. Für Prozesse der Plan/Intentionserkennung stellt eine *partielle* Planrepräsentation einen entscheidenden Faktor dar. *Meta-Beliefs* können dazu eingesetzt werden, *Meta-Reasoning* Prozesse zu steuern und verschiedene *Commitment*-Strategien umzusetzen.

### 4.3 Existierende Architekturen

Es werden konkrete kognitive Architekturen und *BDI*-Systeme vorgestellt, welche sich auf die in Abschnitt 4.1 und Abschnitt 4.2 vorgestellten Modelle stützen. Die praktische Anwendbarkeit der Modelle wird beleuchtet. Hierbei ist insbesondere von Interesse, wie die einzelnen postulierten Mechanismen in einer Architektur zusammenspielen und ein intelligentes, konsistentes Gesamtverhalten erzeugen können.

	<i>Means-Ends-Reasoning</i>	Aktivierungsprozesse	Gedächtnismodell	Commitment
Soar	x			
Act-R		x	Baddeley	
Icarus	x	x		(x)
IDA		x	Baddeley	
ISAC	x		Baddeley	
BDI	x			x

#### 4.3.1 Kognitive Architekturen

##### Soar

*Soar* ist eine der ersten kognitiven Architekturen und ein Kandidat der „*Unified Theory of Cognition*“ (Newell, 1990). Sie wird mit großem Erfolg eingesetzt und weiterentwickelt. Deklarative Fakten werden in *Soar* als **Attribut-Wert-Listen** repräsentiert. Die *Soar*-Repräsentation ist dabei sehr offen und einfach. Für die domänen-spezifischen Inhalte dient das *Langzeitgedächtnis* als Speicher. *Soar* unterstützt drei verschiedene Arten des Langzeitgedächtnisses: *semantisches* Wissen kodiert in Form von deklarativen Strukturen, *prozedurales* Wissen kodiert als Produktionsregeln, und *episodisches* Wissen gespeichert in Form zusammenhängender Episoden (Nuxoll & Laird, 2007).

Der *Entscheidungszyklus* Soars ähnelt im Ansatz der Spezifikation des *BDI-Intepreters* (siehe *Algorithm 1* in Abschnitt 4.2.1). Ein entscheidender Unterschied besteht darin, dass es sich bei *Soar* um ein *Produktionssystem* handelt.

Die strukturellen Komponenten eines **Produktionssystems** lassen sich als Strukturtheorien des menschlichen Gedächtnisses interpretieren (Strube et al., 1996). Der Datenspeicher wird dabei als Pendant des menschlichen Arbeitsgedächtnisses gesehen, während der Produktionenspeicher als Form eines prozeduralen Langzeitgedächtnisses dient. Produktionen selbst bestehen aus Regeln der Form, „*wenn Bedingung, dann Aktion*“. Der Bedingungsteil spezifiziert eine Rei-

he von Tests, die mit dem aktuellen Inhalt des Datenspeichers verglichen werden. Sind sie erfüllt, können die in ihrem Aktionsteil angegebenen Operatoren wirksam werden. Ein Produktionssystem integriert grundlegende psychologische Vorstellungen über parallel und seriell ablaufende Vorgänge der Informationsverarbeitung. In Analogie zu Wahrnehmungsvorgängen verläuft der Auswertungsprozess parallel, während Entscheidungs- und Problemlösevorgänge seriell erfolgen.

Soar unterscheidet zwischen der Architekturebene und dem Einsatz in einer bestimmten Domäne (Lehman et al., 1996). So existieren viele Erweiterungen und Umsetzungen einzelner Modelle menschlicher Fähigkeiten (Rosenbloom et al., 1993). Die Elaborationsphase Soars ist so angelegt, dass der Agent sein sämtliches Wissen aktiviert und in seine Überlegungen einbezieht. Es werden dabei keine speziellen Aktivierungsmechanismen vorgesehen, welche für eine schnellere, kontextsensitive Verarbeitung sorgen könnten.

In Soar stellt ein Operator einen abstrakten Typ dar, welcher durch eine Menge an Regeln implementiert wird. Der Einsatz expliziter **Präferenzen** in Bezug auf die Operatorauswahl erlaubt es, zunächst sämtliche Evidenzen für potentielle Veränderungen des Zustands zu sammeln, bevor eine Entscheidung getroffen wird. Die Präferenzen bestehen aus der Relation „*ist besser als*“ oder aus numerischen Werten. In der *Anwendungsphase* feuert jeweils ein einziger Operator pro Entscheidungszyklus. Dies stellt einen bewusst gewählten Flaschenhals im Rahmen der kognitiven Verarbeitung dar (Jones et al., 2007).

Das modulare **Perception-Motor-Interface** stellt Wahrnehmungsmechanismen zur Verfügung und vermittelt die Ausführung motorischer Handlungen. Es operiert *parallel* zu dem kognitiven Produktionssystem. In Bezug auf die Ausführung paralleler Handlungen innerhalb Soars besteht nur die Möglichkeit, mehrere Aktionen in einem Operator zusammenzufassen. Es existiert jedoch keine allgemeine Form dafür, Handlungen gleichzeitig zu verfolgen.

Die Integration und das automatische Anstoßen von **Lernmechanismen** (*Chunking, Reinforcement Lernen* (Nason & Laird, 2005)) in die Deliberationsprozesse des Agenten stellen einen zentralen Punkt der Soar-Architektur dar.

### **Act-R**

*Act-R* wird als die bisher erfolgreichste kognitive Architektur angesehen (Sun, 2004), welche seit Jahren in einer großen Forschungsgemeinschaft weiter entwickelt wird. Sie wurde bereits eingesetzt, um eine große Bandbreite menschlicher Daten, welche von dem Erlernen von Fertigkeiten bis hin zur Sprachproduktion reichen, erfolgreich zu modellieren und quantitative Messungen von Problemlösungsprozessen vorzunehmen.

Die Act-R Theorie fokussiert historisch gesehen auf höhere Kognition und geht von einem **ganzheitlichen Ansatz** aus (Anderson & Lebiere, 1998; Ander-

son, 1993). Dabei dürfen einzelne Module nicht isoliert betrachtet werden, sondern das Gesamtsystem und das Zusammenspiel der Module müssen im Blick bleiben (Anderson et al., 2004).

Konkret umfasst *Act-R* fünf Module (siehe Abbildung 4.5 nach Anderson et al. (2004)): ein *visuelles Modul*, welches die Ergebnisse komplexer Berechnungen in einem Buffer zugreifbar macht, ein *manuelles Modul*, welches die Hände kontrolliert, ein *deklaratives Modul*, welches Informationen aus den Gedächtnissen abrufen und in einem Buffer zur Verfügung stellt, sowie ein *Ziel-Modul*, welches dafür sorgt, zielbezogene Gedankenstränge in Abwesenheit externer Stimuli aufrechtzuerhalten. Die Koordination der Module wird durch ein zentrales **Produktionssystem** erreicht. Die Produktionsregeln innerhalb des Systems enkodieren kristallisierte Problemlösungsoperatoren als Bedingungs-Aktions-Regeln (Anderson, 2001). Das Auswahlkriterium der Regeln basiert auf der Berechnung eines Nützlichkeitswerts (**Utility**).

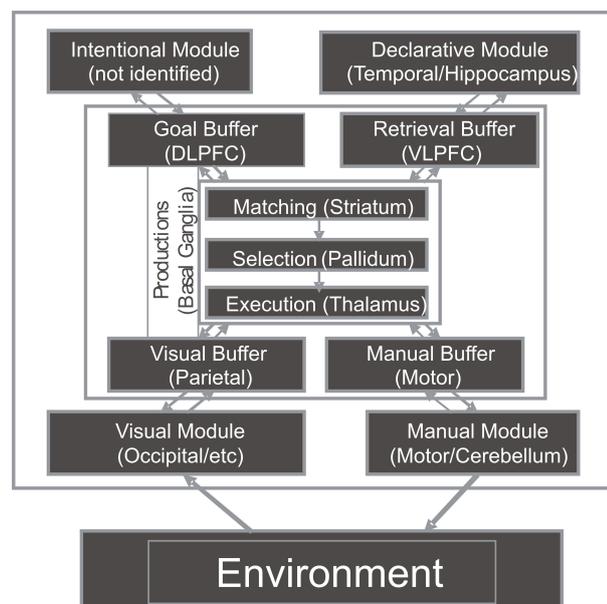


Abbildung 4.5: Informationsverarbeitung in Act-R 5.0. nach (Anderson et al., 2004)

Das Produktionssystem hat lediglich Zugriff auf die Buffer der Module. Dadurch ist die Information in den Buffern gekapselt und die Module kommunizieren nur über die Informationen, die sie den anderen Modulen in ihren Buffern zur Verfügung stellen. Allerdings betonen Anderson et al. (2004), dass es Anhaltspunkte dafür gibt, dass auch direkte Modul-zu-Modul Verbindungen bestehen, welche nicht durch das Produktionssystem gesteuert werden. Die Buffer

verfügen über eine große Ähnlichkeit zu den *Slave*-Systemen aus Baddeleys Arbeitsgedächtnismodell (siehe Abschnitt 4.1.3).

Act-R ist ein Hybridsystem, welches **symbolische** und **subsymbolische** Aspekte in einem Modell vereint. Wissen wird symbolisch repräsentiert, während die Prozesse, die auf dem Wissen operieren, auf einem subsymbolischen Level agieren. Das Act-R-Modell geht von einer Mischung aus *paralleler* sowie *serieller* Verarbeitung aus und operiert dabei *zyklusbasiert*. Das Produktionssystem wird dazu eingesetzt, Muster in den Buffern zu erkennen und darauf basierend Produktionen zu feuern. Innerhalb der Module operieren Prozesse parallel und asynchron. Es existieren jedoch zwei Engpässe. Die Buffer der Module können nur jeweils einen *Chunk* halten und in jedem Zyklus wird nur eine einzige Produktionsregel gefeuert. Dies steht in einem Kontrast zu *Soar* (siehe Abschnitt 4.3.1).

Die subsymbolische Verarbeitung in der Act-R Architektur bietet eine gute Grundlage für die Umsetzung der dynamisch, kontextsensitiven Verarbeitung, die als Anforderung an einen situierten Agenten spezifiziert wurde. Betrachtet man die postulierten Basismechanismen, sind in Act-R sowohl Formen von **Priming**-Mechanismen, die Propagierung von **Zielaktivierung** sowie **Spreading-activation**-Mechanismen vorgesehen. Die Basisaktivierung eines Chunks reflektiert die Zugriffsmöglichkeit und den allgemeinen Nützlichkeitsgrad unabhängig vom Kontext und bestimmt sich anhand der Anzahl der Zugriffe ihrer zeitlichen Abständen. Dabei stellen die  $t_j$  die Zeitspannen dar, welche nach der  $j$ -ten Anwendung des Elements vergangen sind (Anderson et al., 2004).

$$B_i = \ln\left(\sum_j t_j^{-d}\right) \quad (4.12)$$

Von dem aktuellen Ziel geht eine beschränkte Aktivierungsmenge aus (*source activation*), welche zwischen den einzelnen deklarativen Knoten des Zielchunks aufgeteilt wird. Dabei stellt  $W_j$  die Aktivierung ausgehend vom Ziel  $j$  dar und  $S_{ji}$  die Stärke der Verbindung zwischen Ziel und Chunk.

$$A_i = B_i + \sum_j w_j \cdot S_{ji} \quad (4.13)$$

Dies führt dazu, dass *relative* Aktivierung wichtiger ist als die *absolute* Aktivierung.

Ein Kritikpunkt an Act-R stellt die fehlende Strukturierung in verschiedene Gedächtnisformen dar. Eine weitere Schwäche besteht in Bezug auf die Repräsentation und Berücksichtigung zeitlicher Aspekte. Hierfür sind in der Act-R Architektur keine direkten Mechanismen vorgesehen (Muller et al., 2008).

## Icarus

Die *Icarus*-Architektur beruht in besonderem Masse auf Einflüssen der kognitiven Psychologie (Langley & Choi, 2006). Das Ziel der Architektur besteht darin, das menschliche Verhalten nicht quantitativ sondern qualitativ zu reproduzieren. Dabei folgt das Design von Icarus folgenden Prinzipien Intelligenter Systeme (Langley, 2006):

- Kognition ist in Perzeption und Handlung verankert.
- Konzepte und Fertigkeiten stellen verschiedene kognitive Strukturen dar.
- Die Strukturierung des Langzeitgedächtnisses ist hierarchisch.
- Symbolische, kognitive Strukturen werden durch numerische Funktionen moduliert.

Icarus unterhält einen **perzeptuellen Buffer**. Jedes Perzept in diesem kurzlebigen Speicher korrespondiert mit einem bestimmten Element der Umwelt. Es werden der *Typ* des Objekts, ein eindeutiger *Name* und eine Menge an *Attribut-Wert*-Paaren kodiert, welche das Objekt zu dem Zeitpunkt beschreiben (Langley & Choi, 2006).

Icarus zeichnet sich insbesondere durch die Unterscheidung von konzeptuellem und prozeduralem Wissen aus. Einer der Grundmechanismen in der Icarus-Architektur besteht aus der **konzeptuellen Inferenz**, welche gespeicherte Konzepte des Langzeitgedächtnisses mit wahrgenommenen Perzepten und Annahmen vergleicht und die Ergebnisse in das *konzeptuelle Kurzzeitgedächtnis (Beliefs)* einträgt. Dieses enthält dadurch alle gültigen Inferenzen über die aktuelle Situation des Agenten. Das Inferenzmodul operiert auf eine *bottom-up*, Daten-getriebene Art und Weise. Während Perzepte Attribute bestimmter Objekte beschreiben, spezifizieren *Beliefs* die Relationen zwischen Objekten. Jedes Belief besteht dabei aus einem Prädikat und einer Menge symbolischer Argumente, welche sich auf Objekte beziehen (Langley & Choi, 2006). Die Operation des Inferenzmoduls erinnert dabei an den Elaborationsprozess der *Soar*-Architektur.

Das **konzeptuelle Langzeitgedächtnis**, aus welchen der Inferenzmechanismus sein Wissen bezieht, kodiert Konzepte in Form von booleschen Strukturen. Die boolesche Repräsentation des konzeptuellen Langzeitgedächtnisses stellt jedoch eine starke Vereinfachung dar. Bei Menschen besitzen Kategorien zumeist einen graduellen Charakter. Durch die Berücksichtigung einer numerischen Erwartungsfunktion wird der graduelle Charakter menschlicher Prozesse berücksichtigt. Die Funktion wird bei der Berechnung der *Utility*-Werte der Fertigkeiten im Rahmen des Entscheidungsfindungsprozesses eingesetzt. Eine **Fertigkeit (skill)**

besteht aus den folgenden Feldern: *start* (precondition), *requires* (context condition), *ordered/unordered* (subskills), *actions* (primitive skills), *effects*. Betrachtet man nur die *start*- und *effects*-Felder so erinnert die Beschreibung an einen STRIPS-Operator (Fikes & Nilsson, 1971). Die Fertigkeiten des Agenten liegen in einem prozeduralen Langzeitgedächtnis vor.

Icarus operiert in klaren **kognitiven Zyklen**. In jedem Zyklus wird zunächst der perzeptuelle Buffer aktualisiert, dann wird mittels konzeptueller Inferenz im Kurzzeitgedächtnis überprüft, ob weitere Informationen ableitbar sind. Die *Beliefs* enthalten damit nur solche Konzeptinstanzen, welche durch wahrgenommene Elemente untermauert werden. In einem weiteren Schritt werden im prozeduralen Kurzzeitgedächtnis alle anwendbaren Dekompositionen der Fertigkeiten ausgewählt. Es werden alle möglichen Variablenbindungen der Fertigkeiten gebildet und dann zwischen allen Optionen die Fertigkeiten mit den höchsten Nützlichkeitswerten ausgewählt. Für jeden kompletten Pfad berechnet die Architektur den erwarteten Gesamtwert der Nützlichkeitswerte und wählt dann den Kandidaten mit dem höchsten Wert für die Ausführung. Die ausgewählte Fertigkeit wird in einem Gedächtnis vorgehalten und im nächsten Schritt in die Planung einbezogen. In Bezug auf das *Commitment*-Verhalten des Agenten bestimmt ein **Persistenzparameter**, wie stark der Agent bereits vorher geplante Schritte in seine aktuelle Entscheidungsfindung einbezieht (Choi et al., 2004).

Für die Zielmodellierung wird ein **Ziel-Stack** eingesetzt. Ein Ziel wird dabei immer als ein Prädikat des konzeptuellen Gedächtnisses mit den zugehörigen Argumenten repräsentiert. Sowohl Ziele als auch Fertigkeiten müssen immer in den jeweiligen Gedächtnissen grounded sein. Diese theoretische Position steht in Kontrast zu Soar und Act-R, welche deutlich schwächere Abhängigkeiten geltend machen (Langley, 2006). Problemlösen wird als dynamische Komposition bekannter Fertigkeiten in Pläne und deren Ausführung verstanden. Diese Strategie des Problemlösens stellt eine Version der *Means-Ends-Analysis* (Newell & Simon, 1961) dar. Sie unterscheidet sich jedoch von der Standardform dadurch, dass Planung und Ausführung stark integriert sind (Langley & Choi, 2006), was sich auch bei Menschen beobachten lässt (Gunzelmann & Anderson, 2003).

Während Icarus im Bereich der Berücksichtigung des physikalischen Kontextes und in Bezug auf hierarchische Gedächtnisstrukturen eigene Wege geht, decken sich die Annahmen über ein vom Langzeitgedächtnis zu unterscheidendes dynamisches Kurzzeitgedächtnis, dessen Inhalt aus symbolischen Listenstrukturen besteht, mit den Annahmen von Act-R und Soar.

## IDA

IDA steht für *Intelligent Distribution Agent* und beschreibt ein komplexes Modell der **Global Workspace Theory** (Baars, 2002). Diese Theorie geht da-

von aus, dass das Gehirn aus einer Kollektion verteilter, spezialisierter Netzwerke besteht. Diese versuchen Informationen in das Bewusstsein (*global workspace*) eines Agenten zu rücken. Im *Global Workspace* werden konkurrierende sowie kooperierende Netzwerke integriert. Die Netzwerk-Kontexte beeinflussen die bewussten Inhalte und schränken diese ein. Dabei zählen auch Motivationen und Emotionen zu den Kontexten. Die *Global Workspace*-Theorie geht ferner davon aus, dass ausführbare Funktionen als hierarchische Strukturen für Zielkontexte operieren. Bei dem *IDA*-System wird ein Schwerpunkt auf die Interaktion zwischen Bewusstsein und Arbeitsgedächtnis gelegt (Baars & Franklin, 2003). Dabei bauen die Arbeiten auf dem Arbeitsgedächtnis-Modell von Baddeley auf (siehe Abschnitt 4.1.3).

**Codelets** implementieren die unbewussten Mechanismen der *Global Workspace*-Theorie. Sie verfügen über eine große Variabilität und ihre Konzeption orientiert sich an Minskys Agenten (Minsky, 1985). Sie bestehen aus einem einfachen Programm, welches eine bestimmte Aufgabe erfüllt und bei Bedarf aktiv wird. Kognition wird als kontinuierlicher Fluss von *kognitiven Zyklen* verstanden, welche überlappen können und dadurch so erscheinen, als operierten sie parallel. Baars & Franklin (2003) beschreiben den kognitiven Zyklus in neun Schritten. Die Schritte vier bis neun sehen sie als eine Implementation der *Central Executive*. Der erste Schritt des Zyklus besteht aus der *Perzeption*; dabei werden sensorische Stimuli, welche sowohl internen als auch externen Ursprungs sein können, wahrgenommen und interpretiert. Dieser Prozess geschieht unbewusst. Unterschieden wird zwischen der **Early-Perception**, bei welcher eintreffende Sensordaten von spezialisierten *Codelets* verarbeitet werden, und der *Chunk-Perzeption*, bei welcher die Aktivierung der einzelnen Informationsknoten sich stabilisiert und die bedeutungstragenden *Chunks* in größeren *Chunks* zusammengeführt werden (**Chunking**).

Im zweiten Schritt wird das *Perzept* zusammen mit seiner Bedeutung und Teilen der dazugehörigen Daten in einem *pre-conscious*-Buffer abgelegt. Zu diesen Buffern gehören *räumlich-bildliche* und *phonologische* Aspekte. Die relevantesten *Perzepte* werden in einem nächsten Schritt zusammen mit den Assoziationen, die automatisch abgeleitet werden, in den *Episodic Buffer* (*long-term working memory*) eingetragen. An dieser Stelle setzt die *Central Executive* ein. **Attention Codelets** versuchen, Ereignisse unter Einbezug von Langzeitwissen in das Bewusstsein des Agenten zu rücken. *Codelets*, die nicht genügend Aktivierungspotential erhalten, geraten schnell wieder in Vergessenheit. Die Inhalte bleiben jedoch noch eine zeitlang in unbewussten Buffern erhalten und können das Verhalten des Agenten z.B. durch *Priming*-Mechanismen beeinflussen.

Im nächsten Schritt erhält das *Attention Codelet* mit der höchsten Aktivierung Zugriff auf den *Global Workspace* und kann seine Inhalte bekannt geben. Diese Daten werden auch in dem *Episodic Buffer* gespeichert. Die Inhalte des episodischen

schen Gedächtnisses werden in regelmäßigen Abständen in das autobiographische Langzeitgedächtnis übertragen.

In einem nächsten Schritt reagieren relevante *Codelets* auf die Bekanntmachung der Inhalte und steuern das Bewusstsein des Agenten. Die Verhaltens-Codelets binden Variablen, senden Aktivierungssignale an weitere *Codelets* und setzen den **Zielkontext**. Wenn keine passenden *Codelets* vorliegen, wird nicht-routiniertes Problemlösen durch zusätzliche Mechanismen angeregt. Es wird ein Zielverhalten ausgewählt und ausgeführt. Dabei wird die Entscheidung durch die interne Motivation (ausgehende Aktivierung von Zielen), die aktuelle Situation, externe und interne Bedingungen sowie durch die Relationen und die aktuellen Aktivierungswerte der einzelnen Verhaltensweisen geprägt. Die Aktion wird ausgeführt. Sie kann sowohl externe als auch interne Konsequenzen zur Folge haben. Es können auch explizite Erwartungs-*Codelets* agieren, welche die Ausführung von Aktionen überwachen und das Bewusstsein des Agenten auf Fehlverhalten und Störungen lenken.

Das konzeptionelle IDA-Modell ist sehr komplex und sieht ein breites Spektrum an Einflüssen, eine komplexe Gedächtnisstruktur und vielschichtige Interaktionen vor. Allerdings besteht eine Diskrepanz zwischen dem IDA-Modell und der IDA-Implementation eines Softwareagenten. Nicht alles, was in der Konzeption angedacht ist, wird bisher auch in der Implementation berücksichtigt (Franklin & Ferkin, 2006). LIDA stellt eine Erweiterung des IDA-Modells in Bezug auf Lernmechanismen dar (Franklin et al., 2007).

## ISAC

ISAC (*Intelligent Soft Arm Control*) wird an dem „Center for Intelligent Systems“ der Vanderbilt Universität entwickelt und stellt ein Kontrollsystem für einen humanoiden Roboter dar. Es wird eine Methode der modularen Kontrolle vorgestellt, welche ein biologisch inspiriertes Gedächtnismodell einsetzt, um relevante Informationen während der Aufgabenausführung vorzuhalten. Das Modell des Arbeitsgedächtnis greift Ansätze des *Multi-Component Modells* (siehe Abschnitt 4.1.3) auf. Informationsverarbeitung in ISAC ist in eine Multi-Agenten-basierte Softwarearchitektur eingebettet: IMA (*Intelligent Machine Architecture*). ISAC's kognitive Fähigkeiten werden durch eine Sammlung von IMA-Agenten und Gedächtnisstrukturen sowie einen wichtigen Verbindungsagenten dem *Self-Agent*, umgesetzt.

Eine komplexe Gedächtnisstruktur wird dazu eingesetzt, Informationen für die aktuelle Aufgabe bereitzuhalten. Zu diesen Informationen zählen auch eigene Erfahrungen, welche im Entscheidungsprozess eine wichtige Rolle spielen können. Das Kurzzeitgedächtnis speichert sensorische Informationen in einer Struktur, welche sich *Sensory Ego Sphere* nennt. Die gespeicherten Informationen

verfallen mit der Zeit. Das Langzeitgedächtnis speichert Informationen, welche gelernte Fertigkeiten und semantisches Wissen umfassen. Das Arbeitsgedächtnis der ISAC-Architektur ist aus zwei funktional verschiedenen Teilen konzipiert: einer Menge an Arbeitsgedächtnis-Slots, welche von anderen Agenten eingesehen werden können und einem Modul, welches Gedächtnis-Chunks je nach Kontext auswählt und in den Slots platziert. Das Arbeitsgedächtnis arbeitet mit einer Komponente des *Self Agent*, dem *Central Executive Agent* zusammen, um die kognitive Kontrolle umzusetzen, so wie sie beim Menschen zu beobachten ist.

Der *Central Executive Agent* wählt eine kleine Anzahl an Verhaltensweisen aus und lädt sie anhand ihrer Gewichtung in das Arbeitsgedächtnis. Die Entscheidungsgrundlage basiert dabei auf der Relevanz der einzelnen Verhaltensweisen. Bisher basiert die Relevanzberechnung in ISAC jedoch auf einer einfachen Tabelle, in welcher die Nützlichkeit der einzelnen Verhaltensweisen abgeschätzt ist.

Für die Aufgabenbearbeitung sollen der interne Zustand des Roboters sowie seine früheren Erfahrungen, Emotionen und Gefühle berücksichtigt werden. Emotionale Saliens dient sowohl als Belohnungssignal als auch als Filter für eingehende Informationen der komplexen Umwelt. Durch *Reinforcement*-Lernen sowie *Temporal Difference Learning* soll das System zudem in die Lage versetzt werden, Verhaltensweisen aufgrund vorheriger Erfahrung auszuwählen. Damit soll der Berechnungsaufwand sowie die Berechnungskomplexität vereinfacht werden. Dieser Ansatz lässt sich sowohl für routinierte Aufgaben als auch für neue Aufgaben einsetzen. Bisher beziehen sich alle Verhaltensweisen des ISAC-Systems auf Bewegungen des Arms. In Zukunft soll das gesteuerte Verhalten auf weitere Aktuatoren wie den Kopf und die Hände ausgeweitet werden.

Der Ansatz, kognitive Kontrolle unter Berücksichtigung eines dynamischen Arbeitsgedächtnismodells für einen Roboter nutzbar zu machen, erscheint als vielversprechendes Konzept. Während sich die anderen Systeme auf Prozesse der komplexen Deliberation und bewusstes Problemlösen konzentrieren, geht es in der ISAC-Architektur um die Bewegungskontrolle eines Arms. Dennoch lassen sich einige der Konzeptionen auf andere Systeme übertragen. Die konkrete Umsetzung der einzelnen Komponenten ist bisher wenig komplex, sodass sich teilweise sehr einfache Berechnungen hinter kompliziert anmutenden und konzeptionell aufgeladenen Begriffen verbergen. Interessant ist an der ISAC-Architektur insbesondere die Berücksichtigung emotionaler Aspekte.

### 4.3.2 BDI-Systeme

Im Bereich der BDI-Modellierung muss zwischen der zugrunde liegenden Struktur, den BDI-Logiken und den BDI-Implementationen unterschieden werden. In den Abschnitten über mentale Repräsentationen sind die Grundlagen und Basis-

struktur der BDI-Systeme beschrieben worden (siehe Abschnitt 4.2.1). Hier werden konkrete Implementierungen des Ansatzes vorgestellt. Während IRMA (Bratman et al., 1991) die erste Spezifikation einer BDI-Architektur darstellt, ist das *Procedural Reasoning System* (PRS) (Georgeff & Lansky, 1987) die erste und bekannteste Implementierung des BDI-Ansatzes.

**UM-PRS** (Lee et al., 1994) baut auf PRS auf und setzt eine Mehrzahl der Repräsentationen und der verhaltensbasierten Konzepte der originalen PRS-Spezifikation um. Es wurde zudem um strukturierte Repräsentationen prozeduralen Wissens erweitert. Allerdings bestehen in UM-PRS Schwächen im Bereich der Konzeptualisierung von Zielen sowie wichtiger prozeduraler Konstrukte, wie beispielsweise die Möglichkeit der parallelen Ausführung (Huber, 1999). **JAM** stellt eine Weiterentwicklung von UM-PRS dar und leidet nicht unter diesen Einschränkungen. Es unterstützt homöostatische Ziele, eine eindeutige Zielerreichungs-Syntax und Semantik und eine große Bandbreite an prozeduralen Konstrukten (Huber, 1999). JAM stellt eine Hybrid-Architektur für Intelligente Agenten dar, welche neben PRS auch auf den Ideen des *Structured Circuit Semantics* (Lee et al., 1994) sowie des *Act plan interlingua* (Myers & Wilkins, 1997; Wilkins & Myers, 1995) aufbaut.

JAM stellt komplexe Planrepräsentationen zur Verfügung und unterstützt multiple, simultane Ziele sowie Ziel- und Daten-getriebenes Verhalten. Es bietet Strukturen für den Einsatz von *Metalevel Reasoning* sowie von *Utility*-basiertem *Reasoning*. Der JAM-Interpreter zieht dabei alle möglichen Kombinationen von Plänen, Zielen und Variablenbelegungen in Betracht, bevor er sich für die beste Alternative in der gegebenen Situation entscheidet. Weder die originale PRS-Spezifikation noch andere PRS-basierte Implementierungen unterstützen bisher **Utility**-basiertes *Reasoning*. Jeder Plan in JAM kann eine explizite oder implizite Definition einer *Utility*-Berechnung enthalten, welche im Entscheidungsprozess des Interpreters Berücksichtigung findet.

Es existiert eine deklarative *Observer*-Prozedur, die der Agent zwischen der Ausführung der Planschritte aufruft und die es ermöglicht, Aktionen außerhalb des Ziel/Plan basiertem Reasoning des Agenten auszuführen (beispielsweise die Abarbeitung eingegangener Nachrichten). JAM stellt Konstrukte für die simultane Ausführung von **parallelen Aktivitäten** und ein synchronisierendes *Waiting*-Konstrukt. Es besitzt jedoch keinen allgemeinen Mechanismus für die parallele Ausführung von beliebigen Plänen. Die parallele Ausführung von Plänen muss explizit deklariert und angestoßen werden. Es existieren keine Mechanismen, die eine Interaktion zwischen den parallel ausgeführten Plänen überwachen. Auch wenn in JAM die Möglichkeit vorgesehen ist, auf einen Zeitwert zuzugreifen, ist eine Berücksichtigung zeitlicher Aspekte in der Architektur bisher nicht vorgesehen. JAM stellt laut (Huber, 1999) bisher keine komplette Architektur dar, da noch einige architektonische Konstrukte fehlen, wie beispielsweise die Fähigkeit

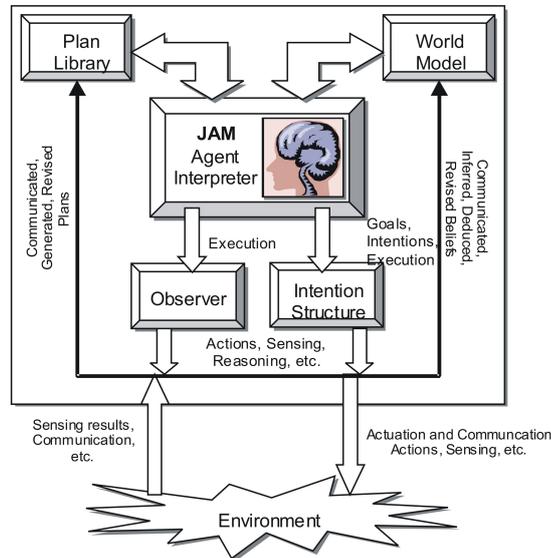


Figure 2. The JAM intelligent agent architecture.

Abbildung 4.6: JAM-Architektur (Huber, 1999)

der Plangenerierung oder des Lernens. Der Autor betont jedoch, dass Erweiterungen geplant sind und in die Architektur integriert werden sollen.

Während JAM im Bereich der Forschung eingesetzt wird, stellt das **JACK Intelligent Agent Framework** (Howden et al., 2001), entwickelt von *Agent Oriented Software*, eine kommerzielle BDI-Implementation dar, welche das Konzept intelligenter Agenten in den Mainstream-Bereich der kommerziellen Softwareentwicklung bringen soll. Der Schwerpunkt liegt auf Erweiterbarkeit und Kompatibilität. Die Architektur ist dabei so offen wie möglich, sodass verschiedene Ansätze und *Plugins* ausprobiert und umgesetzt werden können. In Bezug auf die Realisierung von Teamfähigkeit unterstützt JACK zwei Plugins. Das eine Plugin ist für die BDI-Reasoning Fähigkeiten verantwortlich, das andere dient der Umsetzung von Teamkompetenzen.

**Jadex** ist eine weitere BDI-Architektur, mit der aktuelle Forschungsfragen untersucht werden. Die Architektur von Pokahr et al. (2005b) ähnelt im Prinzip den traditionellen PRS-Systemen, versucht jedoch Einschränkungen der BDI-Systeme zu überwinden, indem sie eine **explizite Zielrepräsentation** unterstützt. Dadurch wird ein Deliberationsalgorithmus ermöglicht, der Zielinteraktionen berücksichtigt. Dieser ist sehr einfach aufgebaut, um in polynomialer Rechenzeit berechenbar zu sein (Pokahr et al., 2005a). Es werden ausschließlich negative Interaktionen zwischen Zielen berücksichtigt, welche durch bilaterale Links repräsen-

tiert werden. Diese müssen von dem Agenten-Entwickler explizit bei der Ziel-Deklaration angegeben werden und bestehen aus *Kardinalitäts-* sowie *Inhibitionseinschränkungen*. Durch diese Einschränkungsbedingungen kann spezifiziert werden, dass zu einem Zeitpunkt nur ein einziges Ziel eines bestimmten Typs existieren darf bzw. dass ein Ziel ein anderes Ziel durch ein Inhibitions-Link daran hindert, aktiv zu werden. Die Autoren wählen einen lokalen Ansatz inhibierender Links anstelle eines globalen *Utility*-Ansatzes, um den Agenten-Entwickler davon zu befreien, Ziele in eine globale Reihenfolge zu bringen. Einerseits vereinfacht dieser Ansatz die Agenten-Entwicklung, andererseits skaliert dieser Ansatz jedoch nicht, wenn neue Pläne zu den Handlungsfähigkeiten des Agenten hinzukommen. Jeder mögliche Konflikt muss von dem Entwickler im Vorhinein antizipiert werden. Die Agenten sind damit nicht in der Lage, Konflikte kontextspezifisch zu erkennen.

Um zwischen angenommenen und aktiv verfolgten Zielen unterscheiden zu können, wurde in der Jadex-Architektur ein **Lebenszyklus** für Ziele eingeführt, welcher den Zielen die Zustände *option*, *active* und *suspended* zubilligt. In Bezug auf den Lebenszyklus eines Ziels können Erzeugung (*create*), Aufgabe (*drop*) und Kontextbedingungen in Form von booleschen Ausdrücken spezifiziert werden. Bei Verletzung einer Kontextbedingung wird die Ausführung des Plans so lange unterbrochen, wie die Bedingung nicht erfüllt ist.

Jadex setzt den Gruppierungsmechanismus der **Capabilities** (Busetta et al., 2000) ein, um die Strukturierung von Wissensbereichen zu unterstützen (siehe auch Abschnitt 4.1.6). Die Verarbeitung von Zielen während der Laufzeit kann durch verschiedene BDI-Flags verfeinert werden: (*Retry*, falls ein Plan scheitert, *Meta-level-Reasoning*, sequentielle/parallele Verarbeitung von Plänen)

Ein weiterer Ansatz, um BDI-Architekturen flexibler zu gestalten, stammt von Chalmers & Gray (2001). Sie schlagen den Einsatz von *Constraints-Logic-Programming*-Techniken vor, um Pläne zunächst auf einer abstrakten Ebene spezifizieren und die genauen Details dann zur Laufzeit festlegen zu können. Durch den Einsatz der *Constraints* sollen alle relevanten Informationen verschiedenen Ursprungs für den Agenten zugreifbar werden, ihm aber genug Autonomie bleiben, um seine Entscheidungen auf Basis der aktuell vorliegenden *Beliefs* und des aktuellen Zustands der Welt zu treffen. Traditionsgemäß wurden *Constraints* als Repräsentation für die Restriktion von Datenfakten eingesetzt. Chalmers und Gray nutzen die deklarative Natur der *Constraints* im Rahmen der BDI-Architektur, um *Desires* als komplexe quantifizierte *Constraints* auszudrücken. Der große Vorteil von *Constraints* besteht darin, dass sie ein einheitliches Repräsentationsformat für Umweltbedingungen, Entscheidungen und Verpflichtungen haben, welches jederzeit dynamisch angepasst werden kann. Der *Constraint-Mechanismus* des Systems von Chalmers und Gray bildet *Beliefs* auf endliche Wertemengen im Problemraum ab und setzt dann *Desires* und Intentionen als Einschränkungen

im Rahmen des *Constraints-Solving* Prozesses ein, um den bestmöglichen Plan auszuwählen und auszuführen. Auch wenn der *Constraints-Logic-Programming*-Ansatz keinen optimalen Planungsmechanismus zur Verfügung stellt, so glauben Chalmers und Gray dennoch, dass er dem Agenten die Möglichkeit bietet, in einer Umwelt der Unsicherheit und ständigen Veränderung planen und erfolgreich agieren zu können (Chalmers & Gray, 2001).

Die vorgestellten Systeme JAM und Jadex ähneln sich in ihrer Umsetzung der Kernkonzepte des BDI-Modells. Sie unterscheiden sich jedoch in ihrer Schwerpunktsetzung. Während Jadex eine komplexere Modellierung von Zielen und dem damit verbundenen Commitmentverhalten vornimmt, besteht in JAM die Besonderheit in der Berücksichtigung von *Utility*-Werten. Damit leisten beide Implementationen einen Beitrag in Bezug auf Erweiterungskonzepte für den BDI-Ansatz. Die komplexere, explizite Zielmodellierung stellt eine Basiskompetenz für die Ausführung und Überwachung paralleler Handlungsstränge dar, unabhängig davon, ob die Handlungen von einem Agenten oder von mehreren ausgeführt werden.

Auch wenn die Spezifikation der *Utility*-Funktionen in JAM sehr frei ist und noch keine konkreten relevanten Faktoren bzw. Mechanismen vorgesehen sind, welche beispielsweise Aspekte der Zielinteraktion oder Aktivierungsprozesse in die *Utility*-Berechnung einbeziehen, so stellt die Integration einer *Utility*-Funktion in die Architektur einen ersten und wichtigen Schritt in die Richtung subsymbolischer Berechnungsprozesse dar.

## 4.4 Diskussion

Betrachtet man, inwieweit kognitive Basismechanismen Eingang in kognitive Architekturen gefunden haben, so wird deutlich, dass zwar einige der Architekturen **subsymbologische** Einflussfaktoren berücksichtigen, jedoch keine der Architekturen alle Faktoren berücksichtigt. Act-R unterstützt als einzige Architektur Aktivierungsprozesse sowohl in Bezug auf deklaratives als auch prozedurales Wissen. Eine Berücksichtigung emotionaler Prozesse ist jedoch bisher in der Grundarchitektur nicht vorgesehen, wenngleich verschiedene Arbeiten dazu existieren, Emotionen in die Act-R Architektur zu integrieren (Cochran et al., 2006). Auch die *Icarus*-Architektur bietet interessante Ansätze in Bezug auf *bottom-up* Aktivierungsprozesse, jedoch fehlt bisher die Berücksichtigung von *top-down* Aktivierungsmechanismen.

Die **Deliberationsprozesse** der vorgestellten Produktionssysteme unterscheiden sich deutlich. Während in Soar die Auswahl der Regeln allein anhand der spezifizierten Vorbedingungen sowie der explizit repräsentierten Präferenzen erfolgt, folgt Act-R ausschließlich dem aktivierungsbasierten Ansatz. Für die situa-

tionssensitive kooperative Interaktion sollten jedoch beide Aspekte berücksichtigt werden. Auch in Bezug auf die **Situietheit** herrscht ein zentraler Unterschied zwischen der Act-R und der Soar-Architektur. Während Soar einen *rationalen Ansatz* verfolgt, in welchem so lange Informationen abgerufen werden und Regeln feuern, bis alle Informationen, über die der Agent verfügt, berücksichtigt wurden, spielen in Act-R die aktuellen Aktivierungswerte bei der Berücksichtigung deklarativen und prozeduralen Wissens eine entscheidende Rolle. Der Agent verfügt in Act-R nicht immer über einen kompletten, perfekten Wissensstand. Je nach Kontext können unterschiedliche Wissensstrukturen aktiviert sein und bei gleichen Impulsen und Situationseinflüssen von außen zu verschiedenen Verhaltensweisen führen.

Der Einbezug von Aktivierungswerten im Vergleich zu der Auswahl von Plänen aufgrund fest definierter Präferenzen und *Utility*-Werte bietet einen entscheidenden Vorteil. Dadurch dass neben den situationalen Fakten auch die Interaktionshistorie und die Verhaltensweisen des Gegenübers mit einbezogen werden; der Agent kann sich automatisch auf Basis der Priming- und Aktivierungsprozesse an die aktuelle Situation und seinen Interaktionspartner anpassen. Dies erscheint gerade in Kooperationssituationen als natürlicheres Verhaltenskonzept als der Ansatz, immer alles perfekt zu wissen. Zudem kann die eingeschränkte Menge der berücksichtigten Fakten auch zu einem *effizienteren* Entscheidungsprozess beitragen.

Betrachtet man die **Repräsentation prozeduralen Wissens** in den verschiedenen Architekturen, so lassen sich Unterschiede in Bezug auf den *Abstraktions-* und *Autonomiegrad* feststellen. Während Produktionsregeln, einmal gefeuert, komplett ausgeführt werden (siehe Act-R und Soar), agieren die *Codelets* der IDA-Architektur in Form von Agenten autonom und parallel. Ein *Codelet* ähnelt zwar einer Produktionsregel und verfügt auch über eine Beschreibung erforderlicher Vorbedingungen und einer Spezifikation der auslösenden Effekte. Es unterscheidet sich jedoch von einer einfachen Produktionsregel durch die Angabe eines Aktivierungswertes und eines numerischen Wertes, welcher die Relevanz des Codelets in der aktuellen Situation beschreibt. Die autonome Ausführung über einen längeren Zeitraum stellt einen weiteren Unterschied dar. Eine Zwischenposition zwischen Produktionsregeln und Codelets nehmen die *Pläne* einer BDI-Architektur ein. Wie auch Produktionsregeln verfügen Pläne über Vorbedingungen und über eine Liste von Effekten. Zusätzlich können jedoch auch *Kontextbedingungen* spezifiziert werden, welche im Rahmen einer parallelen Ausführung das Verhalten des Agenten beeinflussen. Die Pläne sind zudem zwar in die Intensionsstruktur fest eingebunden und durch einen *Zieltyp* und *Commitmentstatus* beschränkt. Sie können im Gegensatz zu einer Produktionsregel aber über mehrere kognitive Zyklen aktiv bleiben. Es wird den Plänen damit ermöglicht, komplexere Berechnungen und Verhaltenssteuerungen über einen längeren Zeitraum

vorzunehmen, was mit einer Produktionsregel allein nicht möglich ist. Andererseits sind die Pläne enger mit der Intensionsstruktur verknüpft als dies bei den *Codelets* der IDA-Architektur der Fall ist.

In Bezug auf die *Abstraktionsstufe* zeichnen sich die Soar-Regeln durch einen sehr feinen Granularitätsgrad aus. Regeln in Act-R besitzen aufgrund der Auswahl einer einzigen Produktionsregel pro Zyklus in der Regel eine komplexere Struktur. Die *Skills* der *Icarus*-Architektur weisen eine große Ähnlichkeit zu den *Achieve*-Plänen einer BDI-Architektur auf. Jedoch werden komplexere Skills nicht direkt spezifiziert sondern setzen sich aus einem *Skill-Path* zusammen, der durch Inferenzprozesse gebildet wird.

Während die kognitiven Architekturen Soar und Act-R sehr genau den Entscheidungsprozess auf unterster Ebene modellieren (inklusive Lernmechanismen), lassen sie die Möglichkeit vermissen, **Commitmentverhaltensweisen** auf höherer Ebene umzusetzen und auf bereits getroffene Entscheidungen zuzugreifen. Commitmentverhalten, das eine entscheidende Rolle in Kooperationsituationen spielt, in denen Planung und Handlung verschachtelt auftreten können, lässt sich durch das reaktive Feuern von Produktionsregeln allein nicht modellieren. Vielmehr muss die Möglichkeit gegeben sein, Zielhierarchien und komplexe Intensionsstrukturen mit kontextspezifischen Einschränkungen und Bindung von Entscheidungsparametern zu repräsentieren. Während dies in BDI-Strukturen einen entscheidenden Aspekt darstellt, werden in Produktionssystemen Variableneinschränkungen und Festlegungen nicht automatisch an Teilziele propagiert.

Sobald der Agent im Rahmen einer Kooperation seinem Gegenüber Rechenschaft darüber ablegen muss, welche Ziele bzw. welche Intentionen er gerade verfolgt, sowie welche Handlungsoptionen er in Erwägung gezogen hat, benötigt er eine **explizite Repräsentation seines mentalen Zustands**. Die Repräsentation der Ziele muss so angelegt werden, dass es dem Agenten ermöglicht wird, auf seinen eigenen mentalen Zustand zuzugreifen. Während Produktionssysteme keine explizite Zielmodellierung auf dieser Ebene unterstützen, gibt es BDI-Systeme die eine komplexe Zielrepräsentation unterstützen. Die Umsetzung einer Repräsentation des *Zieltyps* zusammen mit einer Repräsentation des *Zielzustands* und seines *Commitmentstatus* stellen dabei zentrale Eigenschaften dar. Allerdings fehlt diesen BDI-Systemen die Möglichkeit, Teilentscheidungen zu repräsentieren. Variablen können entweder gebunden oder frei sein, es können jedoch keine Einschränkungen oder Belegungen an ihnen vermerkt werden.

In Bezug auf die Modellierung von Zielinteraktionen unterscheiden sich der *Easy-Deliberation*-Ansatz der *Jadex*-Architektur und die Arbeiten von Thangarajah und Kollegen deutlich in Bezug auf die berücksichtigten Daten, den Anteil der Deliberation und das resultierende Verhalten des Agenten. Während Thangarajah und Kollegen eine komplexe Datenstruktur der Abhängigkeit zwischen Zielen untereinander und den mit ihnen verbundenen Plänen erstellen, werden in dem *Easy-*

*Deliberation*-Ansatz nur für diesen Zweck eingefügte Abhängigkeitslinks und Kardinalitätseinschränkungen betrachtet. Damit lässt sich der *Easy-Deliberation* Ansatz auf alle Zieltypen anwenden, während der Ansatz von Thangarajah und Kollegen nur für *Achieve*-Ziele umgesetzt ist. Im Falle eines Konflikts versucht der Ansatz von Thangarajah und Kollegen diesen durch *Scheduling*-Algorithmen zu beheben, der *Easy-Deliberation*-Ansatz hingegen setzt die Strategie ein, die parallele Ausführung konfligierender Ziele zu unterbinden.

Ein Ansatz das Arbeitsgedächtnismodell Baddeleys in eine kognitive Architektur zu integrieren, ist in der IDA-Architektur beeindruckend umgesetzt. Da noch nicht alles implementiert ist, stellt das IDA-Modell eine Beschreibung auf abstrakter Ebene dar. Für den Einsatz in einem Kooperationsagenten fehlt die Berücksichtigung höherer mentaler Konzepte. Auch die ISAC-Architektur orientiert sich in ihrer Struktur an dem Modell Baddeleys (siehe Abschnitt 4.1.3). Auf welche Weise die einzelnen Prozesse umgesetzt werden können, ist in ihren Arbeiten nicht genau spezifiziert; es werden noch sehr einfache Berechnungsmodelle eingesetzt (z.B. statt Berechnung Abfrage aus einer Tabelle). Ein interessanter Aspekt der ISAC-Architektur besteht aber darin, Emotionen als Salienzfilter und Belohnungssignale einzusetzen. Die Berücksichtigung emotionaler Valenzwerte lässt an das *Controlled-Attention Framework* von Engle und Kollegen denken (siehe Abschnitt 4.1.3).

### Schlussfolgerungen

Die aus der Analyse theoretischer Modelle und konkreter Architekturen gewonnenen Erkenntnisse führen in der vorliegenden Arbeit zu den folgenden Entscheidungen. Da keine der diskutierten Architekturen die Basis für die Erfüllung des kompletten Anforderungskatalogs eines künstlichen Kooperationspartners (siehe Abschnitt 3.7) bietet, wird in der vorliegenden Arbeit eine bestehende Architektur als Ausgangsbasis genommen und diese durch zentrale Erweiterungen und Anpassungen in eine geeignete kognitive Architektur für einen Kooperationspartner überführt.

Als Basis für die Modellierung des mentalen Zustands des Agenten fällt die Entscheidung in dieser Arbeit auf JAM. Denn JAM bietet eine gute Grundlage für die Modellierung des Zusammenspiels von Zielen, Wissen und Ereignissen. Um eine explizite Zielrepräsentation für die Modellierung positiver und negativer Zielinteraktion zu ermöglichen, muss die Zielrepräsentation JAMs deutlich überarbeitet werden. Die Modellierung der geforderten subsymbolischen Prozesse fehlt in der JAM-Architektur. Um reaktive und vorbewusste Verarbeitungsprozesse angemessen zu berücksichtigen, müssen weitere Ebenen einer kognitiven Architektur konzipiert, implementiert und das Zusammenspiel mit der deliberativen Komponente gesichert werden. Der *Utility*-Wert der JAM-Pläne bietet eine

Möglichkeit, Teile subsymbolischer Berechnungen in den Entscheidungsprozess des BDI-Interpreters einfließen zu lassen. In diesem Kontext besteht jedoch Bedarf einer komplexen Modellierung, wie die *Utility*-Werte mit den anderen Faktoren der Priorität, der Aktivierung sowie der Zielinteraktion und eventuell sogar vorliegenden Valenzwerten zu verrechnen sind.

Die starre Menge der *Beliefs* in JAM bedarf einer grundlegenden Überarbeitung. An Stelle der *Beliefs* soll ein dynamisches Arbeitsgedächtnismodell treten, welches in die BDI-Architektur eingebettet wird. Für die Modellierung des Arbeitsgedächtnisses und insbesondere die Modellierung der Zugreifbarkeit von Wissen wird auf die Erkenntnisse der Grundlagenforschung aus der Psychologie zurückgegriffen. Dabei soll insbesondere das *Concentric Model of Working Memory* von Oberauer als Basis der Modellierung dienen. Die theoretischen Modelle der Psychologen machen keine Aussagen über die konkreten Berechnungsfunktionen; sie können demnach nur für die Struktur und die zu berücksichtigenden Mechanismen sowie deren Zusammenspiel herangezogen werden. Für die Modellierung der subsymbolischen Aktivierungsfunktionen wird auf Erfahrungen der *Act-R* Architektur zurückgegriffen. Die subsymbolischen Funktionen *Act-Rs* gilt es dann auf die Ziel- und Intentionsmodellierung der BDI-Architektur zu übertragen und ihre Einflussnahme in den Entscheidungsprozess des Interpreters festzulegen. Bei der Einbettung des Arbeitsgedächtnisses in die BDI-Architektur müssen fehlende Gedächtnisse (Semantisches Langzeitgedächtnis, Episodisches Gedächtnis) berücksichtigt und in Einklang mit der Kognitionsforschung und Psychologie in die Architektur integriert werden. Dafür stellen die klassischen Arbeiten Baddeleys eine Ausgangsbasis dar.

Für die erforderliche Modellierung des *Commitmentverhaltens* im Rahmen einer Kooperation weist die JAM-Architektur große Lücken auf. So ist es in der vorliegenden Arbeit ein Ziel, JAM dahingehend zu erweitern, dass Obligationen und gemeinsame Pläne durch die kognitive Architektur unterstützt werden. Partielle Pläne und Einschränkungen von Handlungsoptionen und Commitment in Bezug auf Handlungsparameter sind in JAM nicht vorgesehen. Im der vorliegenden Arbeit wird die Grundstruktur der Intentionen entscheidend erweitert, um Zwischenergebnisse des Deliberationsprozesses explizit zugreifbar zu machen. So kann der Agent sich seines eigenen mentalen Zustands „bewusst“ werden.

Die im nächsten Kapitel präsentierte CASEC-Architektur bildet die neu entwickelte kognitive Architektur für einen künstlichen Kooperationspartner.

# Kapitel 5

## CASEC-Architektur

Die CASEC-Architektur (Cognitive Architecture for Situated Embodied Communicators) bildet die Basis für die Realisierung eines künstlichen Kooperationspartners. Um die Anforderungen aus *Kapitel 3* umzusetzen, werden basierend auf der theoretischen Analyse des *Kapitels 4* Ansätze der mentalen Zustandsmodellierung mit kognitiven Basismechanismen zusammengeführt. Auf der einen Seite dient die auf hoher Ebene erfolgende Betrachtungsweise der mentalen Zustände dazu, Deliberationsprozesse, Commitmentverhalten, Planungs- und Entscheidungsprozesse zu modellieren. Der Agent sollte sich seiner Ziele und Pläne bewusst sein, so dass er sich im Rahmen einer Kooperation mit seinem Partner adäquat austauschen kann. Andererseits ist eine prozessbezogene Betrachtung für die Modellierung und Simulation eines Agenten unabdingbar. Die beiden Sichtweisen können sich darüber hinaus gegenseitig befruchten, da bei der Modellierung der Basisprozesse das Gesamtbild der mentalen Zustände im Auge behalten wird und andererseits berücksichtigt wird, welche Mechanismen welches Verhalten auslösen und welche Einschränkungen beachtet werden müssen.

### Architektur-Prinzipien

Das konzipierte Kernsystem der CASEC-Architektur integriert *symbolverarbeitende* und *verhaltensbasierte* Ansätze in einer hybriden Systemarchitektur. Dabei werden folgende Fähigkeiten berücksichtigt: Situierte Wahrnehmung, reaktives Verhalten, höhere mentale Prozesse, wie Schlussfolgern und planvolles Handeln, Aufmerksamkeitsmechanismen und motivationale Handlungsbewertung. Die wichtigsten Prinzipien der Architektur sind:

- Verarbeitung: **Parallele** Realisierung von Wahrnehmung, Schlußfolgerungs- und Handlungskomponenten (*Perceive, Reason, Act*) durch **modulare** Struktur (siehe Anforderung **(a1)**)

- Aktionsausführung reaktiver und deliberativer Verhaltensweisen mittels **Behaviors** mit individuellen Prioritätswerten, **kontextsensitive Zielsteuerung** durch den Einsatz von sowohl **Ziel-basierten** als auch **reaktiven Plänen** (siehe Anforderung **(a6)**)
- **Situiertheit**: Ständiger Fluss von Informationen und Aktivierungsprozessen innerhalb der kognitiven Schleife durch Einsatz eines **dynamischen Arbeitsgedächtnisses** (*bottom-up* und *top-down* Aktivierungsprozesse) (siehe Anforderung **(a3)**)
- Modellierung eines **expliziten mentalen Zustands** durch ein **erweitertes BDI-Modul** (Obligationen, gemeinsame Pläne, sowie komplexe *Constraint*-basierte Repräsentation) (siehe Anforderung **(a2)**)
- Repräsentation **modallogischer** Formeln für die Partnermodellierung (siehe Anforderung **(a7)**)
- Berücksichtigung von **Ziel/Plan-Interaktion** (siehe Anforderung **(a6)**)
- Repräsentation prozeduralen Wissens durch Einsatz einer **dynamischen Plan-Bibliothek** mit expliziter Repräsentation von Aktivierungs- und Effort-Werten der Pläne als Basis für Utility-Berechnungen (siehe Anforderungen (a4) und (a5))

## Struktur

Abbildung 5.1 präsentiert das Basismodell der CASEC-Architektur. Der Kreis, unterteilt in eine *Perceive-Reason-Act*-Triade, stellt die interne Verarbeitung des Agenten dar und grenzt ihn von seiner Umwelt ab. Dabei hebt die Dreiteilung die Verzahnung des klassischen *Perceive-Reason-Act* Zyklus hervor. Der direkte Informationsfluss zwischen den Sektoren *Perceive* und *Act* berücksichtigt jedoch, dass reaktives Verhalten entstehen kann, ohne dass zuvor eine Deliberation stattgefunden haben muss und die kognitive Schleife durchlaufen wurde. Damit kann der vorliegende Ansatz als eine Hybridarchitektur charakterisiert werden, die reaktives und deliberatives Verhalten in einer Struktur vereinigt. Die Struktur der kognitiven Komponente geht davon aus, dass kein fester Zyklus von *Wahrnehmen-Schlussfolgern-Handeln* vorliegt, sondern vielmehr die einzelnen Prozesse parallel agieren und der Agent so flexibel und reaktiv bleibt (Wachsmuth & Leßmann, 2002; Leßmann & Wachsmuth, 2003; Fink et al., 2003).

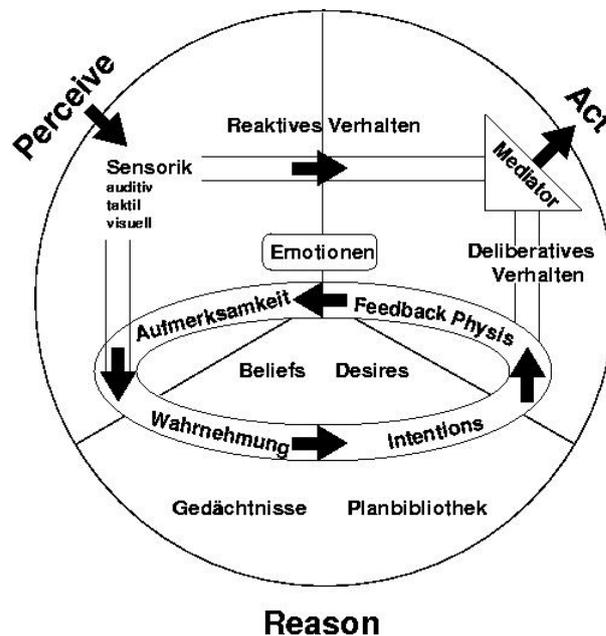


Abbildung 5.1: Basismodell der CASEC-Architektur

### Zentraler Ablauf

Auf der einen Seite können **Sensordaten** (Spracheingabe/auditiv, Körpersensorik/taktil, Szenenwahrnehmung/visuell) direkt ein **reaktives Verhalten** auslösen, welches auf einem niedrigen Abstraktionsniveau operiert und damit schnelle Antwortzeiten ermöglichen. Dazu gehört beispielsweise die Steuerung des Blickverhaltens. Die **kognitive Schleife** (Verarbeitung der Wahrnehmung - Bildung von Intentionen auf Basis der von Beliefs, Desires und Gedächtnisinhalten - Anstoßen der Handlungen- zurückfließende Feedbackinformationen und Ausrichten des Aufmerksamkeitsfokusses) bildet auf der anderen Seite einen Kreislauf, der die interne kognitive Verarbeitung des Agenten betrifft und das Wechselspiel zwischen Datenakquisition und Informationsverarbeitung aufzeigt (Wachsmuth & Leßmann, 2002). Wahrnehmung besteht hier nicht aus der starren Erfassung sensorischer Daten, sondern aus einer situationssensitiven Verarbeitung perzipierter Sinneseindrücke. Das bedeutet, dass eingehende Informationen im aktuellen Kontext der Umwelt, in Bezug auf den Verlauf der aktuellen Interaktion und den internen Zustand des Agenten interpretiert werden.

Die Sensordaten finden Eingang in die kognitive Schleife, werden dabei durch eine **Aufmerksamkeitssteuerung** gefiltert und wechselwirken in Form einer interpretierten und analysierten Wahrnehmung mit verschiedenen **Gedächtnissen**. Diese sind auf unterschiedliche Aspekte spezialisiert und arbeiten auf verschie-

denartigen Repräsentationen, legen aber alle jeweils relevante Fakten auf einem hohen Abstraktionsniveau in den **Beliefs** ab, die den *Episodic Buffer* des Arbeitsgedächtnisses des Agenten darstellen (Theorie siehe 4.1.3, Umsetzung siehe 5.1.2).

Die **kognitive Schleife** des Agenten reagiert nicht nur auf Impulse und Informationen von Außen, sondern reguliert sich selbst, wenn keine weiteren Eingaben erfolgen. Dabei laufen interne Prozesse parallel zu den perzeptionsverarbeitenden Prozessen. Gedächtnisprozesse sorgen durch ein Aktualisieren der Daten dafür, dass nur relevante Informationen im Arbeitsgedächtnis bereitgehalten werden. Sie situieren den Agenten in der Umwelt sowie in der Interaktion (siehe *bottom-up* Basisprozesse 4.1.3).

Nach Ermittlung einer **Intention** und der Schlichtung konfligierender Verhaltensweisen der deliberativen und reaktiven Komponente im **Mediator** kommt es zur Ausführung einer **Aktion**. Sowohl die aktiv ausgeführten Intentionen als auch die aktuell anliegenden und möglicherweise konkurrierenden Verhaltensweisen werden bei den zurückfließenden **Feedbackinformationen** berücksichtigt. Die Rückkopplung der erfolgten Aktionen und Effektorzustände wirkt sich wiederum in Form einer **Aufmerksamkeitssteuerung** auf die Sensorik und Wahrnehmung aus, womit der Zyklus geschlossen wird.

Die Schleife verdeutlicht eine der Kernideen der Architektur, nämlich dass ein ständiger Strom von Informationen zwischen den Sektoren umläuft, der sowohl aktuelle Sensor- und Effektorinformationen als auch interne mentale Zustände einbezieht.

## 5.1 Dynamisches Gedächtnismodell

Im Rahmen der CASEC-Architektur wird ein *dynamisches Gedächtnismodell* eingesetzt. Dabei bezieht sich das Attribut *dynamisch* darauf, dass es sich bei dem Gedächtnismodell nicht um einen passiven Speicher handelt, sondern dass vielmehr Prozesse unabhängig von direkten Zugriffen dynamisch auf den Inhalten des Gedächtnisses operieren und die Inhalte situativ anpassen.

Das *dynamische Gedächtnismodell* versetzt den Agenten in die Lage, auf neue Inferenzen und aktuelle Ereignisse und Signale seines Gegenübers ein angemessenes Antwortverhalten zu produzieren (*gegenseitiges Aufeinandereingehen*, siehe Definition 2 (k1)). Zusätzlich hilft das dynamische Gedächtnismodell dabei, den Agenten in seiner *Umwelt* (durch seine Wahrnehmungen), in der *Interaktionshistorie* (durch sein Arbeitsgedächtnis) und in seinen *Erfahrungen* (durch ein episodisches Langzeitgedächtnis) zu situieren. Durch das dynamische Gedächtnismodell wird eine Informationsbasis für nachfolgende Deliberations- und Entscheidungsmechanismen geschaffen. Erreicht wird dies durch *kognitive Basisme-*

chanismen, welche im Dienste der Gedächtnisse operieren.

### Gedächtnisstruktur

Die Strukturierung des *Arbeitsgedächtnisses* in der CASEC-Architektur (siehe Abbildung 5.2) orientiert sich an dem *Multi-Component Model*. Sie enthält einen *Phonological Loop* und ein rudimentäres *Visual Spatial Memory* (siehe Abschnitt 4.1.3). Die modalitätsspezifischen Komponenten ermöglichen eine parallele Verarbeitung sowie den Einsatz spezialisierter Algorithmen.

Der *Episodic Buffer* nimmt eine Schlüsselposition im Rahmen der Architektur ein. Modelliert wird er durch eine erweiterte Form der *Beliefs*, auf welchen mentale Prozesse in Form eines Interpreters der *BDI*-Struktur operieren. Die *Central Executive* wird damit durch den *Interpreter* ausgefüllt und durch weitere Mechanismen ergänzt. Der *Episodic Buffer* enthält die aktuell relevanten Informationen des *Visual Spatial Memory* und des *Phonological Loop*, aktivierte Elemente des Langzeitgedächtnisses sowie Inferenzen des Agenten. Für die Strukturierung des

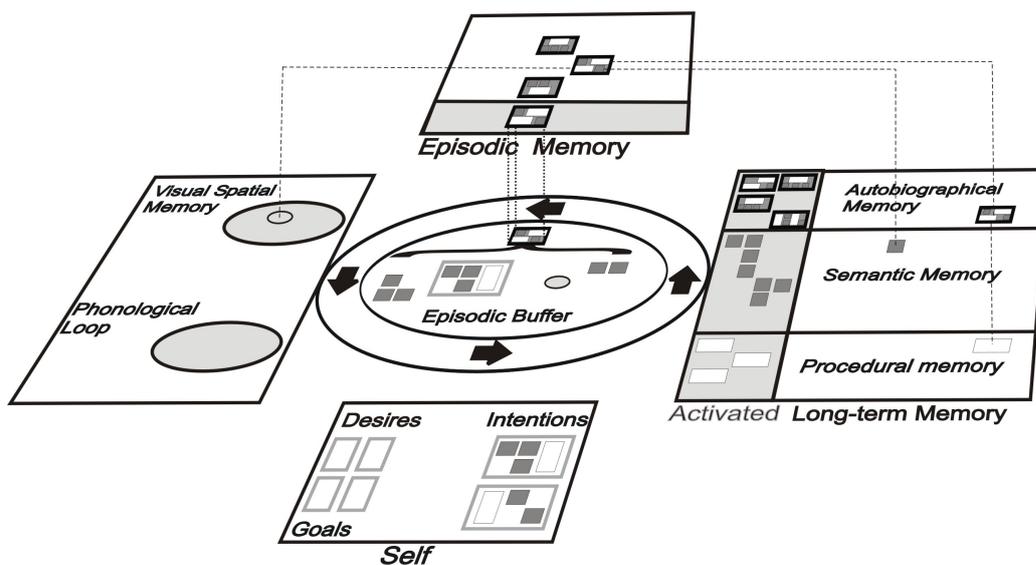


Abbildung 5.2: Gedächtnismodell der CASEC-Architektur

Langzeitgedächtnisses (*Long-term memory, LTM*) folgt die CASEC-Architektur dem Ansatz von Tulving & Donaldson (1972), zwischen *deklarativem* und *prozeduralem* Wissen zu unterscheiden. Dabei wird das deklarative Wissen zusätzlich in das autobiographische sowie semantische Gedächtnis unterteilt. Die jeweiligen Gedächtnisse sind mit einem Buffer versehen, welcher den Zugriff auf Inhalte mit

besonders hoher Aktivierung ermöglicht. Dies entspricht dem dreistufigen Zugriffsmodell Oberauers und Cowans (siehe 4.1.3), wobei diese jedoch in ihren Modellen keine inhaltliche Strukturierung des Langzeitgedächtnisses vorsehen.

Das Episodische Gedächtnis (*Episodic Memory*) spielt eine zentrale Rolle im Rahmen von Interaktionsszenarien, da es den Verlauf der Interaktion sowie die Ereignisstruktur im zeitlichen Rahmen von Minuten und Stunden aufnimmt. Die Konzeption des episodischen Gedächtnisses in der CASEC-Architektur orientiert sich an Arbeiten von Conway (2001). Zusätzlich zu den Gedächtnissen, die sich auf Wissen beziehen, besteht in der Architektur eine Struktur, die sich auf selbstbezogene Aspekte bezieht, eine einfache Form eines *Selbstmodells*. Selbstmodelle sind nach Metzinger (1999) komplexe relationale Strukturen, mit denen informationsverarbeitende Systeme intern auf sich selbst Bezug nehmen können. Sie modellieren zeitliche, räumliche, kausale und logische Relationen anhand des inneren Repertoires, das das jeweilige physische System ihnen zur Verfügung stellt. Sie zeichnen sich dadurch aus, dass das von ihnen modellierte Objekt eben derjenige kognitive Agent ist, für den sie eine wichtige Funktion besitzen, und dadurch, dass dieses Objekt ihnen sowohl über externe als auch über interne Signalquellen gegeben ist. Selbstmodelle sind innere Werkzeuge der Informationsverarbeitung.

Zu dem Selbstmodell der CASEC-Architektur zählen Ziele, Intentionen und Wünsche des Agenten, welche im Rahmen eines BDI-Moduls repräsentiert werden. Auch der *emotionale Zustand* des Agenten lässt sich diesem Modell zuordnen. Prozesse, die von dem Selbstmodell des Agenten ausgehen, sind für das zielorientierte Verhalten des Agenten verantwortlich. Dem Selbstmodell des Agenten werden auch Parameter zugeordnet, welche die Persönlichkeit des Agenten ausmachen. Es ist ein konzeptioneller Container für selbstbezogene Prozesse und Einstellungen.

## 5.1.1 Vorbewusste Zwischenspeicher

### Visual Spatial Memory

Auf unterster Ebene berücksichtigt die CASEC-Architektur vorbereusste Zwischenspeicher (*Preconscious buffer*). Damit orientiert sie sich an der Struktur des *Multi-Component Models* (siehe Abschnitt 4.1.3). In einem *Visual Spatial Memory* wird vermerkt, welche Objekte der Agent gerade wahrnimmt und welche er aufgrund seiner vorherigen Wahrnehmung erwartet wahrzunehmen. Es werden nur die signifikanten Veränderungen der Wahrnehmung propagiert (*bottom-up Filter*), damit nicht eine ständige Informationsflut von den Schlussfolgerungsmechanismen gemeistert werden muss. Andererseits kann der interne *Aufmerksamkeitsfokus* auch auf bestimmte Objekte gelenkt werden (*top-down Filter*). Diese Informationen werden zusätzlich zu den aktuellen Wahrnehmungen weitergelei-

tet. Das Repräsentationsformat besteht aus einem einfachen räumlichen Buffer, in welchem wahrgenommene Objekte anhand ihrer wahrgenommenen Position eingefügt werden.

Neben der Wahrnehmung von Objekten ist auch eine Wahrnehmung für Bewegungen und Abläufe auf unterer Ebene vorgesehen. Dafür werden *Detektoren* eingesetzt, welche beispielsweise Bewegungen des Gegenübers analysieren und nach Mustern durchsuchen. Diese Detektoren werden situationsbezogen aktiviert und zu Netzwerken zusammenschaltet, sodass die Verarbeitung kontextspezifisch abläuft. Als Kontextfaktoren werden interne *Beliefs* und Zustände des Agenten an das *Detektornetzwerk* propagiert. Dabei wird in der CASEC-Architektur auf Arbeiten des PRoSA Systems aufgebaut (Latoschik, 2001), (Biermann & Wachsmuth, 2003). So können beispielsweise *Turn-Taking*-Detektoren in Abhängigkeit des konversationalen Zustands des Agenten ausgerichtet werden (siehe 6.2.2). In diesem Fall erwartet der Agent keine *Taking-Turn*-Geste während sein Gegenüber bereits den Turn hat, und nutzt die Ressourcen, die durch die Deaktivierung der zugehörigen Detektoren frei werden dazu, *Giving-Turn*-Gesten zu erkennen.

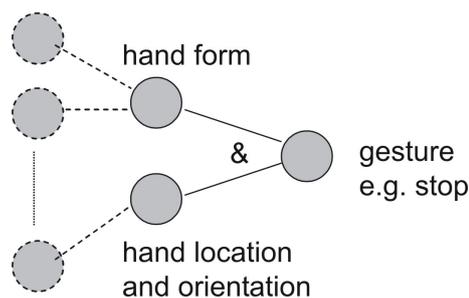


Abbildung 5.3: Beispiel Detektorennetz: Turn-Taking-Detektor

Ein weiterer Mechanismus auf dieser Ebene besteht aus *Perceptual Priming* (siehe Abschnitt 4.1.6). Objekte, die auf dieser Ebene wahrgenommen werden, können Aktivierungsimpulse auslösen und dadurch Elemente im Arbeits- sowie im Langzeitgedächtnis aktivieren (siehe Abschnitt 5.1.5). Ausgehend von dem wahrgenommenen Objekttyp wird ein Impuls an das dynamische Arbeitsgedächtnis gesendet. Ist der Aufmerksamkeitsfokus des Buffers zudem auf ein bestimmtes Attribut gesetzt, so lösen sowohl das Attribut als auch die Werte des wahrgenommenen Attributs weitere Aktivierungsimpulse aus.

### Phonological Loop

Der *Phonological Loop* zählt zum Arbeitsgedächtnis (siehe auch Abbildung 5.2) und ist in der CASEC-Architektur auf symbolischer Ebene konzipiert. Er dient als

Repräsentant der auditiven Verarbeitung, um die Anforderungen an die Strukturierung zu erfüllen und das Zusammenspiel der Komponenten zu gewährleisten. Der *Phonological Loop* besteht aus zwei Teilen: einem Buffer (*phonological store*), welcher die verstandenen Wörter eine gewisse Zeit vorhält, und artikulatorischen Kontrollprozessen (*articulatory control processes*). Der *Phonological Store* agiert als „inneres Ohr“ und hält Informationen in sprachbasierter Form für 1-2 Sekunden vor. Laut der Theorie Baddeleys kommen gesprochene Worte (auch die des Agenten) direkt in den Speicher. Gelesene Wörter müssten erst noch in artikulatorischen Code überführt werden; dies ist in der Architektur bisher nicht vorgesehen. Aufgebaut ist der Buffer in Form eines Ringspeichers mit beschränkter Kapazität. Liegt jedoch ein Wort mit erhöhtem Aktivierungswert in dem aktuell zu beschreibenden Feld vor, so wird der nächstfolgende Speicherplatz überprüft. Zusätzlich operiert ein *Decay*-Mechanismus, welcher die Aktivierungswerte der Wörter mit der Zeit verringert. Möchte der Agent bestimmte Wörter länger im *Phonological Store* vorhalten, so muss er sich des *Rehearsal*-Mechanismus bedienen. Dieser sorgt dafür, dass die Wörter erneut eine Aktivierung erfahren, als wären sie gerade neu in den Speicher eingetragen worden.

Eine frühe Verarbeitung der Elemente des *Phonological Loop* findet in Bezug auf die Referenzauflösung statt. Bevor der Agent eine komplette Äußerung gehört und verarbeitet hat, konzentrieren sich bereits Prozesse darauf, Referenzen aufzulösen und dem Interlokutor Feedback zu liefern. Dies ist insbesondere im Rahmen des *gegenseitigen Aufeinandereingehens* (Abschnitt siehe 2.1.1) von entscheidender Bedeutung.

### 5.1.2 Episodischer Speicher (*Episodic Buffer*) - Beliefs

Das Arbeitsgedächtnis steht im Schnittpunkt der Vereinigung der mentalen, philosophischen und psychologisch-kognitionswissenschaftlichen Modelle. Was im Rahmen philosophischer Modelle als Menge der *Beliefs* bezeichnet wird, wird in der CASEC-Architektur durch das *dynamische Arbeitsgedächtnis* in Form des *Episodic Buffer* modelliert.

In BDI-Modellen werden die *Beliefs* in einer symbolischen Form repräsentiert, welche logische Schlüsse und Abfragen erlaubt und eine Verbindung zu den Zielrepräsentationen ermöglicht. In kognitionswissenschaftlichen Modellen wird dagegen nicht von einem passiven Speicher ausgegangen, sondern dynamische Adaptationsprozesse gehören direkt zu dem Gedächtnismodell.

### 5.1.3 Zugriffstruktur

Die Zugriffsstruktur von Informationen in der CASEC-Architektur baut auf den psychologischen Modellen von Oberauer und Cowan (siehe 4.1.3) auf. So stellen

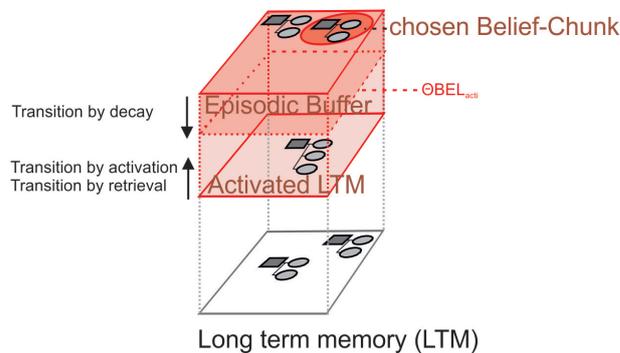


Abbildung 5.4: 3 Teilung: Aktiviertes Langzeitgedächtnis - Episodischer Buffer - Ausgewählter *Belief-Chunk*

die Elemente des *Episodic Buffer* die aktivierten Elemente des Langzeitgedächtnisses (*LTM*) dar, welche sich in der *Region des direkten Zugriffs* befinden. Zusätzlich zu den aktivierten *LTM*-Elementen können Elemente der Wahrnehmung aus den *Visual Spatial Memory* und *Phonological Loop* dem *Episodic Buffer* hinzugefügt werden. Die Elemente des *Episodic Buffer* stellen damit die der kognitiven Operanden der *Central Executive* dar (siehe Abbildung 5.2).

Neben der *Region des direkten Zugriffs* werden auch aktivierte Langzeitgedächtnisinhalte verwaltet, auf welche nicht direkt zugegriffen werden kann, die aber die Verarbeitung beeinflussen, wie in den Modellen Cowans und Oberauers vorgeschlagen. Alle diejenigen Elemente des Langzeitgedächtnisses, welche über einen bestimmten Schwellwert hinaus aktiviert sind, werden in einem speziellen Zugriffsbereich (*Buffer*) des aktivierten *LTM* organisiert (siehe Abbildung 5.4). Der Übergang von einem Bereich in den anderen kann dabei durch explizite *Retrieval*-Aufrufe geschehen, aber auch durch einen automatischen Mechanismus, wenn ein Element einen bestimmten Aktivierungsschwellwert überschreitet. Sowohl in der *Region des direkten Zugriffs* als auch im aktivierten *LTM*-Bereich werden die Aktivierungswerte der Elemente durch *Aktivierungs*- sowie *Decay*-Mechanismen manipuliert und dynamisch angepasst (siehe Abschnitt 5.1.5 und 5.1.6). Liegen im Falle einer Anfrage von Informationen keine geeigneten Kandidaten im *Episodic Buffer* vor, so wird mittels Zugriffsmechanismen versucht aus den darunter liegenden Schichten (dem aktivierten bzw. noch nicht aktivierten *LTM*) entsprechende Informationen abzuleiten. Dafür bestehen automatische *Retrieval*-Mechanismen (siehe Abschnitt 5.1.5).

Die Gedächtniselemente in dem Bereich des aktivierten Langzeitgedächtnisses verändern die Verarbeitungsgeschwindigkeit der *Central executive* nicht, nur die Elemente des *Episodic Buffer* können die Effizienz des Zugriffs beeinflussen,

da sie potentielle Kandidaten der kognitiven Operationen darstellen.

Mit der Unterteilung des Arbeitsgedächtnisses in drei funktional unterschiedliche Regionen bezüglich des Zugriffs (*aktivierter Teil des LTM*, den *Bereich des direkten Zugriffs*, *aktuell gewählter Belief-Chunk*) grenzt sich die CASEC-Architektur von anderen kognitiven Architekturen ab (siehe Abschnitt 4.3.1). In *Soar* liegen nur alle die Produktionen im Bereich des direkten Zugriffs, welche mit dem Problemraum verwandt sind und deren Zugehörigkeit explizit definiert wird. In *Act-R* wird der Aktivierungswert allein durch das Produktionssystem und die Zeit beeinflusst. In der CASEC-Architektur wird dagegen ein weiter greifender Ansatz verfolgt, in welchem eine situationssensitive Verarbeitung als Berechnungs- und Verarbeitungsgrundlage gewählt wird, welche Einflussfaktoren der Wahrnehmung, der Interpretation sowie der Aufgabenausrichtung zusammenbringt.

Die Einteilung in drei Bereiche (direkt auswählbar (ACCESSED), sichtbar (VISIBLE) und prinzipiell zugreifbar (REACHABLE)) ähnelt dem LAKOS-Modell (Wachsmuth, 1988, 1985). Allerdings beziehen sich die Zugriffsoptionen in der CASEC-Architektur auf dynamisch adaptierte Arbeitsgedächtnisstrukturen, wohingegen in LAKOS von permanenten, explizit strukturierten, domänenspezifischen Wissenspaketen ausgegangen wird (siehe Abschnitt 4.1.3). Damit unterscheiden sich sowohl die Basis, auf welcher die Prozesse operieren, als auch die Mechanismen, die die Elemente von einem in einen anderen Zugriffsbereich überführen in CASEC und LAKOS deutlich voneinander.

## Beliefs

Das Repräsentationsformat der Elemente des Arbeitsgedächtnisses folgt in der CASEC-Architektur einem symbolbasierten Ansatz, der noch um verschiedene subsymbolische Mechanismen erweitert wird. Die Grundeinheit für die Repräsentation von Beliefs besteht in der CASEC-Architektur aus Formeln. Als Spezifikationsprache der Beliefs wird auf eine erweiterte Form der Modal-Logik zurückgegriffen. Der eingesetzte Formalismus beruht dabei auf dem „Mögliche Welten“-Ansatz und der KD45-Logik (siehe Anhang A.1). Dabei werden inkonsistente Beliefs ausgeschlossen und es wird davon ausgegangen, dass sich der Agent seiner Beliefs bewusst ist.

Das BDI-Modul der CASEC-Architektur baut zunächst auf JAM und UM-PRS auf, erweitert diese jedoch entscheidend. Die Repräsentation der Beliefs, welche in JAM und UM-PRS aus einer Menge einfacher Relationen mit Attributlisten besteht, wird so erweitert, dass komplexe Belief-Formeln darstellbar werden sowie eine Verschachtelung von Beliefs ermöglicht wird. Die zentralen Erweiterungen sehen folgendermaßen aus:

**Erweiterungen:**

- Einführung von Junktoren ( $\wedge, \vee, \neg$ ) für die Repräsentation komplexer Formeln
- Repräsentation verschachtelter Beliefs
- Repräsentation zeitlicher Attribute (Entstehungszeitpunkt, letzter Zugriff)
- Repräsentation des Ursprungs der Beliefs (Perzeption, Schlussfolgerung, Sprechakt des Interlokutors)
- Aktivierungswert, Salienzwert
- Vernetzung von Formeln, welche sich auf gleiche Entitäten beziehen

Dabei dienen die ersten beiden Erweiterungen dazu, modallogische Formeln repräsentierbar zu machen. Die Erweiterung der Beliefs um individuelle Attribute der Zeitlichkeit und der Aktivierung dienen dazu, den Zugriff auf Beliefs und die Vergleichbarkeit von Beliefs untereinander dynamisch und kontext-sensitiv zu gestalten. Im Weiteren werden die konkreten Datenstrukturen, die in der CASEC-Architektur für die Modellierung der Beliefs eingesetzt werden, in der Z-Notation (ISO/IEC, 2002) vorgestellt.

### 5.1.4 Repräsentation

#### Belief-Chunk

Die Grundeinheit für die Repräsentation eines Beliefs in der CASEC-Architektur stellt der *Belief-Chunk* dar. Mit Hilfe von *Belief-Chunks* werden Prädikate und Relationen zwischen Termen dargestellt. In CASEC können Terme aus konstanten Symbolen (*Values*), Variablen oder der Referenz auf eine konkrete Instanz eines Objekts bzw. einer Person bestehen. Individuelle Instanzen verfügen über einen eindeutigen Identifikationsnamen (ID).

$$\begin{aligned}
 \text{Term} & ::= \text{val}\langle\langle\text{Val}\rangle\rangle \\
 & \quad | \text{var}\langle\langle\text{Var}\rangle\rangle \\
 & \quad | \text{inst}\langle\langle\text{InstanceVar}\rangle\rangle
 \end{aligned}
 \tag{5.1}$$

Prädikate und Relationen werden über *Expressions* definiert. Dabei kann eine *Expression* neben Termen auch *Funktionen* und *Belief-Chunks* enthalten. *Belief-Chunks* enthalten Prädikate, welche wiederum *Expressions* enthalten. Durch die

rekursive Definition ist es möglich, modallogische Formeln in CASEC zu repräsentieren (siehe Abschnitt 5.1.4).

$$\begin{aligned}
 \textit{Expression} & ::= \textit{term}\langle\langle\textit{Term}\rangle\rangle \\
 & \quad | \textit{functor}\langle\langle\textit{FunSym} \times \textit{seq Expression}\rangle\rangle \\
 & \quad | \textit{predicate}\langle\langle\textit{Predicate}\rangle\rangle \\
 & \quad | \textit{beliefChunk}\langle\langle\textit{BeliefChunk}\rangle\rangle
 \end{aligned} \tag{5.2}$$

In der CASEC-Architektur besteht ein **Belief-Chunk** aus einem Relationsnamen (Prädikatsymbol), welcher für die Speicherung und den Abruf als Zugriffsschlüssel dient (*Predicate indexing*, siehe (Russell & Norvig, 2003)), und einer Liste von *Expressions*. Zusätzlich zu der Repräsentation des Inhalts können für jeden *Belief-Chunk* Attribute spezifiziert werden (Entstehungszeitpunkt, letzter Zugriff, Aktivierung, Salienzwert). Diese dienen dazu, kognitive Operationen auf den *Belief-Chunks* zu unterstützen.

$$\textit{ActivationValue} == a : \mathbb{R} \mid a \geq 0 \tag{5.3}$$

$$\textit{SaliencyValue} == s : \mathbb{R} \tag{5.4}$$

(Für die Erweiterung der Z-Notation um  $\mathbb{R}$  siehe (W. Oliveira & Barros, 1998).)

### Schema 5

$\textit{PointInTime}$ $t : \mathbb{R}$ $\textit{startTime} : \mathbb{R}$
$t \geq \textit{startTime}$

$\textit{InitTime}$ $\textit{PointInTime}$
$\textit{startTime} = 0$

### Schema 6

<i>BeliefChunk</i> <i>head</i> : <i>PredSym</i> <i>terms</i> : <i>seq Expression</i> <i>timeStamp</i> : <i>PointInTime</i> <i>lastRetrieved</i> : <i>PointInTime</i> <i>activation</i> : <i>ActivationValue</i> <i>saliency</i> : <i>SaliencyValue</i>
--

### Repräsentation modallogischer Formeln

Gerade für die Partnermodellierung ist es von entscheidender Bedeutung, verschachtelte Beliefs (modallogische Formeln) über den Interaktionspartner repräsentieren zu können. Um die Kluft zwischen modalen BDI-Logiken und realen Implementationen zu verringern (Móra et al., 1999), wird die Komplexität des Repräsentationsformats im Vergleich zu den klassischen BDI-Implementationen in der CASEC-Architektur entscheidend erweitert. Durch die rekursive Definition der *Expressions* (Definition 5.2) und *Belief-Chunks* wird erreicht, dass verschachtelte *Belief-Chunks* dargestellt werden können, d.h. ein *Belief-Chunk* kann als Term einen weiteren *Belief-Chunk* beinhalten.

Eingesetzt wird dieses für die Repräsentation von Modaloperatoren. Der Agent kann durch die Erweiterung beispielsweise Formeln der Struktur: Der Agent  $i$  glaubt, dass sein Gegenüber  $j$  glaubt, dass der Agent  $i$  ein Objekt fokussiert, ( $BEL_i(BEL_j(ATT_i(obj)))$ ), repräsentieren (siehe Abbildung 5.5). Repräsentation der modallogischen Formel als *Belief-Chunk* in der CASEC-Architektur:

$$BEL\ agent\_j\ (ATT\ agent\_i\ \$obj)$$

Dabei wird das erste *belief agent\_i* weggelassen, da dies durch die Zugehörigkeit des *Belief-Chunks* zu der Menge der Beliefs des Agenten  $i$  impliziert ist. Die Unifizierungsfunktion *unify* der CASEC-Architektur (siehe Abschnitt 5.1.7) baut auf der Unifizierungsfunktion von UM-PRS auf wurde aber in Bezug auf verschiedene Aspekte erweitert, beispielsweise kann sie durch rekursive Aufrufe verschachtelte *Belief-Chunks* miteinander unifizieren.

### Junktoren

Negierte Formeln dienen dazu, falsifizierte Fakten darstellen zu können. Zudem können sie dabei helfen, Informationen schnell zugreifbar zu machen. Normalerweise bauen die Schlussfolgerungsmechanismen auf der „*Closed world assumption*“ auf, d.h. falls Beliefs nicht direkt vorliegen oder sich durch Schlussfolgerungen ableiten lassen, wird davon ausgegangen, dass sie nicht gelten. Durch die

## Beispiel verschachtelter Belief

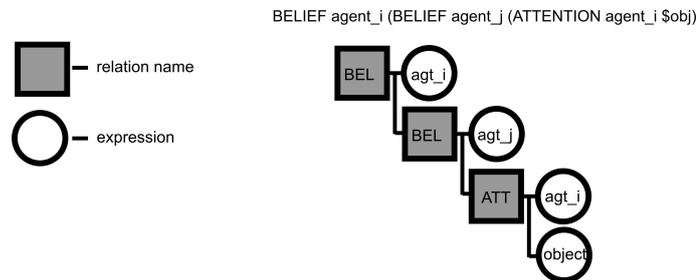


Abbildung 5.5: Verschachtelter Belief in CASEC

Möglichkeit negierte Fakten vorzuhalten, können Abfragen beschleunigt werden: Bei einer Abfrage wird zunächst geprüft, ob das Faktum gilt; wenn dies nicht der Fall ist, wird als nächstes getestet, ob die Negation der Aussage gilt. So müssen dann keine komplexen Schlussfolgerungsmechanismen angestoßen werden.

In der CASEC-Architektur besteht zunächst die Möglichkeit, *Belief-Chunks* oder negierte *Belief-Chunks* als atomare *Belief-Formeln* zu repräsentieren:

**Definition 3** Eine atomare *Belief-Formel*  $\varphi$  besteht aus einem *Belief-Chunk*  $\beta$  oder einem *negierten Belief-Chunk* ( $\neg\beta$ ). *Belief-Chunks* können durch die *Junktoren*  $\vee, \wedge$  zu komplexen *Belief-Formeln* zusammengesetzt werden.

Die konkrete Struktur einer *Belief-Formel* in der CASEC-Architektur ergibt sich damit zu:

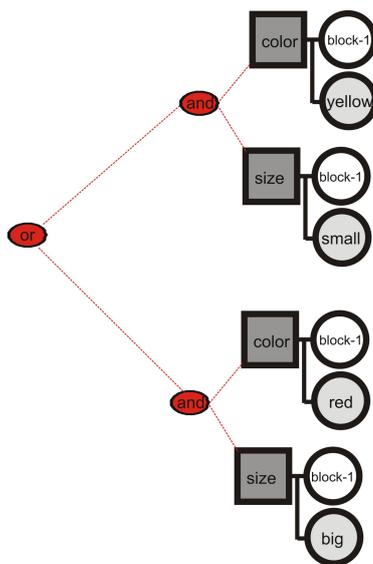
$$\begin{aligned}
 \textit{BeliefFormula} & ::= \langle\langle \textit{BeliefChunk} \rangle\rangle \\
 & \quad | \textit{not} \langle\langle \textit{BeliefChunk} \rangle\rangle \\
 & \quad | \textit{or} \langle\langle \textit{BeliefFormula} \times \textit{BeliefFormula} \rangle\rangle \\
 & \quad | \textit{and} \langle\langle \textit{BeliefFormula} \times \textit{BeliefFormula} \rangle\rangle \quad (5.5)
 \end{aligned}$$

Junktoren dienen dazu, komplexe Formeln darstellen zu können. Einerseits wird dadurch ein mächtigeres Repräsentationsformat unterstützt. Werden jedoch Abfragen vorgenommen, ob eine bestimmte Formel gilt, so muss berücksichtigt werden, ob der *Belief-Chunk* Teil einer komplexen mit Junktoren verknüpften Formel ist, um den Wahrheitswert der Anfrage bestimmen zu können. Da es im Falle von mit AND und OR verknüpfter Formeln nicht immer möglich ist, einen eindeutigen Wert zurückzuliefern, wird die CASEC-Architektur um eine die Abfrage POSSIBLE erweitert.

In Abbildung 5.6 ist ein Beispiel zu sehen, welches auf Abfragen und Wahrheitswerte der *Belief-Formeln* eingeht. Es wird spezifiziert, dass eine Box klein

und gelb oder groß und rot ist. Abfragen in Bezug auf die Farbe der Box werden auf die Wahrheitswerte *true*, *false* und *unknown* abgebildet. Die Abfrage, ob es möglich ist, dass die Box blau bzw. rot ist, wird über den POSSIBLE Abfragemechanismus umgesetzt und auf die Wahrheitswerte *true* und *false* abgebildet.

### Beispiel 2

$$((\text{color "block-1" "yellow"} \wedge (\text{size "block-1" "small"})) \vee ((\text{color "block-1" "red"} \wedge (\text{size "block-1" "large"})))$$


### Abfragen und Wahrheitswerte

FACT (color "block-1" "yellow")  $\Rightarrow$  unknown  
 FACT (color "block-1" "blue")  $\Rightarrow$  false  
 POSSIBLE (color "block-1" "blue")  $\Rightarrow$  false  
 POSSIBLE (color "block-1" "red")  $\Rightarrow$  true

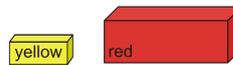


Abbildung 5.6: Abfragen und Wahrheitswerte von mit Junktoren verknüpften Belief-Formeln

## Constraints

Während verschachtelte Beliefs dazu eingesetzt werden, komplexe modallogische Formeln zu repräsentieren, dienen *Constraints* dazu, Formel-Netze zu modellieren, in denen Eigenschaften von Variablen/Termen in einem komplexen Format explizit spezifiziert werden können. *Constraints* ermöglichen es, im Rahmen einer partiellen Zielrepräsentation den Entscheidungsprozess und die potentiellen Kandidaten so offen wie möglich zu halten, sodass im Rahmen des *Means-Ends-Reasoning* nicht nur der Handlungsplan kontextspezifisch gewählt wird, sondern auch die Parameter und involvierten Objekte situationsabhängig bestimmt werden können.

Die Constraintrepräsentation geht auf die klassischen *Constraint Satisfaction Problems* (CSPs) (Russell & Norvig, 2003) zurück. Formell besteht ein CSP aus

einem Tripel  $\langle X, D, C \rangle$ , wobei  $X$  aus einer Menge von Variablen besteht,  $D$  eine Domäne von möglichen Werten darstellt und  $C$  eine Menge von Einschränkungen (*Constraints*) repräsentiert. In der CASEC-Architektur bestehen die Variablen aus Variablen des *Ziel-* bzw. *Belief-Chunks* und die Constraints können explizit in den zugehörigen *Ziel-/Plan-Bindings* definiert werden, in denen auch die aktuellen Variablenbelegungen vermerkt werden.

Die Struktur der Variablen-Repräsentation ist in der CASEC-Architektur so erweitert, dass Variablen mit einem oder mehreren *Constraints* verknüpft werden können. In der CASEC-Architektur werden die folgenden Typen von *Constraints* für Variablen unterstützt.

**Constraint-Typen:**

- *c:Is\_A*
- *c:Is\_Not\_Equal\_To*
- *c:General*
- *c:Exist*
- *c:All*
- *c:No\_Other\_Than*
- *c:And*
- *c:Or*
- *c:Function*

Die Einschränkung *c:Is\_A* dient dazu, den Typ eines Objekts auf eine bestimmte Objektklasse einzuschränken. Möchte man hingegen verhindern, dass ein bestimmtes, individuelles Objekt an eine Variable gebunden wird, so kann der Constrainttyp *c:Is\_Not\_Equal\_To* eingesetzt werden, welcher sich entweder auf den eindeutigen Identifikationsnamen eines Objekts oder den Verweis auf eine Variable des lokalen Kontextes bezieht. Für die Spezifikation allgemeiner Einschränkungen existiert der Constrainttyp *c:General*, welcher mit einer beliebigen Relation gefüllt werden kann. Komplexere Formen einer allgemeinen einschränkenden Relation stellen die folgenden Einschränkungen dar: *c:General\_Exist*, *c:All* und *c:No\_Other\_Than*.

Neben der Spezifikation einer einzelnen Einschränkung besteht auch die Möglichkeit, mehrere *Constraints* zu einer komplexen Einschränkung zu verbin-

den. Dazu dienen die Constrainttypen  $c:And$  und der  $c:Or$ . Neben den logisch geprägten Einschränkungen, welche sich auf Relationen und ableitbares Wissen beziehen, existiert auch die Möglichkeit, eine Variable durch  $c:Function$  einzuschränken, welche berechnet und mit einem spezifizierten Wert verglichen wird.

### Repräsentation der Zeit

Zeit spielt gerade in einer dynamischen Interaktion und Umwelt eine entscheidende Rolle. Die temporale Kongruenz von Ereignissen lässt sich durch *Zeitstempel* repräsentierbar machen. Zum einen dienen Zeitstempel dazu, die Aktualität eines *Belief-Chunks* zu bezeichnen, und zum anderen können zeitliche Abfolgen eine wichtige Rolle beim Schlussfolgern spielen. Die Erweiterung um zeitliche Konzepte stellt daher einen weiteren bedeutsamen Schritt für eine dynamische, kontextsensitive Verarbeitung dar.

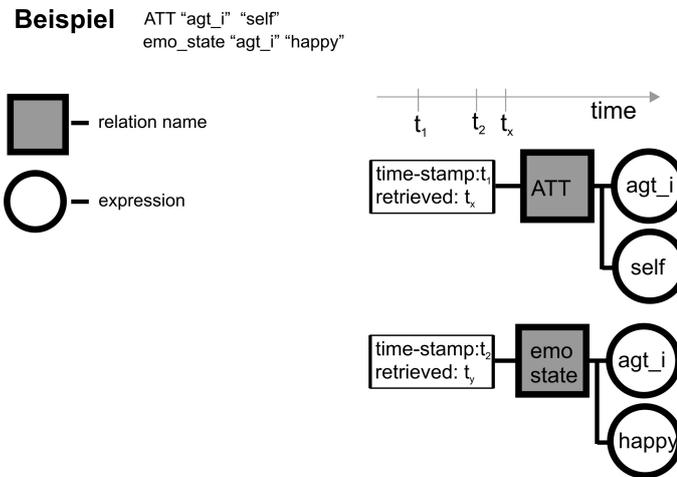
**Definition 4** Sei  $\varphi$  ein *Belief-Chunk*, dann ist auch  $\varphi_{[t1]}$ , ( $t1 = PointInTime$ ) ein *Belief-Chunk*. Die Funktion  $Time(\varphi) = t1$  liefert den *Zeitstempel* des *Belief-Chunks* zurück.

Abbildung 5.7 zeigt ein Beispiel für *Belief-Chunks* mit *Zeitstempeln*. Für Schlussfolgerungsprozesse kann es einen entscheidenden Unterschied machen, ob beispielsweise der Interaktionspartner direkt, nachdem er seinen Aufmerksamkeitsfokus auf den Agenten ausrichtete, einen neuen mentalen Zustand angenommen hat, oder ob der mentale Zustand bereits vorher bestand. Dies kann für Ursache-Wirkung-Analysen und auch für Überlegungen zum Einsatz kommen, ob ein Interaktionspartner ein bestimmtes Wissen zu einem bestimmten Zeitpunkt schon haben konnte.

### *Belief-Chunks* mit Aktivierungswerten

Für das konzipierte dynamische Arbeitsgedächtnis verfügen *Belief-Chunks* in der CASEC-Architektur über individuelle **Aktivierungswerte**. Der Aktivierungswert wirkt sich auf die Präsenz der *Belief-Chunks*, ihre Auswahlwahrscheinlichkeit sowie ihre Verweildauer im Arbeitsgedächtnis aus. Damit geht die CASEC-Architektur in Bezug auf die kognitive Modellierung der Annahmen des Agenten weit über andere BDI-Systeme hinaus. Um Aussagen über die *Beliefs* eines Agenten  $i$  machen zu können, wird eine modallogische Notation eingeführt (siehe Anhang A.1 Definition 19).

**Definition 5** Sei  $(BEL_i \varphi)$  eine Formel, dann ist auch  $(BEL_i \varphi a)$ ,  $a \in \mathbb{R}^+$  eine Formel. Die Funktion  $Acti(BEL_i \varphi a) = a$  liefert den aktuellen Aktivierungswert  $a$  einer Formel zurück.

Abbildung 5.7: Berücksichtigung zeitlicher Aspekte bei *Belief-Chunks*

Durch die Berücksichtigung der Aktivierungswerte wird eine weitere Dimension in die Repräsentation der Beliefsmenge eingeführt, die es ermöglicht, die einzelnen Beliefs zusätzlich zu ihrem Inhalt zu strukturieren (siehe Abbildung 5.8).

Durch die Aktivierungswerte lässt sich eine *Dynamik der Kontextsensitivität* in das *BDI-System* integrieren. Die Aktivierung bezieht sich dabei sowohl auf einzelne Terme als auch auf die Aktivierung eines gesamten *Belief-Chunks*.

Für Terme gilt die folgende Einführung von Aktivierungswerten:

**Definition 6** Für eine gegebene Formel  $\varphi$ , welche aus  $n$  Termen besteht, bezeichne  $t\_set(\varphi)$  die Menge der Terme, welche in  $\varphi$  enthalten sind,  $t\_set(\varphi) := \{e_i \mid e_i \text{ terms of } \varphi, (i = 1, \dots, n)\}$ . Jeder Term  $e_i$  besteht dabei aus einem Wert des

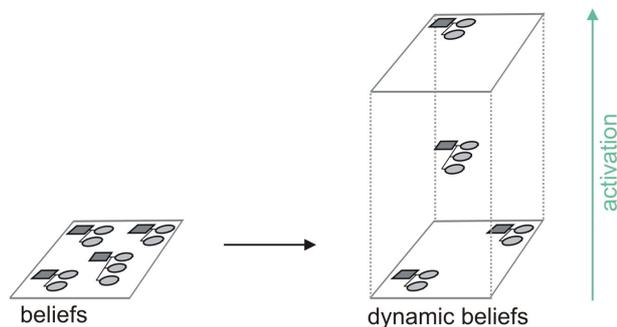


Abbildung 5.8: Erweiterung der Beliefs zu einer dynamischen Beliefmodellierung mit Aktivierungswerten

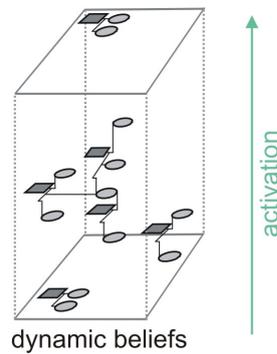


Abbildung 5.9: Darstellung der Aktivierungswerte der Terme

Terms  $\|e_i\|$  und einem Aktivierungswert  $a$ . Daher definieren wir, dass ein Term  $\hat{e}$  aus den folgenden Teilen besteht:  $\hat{e} := (\|e\|, a)$ ,  $a \in \mathbb{R}^+$ . Die Funktion  $Acti(\hat{e}) = a$  liefert den Aktivierungswert des Terms zurück.

Die Gesamtaktivierung eines *Belief-Chunks* berechnet sich aus der normierten Summe der Aktivierungswerte der enthaltenen Terme sowie des eigenen Aktivierungswerts des Prädikatsymbols, welches durch kognitive Prozesse eigene Aktivierungs-Impulse erfahren kann (siehe Abbildung 5.9). Damit folgt die Beziehung zwischen den Aktivierungswerten eines *Belief-Chunks* und den Aktivierungswerten der beinhalteten Terme der folgenden Gleichung, wobei  $\#e_i$  die Anzahl der Terme, ( $i=1, \dots, n$ ) bezeichnet:

$$Acti(\varphi) := \sum \frac{Acti(\hat{e}_i)}{\#e_i}, e_i \in t\_set(\varphi) \quad (5.6)$$

Um die Aktivierungswerte eines *Belief-Chunks* zu manipulieren, führen wir die folgenden Funktionen ein:

Sei  $\varphi$  ein *Belief-Chunk* und  $e_i \in t\_set(\varphi)$  so sei

$$IncreaseActi(e_i, \delta) := (\|e_i\|, a + \delta), \delta \in \mathbb{R}^+ \quad (5.7)$$

$$IncreaseActi(\varphi, \delta) := \forall e_i \in t\_set(\varphi) : IncreaseActi(e_i, \delta), \delta \in \mathbb{R}^+ \quad (5.8)$$

Der Aktivierungswert einer *Belief-Formel* errechnet sich aus dem Mittelwert der Aktivierungswerte der enthaltenen *Belief-Chunks*.

### Aktuelle Beliefs

Die Beliefs eines Agenten  $i$  bestehen aus der Menge der *Belief-Formeln*, die sich aktuell im *Episodic Buffer* seines Arbeitsgedächtnisses befinden.

**Definition 7**  $Beliefs_i := \{b \mid b : BeliefFormula \wedge BEL_i(b)\}$

Die Menge der *aktuellen Beliefs* eines Agenten  $i$  ( $^{cur}Bels_i$ ) wird folgendermaßen definiert:

**Definition 8** Sei  $Beliefs_i$  die gesamte Menge der Beliefs des Agenten  $i$ . Dann gilt  $^{cur}Bels_i := \{b_x \mid b_x \in Beliefs_i \wedge Acti(b_x) > \theta BEL_{acti}\}$

$\theta BEL_{acti}$  repräsentiert dabei einen Schwellwert, welcher dynamisch angepasst wird, sodass immer nur eine begrenzte Anzahl an *Belief-Formeln* in der Menge der  $^{cur}Bels_i$  vorgehalten wird (siehe Abbildung 5.4). Eine Möglichkeit den Schwellwert  $\theta BEL_{acti}$  festzulegen besteht darin, ihn aus dem Mittelwert der Aktivierungswerte aller vorliegenden *Belief-Formeln* zu bestimmen.

$$\theta BEL_{acti} = \frac{\sum_{b_x \in Beliefs_i} Acti(b_x)}{\#b_x \in Beliefs_i} \quad (5.9)$$

Ist die Anzahl der Elemente der Menge  $^{cur}Bels_i$  größer als 10 so kann der Schwellwert  $\theta BEL_{acti}$  sukzessive erhöht, bis die Anzahl der Elemente der Menge  $^{cur}Bels_i \leq 10$  ist. Diese Menge der  $^{cur}Bels_i$  modelliert die *Region des direkten Zugriffs*, wie von Oberauer postuliert (siehe Abschnitt 4.1.3). Abbildung 5.8 illustriert die Erweiterung der klassischen Belief-Modellierung in eine dynamische Modellierung, welche auf Aktivierungswerten und automatischen Adaptationsprozessen beruht. Durch die Einführung von Aktivierungswerten wird eine zusätzliche Dimension in die Belief-Modellierung eingeführt, welche dynamische Filterungsmechanismen ermöglicht. Kognitive Kontrolle und Aufmerksamkeit können damit als emergente Eigenschaften der Informationsrepräsentation im Arbeitsgedächtnis gesehen werden.

Durch die Modellierung der *Beliefs* als dynamisches Arbeitsgedächtnis wird es dem Agenten ermöglicht, zwischen der Relevanz von Fakten zu unterscheiden. In einem klassischen, logischen Inferenzverfahren werden alle Fakten gleich behandelt. Mit Hilfe des dynamischen Arbeitsgedächtnisses kann jedoch die Aktualität/Neuheit der Fakten berücksichtigt werden. Ist die Menge der aktuell vorliegenden Fakten nicht ausreichend, wird zusätzlich Information aus weiteren Gedächtnissen aktiviert (siehe Abschnitt 5.1.7).

### Vernetzungstruktur der Beliefs

Die Repräsentation der Beliefs im Arbeitsgedächtnis greift Ansätze des Modells der semantischen Netze auf. Abbildung 5.11 zeigt ein Beispiel der Konzeption des Arbeitsgedächtnisses als **semantisches Netz**, in welchem die Relationen miteinander verknüpft sind, die sich auf verschiedene Objekte und Konzepte beziehen.

Vorteile der vernetzten Struktur bestehen darin, dass zusammengehörige Informationen auch gemeinsam repräsentiert und abgerufen werden können und dass Aktivierungswerte individuell für jeden einzelnen Term vorgehalten werden und Aktivierungsimpulse auf verwandte Relationen überspringen können (siehe Abschnitt 5.1.5). Damit geht die CASEC-Architektur auf die Kritik der inflexiblen Gruppierungsstruktur der Chunks in Act-R ein (Schultheis et al., 2006).

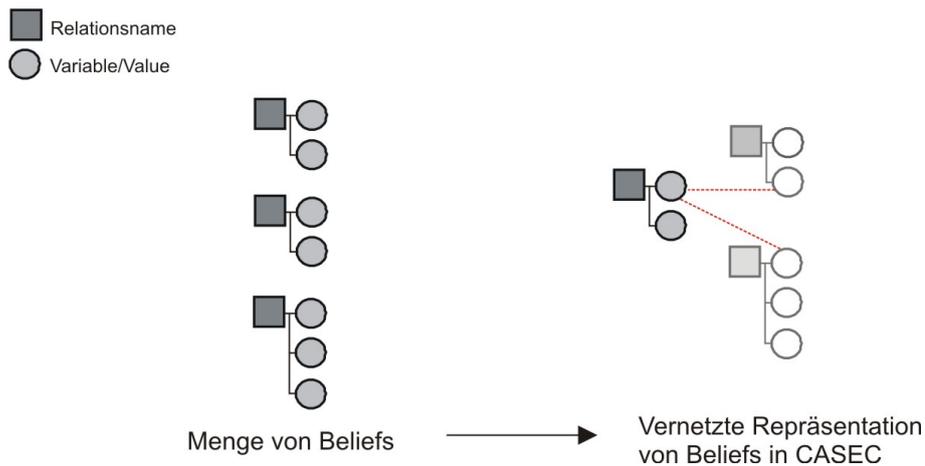


Abbildung 5.10: Vernetzte Repräsentation von *Belief-Chunks*

Neben der Repräsentation der *Belief-Chunks* in Form einer Menge, wird komplementär dazu eine Vernetzungsstruktur aufgebaut, welche die *Belief-Chunks* untereinander verknüpft (siehe Abbildung 5.10). Durch Vorkommnisse eines Wertes oder einer Variablen in mehreren *Belief-Chunks* lassen sich diese in hierarchisch organisierten, netzartigen Strukturen repräsentieren. Dazu wird parallel zu der Liste der *Belief-Chunks* eine Verknüpfungstabelle erstellt. Sie sorgt dafür, dass gleiche Werte bzw. Objekte über Links miteinander verbunden werden und sich so gegenseitig aktivieren und beeinflussen können.

Abbildung 5.11 stellt die klassische Repräsentation der Relationen in Form einer Menge von Relationen der Repräsentation in Form eines Netzes gegenüber.

### 5.1.5 Aktivierungsmechanismen

Abbildung 5.12 veranschaulicht das dynamische Arbeitsgedächtnis der CASEC-Architektur. Die Rhomben repräsentieren die *Belief-Chunks* mit ihren Aktivierungswerten, die im *Episodic Buffer* vorliegen. Die Säulen an der Spitze der Rhomben geben dabei den Grad der Aktivierung des *Belief-Chunks* an. Die großen Pfeile verdeutlichen, welche Komponenten Elemente in den *Episodic Buf-*

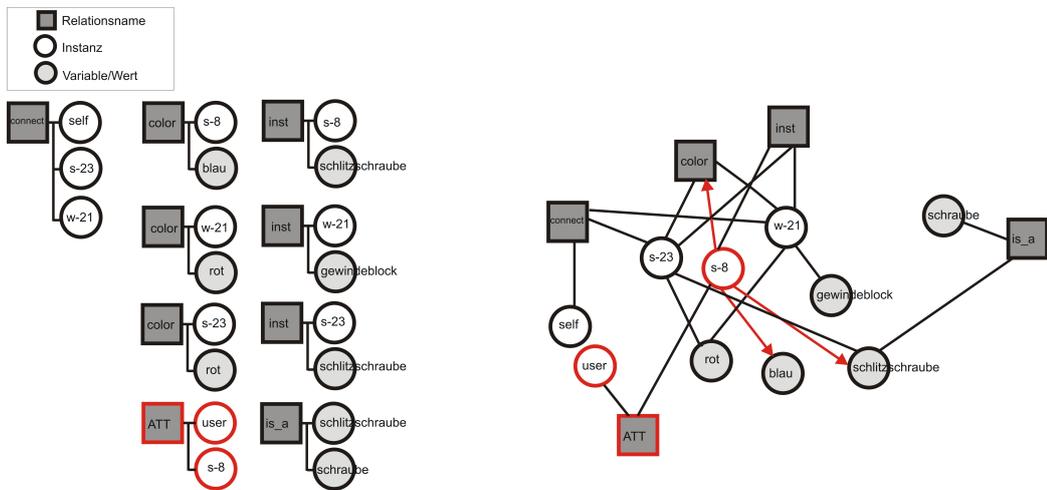


Abbildung 5.11: Liste von Relationen - Netz von Relationen

fer eintragen (*Visual Spatial Memory, Phonological Loop, Long-Term-Memory, Intentions*).

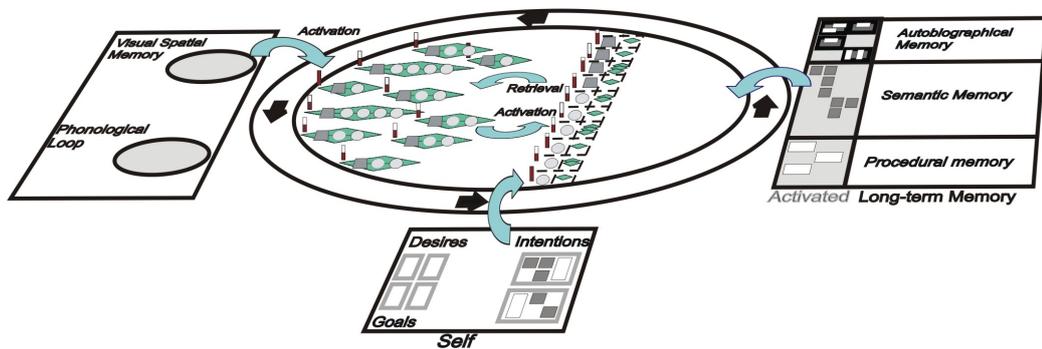


Abbildung 5.12: Dynamisches Arbeitsgedächtnismodell

Die Aktivierungswerte der Elemente des Arbeitsgedächtnisses können durch Ereignisse, interne Verarbeitungsprozesse sowie durch deliberative Entscheidungen des Agenten beeinflusst werden. Zusätzlich zu den automatischen Aktivierungsprozessen, welche die Aktivierungswerte immer dann erhöhen, wenn das Objekt in den Blickfokus des Agenten fällt, erhöhen deliberative Mechanismen die Aktivierungswerte, wenn das Objekt in interne Verarbeitungsprozesse involviert ist. Dadurch werden die Mechanismen, die hinter dem Phänomen der Aufmerksamkeit stehen modelliert. Im Gegensatz zu den Ansätzen, in welchen Aufmerksamkeit durch einen Stack modelliert wird, auf welchen Objekte deliberativ geschoben werden. Der Aufmerksamkeitsfokus des Agenten emergiert aus seinen

Zielen, seinem Verhalten und seiner Interaktion mit der Umwelt. In dem CASEC-Arbeitsgedächtnis werden folgende Mechanismen der Aktivierung berücksichtigt:

**Aktivierungsmechanismen:**

- *Perceptual Priming* - Aktivierung durch perzeptuelle Wahrnehmung
- *Retrieval* - Aktivierung durch deliberative Zugriffe
- *Spreading Activation* - Propagierung von Aktivierung zu verwandten Belief-Chunks
- *Maintenance* - Mechanismen der Aufrechterhaltung zielrelevanter Informationen

**Perceptual Priming**

*Perceptual Priming* stellt eine Form der Aktivierung aufgrund perzeptueller Wahrnehmung dar. Objekte, die im *Visual Spatial Memory* des Agenten wahrgenommen werden (siehe Abschnitt 5.1.1), sorgen für einen Aktivierungsimpuls, der sich sowohl auf relevante Elementen im *Episodic Buffer* als auch auf Elemente im aktivierten Langzeitgedächtnis auswirkt. Immer wenn ein neues Objekt wahrgenommen oder erneut fokussiert wird, wird Information über seine relevanten Attribute sowie seine eindeutige ID in das Arbeitsgedächtnis geschrieben. Die entsprechenden *Belief-Chunks* werden aktualisiert sowie die Aktivierungswerte der involvierten *Belief-Chunks* erhöht. Richtet sich der Blick des Agenten gezielt auf ein Objekt, so erhält es einen zusätzlichen Aktivierungsimpuls und es wird in diesen des aktivierten Langzeitgedächtnisses aufgenommen, sollte es noch nicht Element dieses Bereichs sein.

In der CASEC-Architektur sind drei verschieden starke Aktivierungsimpulse vorgesehen:  $\delta_{small}$ ,  $\delta_{middle}$ ,  $\delta_{large}$ .

Das *Perceptual Priming* lässt sich folgendermaßen fassen: (Definition von  $P$  siehe Anhang A.1, Definition 23, *IncreaseActi* siehe Abschnitt 5.1.4 Def 5.7)

$$\{P(\varphi)\}_i \Rightarrow (BEL_i \varphi) \wedge \forall e_x \in t\_set(\varphi) : IncreaseActi(e_x, \delta_{middle}) \quad (5.10)$$

**Retrieval - Maintenance**

Beliefs werden auch aufgrund von Inferenzprozessen und mentalen Operationen in das Arbeitsgedächtnis eingetragen. Zudem können durch explizite Abfrage-

mechanismen relevante Konzepte und Informationen aktiviert und damit in den *Episodic Buffer* überführt werden. Ein weiterer zentraler Mechanismus besteht in der automatischen Aktivierung durch *Retrieval*. Wird auf die Inhalte des Arbeitsgedächtnisses zugegriffen, beispielsweise im Rahmen einer Suche nach einem bestimmten Inhalt, oder wenn der Agent ein Objekt auswählt oder sich für ein Belief entscheidet, so erhalten alle Kandidaten, welche zunächst als Lösung in Frage kamen, während des Prozesses des Abschätzens einen kleinen Aktivierungsimpuls  $\delta_{small}$ . Stimmen sie mit allen Bedingungen überein und werden ausgewählt, so erhalten sie einen weiteren größeren Impuls  $\delta_{large}$ . ( $\neg$ : *converse operator*, steht in der *Propositional-Dynamic-Logic* dafür, ein Programm rückwärts laufen zu lassen (Lorini et al., 2005); in diesem Fall bezieht er sich auf den *vorherigen* Zustand, *unify* bezieht sich auf den Unifikationsalgorithmus der CASEC-Architektur siehe Abschnitt 5.1.7)

$$\begin{aligned} HAPPENS(BEL_i \varphi) \quad \wedge \quad ((BEL_i \varphi)^- false) \\ \Rightarrow \quad \forall e_x \in t\_set(\varphi) : IncreaseActi(e_x, \delta_{small}) \end{aligned} \quad (5.11)$$

$$\begin{aligned} \{Retrieve_i \varphi\} \quad \Rightarrow \quad \forall \vartheta \in {}^{cur}Bels_i \mid unify(\varphi, \vartheta) : IncreaseActi(\vartheta, \delta_{small}) \wedge \\ \forall \vartheta_x \mid \{Commit_i \vartheta_x\} : IncreaseActi(\vartheta_x, \delta_{large}) \end{aligned} \quad (5.12)$$

Da *Commit* und *Retrieve* komplexe Handlungsausdrücke darstellen, werden geschweifte Klammern eingesetzt, um die Handlungsausdrücke von Prädikaten und Modaloperatoren zu unterscheiden.

Einerseits führt eine von den *Retrieval*-Mechanismen ausgehende Aktivierung dazu, relevante Informationen im Arbeitsgedächtnis zu bewahren. Andererseits ist auch ein eigener Mechanismus vorgesehen, um zielrelevante Informationen im Arbeitsgedächtnis vorzuhalten. Dabei üben die aktuellen Ziele des Agenten einen Einfluss auf die Aktivierung der *Belief-Chunks* und der Ziele aus, wie von E. Miller & Cohen (2001) postuliert. Diese Aktivierungsimpulse sorgen dafür, dass zielrelevante Informationen nicht verloren gehen und im Aufmerksamkeitsfokus bleiben. Dabei werden die *Chunks*, welche in den Intentionen des Agenten vorkommen, auch in das Arbeitsgedächtnis eingetragen und in die allgemeinen Aktivierungsprozesse in gleicher Weise wie die Belief-Elemente des Arbeitsgedächtnisses mit einbezogen. Immer wenn ein Ziel als neues Ziel angenommen oder ausgewählt wird, erhält die entsprechende Zielformel einen Aktivierungsimpuls. Das Annehmen eines Ziels zeigt seine Relevanz in der aktuellen Situation an und wird reflektiert durch die Erhöhung des Aktivierungswerts:

$$\begin{aligned} HAPPENS \quad (CurrentGoal(GOAL_i \varphi)) \\ \Rightarrow \quad \forall e_x \in t\_set(\varphi) : IncreaseActi(e_x, \delta_{middle}) \end{aligned} \quad (5.13)$$

### Spreading Activation

Die Mechanismen der *Spreading Activation* der CASEC-Architektur gehen in ihrem Ursprung auf die Ansätze von Anderson (1983) zurück und bauen auf den Ideen von Engle und Kollegen auf, *Spreading Activation*-Mechanismen als einen beeinflussenden Faktor des *Retrievals* zu sehen (siehe Abschnitt 4.1.3).

In der CASEC-Architektur wird *Spreading Activation* umgesetzt, wobei Impulse jedoch nur bis zu einer bestimmten Tiefe (Defaultwert  $depth = 2$ ) weiter propagiert werden. Erhält ein Chunk einen größeren Aktivierungsimpuls ( $impulse \geq \delta_{large}$ ), so wird dieser an seine Terme weitergegeben. Diese wiederum propagieren die Aktivierung in einem weiteren Schritt an die mit ihnen verlinkten Chunks, welche die entsprechenden Terme enthalten und den Impuls an alle weiteren enthaltenen Terme propagieren. Dabei greift der Mechanismus auf die vernetzte Struktur des Arbeitsgedächtnisses zurück und auf die Repräsentation der einzelnen Terme. Die Aktivierung wird aktuell nur in zwei Schritten propagiert, sodass zwar zusammenhängende *Chunks* sich gegenseitig aktivieren und damit assoziative Verbindungen unterstützt werden, jedoch keine rechenintensive vielschrittige Propagierung vorgenommen wird (siehe auch Beispiel 5.13).

Aufgrund der Propagierung der Aktivierungsimpulse kann ein Aktivierungsschwellwert überschritten werden, der dazu führt, dass ein betroffener *Belief-Chunk* in den jeweiligen nächsten Bereich des Arbeitsgedächtnisses übertritt.

$$\begin{aligned} IncreaseActi(\vartheta_x, \delta_{large}) \Rightarrow \forall \varphi \in^{cur} Bels \mid e_x \in t\_set(\vartheta) \wedge \\ e_x \in (t\_set(\varphi)) : IncreaseActi(\varphi, \delta_{small}) \end{aligned} \quad (5.14)$$

Abbildung 5.13 zeigt ein Beispiel für die Aktivierungspropagierung innerhalb des *Episodic Buffer*. Der *Belief-Chunk* `attention_focus „user“ „s-8“` wurde dem *Episodic Buffer* hinzugefügt. Dies führt in einem ersten Schritt zu der Aktivierung des Relationsnamens `attention_focus` sowie zu der Aktivierung der Entitäten `s-8` und `user`. In einem zweiten Schritt erfahren nun die damit verbundenen *Belief-Chunks* einen Aktivierungsimpuls. In dem vorliegenden Fall bezieht sich dies auf die Relationen `color „s-8“ „blau“` sowie `inst „s-8“ „schlitzschraube“`. In einem dritten Aktivierungsschritt werden die Entitäten dieser *Belief-Chunks* aktiviert. Somit erhöhen sich die Aktivierungswerte der Entitäten `schlitzschraube` und `blau`.

#### 5.1.6 Verfall der Aktivierung (*Decay*)

Neben den Aktivierungsfunktionen wird auch eine *Decay*-Funktion benötigt. Diese sorgt dafür, dass *Chunks*, auf die längere Zeit nicht zugegriffen wird, an Aktivierung verlieren und aus dem aktiven Bereich des Arbeitsgedächtnisses entfernt

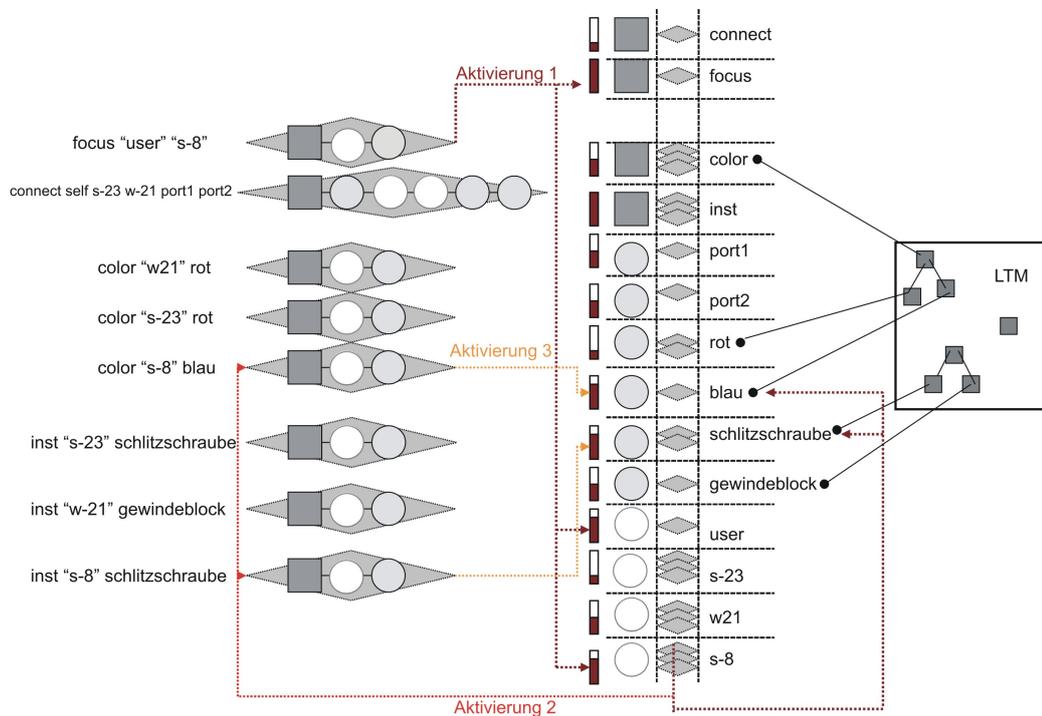


Abbildung 5.13: Aktivierung im Arbeitsgedächtnis

werden. In CASEC wird der Weg einer inkrementellen *Decay*-Funktion gewählt, welche zu jedem Zeitpunkt berechnet werden kann und den Aktivierungswert eines *Chunks* in Abhängigkeit der vergangenen Zeit logarithmisch reduziert. Der Zugriff auf einen *Belief-Chunk* wirkt sich in Form eines Aktivierungsimpulses zur Erhöhung des Aktivierungswerts aus. Um zu verhindern, dass der Aktivierungswert eines *Belief-Chunks* zu groß wird, ist eine Minimumfunktion eingesetzt, welche die Größe der maximalen Aktivierung eines *Belief-Chunks* auf 1 normiert.

**Definition 9**  $f(t + \Delta t) = \min(f(t)e^{-\lambda t}, 1)$ ,  $\lambda=0.2$

Auch in Act-R steigt und fällt die Basisaktivierung mit der Anzahl der Zugriffe und ihren zeitlichen Abständen (siehe Abschnitt 4.3.1, Gleichung 4.12). Für die Berechnung der Basisaktivierung muss jedoch die gesamte Historie der Zugriffe für jeden Chunk vorgehalten werden. Durch die inkrementelle Funktion in CASEC können der logarithmische *Decay* und die Aktivierungsimpulse der Zugriffe angenähert werden, ohne für jeden Chunk die Zugriffshistorie vorhalten zu müssen. Die Berechnung der angenäherten *Decay*-Funktion in CASEC ist zudem nicht so rechenintensiv. Während in Act-R allein der zeitliche Abstand der Zugriffe auf einen Chunk die *Base-level*-Aktivierung beeinflusst, können in

CASEC verschieden starke Aktivierungsimpulse den Aktivierungswert beeinflussen. So wird ein stärkerer Aktivierungsimpuls  $\delta_{large}$  ausgelöst, wenn ein Element des Arbeitsgedächtnisses ausgewählt wird, als wenn lediglich ein erneuter Zugriff auf dasselbe erfolgt ( $\delta_{small}$ ). Auch *Spreading Activation*-Mechanismen werden unterstützt.

Wie auch in ACT-R (siehe Abschnitt 4.3.1, Gleichung 4.13) werden Aktivierungsimpulse des aktuellen Ziels in die Gesamtaktivierung eines *Chunks* einbezogen. Während in ACT-R dafür die Stärke der Verbindungen zwischen dem Ziel-*Chunk* und dem *Belief-Chunk* mit einberechnet werden, wirken sich die Aktivierungsimpulse in CASEC direkt auf die Aktivierung des *Belief-Chunks* aus. Die *Decay*-Rate kann dabei so bestimmt werden, dass sich *Belief-Chunks*, welche durch ein aktuelles Ziel mit Aktivierungsimpulsen der Stärke  $\delta_{middle}$  in einem Takt  $t$  ( $t = 0,5ms$ ) aktiviert werden, im Arbeitsgedächtnis halten, ohne dass der Aufmerksamkeitsfokus direkt auf sie ausgerichtet ist. Abbildung 5.14 zeigt die Aktivierungsfunktion eines *Belief-Chunks*, welches zunächst regelmäßige Aktivierungsimpulse der Stärke  $\delta_{middle} = 0.3$  erhält, dann nicht mehr aktiviert wird und in Vergessenheit gerät, nach einiger Zeit aber durch einen Aktivierungsimpuls der Stärke  $\delta_{large} = 1$  wieder in den Fokus gerät.

Im Gegensatz zu Act-R, in welchem der Aktivierungswert nur im Falle eines Zugriffs direkt in Abhängigkeit des aktuellen Kontextes berechnet wird, werden die Aktivierungswerte in der CASEC-Architektur unabhängig von Zugriffen in festen Abständen berechnet. Damit erfasst der Aktivierungswert die Interaktionshistorie des Elements, und es kann bestimmt werden, wann der *Belief-Chunk* unter den Schwellwert fällt und den Bereich des direkten Zugriffs bzw. das Arbeitsgedächtnis verlassen muss. Durch die von direkten Zugriffen unabhängige Berechnung der Aktivierungswerte können auch Elemente des Arbeitsgedächtnisses in den Aufmerksamkeitsfokus rücken, nach denen nicht aktiv gesucht wurde. Der Schwellwert, bei welchem ein Element des Arbeitsgedächtnisses den Bereich des direkten Zugriffs verlassen muss, liegt bei  $\theta BEL_{acti}$  (siehe Abschnitt 5.1.4). Die Aktivierungs- und *Decay*-Funktionen in der CASEC-Architektur bieten den Vorteil, die Aktivierungsmechanismen in Abhängigkeit der Relevanz der Aktivierungsimpulse regeln können. Die Funktionen ermöglichen die Modellierung von *Priming*-Effekten.

### 5.1.7 Zugriff auf Gedächtnisinhalte (*Retrieval - Unification*)

Der Zugriff auf Elemente des *Episodic Buffer* setzt sich aus einem mehrschrittigen Prozess zusammen. Die Basis dafür liefert ein **Unifikationsalgorithmus** *unify*, welcher für zwei *Belief*-Formeln entscheidet, ob sie bei der gegebenen Variablenbelegung (vorliegende *Bindings*) miteinander unfizierbar sind d.h. ob es eine gültige Belegung gibt, sodass die Variablen der Formeln so durch geeignete

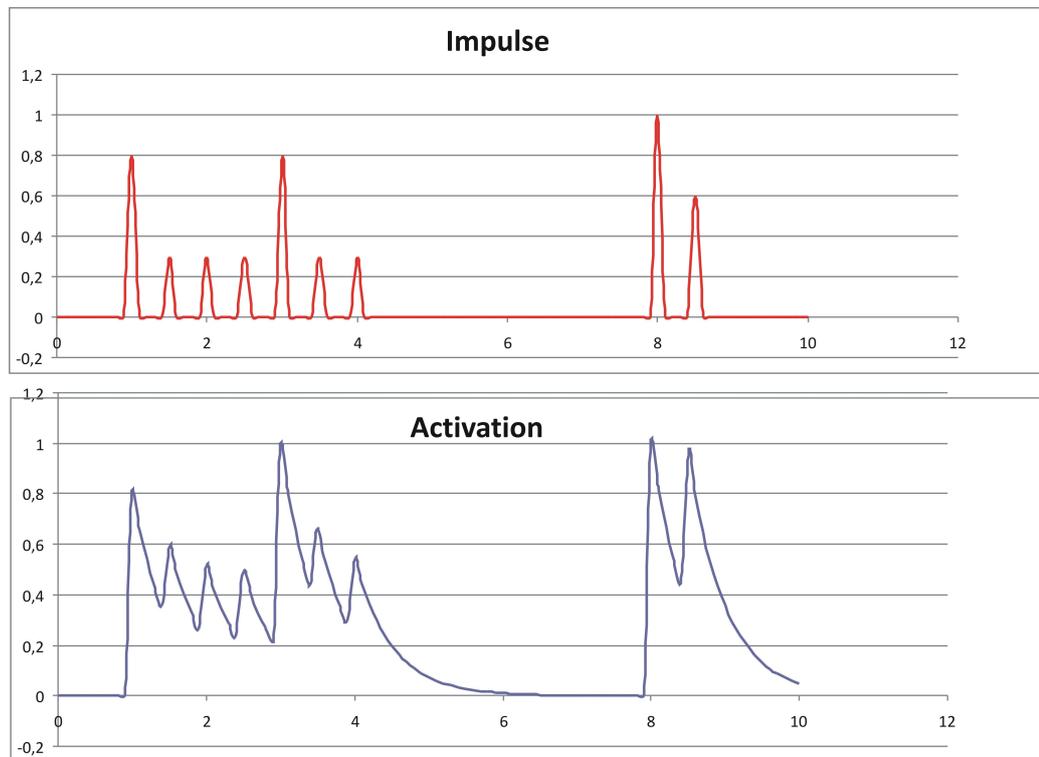


Abbildung 5.14: Impulse, Aktivierungsfunktion und *Decay*

Terme ersetzt werden, dass die resultierenden Ausdrücke gleich sind (siehe Anhang A.2.1 *Algorithm 14*).

Die Zugriffsfunktion nutzt den Unifikationsalgorithmus. Geht man von einer Abfragestrategie aus, bei welcher der erste passende Kandidat zurückgegeben wird, so spielt die Reihenfolge eine entscheidende Rolle, in welcher potentielle Kandidaten geprüft werden. CASEC unterstützt dabei einen **Matching-Algorithmus**, der die Aktivierungswerte der einzelnen *Belief-Chunks* berücksichtigt und damit den kontextsensitiven Zugriff und die Zugriffsgeschwindigkeit von relevanten Elementen modelliert. Aufbauend auf dem *Matching*-Algorithmus unterstützt CASEC verschiedene **Abfragestrategien**. So können bei der Abfrage zusätzlich beispielsweise zeitliche Constraints berücksichtigt werden, z.B. ob ein *Belief* zu einem bestimmten Zeitpunkt bereits vorlag oder nicht.

Im Folgenden werden der Unifikationsalgorithmus, der *Matching*-Algorithmus sowie die verschiedenen Abfragestrategien im Detail vorgestellt.

## Unifikation

Der Unifikationsalgorithmus überprüft als erstes den Relationsnamen und die Stelligkeit der Relationen. Stimmen diese überein, so werden sukzessive die einzelnen Terme der Relation miteinander verglichen. Hierbei ähnelt er dem Unifikationsalgorithmus von JAM und UM-PRS.

Der Unifikationsalgorithmus der CASEC-Architektur ist jedoch um zentrale Kontextaspekte entscheidend erweitert (siehe Anhang A.2.1). Im Rahmen der Unifikation werden auch Negation und gegebenenfalls spezifizierte *Constraints* berücksichtigt und im Falle verschachtelter Relationen wird der Unifikationsalgorithmus rekursiv aufgerufen.

Konkrete Werte können direkt miteinander verglichen werden. Für Variablen werden die Variablenbindungen ausgewertet und gegebenenfalls die *Constraints* abgeglichen. Dabei werden eventuell benötigte Informationen dynamisch aus dem Arbeitsgedächtnis oder weiteren Gedächtnissen abgeleitet, um den Wahrheitswert eines *Constraints* bestimmen zu können. Wurde beispielsweise spezifiziert, dass eine Variable ein Objekt eines bestimmten Typs darstellen soll (*is\_a* Constraint), so wird für die potentiellen Kandidaten abgeleitet, zu welchem Objekttyp sie gehören und ob der angegebene Objekttyp ein Untertyp des gesuchten ist. Diese Informationen werden in das Arbeitsgedächtnis des Agenten eingetragen. Solche Prozesse finden automatisch statt. Nach dem Verifizieren der Constraints wird unter Einbezug der neuen Information die Unifikation fortgesetzt.

## Matching

Eingesetzt wird der Unifikationsalgorithmus im Rahmen des *Match*-Algorithmus (siehe Anhang A.2.2). Dieser übernimmt die Steuerung, darüber für welche Relationen die Unifizierung versucht wird. Nachdem aufgrund der Aktivierungs- und Abfragemechanismen eine Basismenge an Fakten in den aktuellen Beliefs vorliegt, wird im Falle des *Retrieval* der zuvor beschriebene Unifikationsalgorithmus (siehe 5.1.7) eingesetzt, der die *Beliefs-Chunks* mit der gesuchten Abfrage vergleicht. Gibt es keinen passenden Kandidaten, werden automatische Aktivierungsmechanismen angestoßen.

Zunächst werden anhand des *Relationsnamens* die relevanten *Belief-Chunks* aus dem Arbeitsgedächtnis abgerufen. Von diesen Elementen werden die Kandidaten getestet, welche in Bezug auf die Stelligkeit mit dem *Abfragechunk* übereinstimmen und über den höchsten Aktivierungswert verfügen. Dabei werden jedoch nur solche *Belief-Chunks* berücksichtigt, welche noch als potentielle *Kandidaten* gelten und noch nicht im Rahmen der Anfrage deaktiviert wurden. Diese potentiellen *Belief-Chunks* erhalten zunächst einen kleinen Aktivierungsimpuls, da sie als Kandidaten einer Abfrage in den Aufmerksamkeitsfokus gerückt sind.

Danach werden iterativ die einzelnen Terme getestet, d.h. der Abfrage-Chunk wird mit einem Kandidaten-Chunk verglichen und es wird geprüft, ob sie sich unifizieren lassen. Im Falle einer Übereinstimmung des *Belief-Chunk* mit der Anfrage erfährt der *Belief-Chunk* einen größeren Aktivierungsimpuls und geht in den Aufmerksamkeitsfokus im Sinne von Oberauer über (siehe 4.1.3). Unabhängig davon, ob die Elemente unifiziert werden können, wird der Kandidaten-Chunk für weitere Berechnungen bezüglich des aktuellen *Retrieval*-Aufrufs deaktiviert, so dass bei einer weiteren Abfrage das Element mit der nächsthöheren Aktivierung berücksichtigt wird. Dieser Mechanismus wird beispielsweise für Abfragestrategien benötigt, welche eine Menge von *Lösungs-Chunks* zurück liefern.

Damit ist die Abfragestrategie in CASEC im Vergleich zu den anderen Systemen in Bezug auf zwei zentrale Aspekte erweitert. Die Reihenfolge, in der Kandidaten überprüft werden, wird nicht dem Zufall überlassen, sondern hängt von den Aktivierungswerten der *Belief-Chunks* ab. Der Zugriff geschieht also kontextsensitiv. Die Abfrage selbst sendet neue Aktivierungsimpulse aus und beeinflusst damit die Zugreifbarkeit der *Belief-Chunks*.

Abbildung 5.15 zeigt exemplarisch den Prozess eines *Retrieval*-Aufrufs. Zunächst wird anhand des Relationsnamens des Abfrage-Chunks die Menge der potentiellen Kandidaten mit gleichem Relationsnamen aus der Menge der Beliefs abgerufen. In dieser Menge wird dann zunächst der *Belief-Chunk* mit der höchsten Aktivierung überprüft. Der *Belief-Chunk* erhält dabei einen kleinen Aktivierungsimpuls. Da die Unifizierung mit diesem *Belief-Chunk* nicht erfolgreich ist, wird der *Belief-Chunk* im Rahmen dieser Abfrage deaktiviert und der *Belief-Chunk* mit dem nächsthöchsten Aktivierungswert wird getestet. In diesem zweiten Schritt, lässt sich auch keine Unifikation finden. Erst im dritten Schritt wird ein passender *Belief-Chunk* ermittelt.

### Abfragestrategien

Die kontextsensitive Verarbeitung von Informationen und die Adaptation an eine konkrete Aufgabenstellung erfordert eine **Kontrolle der Kodierungs- und Abfragestrategien**, wie sie von Baddeley der *Central Executive* zugeschrieben wird (siehe Abschnitt 4.1.3). Diese wird in der CASEC-Architektur durch variable Abfragemechanismen des Interpreters umgesetzt. Die verschiedenen Abfragestrategien bauen alle auf den Unifikations- und Matchingalgorithmen auf. Zusätzlich können bei den Abfragestrategien weitere Aspekte berücksichtigt werden. Der *Zugriffs*-Mechanismus liefert je nach angewandter Abfragestrategie den besten Kandidaten bzw. eine Menge an besten Kandidaten zurück.

Für den Basisabfragemechanismus  $FACT_i$  operiert die Funktion auf der Menge der aktuellen Beliefs  ${}^{cur}Bels_i$  des Agenten  $i$ . Die Funktion liefert ein Element der aktuellen Beliefs zurück und falls die Menge der aktuellen Beliefs keinen gültigen

Abfrage Chunk: 

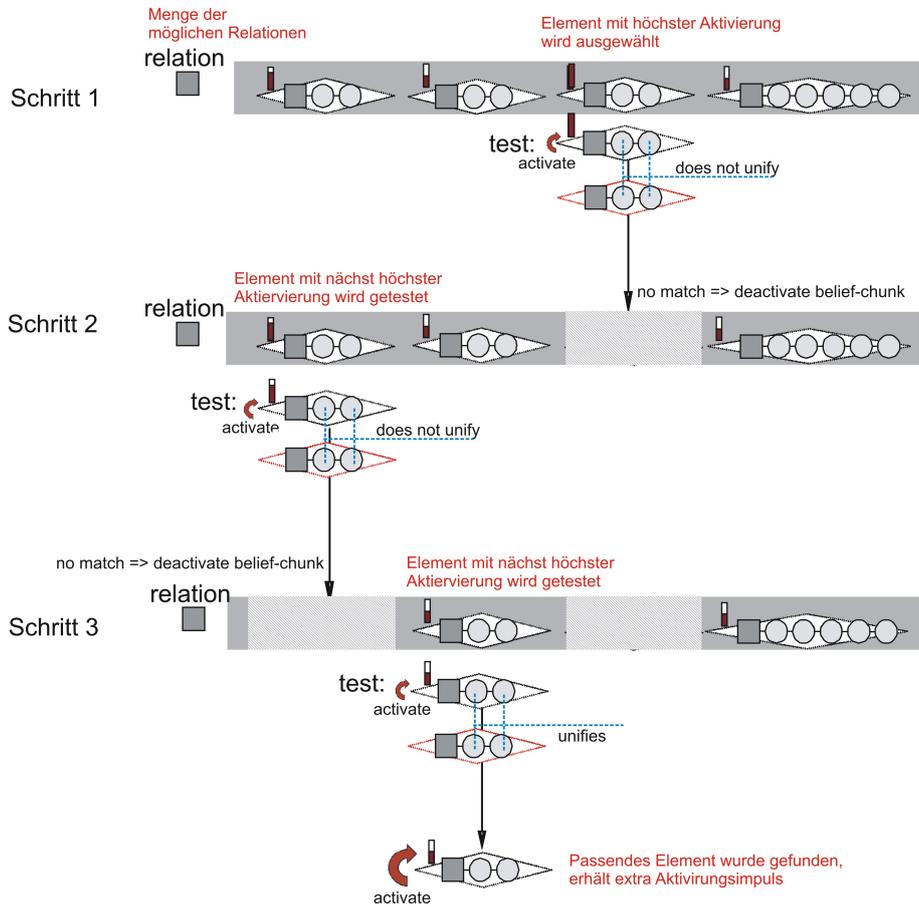


Abbildung 5.15: Arbeitsgedächtnis - Retrieval

Kandidaten enthält, wird die leere Menge zurückgegeben. *unify* bezieht sich auf den Unifikationsalgorithmus der CASEC-Architektur siehe Abschnitt 5.1.7.

$$FACT_i(\varphi) = \begin{cases} \vartheta, & \text{falls } \exists \vartheta \in {}^{cur}Bels_i \wedge unify(\vartheta, \varphi) \\ \emptyset, & \text{sonst} \end{cases} \quad (5.15)$$

Es gibt auch einen Abfragemechanismus unter Berücksichtigung der Aktivierungswerte:

$$FACT_i(\varphi) = \begin{cases} \vartheta, & \text{falls } \exists \vartheta \in {}^{cur}Bels_i \wedge unify(\vartheta, \varphi) \\ & \wedge \forall \beta_x \in {}^{cur}Bels_i \wedge unify(\vartheta, \beta_x) : Acti(\vartheta) \geq Acti(\beta_x) \\ \emptyset, & \text{sonst} \end{cases} \quad (5.16)$$

Neben der ersten Übereinstimmung kann in CASEC auch die Menge aller passenden Kandidaten zurückgegeben werden:

$$FACT\_ALL_i(\varphi) = \{\vartheta \mid \vartheta \in {}^{cur}Bels_i \wedge unify(\vartheta, \varphi)\} \quad (5.17)$$

Darüber hinaus können zeitliche Aspekte berücksichtigt werden. Es kann beispielsweise der *Belief-Chunk* mit dem neusten Zeitstempel abgefragt werden:

$$FACT\_NEWEST_i(\varphi, t) = \begin{cases} \vartheta, & \text{falls } \exists \vartheta \in {}^{cur}Bels_i \wedge unify(\vartheta, \varphi) \\ & \wedge TimeStamp(\vartheta) \leq t \wedge \forall \beta_x \in {}^{cur}Bels_i \\ & \wedge unify(\vartheta, \beta_x) : TimeStamp(\vartheta) \leq TimeStamp(\beta_x) \\ \emptyset, & \text{sonst} \end{cases} \quad (5.18)$$

Ferner besteht die Möglichkeit, nur *Belief-Chunks* abzufragen, welche einen spezifizierten *Neuheitswert* besitzen (FACT\_NEWER) und demnach nach einem spezifizierten Zeitpunkt entstanden sind:

$$FACT\_NEWER_i(\varphi, t) \rightarrow \{\vartheta \mid \vartheta \in {}^{cur}Bels_i \wedge unify(\vartheta, \varphi) \\ \wedge TimeStamp(\vartheta) \leq t\} \quad (5.19)$$

### Automatische Aktivierung

Normalerweise greifen die Abfragemechanismen zunächst auf Informationen innerhalb der Region des direkten Zugriffs (*Episodic Buffer*) zurück. Aktivierungsimpulse werden jedoch auch zusätzlich in den aktivierten Teil des Langzeitgedächtnisses übertragen. Liegt kein passender Kandidat im *Episodic Buffer* vor, so wird die Abfrage an das aktivierte Langzeitgedächtnis respektive die untergeordneten *Slave-Systeme* weitergeleitet und versucht, die Information aus diesen Modulen abzuleiten. Handelt es sich um allgemeines, semantisches Wissen, so wird das semantische Langzeitgedächtnis nach passenden Informationen durchsucht. Bezieht sich die Information auf das Attribut eines speziellen Individuums, wird der Wahrnehmungsapparat des Agenten befragt.

### Mentale Manipulation

Das mentale Manipulieren der Beliefs im *Episodic Buffer* geschieht durch die Ausführung von primitiven Planaktionen. Diese können Schlussfolgerungsmechanismen umfassen und zu der Entstehung neuer Beliefs führen. Dabei können Variablen direkt mit Werten belegt, durch die Spezifikationen von *Constraints* eingeschränkt, oder neue Fakten abgeleitet und dem Wissenstand des Agenten hinzugefügt werden.

**Mentale Operationen:**

- ASSERT
- RETRIEVE
- RETRACT
- UPDATE
- CONSTRAIN
- ASSIGN
- TEST
- WAIT
- POST GOAL

Neben dem Manipulieren des aktuellen Wissensstandes des Agenten durch Assertieren und Entfernen von *Belief-Chunks*, können auch Entscheidungen in Handlungsplänen durch lokale *Bindings* repräsentiert werden. *Bindings* umfassen sowohl die Variablenbindungen der Ziel-Chunks bzw. Plan-Chunks als auch lokale Variablen, durch welche Entscheidungen repräsentiert werden können. Der Einsatz von *Constraints* erlaubt die Berücksichtigung von graduellen Entscheidungen. Durch die explizite Repräsentation wird der Agent in die Lage versetzt, sich der Entscheidungen bewusst zu werden und über seine Entscheidungen Auskunft zu erteilen. Die Beliefs werden in erster Linie in dem globalen Arbeitsgedächtnis vorgehalten. Es besteht jedoch auch die Möglichkeit, Zwischenergebnisse in Rahmen von lokalen Plänen zu vermerken.

**Attribute**

Die individuellen Attribute eines *Belief-Chunks* setzen sich neben seiner *Aktivierung* zusammen aus einem Zeitstempel (*time-stamp*) (Neuheitswert, wann wurde es in die Beliefs eingetragen), einem Wert, der den letzten Zugriff auf das Element vermerkt (*last-retrieved*) und einem *Saliency*-Wert - welche Valenz bzw. Salienz besitzt das Element.

### 5.1.8 Episodisches Gedächtnis

Das *Episodische Gedächtnis* dient als Kurzzeitgedächtnis für Ereignisse und Handlungen. Im Rahmen einer Interaktionssituation fungiert es zudem als *Dialoghistorie*. Der Agent benötigt ein Gedächtnis, um den Verlauf der Interaktion und Kooperation mitprotokollieren zu können, damit er Referenzen auflösen, aber auch Sackgassen erkennen und vermeiden kann. Die Einträge des *Episodischen Gedächtnisses* bestehen aus *Interaction-Moves* (siehe Abschnitt 3.1) und Ereignissen; sie werden entlang einer Achse der Intentionalität und der Zeit organisiert. Das *Episodische Gedächtnis* ist von dem *Episodic Buffer* zu unterscheiden. Während die Verweildauer von Informationen innerhalb des *Episodic Buffer* aus wenigen Sekunden bis Minuten besteht, sind es bei der Speicherdauer im mittelfristigen episodischen Gedächtnis Minuten bis mehrere Stunden (Conway & Pleydell-Pearce, 2000). Ericsson & Kintsch (1995) gehen davon aus, dass eine episodische Repräsentation der Diskursinhalte aufgebaut wird. Das episodische Diskursgedächtnis wird während des Verstehens aufgebaut, und die im momentanen Fokus der Aufmerksamkeit befindlichen Diskurselemente bieten einen raschen Zugriff auf die Strukturen des Langzeitgedächtnisses.

### 5.1.9 Semantisches Langzeitgedächtnis

Das semantische Langzeitgedächtnis ist in Form eines *semantischen Netzes* konzipiert. In diesem Netz sind Knoten für Konzepte, Objekte und Eigenschaften angeordnet. Zusammengehörige Konzepte sind über eine *is\_a*-Hierarchie verbunden. Zusätzlich gibt es *inst*-Knoten, welche angeben, dass ein Objekt eine Instanz einer bestimmten Objektklasse ist. Über *is\_a*- und *inst*-Verbindungen wird eine *Vererbungshierarchie* etabliert. Zusätzlich können Objektknoten über Attributverbindungen verfügen. Diese sind jeweils mit einem konkreten Wert verbunden, der jeweils wieder von einer Eigenschaftsklasse abgeleitet ist. *Part\_of*-Beziehungen und Rollenbeziehungen zwischen Knoten spiegeln Teil-Ganzes-Beziehungen wider und spezifizieren Zusammenhänge zwischen einzelnen Knoten. Als Eintrittspunkte in das semantische Netz gelten die Knoten, welche über eine besonders hohe Aktivierung verfügen. So können bereits aktivierte Gebiete des semantischen Netzes leichter betreten und durchsucht werden. Diese Knoten werden in einem zusätzlichen Bereich verwaltet, der den aktivierten Teil des semantischen Langzeitgedächtnisses repräsentiert (siehe 5.1.3). Neben dem Langzeitwissen speichert dieser Bereich auch Zwischenergebnisse, welche eventuell nicht mehr direkt im Arbeitsgedächtnis vorliegen, jedoch ihre Spuren in der Erinnerungsstruktur hinterlassen haben und aus dem Bereich leichter wieder aktiviert werden können. Aus dem semantischen Netz können *Belief-Chunks* dynamisch abgeleitet werden, die dann die jeweils relevante Abstraktionsstufe besitzen. Das Konzept der *relevanten*

*Abstraktionsstufe* soll durch das folgende Beispiel verdeutlicht werden.

Wird über vorliegende Objekte nur als „Baufixteile“ gesprochen und spielt der genaue Typ dabei aktuell keine Rolle, so können die Objekte als Instanz eines Baufixteils behandelt werden. Geht es jedoch um eine genaue Typbezeichnung, so wird diese aus dem semantischem Wissen abgefragt und aktiviert. Das Objekt wird im Folgenden dann als eine Instanz des konkreten Typs im Arbeitsgedächtnis geführt.

## 5.2 Kognitive Schleife

Auf den Gedächtnisstrukturen operieren die kognitiven Prozesse. Sie werden in der CASEC-Architektur in der kognitiven Schleife zusammengefasst. Zu ihnen zählen:

- Mechanismen der Zielauswahl (Deliberation)
- Generierung von Handlungsoptionen
- Auswahl eines Plans (*Means-Ends-Reasoning*)
- Mentale Operationen
- Instantiierung eines ausführbaren *Behavior*

### Variable Zielgerichtetheit

Deliberation besteht aus dem Prozess, etwas genau zu durchdenken und Eventualitäten zu berücksichtigen. In der Informatik wird der Begriff für den Entscheidungsprozess zwischen den zu verfolgenden Zielen verwendet. Insofern spielen bei der Deliberation Wünsche, Ziele und die Bildung von Präferenzen und Prioritäten eine Rolle. Es muss eine Gewichtung der einzelnen Zielsetzungen vorgenommen werden. In der CASEC-Architektur wird dieser Prozess aus einem Zusammenspiel von *top-down*- und *bottum-up*-Mechanismen umgesetzt, welche zu den kognitiven Basisprozessen zählen.

## 5.2.1 Ziele in CASEC

### Zielursprung - Zielentstehung

In der CASEC-Architektur wird zwischen explizit aufgeworfenen Zielen (*Goals*) und reaktiven Zielen (*Conclude-Goals*) unterschieden (siehe auch JAM 4.3.1). Während *Conclude-Goals* sich in ihrem Ursprung nicht auf explizite Ziele des Agenten beziehen, sondern reaktiv auf Fakten, Erkenntnisse oder Wahrnehmungen eingehen und eine kontextsensitive Bearbeitung von Zielen organisieren, zielen die *Goals* direkt auf das Erreichen eines spezifizierten Weltzustands oder die Ausführung einer bestimmten Handlung ab. *Conclude-Goals* können für Mechanismen der vorausschauenden Kontrolle eingesetzt werden; sie stellen Ziele dar, die zu einem geeigneten Zeitpunkt aktiv werden (siehe Kognitive Kontrolle, Abschnitt 4.1.1).

Eine besondere Form der Ziele stellen *Obligationen* dar, welche in CASEC explizit als besondere Form der *Goals* repräsentiert werden.

### Zielspezifikation - Zielrepräsentation

Die CASEC-Architektur verfolgt den Ansatz *expliziter Zielrepräsentation*. Damit wird der Agent in die Lage versetzt, über seine Ziele Auskunft zu erteilen.

Ein Ziel besteht dabei aus der Spezifikation eines Zieltyps (*Goal Type*) und einer Zielformel (*Formula  $\phi$* ) (siehe Abbildung 5.16). Die Zielformel baut in ihrer Repräsentation auf dem Repräsentationsformat der *Belief-Formel* auf (siehe Abschnitt 5.1.4) und profitiert von allen Erweiterungen der *Belief-Formel*. Die *Constraint*-Repräsentation wird dazu genutzt, Ziele in einer abstrakten, unterspezifizierten Form repräsentieren zu können, welche dann in einer Interaktion oder im Rahmen eines Lösungsprozesses konkretisiert und dynamisch angepasst werden kann. Dies ermöglicht ein kontextsensitives, situiertes Vorgehen im Umgang mit Zielen und deren Erfüllung.

Für die Modellierung der Ziele wird zusammen mit der Zielformel eine Liste der Variablenbelegungen (*Binding*) verwaltet, welche die Werte der gebundenen Variablen sowie die spezifizierten *Constraints* der einzelnen Variablen umfasst. Dabei können die Werte einer Variablen aus ungebunden (*UNBOUND*), einem konkreten Term oder aus einem ungebundenen Wert, der durch *Constraints* eingeschränkt ist, bestehen (siehe Abbildung 5.16).

### Zieltypen

Die in der CASEC-Architektur berücksichtigten *Zieltypen* bestehen aus *Achieve*, *Perform*, *Maintain*, *Query* und *Conclude* (siehe Abschnitt 4.2.4). Dabei ist

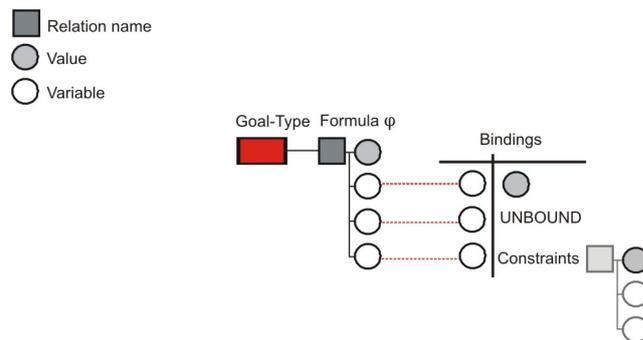


Abbildung 5.16: Goal-Repräsentation

die Unterscheidung zwischen *Achieve* und *Perform* und dem damit verbundenen Commitmentverhalten die Modellierungsgrundlage für *intend-to-* und *intend-that*-Ziele, welche im Rahmen der Kooperationsmodellierung gefordert wird (siehe Abschnitt 2.1.3). *Achieve*-Ziele werden für das Erreichen eines bestimmten Weltzustands eingesetzt, wohingegen *Perform*-Ziele handlungsorientiert sind und der Prozess der Ausführung im Vordergrund steht. Im Gegensatz zu den anderen Zieltypen sind *Conclude*-Ziele *Ereignis-getrieben*, d.h. sie agieren in Form eines Plans, welcher eine spezifizierte Formel überwacht. Bei Erfüllung der Formel wird der Plan aktiv und der *Plan-Body* wird ausgeführt (siehe Abschnitt 5.2.5).

### Zielaktivierung

E. Miller & Cohen (2001) sehen die kognitive Kontrolle als eine *aktivitätsbasierte* Steuerung welche dazu dient, Regularitäten und Ziele sowie Aufgaben bezogene Regeln zu erkennen und zu aktivieren. Daher werden in der CASEC-Architektur wie bei den *Belief-Chunks* zeitliche Aspekte und Aktivierungswerte für Ziele integriert. Durch die Berücksichtigung eines Aktivierungswertes ist es möglich, die Aktualität eines Ziels zu modellieren und eine kontextsensitive Verarbeitung zu gewährleisten, die auf den aktuellen Ziel- und Wissenszustand des Agenten zugeschnitten ist. Der Aktivierungswert dient dazu, den Suchraum einzuschränken und den Aufmerksamkeitsfokus des Agenten auf aktuell relevante Ziele zu lenken. Im Rahmen von Zielinteraktionen dient der Aktivierungswert dazu, die Menge der zu berücksichtigenden Ziele in einem angemessenen Rahmen zu halten.

Ziele besitzen neben einem Aktivierungswert einen Zeitstempel, der angibt, wann sie entstanden sind (*time-stamp*), und einen weiteren Zeitstempel, der angibt, wann sie das letzte Mal ausgeführt wurden bzw. wann der letzte Zugriff erfolgte (*last-retrieved*). Die Berechnung des Aktivierungswertes greift auf die

Informationen der Zeitstempel zurück. Zusätzlich werden, wie bei den *Belief-Chunks* des Arbeitsgedächtnisses, weitere Aktivierungsimpulse in die Berechnung mit einbezogen. Die Berechnung des Aktivierungswertes eines Ziels geschieht in enger Abhängigkeit vom Arbeitsgedächtnis und den damit verbundenen Aktivierungsimpulsen. Der Aktivierungswert errechnet sich aus einem internen Wert des Ziels sowie dem Aktivierungswert der Zielformel. Der zielinterne Wert bestimmt sich aus der Anzahl und dem zeitlichen Abstand der Zugriffe auf das Ziel sowie aus expliziten Aktivierungsimpulsen der interagierenden Ziele (siehe Abbildung 5.17). Die verschiedenen Formen der Zielinteraktion werden im nächsten Abschnitt beschrieben.

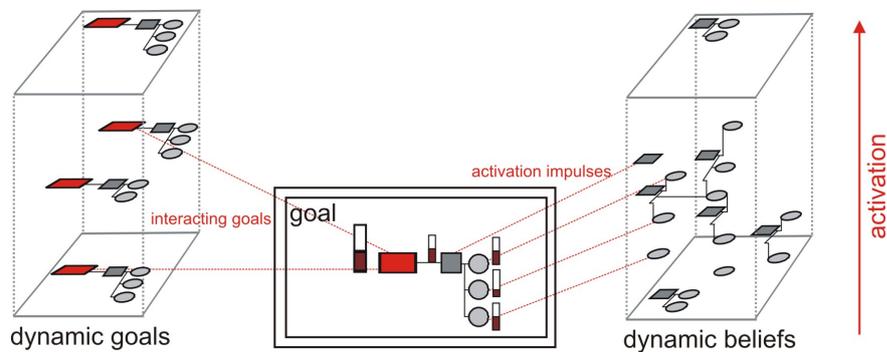


Abbildung 5.17: Einflüsse der Zielaktivierung

Die Berechnung der Aktivierungsimpulse eines Ziels  $g$  bestimmen sich zu:

$$\begin{aligned} \text{Acti\_Impulse}(g) := & \text{Impulse bei Interaktion mit aktiven Zielen} \\ & + \text{Impulse bei Auswahl als Intention} \\ & + \text{Impulse bei Aenderung des Zielzustands} \end{aligned} \quad (5.20)$$

Formal gefasst sehen die Aktivierungsregeln der einzelnen Terme folgendermaßen aus:

Berechnung der Impulse bei Interaktion mit aktiven Zielen:

$$\begin{aligned} \text{HAPPENS} \quad ( & \gamma \in {}^{cur}\text{Goals}_i \wedge \text{CurrentGoal}(g) \wedge \text{support}(\gamma, g)) \\ \Rightarrow & \text{IncreaseActi}(\gamma, \delta_{small}) \end{aligned} \quad (5.21)$$

Impulse bei der Auswahl des Ziels als Intention:

$$\begin{aligned} \text{HAPPENS} \quad ( & \gamma \in {}^{cur}\text{Goals}_i \wedge \text{CurrentGoal}(\gamma)) \\ \Rightarrow & \text{IncreaseActi}(\gamma, \delta_{large}) \end{aligned} \quad (5.22)$$

Impulse bei Änderung des Zielzustands:

$$\begin{aligned} \text{HAPPENS } ( \gamma \in {}^{cur}\text{Goals}_i \wedge \text{GoalState}(\gamma)^- \neq \text{GoalState}(\gamma) ) \\ \Rightarrow \text{IncreaseActi}(\gamma, \delta_{middle}) \end{aligned} \quad (5.23)$$

Die Gesamtformel der Aktivierung berücksichtigt den Aktivierungswert der Zielformel sowie die Aktivierungsimpulse:

$$\begin{aligned} \text{Acti}(g(\varphi)) &:= \text{Acti}(\varphi) \\ &+ \sum_{g \in \text{interactinggoals}} \text{Acti\_Impulse}(goal) \end{aligned} \quad (5.24)$$

Der erste Term ( $\text{Acti}(\varphi)$ ) stellt den Aktivierungswert der Zielformel dar und wird aus dem Arbeitsgedächtnis abgelesen. Der zweite, dritte und vierte Term bezieht sich auf weitere Aktivierungsimpulse, beispielsweise Impulse im Falle einer Adaptation des Goals als aktuelle Intention, Impulse im Falle einer Interaktion mit anderen Zielen oder Impulse bei Veränderung des aktuellen Zielzustands. Auf die Formel wird, wie auch bei den *Beliefs*, eine *Decay*-Funktion angewendet (siehe 5.1.6). Bei der Ziel-Modellierung wird in entsprechender Weise wie beim dynamischen Arbeitsgedächtnis der Ansatz verfolgt, nur die aktuellen relevanten Elemente zu berücksichtigen, denn selbst Menschen sind nicht in der Lage, alle ihre Ziele gleichzeitig zu berücksichtigen. Der Aktivierungswert kodiert die Aktualität des Ziels in der aktuellen Situation. Analog zum Arbeitsgedächtnis gilt: Fällt der Aktivierungswert eines Ziels unter den Schwellwert  $\theta\text{GOAL}_{acti}$ , wird das Ziel aus der Menge der *aktuellen Ziele*  ${}^{cur}\text{Goals}_i$  entfernt und in einem Zwischenspeicher vorgehalten (siehe Abbildung 5.18).

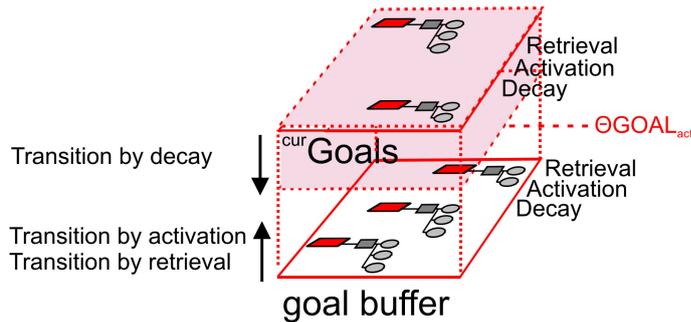


Abbildung 5.18: Modellierung der aktuellen Ziele

Die Ziele eines Agenten bestehen aus der Menge aller Ziele:

**Definition 10**  $\text{Goals}_i := \{ \gamma \mid \gamma : \text{GoalFormula} \wedge \text{GOAL}_i(\gamma) \}$

Die Menge der *aktuellen Ziele* eines Agenten  $i$  ( $^{cur}Goals_i$ ) ist folgendermaßen definiert:

**Definition 11** Sei  $Goals_i$  die gesamte Menge der Ziele des Agenten  $i$ . Dann gilt  $^{cur}Goals_i := \{g_x \mid g_x \in Goals_i \wedge Acti(g_x) > \theta GOAL_{acti}\}$ . Es gilt  $^{cur}Goals_i \subseteq Goals_i$ .

### Zielinteraktion

Ziele sollten nicht nur einzeln betrachtet werden, sondern der Zusammenhang zwischen Zielen und ihr Zusammenspiel müssen berücksichtigt werden. Insbesondere im Rahmen einer parallelen Verarbeitung von Zielen kann es zu Interaktionen zwischen Zielen kommen, die es zu berücksichtigen gilt. Auch innerhalb einer Kooperation müssen Konflikte zwischen den Zielen der Interaktionspartner erkannt und behoben werden. Die CASEC-Architektur bietet die Möglichkeit, sowohl *positive* als auch *negative* Interaktionen zu berücksichtigen. Des Weiteren können Einschränkungen in Bezug auf die zeitliche Abfolge der Ziele modelliert werden.

**Typisierte Zielverknüpfungen** Durch typisierte Zielverknüpfungen besteht die Möglichkeit, neben einfachen Zielen auch *Task-Networks* zu repräsentieren (Sardina et al., 2006). Ein Aufgabenplan (*Task-Network*) repräsentiert ein Problem, das es zu lösen gilt, bzw. ein Ziel, das der Agent erreichen möchte. Dabei kann zwischen einfachen Zielen (*Goal-Tasks*), ausführbaren Aktionen (einfache Aufgaben) (*Primitive-Tasks*) und komplexen Zielstrukturen (*Compound-Tasks*) unterschieden werden. In den weit verbreiteten BDI-Implementationen sind nur einfache Ziele sowie einfache Aufgaben repräsentierbar. In ihnen ist es auch nicht möglich, den Wertebereich einer Variablen einzuschränken. In der CASEC-Architektur können auch komplexe Zielstrukturen dargestellt und verarbeitet werden. Dies ist insbesondere im Rahmen gemeinsamer Ziele und Pläne von entscheidender Bedeutung (siehe Abschnitt 2.1.3).

Umgesetzt werden komplexe Zielstrukturen unter anderem durch einschränkende *Constraint-Links*, mit denen Ziele untereinander verbunden werden können, um die Einhaltung ihrer Reihenfolge zu gewährleisten. Jedes Ziel verfügt dafür über ein Feld, in welchem eine Liste von typisierten Links auf weitere Ziele vorgehalten werden kann. Die Links bestehen aus der Information, ob die Ziele in einer (*OR*)-, (*BEFORE*)- oder (*AFTER*)-Beziehung stehen, und aus einem Link auf das jeweilige Bezugsziel. Beim Test ob ein Ziel als Intention angenommen werden soll, wird überprüft, ob die getypten Zielverknüpfungen gültig sind. Nur wenn diese erfüllt sind, kann das Ziel als Intention fungieren. Im Falle des Erreichens eines Ziels wird berücksichtigt, ob das Ziel mit einem weiteren Ziel durch

eine OR-Verknüpfung verbunden ist; in diesem Fall kann das Bezugsziel geprüft und gelöscht werden.

$$\begin{aligned}
 \text{goalConstraintLinks} & ::= \text{BEFORE seq GOAL} \\
 & \quad | \text{AFTER seq GOAL} \\
 & \quad | \text{OR seq GOAL}
 \end{aligned} \tag{5.25}$$

**Positive und negative Zielinteraktion** Zwischen Zielen kann es sowohl zu positiver als auch negativer Interaktion kommen (siehe 4.2.6). Interaktionen können sowohl mit Zielen selbst als auch mit den Plänen, welche als Intention eines Ziels fungieren, auftreten. Unter *positiver Zielinteraktion* wird die Erfüllung eines Teilziels bzw. die Erfüllung einer Vor- oder Kontextbedingung eines Plans verstanden. *Negative Interaktion* besteht aus einem Konflikt zwischen den Zielformeln mehrerer Ziele oder aus einem Konflikt zwischen der Zielformel und den Vor- bzw. Kontextbedingungen einer Intention.

**Definition 12** Sei  $\gamma$  ein Ziel und  $\gamma \in {}^{cur}\text{Goals}$  und sei  $i$  die zugehörige Intention zu  $\gamma$ . Sei  $(\text{Intent} : \text{Goal} \rightarrow \text{Plan})$  eine Funktion. Dann gilt  $\text{Intent}(\gamma) = i$ .

**Definition 13** Sei  $i$  eine Intention. Dann liefert die Funktion  $\text{PreC}(i)$  die Menge der zugehörigen Vorbedingungen von  $i$ , die Funktion  $\text{ContextC}(i)$  die Menge der spezifizierten Kontextbedingungen von  $i$  und die Funktion  $\text{Effects}(i)$  die Menge der spezifizierten Effekten.

Die Relation *supports* bestimmt für zwei Ziele, ob in Bezug auf Vorbedingungen, Kontextbedingungen und Effekte der Ziele eine unterstützende Beziehung gilt. Dafür wird überprüft, ob die Zielformeln miteinander unifiziert werden können, und ob eine der Zielformeln zu der Menge der Vor- bzw. Kontextbedingungen der anderen Zielformel gehört. Desweiteren wird überprüft, ob es Formeln gibt, die sowohl zu der Menge der Effekte der zugehörigen Intention eines Ziels als auch zu der Menge der Vor- bzw. Kontextbedingungen des anderen Ziels gehören. (*unify* bezieht sich auf den Unifikationsalgorithmus der CASEC-Architektur siehe Abschnitt 5.1.7)

$$\begin{aligned}
 \text{supports}(g_i(\varphi), g_j(\gamma)) & ::= \text{unify}(\varphi, \gamma) \\
 & \quad \vee \text{unify}(\varphi, \vartheta_x) \wedge \vartheta_x \in \text{PreC}(\text{Intent}(g_j)) \\
 & \quad \vee \text{unify}(\varphi, \vartheta_y) \wedge \vartheta_y \in \text{ContextC}(\text{Intent}(g_j)) \\
 & \quad \vee (\{\text{Effects}(\text{Intent}(g_i))\} \cap \\
 & \quad \quad (\{\text{PreC}(\text{Intent}(g_j))\} \cup \{\text{ContextC}(\text{Intent}(g_j))\}) \neq \emptyset)
 \end{aligned} \tag{5.26}$$

Die Menge der Ziele ( $Support_{goals}(\gamma)$ ), welche ein Ziel  $\gamma$  unterstützen, besteht aus den aktivierten Zielen der Intentionstruktur, welche in einer positiven Interaktionsbeziehung zu dem Ziel  $\gamma$  stehen.

$$Support_{goals}(\gamma) := \{g_i \mid g_i \in {}^{cur}Goals \wedge supports(g_i, \gamma)\} \quad (5.27)$$

Die Relation *conflicts* wird analog zu der *supports*-Relation (5.26) bestimmt:

$$\begin{aligned} \mathbf{conflicts}(g_i(\varphi), g_j(\gamma)) &:= \mathbf{unify}(\neg\varphi, \gamma) \\ &\vee \mathbf{unify}(\neg\varphi, \vartheta_x) \wedge \vartheta_x \in \mathbf{PreC}(\mathbf{Intent}(g_j)) \\ &\vee \mathbf{unify}(\neg\varphi, \vartheta_y) \wedge \vartheta_y \in \mathbf{ContextC}(\mathbf{Intent}(g_j)) \\ &\vee (\neg\{\mathbf{Effects}(\mathbf{Intent}(g_i))\} \cap \\ &\quad (\{\mathbf{PreC}(\mathbf{Intent}(g_j))\} \cup \{\mathbf{ContextC}(\mathbf{Intent}(g_j))\}) \neq \emptyset) \end{aligned} \quad (5.28)$$

Analog wird die Menge der konfligierenden Ziele ( $Conflict_{goals}(\gamma)$ ) in Bezug auf ein Ziel  $\gamma$  definiert; diese besteht aus all den Zielen, welche in einer negativen Interaktionsbeziehung zu dem Ziel  $\gamma$  stehen.

$$Conflict_{goals}(\gamma) := \{g_i \mid g_i \in {}^{cur}Goals \wedge conflicts(g_i, \gamma)\} \quad (5.29)$$

**Konfliktbestimmung** Die Konfliktbestimmung zwischen Zielen untereinander und zwischen Zielen und Plänen kann unterschiedlich detailliert ausfallen, je nachdem welche Bedingungen überprüft werden sollen. Dies kann durch das entsprechende Flag festgelegt werden. Dafür lassen sich die Bedingungen der positiven und negativen Zielinteraktion in ihre einzelnen Faktoren zerlegen:

$$\mathbf{direct\_supp}(g_i(\varphi), g_j(\gamma)) := \mathbf{unify}(\varphi, \gamma) \quad (5.30)$$

$$\mathbf{preCond\_supp}(g_i(\varphi), g_j(\gamma)) := \mathbf{unify}(\varphi, \vartheta_x) \wedge \vartheta_x \in \mathbf{PreC}(\mathbf{Intent}(g_j)) \quad (5.31)$$

$$\mathbf{contextCond\_supp}(g_i(\varphi), g_j(\gamma)) := \mathbf{unify}(\varphi, \vartheta_x) \wedge \vartheta_x \in \mathbf{ContextC}(\mathbf{Intent}(g_j)) \quad (5.32)$$

$$\begin{aligned} \mathbf{effect\_supp}(g_i(\varphi), g_j(\gamma)) &:= (\{\mathbf{Effects}(\mathbf{Intent}(g_i))\} \\ &\cap (\{\mathbf{PreC}(\mathbf{Intent}(g_j))\} \cup \{\mathbf{ContextC}(\mathbf{Intent}(g_j))\}) \\ &\neq \emptyset) \end{aligned} \quad (5.33)$$

Die Bedingungen von  $direct\_confl(g_i, g_j)$ ,  $preCond\_confl(g_i, g_j)$ ,  $context\_Confl(g_i, g_j)$ ,  $effect\_Confl(g_i, g_j)$  lassen sich analog zu 5.30 - 5.33 definieren.

Algorithm 2 zeigt die möglichen Testaufrufe. Für die Bestimmung der zeitlichen Beschränkungen (*checkGoalConstraints*) werden die typisierten Zielverknüpfungen überprüft. Übergeben wird dem Algorithmus das zu testende Ziel und eine Menge von zu testenden Zielen oder Intentionen. Dabei kann die Testmenge aus nur der aktuellen Intention (*Current Intention Test*), allen Zielen mit einer höheren Priorität (*Higher Priority Test*) oder aus allen Zielen der Menge  $^{cur}Goals$  (*Complex Conflict Test*) bestehen. Die zu berücksichtigende Strategie und die Testmenge, können explizit festgelegt, aber auch durch *Meta-Level*-Prozesse dynamisch gesetzt werden.

---

**Algorithm 2** conflictSupportDetection
 

---

```

1: procedure CONFLICT_INTERACTION(Goal g, GoalSet testSet)
2:   if (considerGoalConstraint == true) then
3:     checkGoalConstraints(g, testSet)
4:   end if
5:   if (considerDirectSupport == true) then
6:     checkDirectSupport(g, testSet)
7:   end if
8:   if (considerPreconditionSupport == true) then
9:     checkPreconditionSupport(g, testSet)
10:  end if
11:  if (considerContextSupport == true) then
12:    checkContextSupport(g, testSet)
13:  end if
14:  if (considerDirectConflict == true) then
15:    checkDirectConflict(g, testSet)
16:  end if
17:  if (considerPreconditionConflict == true) then
18:    checkPreconditionConflict(g, testSet)
19:  end if
20:  if (considerContextConflict == true) then
21:    checkContextConflict(g, testSet)
22:  end if
23: end procedure

```

---

Abbildung 5.19 zeigt exemplarisch die Zielinteraktion. In dem Beispiel wird die *Zielformel* des zu überprüfenden Ziels mit allen Zielformeln der aktuellen Ziele aus der Menge der  $^{cur}Goals$  auf mögliche Konflikte verglichen (Schritt 1). In

Bezug auf die aktuellen Intentionen werden sowohl die Zielformel des Ursprungsziels der Intention als auch die aktuellen Teilziele berücksichtigt. In einem weiteren Schritt werden die Kontextbedingungen der zugehörigen Pläne berücksichtigt (Schritt 2). Dies stellt eine umfangreiche Berücksichtigung möglicher Konflikte dar (*Complex Conflict Test*).

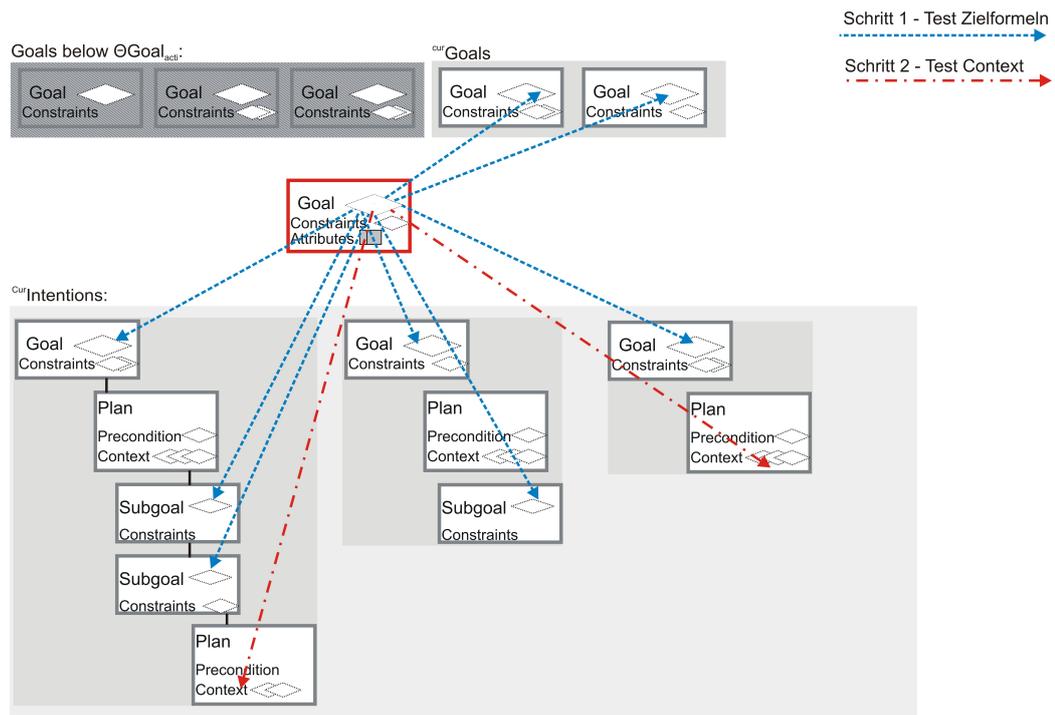


Abbildung 5.19: Zielinteraktion

**Einsatz Konfliktbestimmung** Die Methoden, welche für die Bestimmung von Konflikten dienen, werden in der CASEC-Architektur in verschiedenen Kontexten eingesetzt. Im Rahmen der **parallelen Verfolgung** von multiplen Zielen dient die Konfliktdetektion als Grundlage für die Entscheidung, ob ein Ziel als Intention angenommen werden kann, wenn sich die aktuelle Intention in einem Wartezustand befindet (siehe Abschnitt 5.2.6). Die Ergebnisse der Zielinteraktion werden außerdem für die Berechnung des **Desirability**-Werts eingesetzt. Dabei kann sowohl positive als auch negative Zielinteraktion berücksichtigt werden. Die Detektion von Konflikten spielt ferner auch für die Aufstellung eines **gemeinsamen Plans** im Rahmen einer Kooperation eine wichtige Rolle. Auch im Bereich der **Intentions/Planerkennung** sind die Datenstrukturen und Methoden der Interaktionsberechnungen sehr bedeutsam (siehe Abschnitt 5.2.7).

### Gemeinsame Ziele

Im Rahmen einer Kooperation spielen gemeinsame Ziele eine zentrale Rolle. Die CASEC-Architektur unterstützt die explizite Repräsentation gemeinsamer Ziele. Für ein Ziel kann angegeben werden, dass es sich um ein gemeinsames Ziel handelt. Außerdem kann das zugehörige Team, welches gemeinsam das Ziel verfolgt, bei der Zielspezifikation angegeben werden. Es besteht auch die Möglichkeit, neben festgelegten Variablenbelegungen potentielle Variablenbelegungen vorzumerken, die der Agent als gute Lösungen ansieht. Diese müssen dann noch mit dem (oder eventuell mehreren) Teamkollegen abgestimmt werden. Die Modellierung gemeinsamer Pläne und die Repräsentation damit verbundener Daten ist ein zentraler Aspekt für die Modellierung von *SharedPlans* im Rahmen einer Kooperation (siehe Abschnitt 6.4.1).

### 5.2.2 Obligationen in CASEC

*Obligationen* werden als eine besondere Zielform modelliert. Sie besitzen die gleiche Repräsentationsstruktur wie ein *Goal* und sind in den Deliberationsprozess der kognitiven Schleife integriert. Der Ansatz, Obligationen und Ziele durch die gleiche Struktur und die Repräsentation eines numerischen Prioritätswerts auf eine gemeinsame vergleichbare Skala abzubilden, greift den Ansatz des B-DOING-Modells und der BOID-Architektur auf (siehe 4.2.6). Zusätzlich enthalten Obligationen in der CASEC-Architektur jedoch noch einen Link zu ihrem Entstehungsursprung. Dieses Zusatzattribut kann von *Meta-Level*-Plänen dazu genutzt werden, Obligationen gesondert zu berücksichtigen. Damit können mit der CASEC-Architektur beispielsweise Mechanismen der sozialen Kommunikation umgesetzt werden, welche als interaktive Basisaustattung fungieren und unabhängig vom eingesetzten Szenario in Kooperationssituationen von entscheidender Bedeutung sind.

Der Vorteil explizit repräsentierter Obligationen besteht darin, dass neue soziale Regeln leicht in die Architektur zu integrieren sind. Da sie im normalen Deliberationsprozess des Agenten berücksichtigt werden, profitieren sie von sämtlichen Basismechanismen und Berechnungsstrukturen der kognitiven Schleife. Der Agent ist durch die explizit repräsentierten Obligationen in der Lage, bewusst gegen Obligationen zu verstoßen; er kann auch ihre Priorität und Gewichtung der jeweiligen Situation anpassen.

**Beispiel** Obligationen werden durch *Conclude*-Ziele umgesetzt. Diese überwachen etwa Änderungen des konversationalen Zustands und werfen im Falle einer Änderung kontextsensitiv ein Ziel auf, um auf die Änderung angemessen einzugehen. Der Agent hat beispielsweise die Obligation auf ein *Wanting-Turn* seines

Gegenübers einzugehen. Er ist nicht dazu verpflichtet, den Turn abzugeben, aber er sollte signalisieren, dass er das Signal zur Kenntnis genommen hat.

Obligationen können dazu eingesetzt werden, die Regeln des Kontaktmanagements (siehe Abschnitt 3.3.2) und weitere Diskursregeln (siehe Abschnitt 3.3.4) umzusetzen (siehe Abschnitt 6.2.1 und 6.2.2).

### 5.2.3 Deliberation

Deliberation besteht aus dem Entscheidungsprozess, welche Ziele verfolgt werden sollen. In diesen Prozess gehen verschiedene Einflussfaktoren ein. Dabei lassen sich *Filter* und *Reihenfolge* der Berechnung unterscheiden. Es können verschiedene *Meta-Strategien* eingesetzt werden, um den Entscheidungsprozess des Agenten zu konfigurieren.

**Filter** bestehen aus Bedingungen, deren Nichterfüllung zum Ausschluss der betroffenen Ziele führt. Filter können die Menge der Optionen signifikant reduzieren. Dabei können sich Filter sowohl auf den Inhalt eines Ziels beziehen als auch auf bestimmte Zielattribute. Beispielsweise kann ein Ziel aufgrund seines Inhalts ausgeschlossen, wenn es in einer Konfliktbeziehung zur aktuellen Intention steht und zudem über einen geringeren Prioritätswert verfügt. Ein Beispiel für den Ausschluss eines Ziel aufgrund eines Zielattributes stellen die  $^{curr}Goals_i$  dar, bei denen Ziele aufgrund eines zu geringen Aktivierungswertes aus der Menge der aktuellen Ziele ausgeschlossen werden.

Die **Reihenfolge** einer Berechnung kann das Ergebnis entscheidend beeinflussen. Der Reihenfolgeeffekt kann als eine besondere Form eines Filters wirken. Wird bei einer Berechnung nur eine bestimmte Suchtiefe berücksichtigt bzw. wird nur eine beschränkte Menge an Berechnungszeit aufgewandt, so entscheidet die Reihenfolge darüber, welche Ziele berücksichtigt werden und welche von den Berechnungen ausgeschlossen werden. Im Falle eines rationalen Agenten kann davon ausgegangen werden, dass Entscheidungen immer unter einem beschränktem Zeitaufwand gefällt werden. Eine geschickt gewählte Reihenfolge kann den Rechenaufwand reduzieren und der Agent kann infolgedessen effektiver agieren. Die Berücksichtigung der Suchtiefe bzw. der Zeit, die der Agent für seinen Entscheidungsprozess einsetzt, kann auch durch den mentalen Zustand des Agenten beeinflusst werden. Ein Beispiel dafür stellt die Beziehung zwischen der Suchstrategie und dem emotionalen Zustand des Agenten dar (siehe 5.2.9).

Im Weiteren werden die verschiedenen Faktoren des Entscheidungsprozesses vorgestellt.

### Entscheidungsfaktoren

**Priorität** Der Prioritätswert repräsentiert einen Basiswert, der angibt, wie wichtig ein Ziel ist. Er wird bei der Instantiierung eines Ziels explizit gesetzt. Es besteht auch die Möglichkeit, eine Prioritätsfunktion  $PriorityFkt(g) \rightarrow \mathbb{R}^+$  zu spezifizieren, die dann im aktuellen Kontext ausgewertet wird.

$$\begin{aligned}
 Prio & : Goal \rightarrow \mathbb{R}^+ \\
 Prio(g_i) & = \begin{cases} P \\ PriorityFkt(g_i) \end{cases} \quad (5.34)
 \end{aligned}$$

Der Prioritätswert wird innerhalb einer Intention an Teilziele vererbt und übt insofern einen entscheidenden Einfluss auf die Zielstruktur aus. Der Prioritätswert repräsentiert die generelle Wichtigkeit eines Ziels. Es besteht die Möglichkeit, die Deliberationstrategie des Agenten durch Meta-Pläne auf alleinige Berücksichtigung des Prioritätswerts zu setzen. Damit handelt der Agent dann rational; er verfolgt immer sein wichtigstes Ziel, jedoch ist er in diesem Fall blind gegenüber möglichen Zielinteraktionen.

**Desirability** Der *Desirability*-Wert dient dazu, Ziele in ihrem Gesamtkontext zu betrachten und Interaktionen mit anderen Zielen abzuschätzen. Dabei können sowohl positive als auch negative Zielinteraktionen berücksichtigt werden (siehe Abschnitt *Positive und negative Zielinteraktion* sowie Gleichungen 5.2.1 und Gleichungen 5.26 und 5.28).

$$D_{goal}(g_i) := \sum_{g_s \in Support_{goals}(g_i)} Prio(g_s) - \sum_{g_c \in Conflict_{goals}(g_i)} Prio(g_c) \quad (5.35)$$

In der CASEC-Architektur wird der *Desirability*-Wert nur als Entscheidungsgrundlage für untereinander vergleichbare Ziele eingesetzt. Übersteigt der Differenzbetrag der Prioritätswerte eine Schwelle  $\Delta\chi$  (siehe Gleichung 5.36 und Abbildung 5.20), ist also das eine Ziel signifikant wichtiger als das andere, so ist es nicht sinnvoll, die Ziele mit geringerer Priorität zu verfolgen, auch wenn diese sich sehr positiv ergänzen würden. Der Agent würde unter solchen Umständen eventuell sein wichtigstes Ziel immer zurückstellen, was nicht dem Verhalten eines rationalen Agenten entspräche. Der *Desirability*-Wert dient demnach nur als eine Größe, um mögliche Konflikte zwischen gleichrangigen Zielen zu lösen und einen richtungweisenden Einfluss auf die Deliberation auszuüben. Formal stellt läßt sich dies folgendermaßen fassen:

Sei  $i$  ein Agent,  $Desirability(g_a) : Goal \rightarrow \mathbb{R}^+$ ,  $g_a, g_b \in GoalSet \subseteq Goals_i$  (5.36)

$$Desirability(g_a) = \begin{cases} D(g_a), & \exists g_b \in GoalSet : \|Prio(g_a) - Prio(g_b)\| \leq \Delta\chi \\ 0, & \text{sonst} \end{cases}$$

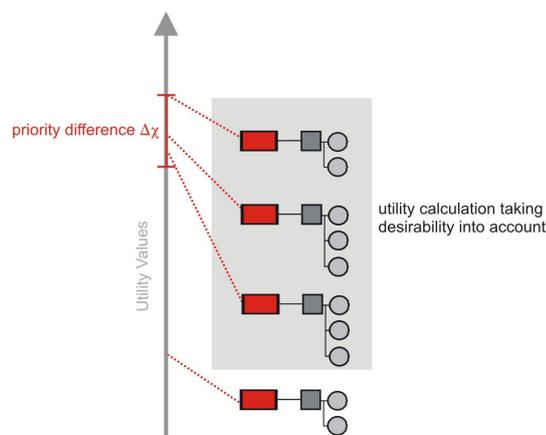


Abbildung 5.20: Bedingung für den Einbezug des Desirability-Werts in die Utility-Berechnung

**Utility** Der *Utility*-Wert eines Ziels fasst die verschiedenen Entscheidungsfaktoren zu einem Gesamtwert zusammen. Dabei stellt der Prioritätswert den größten Faktor dar. Wie bei den Aktivierungswerten des Arbeitsgedächtnisses dienen die zusätzlichen Entscheidungsfaktoren der CASEC-Architektur dazu, den Entscheidungsprozess positiv zu beeinflussen und einen richtungweisenden Einfluss auszuüben, nicht aber dazu, die ursprünglichen Entscheidungsgrundlagen komplett außer Kraft zu setzen.

Die Berechnung des Prioritätswerts selbst liefert den Basiswert der *Utility*-Berechnung. Je nach der Deliberationsstrategie des Agenten kann dieser Wert auch als alleiniger Wert berücksichtigt werden. Es ist aber auch möglich, die *Desirability* in den Deliberationsprozess mit einzubeziehen. Die Berechnung des *Utility*-Werts eines Ziels ist in der CASEC-Architektur konfigurierbar und kann je nach Anwendungsszenario durch *Meta-Reasoning* Prozesse gestaltet werden. Diese Prozesse setzen die jeweiligen Gewichte in der *Utility*-Berechnung:

$$Utility_{goal}(g) := Priority_{goal}(g) + \frac{Desirability_{goal}(g)}{G_{desirability}} \tag{5.37}$$

In der vorliegenden Arbeit wird ein Defaultwert von  $G_{desirability} = 2$  vorgeschlagen. Dadurch wird die Desirability-Berechnung zwar in den Entscheidungsprozess miteinbezogen, aber der Prioritätswert gibt dennoch den größeren Ausschlag.

Abbildung 5.21 bietet eine Übersicht über die Einflussfaktoren, die bei der jeweiligen Berechnung der Entscheidungsfaktoren eingesetzt werden. Der *Filter* bei der Berechnung des *Priority*-Werts besteht aus dem Aktivierungswert eines Ziels. Nur für Ziele, welche in den  $^{cur}Goals_i$  vorliegen, wird ein Prioritätswert bestimmt. Für die *Reihenfolge* der Prioritätsberechnung wird der Aktivierungswert als Grundlage genommen. Der Prioritätswert wird mit keinem Gewicht verrechnet, das Ziel mit dem höchsten Prioritätswert besitzt die höchste Priorität.

Ein *Filter*, welcher bei der Berechnung des *Desirability*-Werts Einsatz findet, ist der Abstand zwischen den Kandidaten mit den höchsten Prioritätswerten. Liegt der Abstand der Prioritätswerte unter der Schwelle  $\Delta\chi$  so wird der *Desirability*-Wert der Kandidaten bestimmt. Die *Gewichtung*, die darüber entscheidet, welches Ziel den höchsten Wert erhält, besteht aus der Differenz der Prioritäten aller unterstützenden und konfligierenden Ziele.

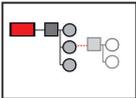
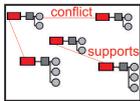
	Reihenfolge der Berechnung	Filter	Gewichtung
 Priorität Relevanz	Zielaktivierung	$^{cur}Goals_i$	höchste Priorität
 Desirability Zielinteraktion		Prioritätsdiff > $\theta$	$\Sigma support - conflict$ $G_{desirability}$

Abbildung 5.21: Entscheidungsgrundlagen Deliberation

Nachdem alle Attribute eines Ziels vorgestellt wurden, wird hier noch einmal die Basisstruktur eines Ziels in der CASEC Architektur präsentiert:

*Goal*


---

*goalType* : *Achieve* — *Perform* — *Maintain* — *Query* — *Conclude*  
*goalFormula* : *Formula*  
*goalBinding* : *Binding with Constraints*  
*goalStatus* : *GoalState*  
*timeStamp* : *TimePoint*  
*lastRetrieved* : *TimePoint*  
*lastRuntime* : *TimePoint*  
*topLevelGoal* : *Goal*  
*subGoal* : *seq Goal*  
*interruptGoal* : *seq Goal*  
*goalConstraintLinks* : *seq goalConstraintLink*  
*supportingGoals* : *seq GOAL*  
*conflictGoals* : *seq GOAL*  
*priority* : *RealValue*  
*desirability* : *RealValue*  
*utility* : *RealValue*  
*goalOrigin* : *Obligation*  
*shared* : *Bool*  
*team* : *seq Name*  
*intention* : *Plan*

---

### 5.2.4 Handlungsoptionsgenerierung (*Means-Ends-Reasoning*)

In der CASEC-Architektur wird eine Form des *Practical Reasoning* eingesetzt. Das bedeutet, dass zunächst in einem Deliberationsprozess bestimmt wird, welche Ziele verfolgt werden sollen. In einem nachfolgenden *Means-Ends-Reasoning* Prozess wird berechnet, auf welche Weise und durch welche Handlungen die Ziele erreicht werden sollen. Der Prozess unterbreitet eine Auswahl an Handlungsoptionen jeweils mit spezifiziertem Utility-Wert, welcher zusammen mit der Ziel-Utility die Entscheidungsgrundlage in der CASEC-Architektur bildet.

Der Algorithmus, der darüber entscheidet, welches Ziel als Intention verfolgt werden soll, operiert auf dem Intentions-Stack (siehe Abbildung 5.22). Die Intentionsstruktur besteht aus den aktuellen Zielen und Intentionen des Agenten. Diese lassen sich in einzelne *Threads* unterteilen. Die Ziele und Intentionen können hierarchisch gestaffelt sein und sich in ein oder mehrere Teilziele gliedern. Die jeweiligen *Top-level*-Ziele spannen einen Kontext auf, in welchem die Teilziele bearbeitet werden sollten. Der Kontext umfasst dabei Variablenbelegungen, Verpflichtungen und Entscheidungen.

Für alle aktiven Ziele der aktuellen Intentionsstruktur, für die ein anwendbarer

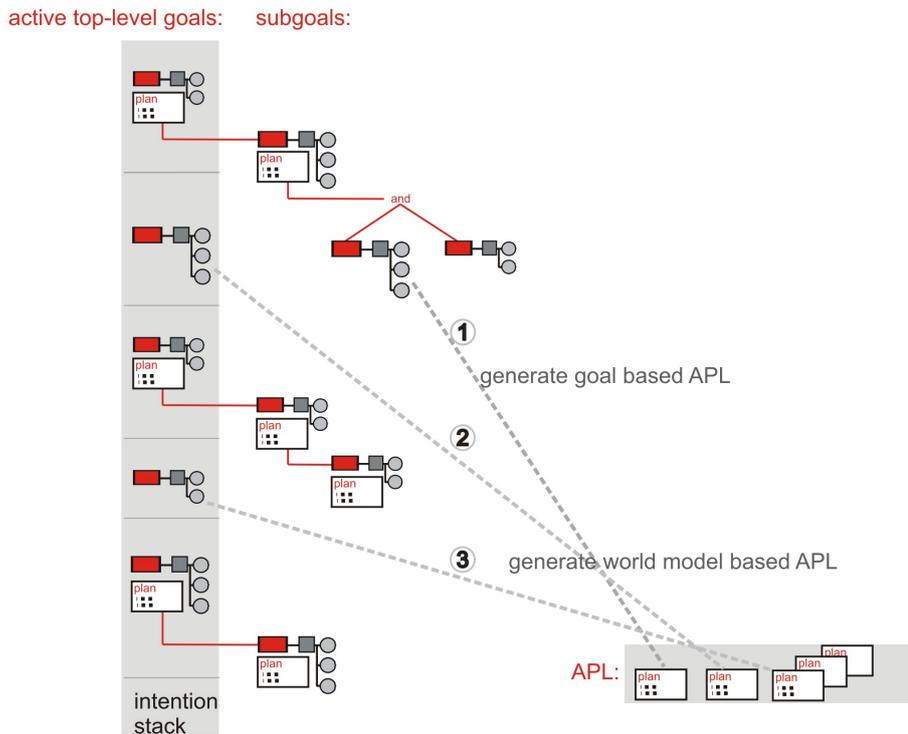


Abbildung 5.22: Intentions-Stack

Plan existiert, werden mögliche Handlungsoptionen in die Liste der anwendbaren Pläne (*Applicable Plan List*, *APL*) instantiiert (siehe Abbildung 5.22). Dabei werden auch verschiedene Variablenbelegungen berücksichtigt. Das heißt, die Handlungsoptionen umfassen sowohl verschiedene Handlungsformen als auch verschiedene Ausprägungen in Bezug auf die Attribute der Handlungen. Für die Auswahl der Handlungsoptionen werden Filter berücksichtigt, um ungültige Handlungsoptionen auszuschließen. Die Liste der Pläne wird nach den *Utility*-Werten der einzelnen Optionen sortiert.

Die Generierung möglicher Handlungsoptionen wird sowohl durch die direkte Instantiierung von Zielen als auch reaktiv durch Veränderungen des Weltzustands angestoßen. Alle potentiellen Intentionen werden analog zu UM-PRS gemeinsam in der (*APL*) verwaltet (siehe Anhang A.2.3 *Algorithm 16*, welche die Algorithmen 17 und 18) einsetzt.

### 5.2.5 Pläne in CASEC

Während das Arbeitsgedächtnis in Form der *Beliefs* konzeptuelles bzw. semantisches Wissen des Agenten repräsentiert, enthält die *Planbibliothek* der CASEC-

Architektur prozedurales Wissen. Handlungsoptionen werden in der CASEC-Architektur in einer Planbibliothek verwaltet. Analog zu kognitiven Gedächtnismodellen existiert ein prozedurales Arbeitsgedächtnis sowie ein prozedurales Langzeitgedächtnis. Die Elemente der prozeduralen Gedächtnisse bestehen aus einzelnen Plänen. Diese können einfache Handlungen, aber auch komplexe Handlungsmuster repräsentieren. Pläne stellen eine Verbindungsstelle zwischen deliberativen Entscheidungen und *Behaviors* dar, die aus den Deliberationsprozessen aus instantiiert werden und dann auf der Verhaltensebene ablaufen. Für einen effizienten Einsatz einer Planbibliothek müssen verschiedene Aspekte eines Plans repräsentiert werden, damit diese in den Entscheidungsprozess mit einbezogen werden können.

Genau wie deklarative Fakten enthalten auch Pläne der Planbibliothek *Aktivierungswerte*. Diese helfen, schneller einen passenden Plan zu finden, und schränken den Suchraum der Pläne ein. Zusätzlich können die Pläne einen *Salien- cy-Wert* enthalten. Ein solcher Marker kann beispielsweise einen negativen Wert enthalten, wenn der Plan zuvor fehlgeschlagen ist. Der Aktivierungswert kodiert, wie aktiv und passend der Plan in der aktuellen Situation ist, d.h. beispielsweise, ob er abgerufen und in Erwägung gezogen wurde, wobei ein Basisaktivierungswert seinen globalen Einsatz charakterisiert.

Auf höchster Ebene stehen abstrakte Ziele und komplexe Zielzustände, die es zu erreichen gilt. Pläne stellen auf einer darunter liegenden Ebene bereits Konkretisierungen dar und beschreiben in einem Ablaufplan, wie ein Zustand erreicht oder eine Handlung ausgeführt werden kann. Die Pläne können aus Teilplänen bestehen, die strukturiert über einen längeren Zeitraum abgearbeitet werden. Auf unterster Ebene existieren Pläne, welche direkt mit *Behaviors* oder *mentalen Operationen* verknüpft sind. Diese wirken ähnlich wie Produktionsregeln, indem sie, einmal angestoßen, ihre Aktion ausführen, ohne sich in Untereinheiten zerlegen zu lassen. Durch solche Pläne werden *Behavior* direkt an die Planbibliothek angeschlossen. Ein großer Vorteil besteht darin, dass der Agent durch den deklarativen Aspekt eines Planes weiß, zu welchen Handlungen er prinzipiell in der Lage ist. Die Planbibliothek spiegelt die Fähigkeiten eines Agenten wider.

$$PriorityValue == p : \mathbb{R} \mid p \geq 0 \quad (5.38)$$

$$DesirabilityValue == d : \mathbb{R} \mid d \geq 0 \quad (5.39)$$

$$EffortValue == e : \mathbb{R} \mid e \geq 0 \quad (5.40)$$

$$\begin{aligned} planType ::= & \text{ACHIEVE} \\ & \mid \text{PERFORM} \end{aligned} \quad (5.41)$$

**Schema 7**

<i>Plan</i>
<i>purpose</i> : <i>ActionFormula</i>
<i>planType</i> : <i>PlanType</i>
<i>preCondition</i> : <i>seq Condition</i>
<i>contextCondition</i> : <i>seq Condition</i>
<i>effects</i> : <i>ActionFormula</i>
<i>failure</i> : <i>ActionFormula</i>
<i>priority</i> : <i>PriorityValue</i>
<i>timeTag</i> : <i>PointInTime</i>
<i>lastRetrieved</i> : <i>PointInTime</i>
<i>activation</i> : <i>ActivationValue</i>
<i>saliency</i> : <i>SaliencyValue</i>
<i>meanEffort</i> : <i>EffortValue</i>
<i>highestEffort</i> : <i>EffortValue</i>
<i>lowestEffort</i> : <i>EffortValue</i>
<i>effort</i> : <i>EffortFunction</i>
<i>desirability</i> : <i>DesirabilityValue</i>
<i>subgoals</i> : <i>seq Goals</i>
<i>possibleSubgoals</i> : <i>seq Goals</i>
<i>attributes</i> : <i>seq Value</i>

Für jeden Plan (siehe Schema 7) ist eine **ActionFormula** gegeben, die das Anwendungsziel des Plans in Form einer ausführbaren Handlung oder eines zu erreichenden Weltzustands (*Belief-Formel*) spezifiziert. Daneben gibt es die Möglichkeit, Vorbedingungen, die vor der Annahme eines Plans gültig sein müssen, und Kontextbedingungen, die während der gesamten Ausführung des Plans gelten müssen, zu spezifizieren. Wird eine Kontextbedingung verletzt, so bricht der Plan mit der Ausführung einer *Failure-Section* ab.

Die **Vor- und Kontextbedingungen** werden als Liste geführt und können aus Abfragen des Weltzustands (*COND\_FACT*, *COND\_RETRIEVE*), des Zielzustands des Agenten (*COND\_GOAL*) sowie aus der Spezifikation von berechenbaren Ausdrücken (*COND\_EXP*) bestehen (siehe Definition 5.42).

**Schema 8**

<i>Condition</i>
<i>type</i> : <i>conditionType</i>
<i>activeValue</i> : <i>bool</i>

$$\begin{aligned}
conditionType & ::= COND\_GOAL \\
& | COND\_EXP \\
& | COND\_FACT \\
& | COND\_RETRIEVE \qquad (5.42)
\end{aligned}$$

Die Spezifikation von **Effekten**, die mit der Ausführung des Plans einhergehen, ist von entscheidender Bedeutung für die Berechnung von Ziel-Plan-Interaktionen. Der **Prioritätswert** eines Plans wird analog zu dem Prioritätswert eines Ziels durch einen Wert bzw. durch eine Funktion angegeben (siehe Abschnitt 5.2.3).

**Aktivierung** Analog zu Zielen und *Belief-Formeln* verfügen auch Pläne über einen *Aktivierungswert*, der dafür sorgt, dass die Menge der anwendbaren Pläne in einem angemessenen Rahmen bleibt. Sinkt der Aktivierungswert unter den Schwellwert  $\theta PLAN_{acti}$ , so wird der Plan aus den aktuell vorgehaltenen Plänen entfernt, er liegt jedoch weiter im prozeduralen Langzeitgedächtnis des Agenten vor und kann bei Bedarf aktiviert werden.

$$Acti(plan_i) := \sum Impulses(plan_i) \qquad (5.43)$$

Die Aktivierung eines Plans bestimmt sich aufgrund der Zugriffe auf den Plan. Zusätzlich werden Aktivierungsimpulse im Falle einer erfolgreichen Ausführung eines Plans ausgesendet. Auch Handlungsvorschläge des Interaktionspartners, welche einem Plan zugeordnet werden können, sorgen für zusätzliche Aktivierungsimpulse. Analog zu den Beliefs und Zielen wirkt eine *Decay*-Funktion (siehe Abschnitt 5.1.6).

**Aufwand (Effort)** Um Pläne genauer charakterisieren zu können, wird in der Planrepräsentation der CASEC-Architektur eine neue Idee umgesetzt. Neben dem Prioritätswert kann auch ein *Effort*-Wert für die einzelnen Pläne spezifiziert und in den Entscheidungsprozess mit einbezogen werden.

Der *Effort*-Wert gibt an, wie anstrengend und aufwendig es ist, einen Plan auszuführen und welche Kosten durch die Ausführung des Plans entstehen. Konkrete Informationen können dabei jedoch nur für die unterste Plan-Ebene angegeben werden. Pläne, die direkt mit einem *Behavior* verbunden sind und diese instantiiieren, können eine Abschätzung des Aufwands durch Anstoßen des *Behavior* erfahren und dann weiterleiten. Neben konkreten Werten kann für einen Plan auch eine *Effort*-Funktion ( $EffortFKT \rightarrow \mathbb{R}^+$ ) angegeben werden. Höher in der Hierarchie stehende Pläne sind auf das Propagieren von Abschätzungen der Teilpläne angewiesen. Je nach aktueller Planungsmethode des Interpreters und in

Abhängigkeit des emotionalen Zustands des Agenten werden mehr oder weniger detaillierte Pläne aufgestellt und berücksichtigt. So kann es passieren, dass eine sehr „blauäugige“ Abschätzung bei „guter Laune“ vorgenommen wird, in welcher nicht bedacht wird, wie „anstrengend“ in Bezug auf den Aufwand einige Handlungen sind (siehe auch Abschnitt 5.2.9).

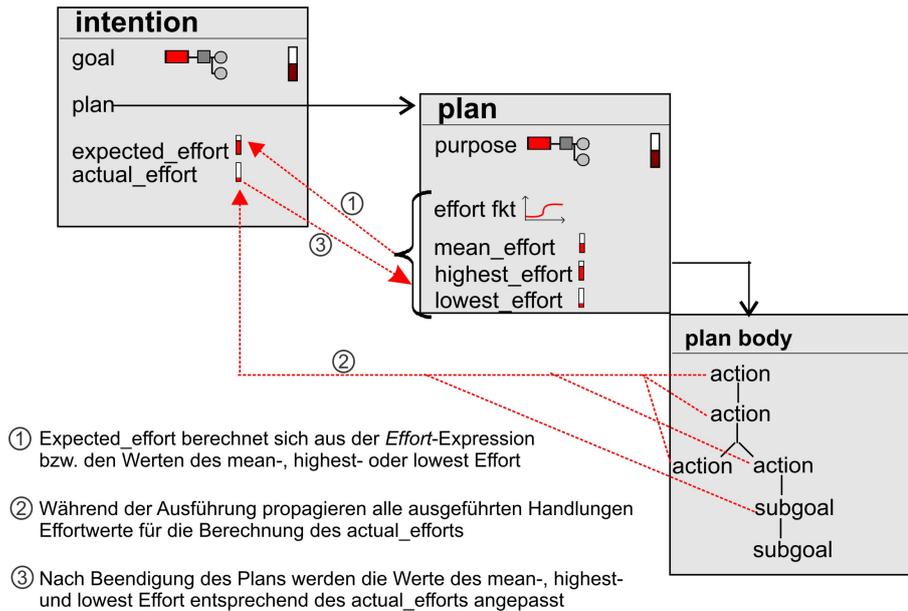


Abbildung 5.23: Effort-Berechnung

$$Effort_{exp}(act_x) = \begin{cases} EffortFKT(act_x), & \text{falls } EffortFKT(act_x) \text{ def} \\ \frac{LowestEffort(act_x) + MeanEffort(act_x)}{2}, & \text{falls positive emotional state} \\ \frac{HighestEffort(act_x) + MeanEffort(act_x)}{2}, & \text{falls negative emotional state} \end{cases} \quad (5.44)$$

$$Effort_{exp}(plan_x) = \sum_{act_i \in Actions(plan_x)} Effort_{exp}(act_i) \quad (5.45)$$

Während der Erwartungswert des *Expected-Effort* für den Entscheidungsprozess des Agenten eingesetzt wird, wird für jede Intention ein aktueller *Effort*-Wert vorgehalten, der den tatsächlichen *Effort* in der aktuellen Situation repräsentiert. Dieser dient dazu, unproduktiv arbeitende Pläne erkennen zu können. Die *Effort*-Werte werden in regelmäßigen Abständen überprüft und mit einem Erwartungswert verglichen, der bei der Instantiierung eines Plans geschätzt und vermerkt

wird. Aufgrund von nicht erfüllten Erwartungen, d.h. der *Expected-Effort*-Wert wird weit übertroffen, kann es zu einer „Frustration“ kommen und ein Plan abgebrochen werden.

Für die Berechnung des *Effort*-Werts wird auf eine Liste der Teilziele eines Plans zugegriffen. Diese Liste werden beim Einlesen der Planbibliothek gebildet.

Zusätzlich zu den definitiven Teilzielen wird eine Liste der möglichen Teilziele verwaltet, die im Rahmen des Plans aufgeworfen werden können, aber nicht zwangsläufig müssen. Der Zugriff auf die mit einem Plan verbundenen Teilziele, stellt einen wichtigen Aspekt für das *Means-Ends-Reasoning* dar. Zudem dient er bei der Plan- bzw. Intentionserkennung dazu, feststellen zu können, ob eine Aktion zu einem Plan beiträgt (siehe Abschnitt 5.2.7).

**Entscheidungsgrundlagen** Analog zu den Entscheidungsfaktoren bei der Berechnung der Deliberation gelten folgende Faktoren im *Means-Ends-Reasoning*-Prozess.

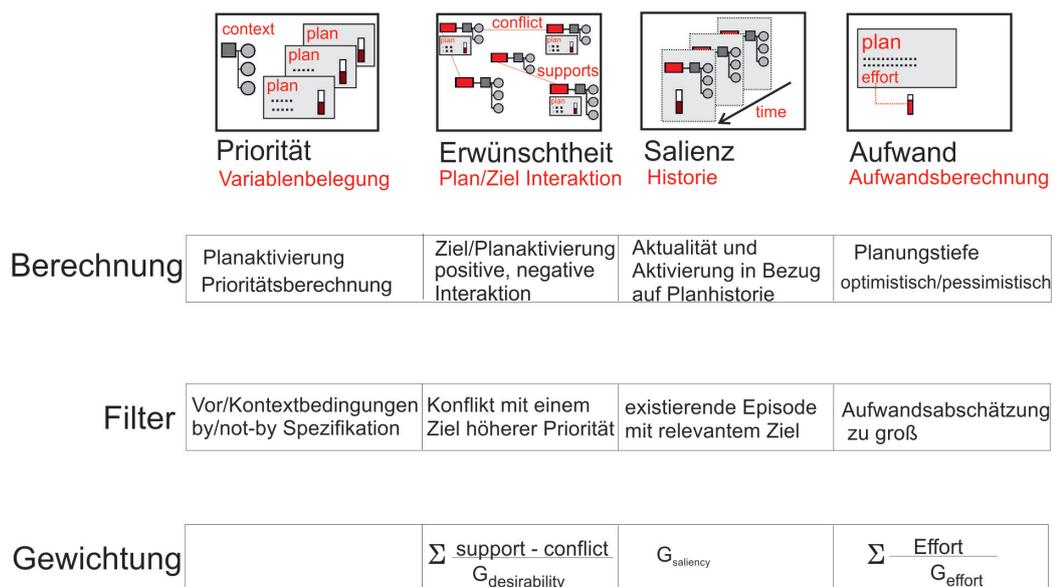


Abbildung 5.24: Entscheidungsgrundlage Pläne

**Desirability (Erwünschtheit)** Der *Desirability*-Wert eines Plans wird analog zu dem *Desirability*-Wert eines Ziels bestimmt (siehe Abschnitt 5.2.3). Sodann werden bei der Berechnung des *Desirability*-Werts neben der Plan-Formel zusätzlich die Effekte und Vor- sowie Kontextbedingungen berücksichtigt.

**Saliency (Salienz)** Einen weiteren Faktor stellt die *Saliency* eines Plans dar. Es geht darum, nicht nur den aktuellen Kontext zu berücksichtigen, sondern auch Erfahrungen des Agenten aus früheren Situationen in seine Deliberationsprozesse mit einzubeziehen. Die Salienz kann dazu eingesetzt werden, Erfahrungen aus ähnlichen Situationen in diesem Wert widerzuspiegeln (siehe auch Abschnitt 5.2.5).

$$Saliency_{plan}(plan_i) := \sum_{situations} \frac{\#Success(plan_i) - \#Failure(plan_i)}{\#situations} \quad (5.46)$$

Für die Berechnung des *Saliency*-Werts wird auf das episodische Gedächtnis zugegriffen. Die *Reihenfolge*, die bei dem Abrufen der relevanten Episoden die Berechnung bestimmt, wird durch den *Recency*-Effekt bestimmt, da das Episodische Gedächtnisses entlang der Zeitachse durchsucht wird. Der *Filter* bei der Berechnung des *Saliency*-Werts besteht aus den aktivierten und zugreifbaren Elementen des episodischen Gedächtnis. Nur Ziele, welche in der Vergangenheit eine hohe Priorität und Aktivierung aufwiesen, werden in dem episodischen Gedächtnis gespeichert und können dann in ähnlichen Situationen einen Einfluss ausüben. Die *Gewichtung* des *Saliency*-Werts kann durch einen Wert  $G_{saliency}$  festgelegt werden.

**Utility eines Plans** Während die Priorität des Ziels, seine Aktivierung und sein *Desirability*-Wert dazu genutzt werden, um zu entscheiden, welches Ziel verfolgt werden soll (*Deliberation*), gibt der *Utility*-Wert eines Plans an, wie gut ein bestimmter Plan in einer gegebenen Situation eingesetzt werden kann; der Wert bezieht sich dabei auf den *Means-Ends-Reasoning*-Prozess.

Bei der Berechnung des *Utility*-Wert eines Plans können neben dem Prioritätswert auch ein *Effort*-Wert (Aufwand, Kosten), ein *Desirability*-Wert und ein *Saliency*-Wert einbezogen werden.

$$\begin{aligned} Utility_{plan}(p_x) &:= Priority_{plan}(p_x) + Desirability_{plan}(p_x) \\ &- (G_{effort} * \sum Exp\ effort(p_x) + G_{saliency} * Saliency_{plan}(p_x)) \end{aligned} \quad (5.47)$$

### 5.2.6 Intentionen in CASEC

Eine Intention besteht aus einem Ziel und der Entscheidung, dieses auf eine bestimmte Art und Weise zu erreichen. Genau wie Bratman (1987) werden Intentionen dabei in der vorliegenden Arbeit als nicht reduzierbar auf *Beliefs* und Ziele sondern als primitive Modaloperatoren angesehen. Intentionen lassen sich aber

folgendermaßen aufspalten (Die Modaloperatoren PLAN, COMMIT werden an dieser Stelle nicht formal definiert.):

**Definition 14** Eine Intention lässt sich in das jeweilige Ziel, den angenommenen Plan und das Commitment in Bezug auf den Plan aufspalten:

$$\begin{aligned} (INTEND_i \varphi) ::= & (GOAL_i \varphi) \wedge (PLAN_i \varphi) \\ & \wedge (COMMIT_i((GOAL_i \varphi), (PLAN_i \varphi))) \end{aligned} \quad (5.48)$$

Während die *Commitment*-Strategie nicht in einem direkten Verhältnis zum Aufmerksamkeitsfokus des Agenten steht, stellen die Parameter der Ziel- und Planformel direkte Bezugspunkte dar. Um sich auf die objektbezogenen Aspekte der Formeln zu konzentrieren, wird die Funktion  $t\_set$  (siehe Definition 6 Abschnitt 5.1.4) eingesetzt.

**Definition 15** Die Menge der Terme einer modallogischen Formel besteht aus der Vereinigung der Menge der Terme der involvierten Formeln:  $t\_set(GOAL_i \varphi) := t\_set(\varphi)$ ,  $t\_set(PLAN_i \varphi) := t\_set(\varphi)$ . Die Menge der Terme einer Intention leitet sich aus der Menge der Terme des aktuellen Ziels ab:  $t\_set(INTEND_i \varphi) := t\_set(CurrentGoal_i(INTEND_i \varphi))$ .

Dabei besteht das *CurrentGoal* aus dem am höchsten aktivierten Ziel und aus der Menge der mit der aktuellen Intention assoziierten Ziele. Diese Menge umfasst die Zielformel selbst sowie die im Rahmen der Ausführung instantiierten Teilziele.

*Intention*

*goal* : goalFormula  
*commiment* : goalType  
*plan* : plan  
*expectedEffort* : value  
*actualEffort* : value  
*utility* : value

Die *Utility* einer Intention setzt sich aus dem *Utility*-Wert des Ziels und dem *Utility*-Wert des zugehörigen Plans zusammen.

$$Utility_{intention}(intent_i) := Utility_{goal}(intent_i) + Utility_{plan}(intent_i) \quad (5.49)$$

Der Vorteil der gemeinsamen Berücksichtigung besteht darin, dass so opportunistisches Verhalten ermöglicht und gefördert wird. Der Prozess der Generierung der Handlungsoptionen unterliegt emotionalen und deliberativen Einflüssen.

### Parallele Zielverfolgung

Für eine parallele Zielverfolgung spielt nicht nur der *Utility*-Wert einer Intention eine wichtige Rolle, es gilt auch den *Zustand* des zugehörigen Ziels zu berücksichtigen. Wurde eine Intention als wichtigste Intention ausgewählt, so wird ihr aktueller Zustand überprüft. Um eine parallele Ausführung von Intentionen zu ermöglichen, wird die Menge der möglichen Zielzustände in der CASEC-Architektur um einen *aktiven Wartezustand* erweitert.

Ein Ziel kann damit in CASEC die folgenden Zustände annehmen:

$$\begin{aligned} \textit{GoalState} ::= & \textit{isUntried} \mid \textit{isActive} \mid \textit{isAbandoned} \mid \textit{isSuccess} \\ & \mid \textit{isFailure} \mid \textit{isBlocked} \mid \textit{isWaiting} \end{aligned} \quad (5.50)$$

Der Zustand *isUntried* charakterisiert Ziele, für die noch keine Intention gebildet wurde. Nur die Ziele, die gerade ausgeführt werden, besitzen den Status *isActive*. Ein Ziel, das bewusst aufgegeben wird, erhält den Status *isAbandoned*. Ein Ziel, das erfolgreich abgearbeitet wurde, erhält den Status *isSuccess*, ein gescheitertes den Status *isFailure*. Ein Ziel, für das wegen nicht erfüllter Zielrelationen noch keine Intention gebildet werden konnte, erhält den Status *isBlocked*.

Der Zustand *isWaiting* stellt einen neu eingeführten Zustand für parallele Zielverarbeitung in der CASEC-Architektur dar. Für diese Erweiterung wird eine Form des *aktiven Wartens* eingeführt. Wenn eine aktuelle Intention blockiert oder sich im wartenden Zustand befindet, kann eine Intention mit geringerer *Utility* verfolgt werden, sofern diese zu keinem Konflikt mit der eigentlich aktuellen Intention führt. Um dies zu prüfen, wird auf die Funktion der Zielinteraktion (Definition 5.28, siehe Abschnitt 5.2.1) zurückgegriffen.

---

#### Algorithm 3 active\_waiting()

---

```

1: procedure ACTIVE_WAITING()
2:   if ( (goalState (currentIntention) == isBlocked)  $\vee$  (goalState (currentIntention) == isWaiting) ) then
3:     while no intentional commitment do
4:       nextIntention  $\leftarrow$  get intention with next highest utility
5:       if testConflict (currentIntention, nextIntention) == false then
6:         commit to next intention
7:       end if
8:     end while
9:   end if
10: end procedure

```

---

Pläne werden oft als Auslöser von *Behavior* instantiiert, welche einfache Handlungen umsetzen. Als solche müssen die zugehörigen Intentionen auf die

Ausführung der angestoßenen *Behavior* warten. In der CASEC-Architektur ist es durch das aktive Warten möglich, während der Abwicklung des *Behavior* simultan weitere kognitive Operationen auszuführen. In jedem kognitiven Zyklus wird dann überprüft, ob die Intention mit dem eigentlich höchsten *Utility*-Wert wieder Berechnungen und mentale Manipulationen vornehmen möchte.

### Umgang mit Konflikten

In der CASEC-Architektur leitet sich der Agent aus dem deklarativen Wissen seiner Handlungsoptionen (Vorbedingungen, Kontextbedingungen sowie Effekte) die möglichen Konflikte aktiv selber ab. Dadurch ist der Agent in der Lage, kontextspezifische Konflikte zu erkennen. Die vorgesehenen Mechanismen der Konfliktdetektion lassen sich sowohl auf Ziele als auch auf Pläne anwenden und ermöglichen so die Berücksichtigung des situationalen Kontextes. In der CASEC-Architektur werden beim Einlesen von Planbibliotheken Datenstrukturen vorgesehen, welche Informationen über Effekte, Ziele und potentielle Teilziele vorhalten. Die Konflikte selbst werden dann dynamisch während der Laufzeit ermittelt.

In der CASEC-Architektur wird die parallele Ausführung von Zielen als entscheidender Vorteil gesehen; kommt es jedoch zu Konflikten, wird das Ziel mit der höheren Priorität weiter verfolgt und das andere Ziel muss mit der Ausführung warten, bis kein Konflikt mehr besteht. Damit stellt der Ansatz der CASEC-Architektur einen Kompromiss zwischen dem effektiveren Agieren des Agenten durch parallele Ausführung von Zielen unter Berücksichtigung von Zielinteraktionen und den Kosten für ein komplexes *Scheduling* dar.

### 5.2.7 Interaktionsverarbeitung

Insbesondere in Kooperationsszenarien stellt die Interpretation der Handlungen und Äußerungen des Gegenübers einen wichtigen Aspekt dar. Neben der eigenen Zielverfolgung dient daher die Intensionsstruktur des Agenten als Basis für die Verarbeitung von Interaktionsbeiträgen. Die Modellierung der **Intensionsstruktur** sorgt für ein tiefer gehendes „Verständnis“ des Agenten für die Situation. Handlungen des Gegenübers werden nicht als einzelne losgelöste Aktionen interpretiert, sondern besitzen einen Hintergrund z.B. im Kontext eines komplexeren Plans. Das Einordnen in die *Intensionsstruktur* ermöglicht eine kontextsensitive Interpretation. Dabei spielen die Aktivierungswerte des dynamischen Arbeitsgedächtnisses eine entscheidende Rolle, da sie den Suchraum einschränken und erste Hinweise darauf liefern, welcher Kontext relevant sein könnte.

Für die Interaktionsverarbeitung in der CASEC-Architektur können zusätzlich zu den normalen Teilzielen auch sogenannte *Interrupt-Goals* einem Ziel zugeordnet werden. Diese können während der Interaktion des Agenten mit seiner

Umwelt oder seinem Interaktionspartner als einfache Ziele entstehen. Thematisch lässt sie sich ein solches Ziel oft einem *Intention-Thread* zuordnen, auch wenn der Entstehungsursprung auf andere Ziele und Mechanismen zurückzuführen ist (beispielsweise Obligationen). Diese Ziele werden durch den inhaltlichen Bezug auf ein bestehendes Ziel oder eine bestehende Intention dem jeweiligen *Intention-Thread* zugeordnet und als eine besondere Form eines Teilziels eingefügt.

Die Abbildung 5.25 zeigt den aktuellen Intentionstack des Agenten und das Einfügen eines *Interrupt-Goals*, für welches ein passender Kontext ermittelt werden konnte. Beispielsweise wird so versucht, Zwischenfragen dem jeweiligen In-

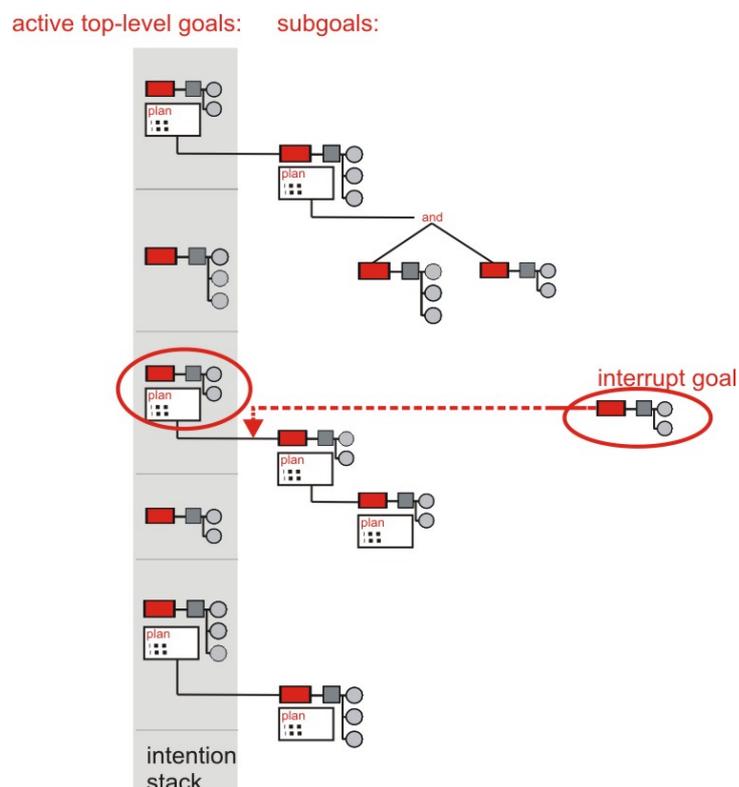


Abbildung 5.25: Einfügen eines *Interrupt-Goals* in den Intentionstack eines Agenten

tentionskontext zuzuordnen und in diesem Rahmen (unter der passenden Variablenbelegung und Interpretation) zu bearbeiten. Für alle eingehenden *Interaction-Moves* und Ereignisse wird versucht, diese im Rahmen der Intentionsstruktur zu interpretieren und in einen größeren Kontext zu setzen. Kontexte können dabei aus Zielen, Intentionen und insbesondere aus *gemeinsamen* Zielen bestehen. Beispielsweise können auch Ziele des Gegenübers als mögliche Kontextinterpretatio-

nen fungieren.

### **Intentionserkennung**

Während in den meisten Ansätzen zur Intentionserkennung von eingeschränkten Stacks der zugreifbaren Dialog-Einheiten ausgegangen wird (Lochbaum, 1995; Traum, 1996), bauen Balkanski & Hurault-Plantet (2000) in ihrem Ansatz zusätzlich auf einer Rezept-Graph-Struktur (*R-Graph*) auf. Die CASEC-Architektur geht darüber hinaus, indem für die Interpretation sowohl das episodische Gedächtnis, das dynamische Arbeitsgedächtnis als auch die aktuelle Intensionsstruktur berücksichtigt werden.

In dem Modell von Lochbaum dienen die Elemente auf dem Stack dazu, den Fokus auf den aktuellen Kontext zu lenken. Der R-Graph speichert die Akte, auf welche sich die Agenten geeinigt haben, während des gesamten Dialogs und sorgt dafür, dass die Agenten sich nicht auf konfligierende Handlungen festlegen. Es bestehen damit zwei getrennte Repräsentationen, der R-Graph und die Intensionsstruktur des Agenten. Die Intensionsstruktur der CASEC-Architektur hingegen umfasst beide Aspekte. Der Agent ist in der Lage, in seiner Intensionsstruktur auch Intentionen seines Gegenübers aufzunehmen. Im übertragenen Sinn verfolgt der Agent auch diese Ziele und Pläne als Intentionen, auch wenn er sie nicht direkt ausführt. So können Handlungen des Gegenüber während einer Kooperation leicht im eigenen Plan berücksichtigt werden.

Während ein Rezept nur eine Ebene der Handlungsdekomposition enthält, kann ein R-Graph mehrere Level enthalten und dabei Informationen in Bezug auf gebundene Parameter, Akteure sowie zeitliche Einschränkungen beinhalten. Dies gilt auch für die Umsetzung der Intensionsstruktur in der CASEC-Architektur. Lochbaum fordert, dass ein Algorithmus für die Modellierung der Planerkennung in einem Diskurs in der Lage sein muss, das Schlussfolgern über hypothetische und nur teilweise spezifizierte Handlungen zu praktizieren. Da im Rahmen der Intensionsstruktur des Agenten Variablen, *Constraints* sowie partielle Pläne repräsentierbar sind, wird diese Forderung in der CASEC-Architektur erfüllt.

Die Intensionsstruktur ermöglicht es, Ziele zu spezifizieren, die noch nicht verfolgt werden, aber Teilziele im Rahmen eines Plans oder Ziels darstellen. Der Agent kann sowohl angenommene Pläne mit Variablenbelegungen als auch potentielle Belegungen (*identified*) repräsentieren. Diese bieten mögliche Kontexte für eingehende *Interaction-Moves* des Gegenübers. Damit der *Intensionsstack* handhabbar und kognitiv plausibel bleibt, werden abgeschlossene Ziele in das *Episodische Gedächtnis* und Ziele, welche längere Zeit nicht behandelt wurden, in den *GoalBuffer* (siehe Abschnitt 5.2.1) ausgelagert; die Ziele können bei Bedarf wieder aktiviert werden.

**Korrelationswert** Um zu erkennen, zu welcher Intention bzw. welchem Ziel ein eingehendes Ereignis gehört, wird ein *Korrelationswert* berechnet. Dieser wird für die jeweilige Zielformel und die interpretierte Formel des *Interaction-Moves* bestimmt. Der Korrelationswert einer Formel setzt sich aus der Summe der Korrelationswerte der einzelnen Terme und der Korrelation der Prädikate bzw. Relationsnamen zusammen.

Sei  $\varphi$  eine *Formel*,  $t_x$  ein Term. Für jeden Term der Formel  $\varphi$  wird der Korrelationswert mit dem Term  $t_x$  bestimmt. Der Korrelationswert des Terms  $t_x$  in Bezug auf die Formel  $\varphi$  besteht dann aus dem maximalen Korrelationswert der Terme der Formel  $\varphi$  mit Term  $t_x$ .

$$\text{Correlation}(\varphi, t_x) = \max(\text{Corr}(t_j, t_x), t_j \in t\_set(\varphi)) \quad (5.51)$$

Die Identität hat das stärkste Gewicht, gefolgt von einer *is\_a*-Beziehung bis hin zu einer *part\_of*-Beziehung. Die Funktion zur Berechnung der Korrelationswerte besteht aus den folgenden Wertezuweisungen (*unify* bezieht sich auf den Unifikationsalgorithmus der CASEC-Architektur siehe Abschnitt 5.1.7):

$$\text{Corr}(t_j, t_x) = \begin{cases} 10, & \text{falls } \text{instance}(t_j) \wedge \text{instance}(t_x) \wedge \text{identity}(t_j, t_x) \\ 1, & \text{falls } \text{unify}(t_j, t_x) \\ 0.5, & \text{falls } \text{part\_of}(t_j, t_x) \\ 0, & \text{sonst} \end{cases} \quad (5.52)$$

Um Wissen über *is\_a*-Relationen abrufen zu können, wird auf das semantische Wissen des Arbeits- und des Langzeitgedächtnisses zurückgegriffen, sowie auf das prozedurale Gedächtnis des Agenten.

In Bezug auf die Berücksichtigung spezifizierter *Constraints* der Terme wird auch bei Teilübereinstimmungen (der Typ passt, aber nicht alle spezifizierten *Constraints* werden erfüllt) der Korrelationswert erhöht, wenn auch um einen geringeren Wert. So wird für jedes Constraint, das erfüllt wird ein Wert von 0.3 auf den Korrelationswert addiert, für jedes Constraint jedoch, das verletzt wird, wird der Wert von 0.1 von dem Korrelationswert abgezogen:

$$\text{Corr}(t_j, t_x) = \left( \sum_{\forall c_i \in \text{Constraints}_{t_x} | c_i(t_j) = \text{true}} 0.3 \right) + \left( \sum_{\forall c_i \in \text{Constraints}_{t_x} | c_i(t_j) = \text{false}} -0.1 \right) \quad (5.53)$$

Damit die Berechnung nicht zu rechenintensiv und daher zu langsam wird, wird nach den Aktivierungswerten der einzelnen Kontextkandidaten vorgegangen, und es wird auch nur eine gewisse Tiefe berücksichtigt. Liegt danach kein zufriedenstellender Korrelationswert vor, so wird die Tiefe nochmals vergrößert.

Konnte immer noch kein Kontext ermittelt werden, so wird die Handlung bzw. das Ereignis singular eingetrag und der Agent stellt gegebenenfalls eine Rückfrage.

Die Reihenfolge, in welcher die Kandidaten für die Kontextzuordnung eingehender *Interaction-Moves* berücksichtigt werden, gestaltet sich wie in dem Pseudocode definiert.

---

**Algorithm 4** find\_context

---

```

find_context(Relation* formula)           ▷ calculate correlation to find context
1: corrResult1 = check_current_goal(formula)
2: corrResult2 = check_topLevelGoalOf_current_goal(formula)
3: corrResult3 = check_shared_goal(TeamMember member, formula)
4: corrResult4 = check_topLevelGoalOf_shared_goal(formula)
5: corrResult5 = check_possible_condition_current_goal(formula)
6: corrResult6 = check_possible_condition_shared_goal(TeamMember member, formula)
7: return max (corrResult1, corrResult2, corrResult3, corrResult4, corrResult5, corrResult6)

```

---

## 5.2.8 Commitmentverhalten

In der CASEC-Architektur wird zwischen *Top-level-Goals* und Teilzielen (*Subgoals*) unterschieden. Während *Top-level-Goals* beim Scheitern eines Plans nicht vom Intensionsstack genommen werden und persistent sind, werden *Subgoals* nicht automatisch wieder aufgeworfen. *Top-level-Goals* können nur durch explizite mentale Operationen aufgegeben werden. Wenn ein *Top-level-Goal* aufgegeben wird, dann werden automatisch die damit verbundenen Subgoals nicht weiter verfolgt.

Um das *Commitmentverhalten* an die Relevanz der einzelnen Ziele anpassen zu können, ermöglicht die CASEC-Architektur eine Kennzeichnung von Teilzielen; diese können mit den Flags *critical* bzw. *enabling* versehen werden. Die Flags üben einen Einfluss auf das Commitmentverhalten in Bezug auf die Teilziele aus. Ein Teilziel, bei dem das Flag *critical* gesetzt ist, wird nur dann aufgegeben, wenn es neben der gescheiterten keine weitere Handlungsalternativen gibt. Andernfalls wird der gescheiterte Versuch in die Liste der *Failure-Attempts* eingetragen und das Ziel bleibt als Teilziel bestehen, für welches nach einem weiteren passenden Plan gesucht wird. *Enabling-Goals* hingegen sind als nicht persistente Ziele umgesetzt.

Die *Flags (critical/enabling)* der Teilziele werden aufgrund des *Entstehungsursprungs* der Teilziele gesetzt. Je nachdem, ob es sich bei einem Teilziel um ein explizites Teilziel handelt, oder ob es ein Ziel für die Erfüllung einer Bedingung

im Planungsprozess darstellt, wird das Ziel mit einem *critical*-Flag bzw. *enabling*-Flag versehen.

$$Flag(g_i, plan_x) = \begin{cases} critical, & \text{falls } g_i \in Subgoals(plan_x) \\ enabling, & \text{falls } g_i \in PossibleSubgoals(plan_x) \\ enabling, & \text{falls } GoalFormula(g_i) \in PreConditions(plan_x) \vee \\ & GoalFormula(g_i) \in ContextConditions(plan_x) \end{cases} \quad (5.54)$$

Die Flags können im Rahmen von *Meta-Reasoning* Prozessen abgefragt und nutzbar gemacht werden.

### 5.2.9 Einflussnahme von Emotionen

Zu den kognitiven Basismechanismen zählt auch die Einflussnahme des emotionalen Zustands des Agenten auf seine Verarbeitungsprozesse. Dabei können insbesondere Strategien des Planens und der Entscheidungsfindung vom aktuellen emotionalen Zustand beeinflusst werden (Becker et al., 2006). Die Idee ist dabei die folgende: Ein negativer emotionaler Zustand wird zumeist die Problemlösungsstrategien einer Person so beeinflussen, dass eine lokale *bottom-up*-Verarbeitung eingesetzt wird, wohingegen eine positive Stimmung der Person in der Regel zu einer globalen *top-down*-Strategie führt.

Um zu verdeutlichen, an welchen Stellen Emotionen einen Einfluss auf den deliberativen Entscheidungsprozess des Agenten ausüben können, wird der BDI-Algorithmus (Rao & Georgeff, 1995) als Pseudocode präsentiert (siehe *Algorithm 5*) und der potentielle Einfluss der Emotionen wird im Einzelnen vorgestellt. In der vierten Zeile, in welcher die möglichen Handlungsoptionen aufgestellt werden, können Emotionen als *Filter* fungieren und nur solche Handlungsoptionen als mögliche durchlassen, welche mit dem emotionalen Zustand des Agenten übereinstimmen.

Zusätzlich kann der Prozess, in welchem die *Utility*-Werte der Pläne berechnet werden, entweder von optimistischen Abschätzungen der *Effort*-Werte (siehe 5.2.5) ausgehen oder pessimistischere Werte als Berechnungsgrundlage nehmen. Positive emotionale Stimmung korreliert dabei mit *optimistischerer Abschätzung*, negative emotionale Stimmung mit einer exakteren, *pessimistischeren Abschätzung*.

---

**Algorithm 5** BDI-interpreter

---

```

1: procedure THINK
2:   initialize-state()
3:   loop
4:     options  $\leftarrow$  option_generator(event-queue)
5:     selected_options  $\leftarrow$  deliberate(options)
6:     update-intentions(selected_options)
7:     execute()
8:     get_new_external_events()
9:     drop_successful_attitudes()
10:    drop_impossible_attitudes()
11:   end loop
12: end procedure

```

---

In der fünften Zeile kann die Entscheidung, welche der Intentionen als nächste angenommen wird, auch von dem emotionalen Zustand des Agenten beeinflusst werden. Der Algorithmus zur Berechnung der *Desirability* eines Ziels kann die Zielinteraktionen unterschiedlich berücksichtigen, wenn mehrere Ziele simultan verfolgt werden. Der Agent kann negative Interaktionen wie beispielsweise Ressourcenkonflikte zwischen den Zielen ebenso wie positive Zielinteraktionen in seine Überlegungen miteinbeziehen (siehe Abschnitt 5.2.1). Da die Berechnung der Interaktionen Zeit in Anspruch nimmt, kann der Agent die Berechnung nicht in beliebigen Berechnungstiefen vornehmen. Deshalb wird in der vorliegenden Arbeit vorgeschlagen, die Mechanismen in Zeile 5 vom emotionalen Zustand des Agenten abhängig zu machen. *Negative Emotionen* erscheinen dabei zu einer verengten Sichtweise während des Problemlösungsprozesses zu führen, während *positive Emotionen* eine breitere Lösungssuche veranlassen, welche multiple Ziele simultan berücksichtigt (Sloman, 1987).

Dieser Effekt kann dadurch modelliert werden, dass der extra Aufwand bei positiver Zielinteraktion nur dann berücksichtigt wird, wenn sich der Agent in einem positiven emotionalen Zustand befindet. In einem negativen emotionalen Zustand wird der Agent positive Interaktionen unberücksichtigt lassen und seinen Fokus allein auf sein wichtigstes Ziel lenken. Sein Verhalten erscheint dadurch engstirniger (*narrow-minded*).

In Zeile 10 des Algorithmus geht es um das Aufgeben unmöglich gewordener Ziele. Dieser Algorithmus kann so angepasst werden, dass auch Pläne, welche in ihrer Ausführung sehr schlecht abschneiden, überdacht und gegebenenfalls aufgegeben werden. Dabei spielt der *Expected-Effort*-Wert eine entscheidende Rolle (siehe Abschnitt 5.2.5). Der Entscheidungsprozess lässt sich ebenfalls von dem emotionalen Zustand des Agenten beeinflussen. Wenn der Agent sich in einem

negativen emotionalen Zustand befindet, wird er den kompletten Plan aufgeben und versuchen neu zu planen. Im einem positiven emotionalen Zustand wird der Agent „geduldiger“ sein und versuchen, das Teilziel auf einem anderen Weg zu erreichen. D.h. der Agent geht aufgrund seines positiven emotionalen Zustands davon aus, dass sein Verhalten in der aktuellen Situation einen erfolgreichen Eindruck macht und wird deshalb, soweit dies möglich ist, keine größeren Veränderungen in seinen Verhaltensweisen vornehmen.

Die verschiedenen Stellen der Einflussnahme des emotionalen Zustands des Agenten zeigen, dass Emotionen als Maß für das erfolgreiche Meistern von Situationen den Deliberations- und *Means-Ends-Reasoning*-Prozess des Agenten lenken können. Dabei können Emotionen als Einflussfaktoren auf verschiedenen Ebenen helfen. Der Ansatz, den emotionalen Zustand als Regulator für verschiedene Formen der Informationsverarbeitung einzusetzen, baut auf den Arbeiten von J. R. Gray (2004) auf.

### 5.2.10 Aufmerksamkeitsfokus - *Joint Attention*

Der Aufmerksamkeitsfokus des Agenten steht in einem engen Zusammenhang mit den aktivierten Elementen seiner Gedächtnisse. Aktivierungswerte dienen in der CASEC-Architektur als ein Maß für Salienz und aktuelle Relevanz. Ein Objekt mit einem höheren Aktivierungswert ist salienter als ein Objekt mit einem geringeren Aktivierungswert. Es gibt verschiedene Mechanismen, die die Aktivierungswerte der Gedächtniselemente beeinflussen (siehe Abschnitt 5.1.5).

Die Menge der  $^{cur}Bels_i$  modelliert die *Regions des direkten Zugriffs* nach Oberauer (siehe Abschnitt 4.1.3). In Abhängigkeit des aktuellen Verarbeitungsschritts korrespondiert entweder eine neu abgeleitete *Belief-Formel*, eine ausgewählte Intention oder eine ausgeführte Handlung eines Plans mit dem Aufmerksamkeitsfokus des Agenten. In der CASEC-Architektur werden die aktuelle *Belief-Formel* und die aktuelle Intention über Aktivierungswerte definiert. Der aktuelle Planschritt korrespondiert mit der Aktion des aktuell verfolgten Plans. Dieser besteht aus einem azyklischen Graph verschachtelter Ziele.

#### Definition 16

$$\begin{aligned}
 ^{cur}BEL_i &:= \{b_x \mid b_x \in ^{cur}Bels_i \wedge \forall b \in ^{cur}Bels_i \wedge b \neq b_x : Acti(b_x) > Acti(b)\} \\
 ^{cur}INT_i &:= \{n_x \mid n_x \in Ints_i \wedge \forall n \in Ints_i \wedge n \neq n_x : Acti(n_x) > Acti(n)\} \\
 ^{cur}PLAN\_STEP_i &:= \{action_x \mid (COMMIT_i(GOAL_i \varphi), ^{cur}PLAN_i \varphi) \\
 &\wedge action_x \in Acy\_graph(^{cur}PLAN_i \varphi) \wedge (HAPPENS_i action_x)\}
 \end{aligned}$$

Da die Prozesse der Perzeption und Kognition gleichzeitig und parallel ablaufen, vereinigen wir die drei Aspekte im Konzept des aktuellen Aufmerksamkeitsfokus:

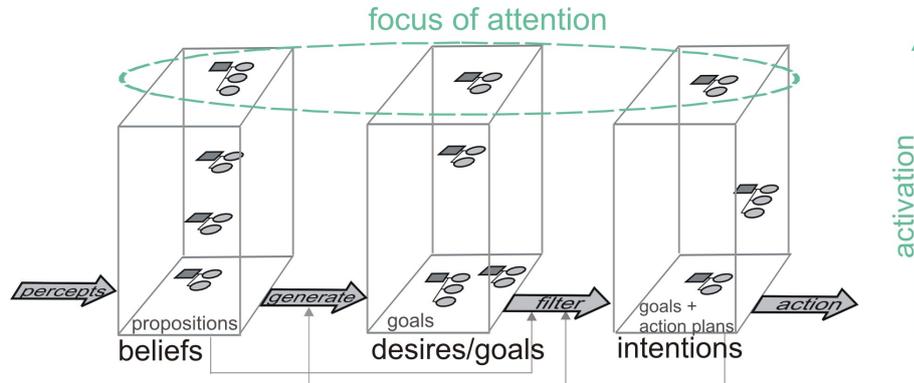


Abbildung 5.26: BDI und Aufmerksamkeitsfokus

**Definition 17**

$$ATT_i := \{t\_set({}^{cur}BEL_i) \cup t\_set({}^{cur}INT_i) \cup t\_set({}^{cur}PLAN\_STEP_i)\}$$

Der Aufmerksamkeitsfokus ist Teil des dynamischen Arbeitsgedächtnisses und wird durch die sich ändernden Beliefs und intentionalen Zustände des Agenten moduliert. Abbildung 5.26 zeigt, wie sich die Idee der Modellierung des Aufmerksamkeitsfokus im klassischen BDI-Modell, das jedoch um Aktivierungswerte erweitert wurde, einfügt.

**Joint Attention**

Aufbauend auf der Definition des Aufmerksamkeitsfokus lässt sich das Phänomen der gemeinsamen Aufmerksamkeit (*Joint Attention*) definieren. In Übereinstimmung mit Tomasello et al. (2005) wird *Joint Attention* in der vorliegenden Arbeit als intentionaler Prozess gesehen. Im Folgenden wird der mentale Zustand beschrieben, der für einen Agenten  $i$  erforderlich ist, um an einen Zustand der *Joint Attention* zu glauben, während er gemeinschaftlich mit seinem Interlokutor  $j$  ein Zielobjekt  $\vartheta$  fokussiert (siehe Abbildung 5.27 und (Pfeiffer-Leßmann & Wachsmuth, 2009)).

**Definition 18** ( $BEL_i(JOINT\_ATT\ i\ j\ \vartheta)$ ) iff

- (1) (*being aware of other*)  $BEL_i(ATT_j\ \vartheta) \wedge BEL_i(INTEND_j(ATT_j\ \vartheta))$
- (2) (*ascribing goal*)  $BEL_i(GOAL_j(ATT_i\ \vartheta \wedge ATT_j\ \vartheta))$
- (3) (*adopting goal*)  $GOAL_i(ATT_i\ \vartheta \wedge ATT_j\ \vartheta)$

(4) (*feedback*)  $BEL_i(BEL_j(ATT_i j))$

(5) (*focus state*)  $HAPPENS(\{T(\vartheta)\}_i \wedge \{P(< \{T(\vartheta)\} >_j)\}_i)$

- (1) (**being aware of other**) Durch die explizite Repräsentation des Beliefs  $BEL_i(ATT_j \vartheta)$  über den Aufmerksamkeitsfokus des Interlokutors, wird die Anforderung (r1) (siehe Abschnitt 3.2.3) erfüllt. Um auch Anforderung (r3) zu erfüllen, muss der Agent zusätzlich inferieren, ob der Interlokutor seinen Aufmerksamkeitsfokus intentional auf  $\vartheta$  lenkt  $BEL_i(INTEND_j(ATT_j \vartheta))$ .
- (2) (**ascribing goal**) Der Agent  $i$  muss daran glauben, dass der Agent  $j$  das Ziel verfolgt, dass beide Agenten ihre Aufmerksamkeit auf das Ziel ausrichten  $BEL_i(GOAL_j(ATT_i \vartheta \wedge ATT_j \vartheta))$ . Dieses Belief kann durch ein *initiate-act* des Agenten  $j$  beispielsweise durch Blick-Alternation (*gaze-alternation*) ausgelöst werden.
- (3) (**adopting goal**) Der Agent muss dann das Ziel des Interlokutors übernehmen  $GOAL_i(ATT_i \vartheta \wedge ATT_j \vartheta)$ . Um die Anforderungen (r4) und (r5) zu erfüllen (siehe Abschnitt 3.2.3), muss der Agent  $i$  als Rezipient einen von seinem Interlokutor wahrnehmbaren *respond-act* ausführen.
- (4) (**feedback**) Aber für ein gemeinsames Belief ist ein weiterer *respond-act* des Initiators  $j$  erforderlich, sodass der Agent  $i$  folgendes Belief annimmt  $BEL_i(BEL_j(ATT_i j))$ .
- (5) (**focus state**) Wenn der Agent  $i$  seinen Aufmerksamkeitsfokus auf das Ziel  $(T(\vartheta)_i)$  ausrichtet, während er wahrnimmt, dass sein Interlokutor ebenfalls das Ziel fokussiert  $(P(< T(\vartheta)_j >_i))$ , dann ist aus Sicht des Agenten  $i$  ein Zustand der *Joint Attention* erreicht. (Für eine Definition von  $T$  (*Test-if*) und  $P$  (*Perceive-that*) siehe Anhang Abschnitt A.1, Definition 23 und Definition 24).

## 5.3 Zusammenfassung CASEC-Architektur

Die CASEC-Architektur aktualisiert das klassische BDI-Paradigma mit aktuellen Ergebnissen der Kognitionswissenschaft. Dabei werden wesentliche Neuerungen auf verschiedenen Ebenen eingesetzt, um den Agenten zu befähigen, situativ effektiv zu agieren und dabei den komplexen Gesamtkontext mit seinen Interaktionspartner im Blick zu behalten.

Für einen kognitiven Agenten spielen der Einsatz expliziter Repräsentationen sowie dynamischer Adaptationsprozesse eine zentrale Rolle, welche zum einen

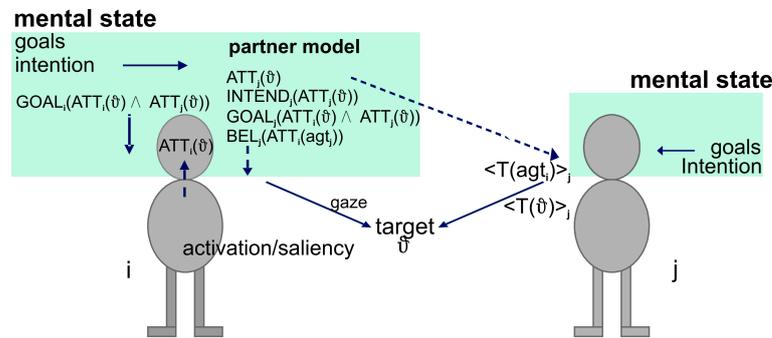


Abbildung 5.27: Joint Attention aus der Perspektive des Agenten  $i$

Meta-Reasoning Prozesse auf höherer Ebene ermöglichen (*top-down Prozesse*) und zum anderen die Modellierung von Salienz und Relevanz in der jeweiligen Situation herbeiführen (*bottom-up Prozesse*).

Die CASEC-Architektur enthält im Rahmen der Adaptationsprozesse Aktivierungsmechanismen sowie Salienzberechnungen. Sie verwirklichen

- die Modellierung eines kontextsensitiven Weltmodells,
- die Situations-sensitive Vorhaltung von prozeduralem Wissen,
- dynamische Anpassungsmechanismen der Ziel- und Intentionsstruktur,
- die Unterstützung zeitlicher Aspekte.

Die Aktivierungsprozesse spielen eine zentrale Rolle für die Modellierung kognitionswissenschaftlich orientierter Gedächtnisstrukturen, welche das Verhalten des Agenten möglichst menschenähnlich gestalten. Dadurch wird von dem Ziel eines allwissenden Agenten mit perfektem Wissensstand Abstand genommen. Stattdessen wird ein natürlicher Interaktionspartner modelliert, welcher kontextsensitiv operiert und eine natürliche Auflösung von Referenzen auf der Basis seines mentalen Zustands inkrementell im Dialog vornimmt.

#### CASEC:

- (ex1) Aktivierungswerte und Mechanismen für Beliefs (siehe 5.1.4)
- (ex2) Aktivierungswerte und Mechanismen für Ziele und Intentionen (siehe 5.2.1)
- (ex3) Aktivierungswerte und Mechanismen für Pläne (siehe 5.2.5)

Der kognitive Agent soll sich nicht allein auf die Umsetzung seines wichtigsten Ziels versteifen, sondern seinen gesamten Zielzustand als Entscheidungsbasis nehmen und dabei zukunftsorientiert Interaktionen und Konflikte sowohl zwischen aktuellen Zielen als auch zwischen Intentionen und Handlungsplänen berücksichtigen. Um das zu verwirklichen werden in der CASEC-Architektur neue Wege beschritten. Die Repräsentation prozeduraler Fähigkeiten erfolgt auf angemessenem Abstraktionsniveau, damit

- der Agent Auskunft über seine Pläne erteilen kann,
- komplexe Zielstrukturen verarbeitet werden können,
- eine parallele Verarbeitung von Zielen unter Berücksichtigung von Interaktionen und Folgen von Handlungen in Bezug auf Aufwand und Kosten vorgenommen werden kann.

Die explizite Repräsentation der Ziele und Pläne umfasst dabei folgende Attribute und Aspekte:

**CASEC:**

- (ex4) Zielattribute (Obligationen, Zeit) (siehe Abschnitte 5.1.4, 5.2.1)
- (ex5) Zielrelationen und typisierte Zielverknüpfungen (BEFORE, AFTER, OR) (siehe 5.5)
- (ex6) gemeinsame Ziele (siehe 5.2.1)
- (ex7) Effort (siehe 5.2.5)
- (ex8) Effekte (siehe 5.2.1)

Aufbauend auf der Erweiterung der Aktivierungswerte und expliziten Repräsentation von Attributen der Beliefs, Ziele und Pläne wird eine erweiterte Modellierung von Zugriffsmöglichkeiten und Abfragestrategien realisiert, welche zu einer kontextsensitiven Informationsverarbeitung führt.

Durch Modellierung verschiedener gewichteter Entscheidungsfaktoren sowohl auf *Deliberations-* als auch *Means-Ends-Reasoning*-Ebene sind verschiedene komplexe Entscheidungsstrategien modellierbar, die auch während einer Interaktion dynamisch angepasst werden können. Es gibt folgende Erweiterungen in Bezug auf die Entscheidungsfaktoren:

**CASEC:**

(ex9)  $Priority_{goal}$ ,  $Priority_{plan}$

(ex10)  $Desirability_{goal}$  (siehe Funktion 5.35),  $Desirability_{plan}$  (siehe Funktion 5.2.5)

(ex11)  $Saliency$  (siehe Abschnitt 5.2.5)

(ex12)  $Utility_{goal}$  (siehe Funktion 5.37),  $Utility_{plan}$  (siehe Funktion 5.47),  $Utility_{intention}$  (siehe Funktion 5.49)

Des Weiteren können in der CASEC-Architektur komplexere Zielrepräsentationen vorgenommen werden. Durch die *Constraint*-basierte Repräsentation von Zielen können abstrakte Ziele, Teilentscheidungen, eingegangene *Commitments* sowie zu berücksichtigende Einschränkungen unterstützt werden. Durch getypte Zielverknüpfungen und den Einsatz von Junktoren können im Gegensatz zu der ausschließlichen Bearbeitung singulärer Ziele komplexe Zielstrukturen verarbeitet werden.

**CASEC:**

(ex13) *Constraint*-basierte Repräsentation (siehe Abschnitt 5.1.4)

(ex14) Commitment (siehe Abschnitt 5.2.5)

(ex15) Junktoren und komplexe Zielstrukturen (siehe Abschnitt 5.2.1)

Im Rahmen der Modellierung kooperativen Verhaltens bietet der Einsatz der Intentionstruktur nicht nur die Basis für agenteninterne Deliberation sondern auch die komplexe Basisstruktur für Interaktionsverarbeitung. Durch Modellierung von Relevanz durch Aktivierungs- und Salienzwerte werden Adaptationsprozesse an den Interaktionspartner realisiert. Die Verwendung verschachtelter *Beliefs* stellt eine wichtige Grundlage für die Partnermodellierung dar, insbesondere, um *Beliefs* über den mentalen Zustand des Interaktionspartners repräsentieren zu können.

**CASEC:**

(ex16) erweiterte Intentionstruktur (siehe Abschnitt 5.2.7)

(ex17) Salienz und Aktivierung als Basis für Adaptationsprozesse

**(ex18)** Repräsentation modallogischer Formeln für die Partnermodellierung (siehe Abschnitt 5.1.4)

Die eingesetzte modallogische formale Spezifikation des mentalen Zustands des Agenten ermöglicht eine klare Verhaltensbeschreibung, auf deren Basis weitere Anforderungen und Definitionen erfolgen können. Die explizite Repräsentation von Salienz durch Aktivierungswerte erlaubt eine Definition übergeordneter Konzepte wie des Aufmerksamkeitsfokus und der *Joint Attention* (siehe Abschnitt 5.2.10).

Im Weiteren wird gezeigt, wie sich die CASEC-Architektur für die Realisierung eines virtuellen Kooperationspartners einsetzen lässt.



## **Teil III**

# **Realisierung eines künstlichen Kooperationspartners**



# Kapitel 6

## *cMax* - Ein virtueller Kooperationspartner

Der Agent *cooperative Max* (*cMax*) realisiert einen virtuellen Kooperationspartner auf Basis der CASEC-Architektur (siehe Kapitel 5). Eine Übersicht der entwickelten Komponenten im Rahmen des SFB 360 und deren Zusammenspiel zeigt Abbildung 6.1. Wie *cMax* die im BCTC-Modell (siehe Kapitel 3) geforderten Verhaltensweisen umsetzt, wird im Folgenden für jede Ebene des Modells anhand der Mechanismen und der resultierenden Fähigkeiten genau erläutert.

Die Präsenz von *cMax* wird visuell und auditiv durch *Virtual Reality*-Techniken realisiert. Durch die fehlenden physikalischen Beschränkungen ist es möglich, einen Agenten darzustellen, der sich relativ natürlich bewegen und innerhalb eines Dialogszenarios menschenähnliches Verhalten approximieren kann. Zur Kommunikation mit der Außenwelt verfügt *cMax* über mehrkanalige Ein- (Sprache, Blicke, Gestik) (Fink et al., 1998; Pfeiffer, 2008; Latoschik, 2001) und Ausgabemöglichkeiten (Audio und Video, Animation des Agenten) (Kopp & Wachsmuth, 2004). Während der Entwurf der CASEC-Architektur allgemein auf kommunizierende und kooperierende Agenten ausgelegt ist, was insbesondere Roboter mit einschließt, kann der volle Umfang der Interaktionsfähigkeiten derzeit nur in der Virtuellen Realität demonstriert werden.

Im folgenden Kapitel werden die in CASEC realisierten kooperativen Fähigkeiten an konkreten Beispielen zu verschiedenen Ebenen der Kooperation demonstriert. Auf dem *Behavior Layer* werden Mechanismen der Aufmerksamkeitssteuerung innerhalb des Agenten bis hin zur kooperativen Aufmerksamkeitsaushandlung im Rahmen der *Joint Attention* betrachtet. Repräsentativ für den *Conversational Layer* werden die Themen Kontaktmanagement, Turn-Taking und Dialogkompetenz detailliert vorgestellt und damit Kooperation auf der Ebene von einzelnen Dialogakten realisiert. Auf dem *Task Layer* wird auf die Anbindung externer Planprozesse eingegangen, mit deren Hilfe dynamisch situiertes Problemlösen

umgesetzt wird. Schließlich wird auf dem *Collaboration Layer* gezeigt, wie mit den mächtigen *SharedPlans* die komplexeste Form der Kooperation realisiert werden kann.

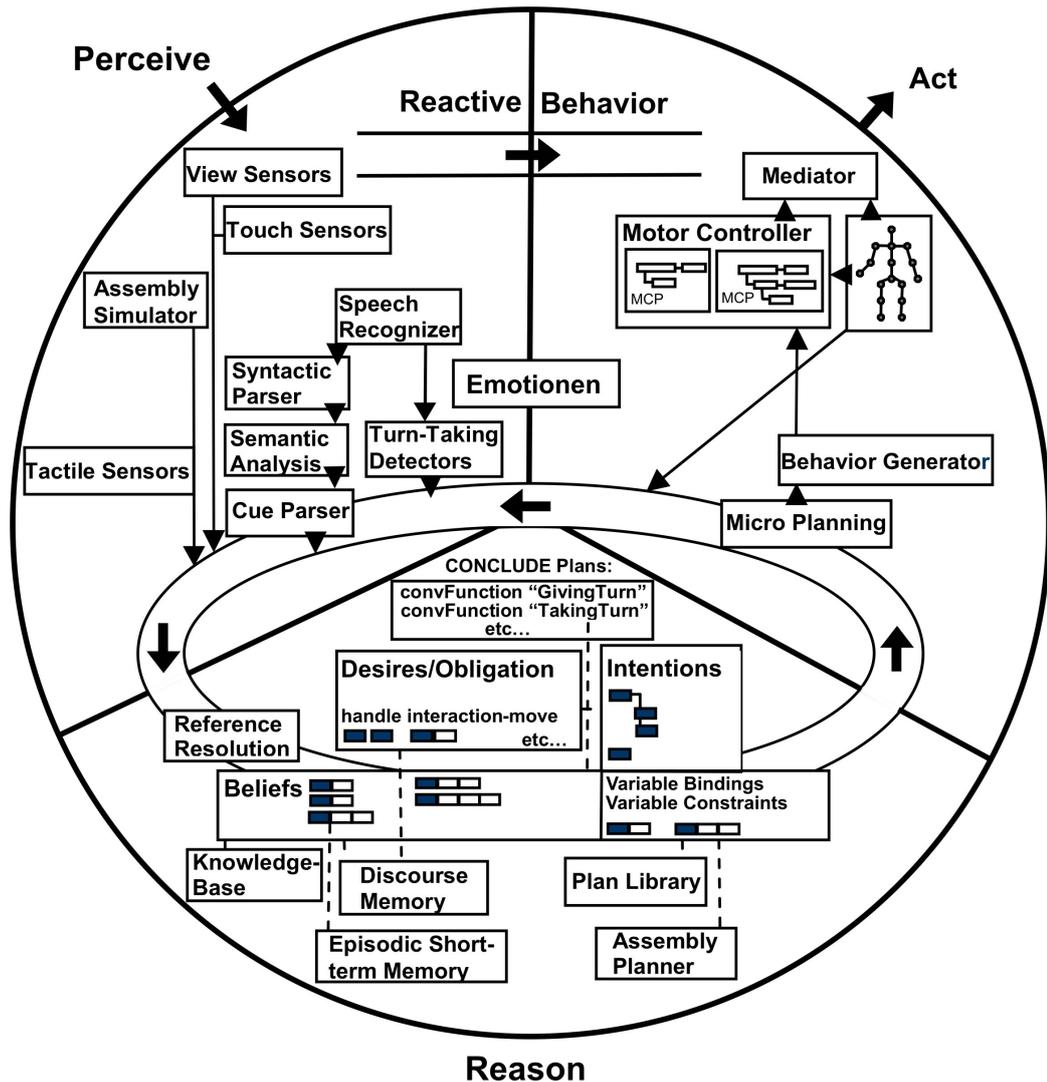


Abbildung 6.1: Detaillierte Übersicht der CASEC-Architektur nach (Leßmann et al., 2006, S.316)

## 6.1 Behavior Layer

### 6.1.1 Aufmerksamkeitsmechanismen

Aufmerksamkeitssignale spielen in einer Kooperationssituation eine wichtige Rolle. Einerseits ist es hilfreich, den Aufmerksamkeitsfokus seines Kooperationspartners zu kennen, und andererseits sollte der Agent selbst einen interessierten Eindruck vermitteln. Im Rahmen des *cMax* werden drei Verarbeitungsmechanismen in Bezug auf drei Formen von Aufmerksamkeitssignalen berücksichtigt.

- (1) **Feedback** Diese Aufmerksamkeitssignale entstehen während der Verarbeitung der Interaktionsbeiträge des Gegenübers als eine Form von Feedbacksignalen.
- (2) **Spiegel der internen Prozesse** Diese Aufmerksamkeitssignale werden durch interne Prozesse generiert. Wenn der Agent über Alternativen nachdenkt, beispielsweise welches Objekt er für eine Handlung benutzen soll, so kann er dies durch Blickbewegungen sichtbar machen.
- (3) **Joint Attention** Diese Aufmerksamkeitssignale stehen in einem direkten Zusammenhang mit den Aufmerksamkeitssignalen des Gegenübers und beziehen sich auf *Joint Attention*.

Ein Indikator für den Aufmerksamkeitsfokus eines Agenten ist sein Blickfokus. Das Blickverhalten des *cMax* wird durch *Behaviors* und *Motorprogramme* auf unterer Ebene gesteuert. Dabei verfügen die einzelnen Motorprogramme über spezifizierte Prioritäten (Kopp & Wachsmuth, 2004). Angestoßen werden diese Behaviors durch verschiedene Prozesse. Im Folgenden werden die drei Mechanismen detaillierter vorgestellt.

#### Aufmerksamkeitssignale als Feedback

Aufmerksamkeitssignale bieten eine Form des Feedback bei der Verarbeitung sprachlicher Äußerungen. Dazu wird im *cMax* ein *Cue-Processor* eingesetzt. Um den Anforderungen des *gegenseitigen Aufeinandereingehens* (siehe Abschnitt 2.1.1) gerecht zu werden, wird eine *inkrementelle Verarbeitung* vorgenommen. Der Parser ist in der Lage, zwei Formen der Eingabe zu verarbeiten und baut auf den Daten des textuellen Präprozessors auf. (Jung & Wachsmuth, 1995, 1998). Zum einen operiert er auf *Word-Messages*, die ein erkanntes Wort umfassen. Diese Verarbeitung wird in dem Agenten dazu eingesetzt, Referenzen inkrementell zu analysieren und dem Gegenüber schritthaltend einen Einblick in die interne

Verarbeitung des Agenten zu gewähren. Der Agent gibt durch Ausrichten seines Blickfokusses einen Hinweis darauf, dass er eine Referenz seines Gegenübers gehört hat und sie analysiert, um die Referenz aufzulösen.

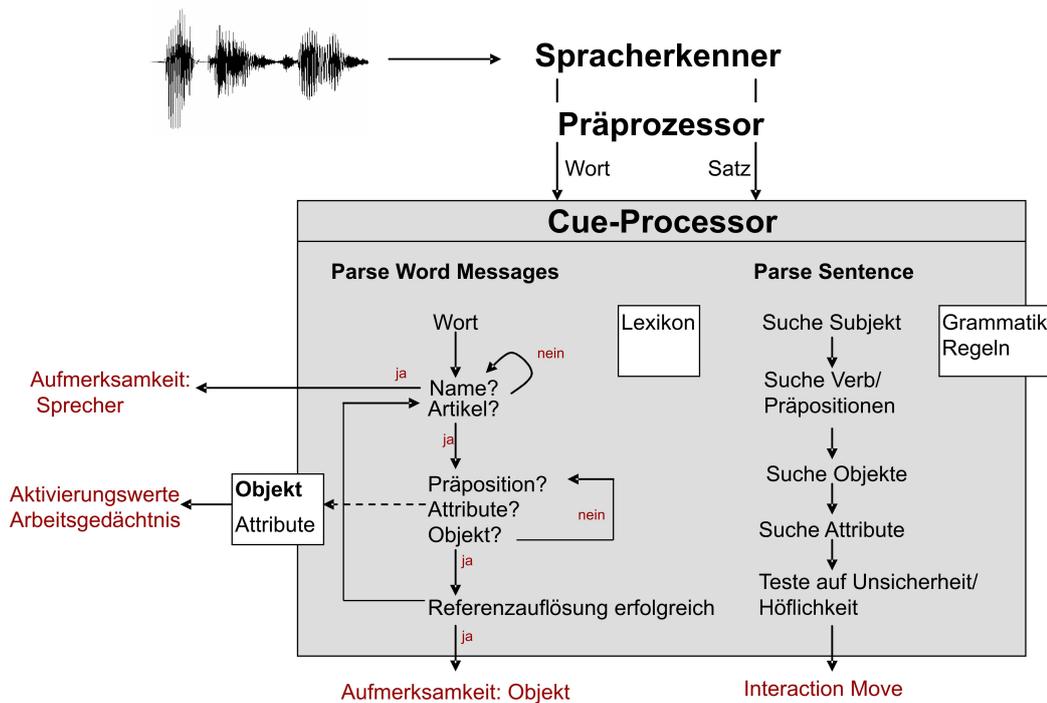
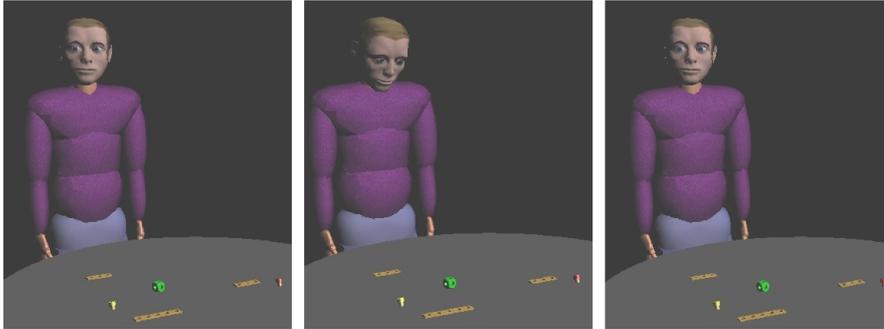


Abbildung 6.2: Verarbeitungsmechanismen des *Cue-Processor*

Da Menschen in der Regel besonders sensitiv auf das Nennen ihres Namens reagieren, werden die eingehenden *Word-Messages* daraufhin überprüft, ob der Agent direkt mit seinem Namen angesprochen wurde. Wird der Name erkannt, so wird der Aufmerksamkeitsfokus und der Blick des Agenten direkt auf den Sprecher ausgerichtet. Ansonsten wird nach Eigennamen bzw. einem Artikel als Indiz auf anschließend genannte Objekt gesucht. Für die Erkennung von Worttypen greift der Parser auf ein Lexikon zurück. Wird ein Artikel gefunden, so werden die darauf folgenden Worte nach Attributen und Präpositionen durchsucht, welche dann in einer Datenstruktur vorgehalten werden. Dies geschieht so lange, bis der *Parser* einen Objektnamen bzw. Objekttyp (Nomen) erkennt und eine Referenzauflösung anstößt. Die Referenzauflösung kann aus einer einfachen Suche im Arbeitsgedächtnis bestehen oder eine komplexe Referenzauflösung (Jung & Wachsmuth, 1995; Pfeiffer & Latoschik, 2004) anstoßen. Ist die Referenzauflösung erfolgreich, so wird ein Aufmerksamkeitsimpuls auf das bzw. die erkannten Objekte

ausgelöst. Befinden sich die Objekte in der Umwelt des Agenten, wird ein Blickfokussierungsverhalten angestoßen. Abbildung 6.3 zeigt ein Beispiel der Blickfokussierung während einer Referenzauflösung (Leßmann & Kopp, 2005).



**User:** Wir könnten diese Schraube [*Blickfokus Schraube*] in eine Leiste stecken.

**cMax:** [*guckt zur rechten Schraube, dann zum Interlokutor*] Okay.

Abbildung 6.3: Beispiel Blickfokussierung während der Referenzauflösung siehe auch (Leßmann & Kopp, 2005)

### Aufmerksamkeitssignale aus internen Prozessen

Für die Generierung von Aufmerksamkeitssignalen, welche ihre Ursprünge in der internen Verarbeitung des Agenten besitzen, wird auf das dynamische Arbeitsgedächtnis des Agenten zurückgegriffen (siehe Abschnitt 4.1.3). Es existiert ein expliziter Plan für die Referenzauflösung bzw. das Verifizieren eines Objekts in der Planbibliothek des Agenten (*PERFORM verify\_object \$obj \$candidate*). Diesem Plan wird das Objekt und der potentielle Objektkandidat übergeben. Im *Plan-Body* werden die grundlegenden *Unify/Match*-Mechanismen des Arbeitsgedächtnisses (siehe Abschnitt 5.1.7) dazu genutzt, den Objektkandidaten zu verifizieren und mit dem Objekt abzugleichen. Dabei werden insbesondere die spezifizierten *Constraints* des Objekts berücksichtigt. Zusätzlich verwaltet dieser Plan Mechanismen für die Produktion von Aufmerksamkeitssignalen. Der Plan findet sowohl bei der Verifizierung einer extern genannten Referenz Einsatz als auch bei der Abwägung des Agent zwischen Alternativen. Konnte eine Referenz erfolgreich aufgelöst werden bzw. wurde ein passender Kandidat gefunden, so wird ein kleiner positiver Impuls an das Emotionssystem des Agenten gesendet. Für die Modellierung der Emotionen wird ein dynamisches Emotionssystem eingesetzt (Becker et al., 2004).

Damit der Agent die Objekte nicht bei jeglicher Nennung fokussiert, was sehr unnatürlich wirken würde, ist eine *Heuristik* vorgesehen: Wurde ein Objekt bereits kurze Zeit vorher (Default: Zeitintervall = 10s) fokussiert, so wird der Agent das Objekt bei bloßer Nennung bzw. Erwägung nicht erneut fokussieren. Es können jedoch andere aufmerksamkeitsorientierte Mechanismen eine Fokussierung auf das Objekt auslösen. Wurde längere Zeit nicht auf das Objekt referenziert, was sich anhand des *last\_retrieved* und dem Aktivierungswert des Objektkandidaten ablesen lässt, so werden auch wieder Blickfokussierungen aufgrund bloßer Nennung auf das Objekt zugelassen. Das Arbeitsgedächtnis hält die benötigten Informationen für ein natürliches Blickverhalten bereit und sorgt für ein dynamisches, situierendes Aufmerksamkeitsverhalten.

## 6.1.2 Joint Attention

### Detektierung des Aufmerksamkeitsfokus des Interlokutors

Als ein Indikator für den Aufmerksamkeitsfokus des menschlichen Interaktionspartners wird der **Blick des Interlokutors** verfolgt, da die Blickverfolgung einen wichtigen Bestandteil gemeinsamer Aufmerksamkeit darstellt. Technisch wird ein Eyetracking System mit einem optischen Motion-Tracking-System kombiniert, um die Blickrichtung im Raum und letztendlich die betrachteten Objekte zu ermitteln (Pfeiffer, 2008).

Um die Objekte zu bestimmen, welche in dem **Aufmerksamkeitsfokus** des menschlichen Interaktanten liegen, wird ein Kegel von  $2,5^\circ$  eingesetzt (siehe (Pfeiffer, 2008)). Zusätzlich zu der Begrenzung des Kegels werden Aktivierungswerte des Arbeitsgedächtnisses in die Berechnung der potentiellen Kandidaten miteinbezogen (siehe Abbildung 6.4). Dadurch wird nicht nur der Blickwinkel des Interaktanten berücksichtigt, sondern auch der situationale Kontext (Pfeiffer-Leßmann & Wachsmuth, 2008). Die Bestimmung des ausgewählten Objekts wird durch **Histogrammberechnungen** vorgenommen. Abweichend von Pfeiffer (2008), der generelle Detektion behandelt, werden hier längere, bewusster eingesetzte Fixationen betrachtet. Daher wird die erforderliche Dauer einer Fixation erhöht. Ein Objekt wird detektiert und dem Aufmerksamkeitsfokus des Interlokutors zugeordnet, wenn es mindestens für 400 ms in einem 600 ms Zeitrahmen fokussiert wurde. Zusätzlich zu dieser Heuristik können weitere Kommunikationskanäle genutzt werden; so können Zeigegesten oder verbale Äußerungen als intentionale Hinweise dafür interpretiert werden, dass die Aufmerksamkeit auf ein bestimmtes Objekt gerichtet wird.

Wenn aufmerksamkeitsbezogene Signale eine gewisse Intensität übersteigen, wird ein daraus resultierendes *Belief-Chunk* (*ATT \$interlocutor \$obj*) in das Arbeitsgedächtnis eingetragen oder der Aktivierungswert und Zeitstempel eines be-

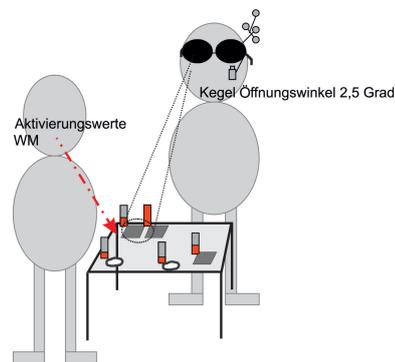


Abbildung 6.4: Aufmerksamkeitsdetektion

stehenden *Belief-Chunks* erhöht (siehe Abbildung 6.5). Durch den Einsatz des dynamischen Arbeitsgedächtnisses führt die Assertierung der *Belief-Chunks* zu Aktivierungsimpulsen an das involvierte Objekt (siehe Abschnitt 5.1.5). Der erhöhte Aktivierungswert sorgt dafür, dass das Objekt bei Referenzauflösungs- und Entscheidungsprozessen bevorzugt wird.



Abbildung 6.5: *cMax* glaubt, dass der Interlokutor seinen Aufmerksamkeitsfokus auf einen bestimmten Park mit der Identifikationsnummer (*5:Park*) ausgerichtet hat

Da aber nur *intentionales* Fokussieren als eine Aufforderung gemeinsamer Aufmerksamkeit gesehen wird (siehe Anforderung (**r3**), Abschnitt 3.2.3), folgt die Modellierung derjenigen von Ogasawara et al. (2005): Die Intensität der direkt wahrnehmbaren Verhaltensweisen wird als Indikator auf die Intention des

Interaktanten interpretiert. Die Intensität wird anhand des benutzten Kommunikationskanals und der Redundanz der aufmerksamkeitsbezogenen Signale des Interlokutors bestimmt. Im Falle der Blickfokussierung gilt, dass der Interlokutor das Objekt in einem Zeitfenster von 5s mehrfach deutlich fokussiert haben muss oder das Objekt für mindestens 3s fokussiert haben muss, damit von einer intentionalen Blickfokussierung ausgegangen wird (für die Definition von *Dur* (*Duration*) siehe A.1 Definition 26):

$$\begin{aligned} & ((\{P(< T(\varphi) >_j)\}_i)_{[t_1]}; (\{P(< T(\varphi) >_j)\}_i)_{[t_2]} \wedge (t_2 - t_1 \leq 5s) \\ & \vee (Dur(\{P(< T(\varphi) >_j)\}_i) \geq 3s) \Rightarrow (BEL_i(INTEND_j(ATT_j \varphi))) \end{aligned} \quad (6.1)$$

Die zeitlichen Bedingungen werden unter Berücksichtigung von Tasker & Schmidt (2008) gewählt. Sie betonen, dass eine Episode der *Joint Attention* zeitlichen Einschränkungen unterliegt. So muss die Antwort eines Interaktanten laut Nadel & Camaioni (1993) in einem Zeitfenster von 5s liegen. Spencer (2000) postulieren ein Zeitfenster von 3s.

Damit der Agent seinem Interaktionspartner das Ziel zuspricht, gemeinsame Aufmerksamkeit eingehen zu wollen, wird die folgende Heuristik angewendet:

$$\begin{aligned} & (BEL_i(INTEND_j(ATT_j \varphi)))_{[t_1]}; (BEL_i(INTEND_j(ATT_j i)))_{[t_2]}; \\ & (BEL_i(INTEND_j(ATT_j \varphi)))_{[t_3]}; (BEL_i(INTEND_j(ATT_j i)))_{[t_4]} \wedge (t_4 - t_1 \leq 20s) \\ & \wedge (t_1 \leq t_2 \leq t_3 \leq t_4) \Rightarrow (BEL_i(GOAL_j(ATT_i \varphi)) \wedge (ATT_j \varphi)) \end{aligned} \quad (6.2)$$

Ein Objekt muss mehrfach im Aufmerksamkeitsfokus liegen, und es müssen zusätzlich kurze Blicke auf den Interaktionspartner ausgerichtet werden (triadische Interaktion). Andernfalls könnte es sein, dass der Interlokutor das Objekt aus einem anderen Grund fokussiert und nicht, um den Aufmerksamkeitsfokus seines Gegenübers auf das Objekt zu lenken.

Das Verhalten der Blickalternierung (*gaze alternation*) wird dann als eine Initiierungshandlung (*initiate-act*) interpretiert. Wenn der Aktivierungswert eines Aufmerksamkeitsfokus-Beliefs einen Schwellwert übersteigt und der Interlokutor interaktive Blicke eingesetzt hat, entsteht in dem Arbeitsgedächtnis des Agenten das Belief (*GOAL \$interlocutor (ATT „self“ \$object)*) (siehe Definition 18 (2)), was zu der Aktivierung eines *Conclude-Plans* führt. Dadurch wird dem Agenten die Intention seines Gegenübers „bewusst“ und er kann entscheiden, wie er darauf reagieren möchte. Beispielsweise kann er das Ziel aufgreifen, das gleiche Objekt fokussieren und einen Zustand *gemeinsamer Aufmerksamkeit* herbeiführen.

### **Manipulation des Aufmerksamkeitsfokus des Interlokutors**

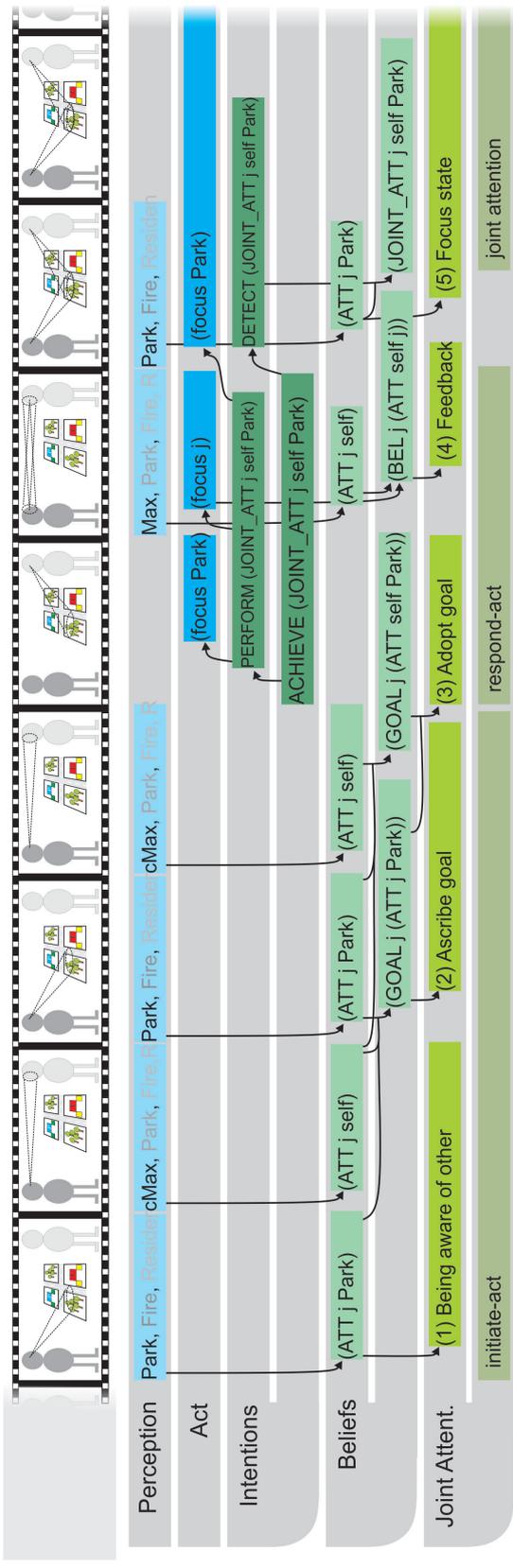
Während die Detektion des Aufmerksamkeitsfokus des Gegenübers als Vorbedingung für das Erreichen gemeinsamer Aufmerksamkeit gelten kann, verfügt der Agent auch über pro-aktive Mechanismen, um den Aufmerksamkeitsfokus seines Interaktionspartners zu manipulieren. Als verkörperter, konversationaler Agent verfügt *cMax* über verschiedene Möglichkeiten aufmerksamkeitsbezogene Hinweise auszusenden. Zu diesen zählen intentional eingesetztes Blickverhalten, Zeigegesten und verbale Ausdrücke (Leßmann et al., 2006).

Wenn der Agent den Aufmerksamkeitsfokus seines Gegenübers auf ein Objekt lenken möchte, so wird dies durch ein explizit repräsentiertes Ziel (*ACHIEVE ATT \$interlocutor \$object*) angestoßen. Die Entscheidungsfindung für einen Handlungsplan, den Aufmerksamkeitsfokus des Interlokutors zu manipulieren, geschieht nach den spezifizierten Mechanismen des *Means-Ends-Reasoning* unter Berücksichtigung der Aktivierungswerte und des *Utility*-Werts (siehe Abschnitt 5.2.4).

Der Plan mit dem geringsten *Effort* besteht aus der Aussendung von Aufmerksamkeitssignalen durch Blickfokussierung. Wenn dies nicht zu einem Erfolg führt, werden offensichtlichere Signale eingesetzt. Wegen des Zieltyps *Achieve* wird der Plan nur so lange ausgeführt, bis der Aufmerksamkeitsfokus des Gegenübers auf das richtige Objekt gerichtet ist. Sobald der Agent glaubt (*ATT \$interlocutor \$obj*), ist das Ziel erreicht und der Agent konzentriert sich auf das nächste Ziel seiner Agenda. Scheitert der Plan und der Interaktionspartner reagiert trotz des produzierten Verhaltens nicht in angemessener Weise, wird erneut ein Handlungsplan bestimmt und der Agent produziert z.B. eine verbale Äußerung, durch die er sein Ziel explizit macht.

### **Erreichen von *Joint Attention***

Nachdem beschrieben wurde, wie der Agent durch Erkennungsprozesse und Handlungen die einzelnen Bedingungen an den mentalen Zustand (siehe Definition 18, Abschnitt 5.2.10) erfüllen kann, wird im Folgenden gezeigt, wie die Erkennungsprozesse und Handlungen koordiniert werden müssen, um *Joint Attention* zu erreichen.



**Erläuterungen:**

Die erste Spur (*Perception*) zeigt die von *cMax* detektierten Wahrnehmungshandlungen des Interlokutors *j*. In der zweiten Spur (*Act*) sind die relevanten von außen wahrnehmbaren *Perceive*-Handlungen von *cMax* dargestellt.

In der dritten Spur (*Intentions*) werden die für *Joint Attention* relevanten Intentionen des Agenten *cMax* aufgezeigt.

Die vierte Spur (*Beliefs*) zeigt die relevanten Beliefs des Agenten *cMax*. Dabei sind die aus einfachen Beliefs abgeleiteten modallogischen Beliefs, die sich auf die Beliefs und Ziele des Interlokutors *j* beziehen, in einer extra Spur angegeben. Die Pfeile zeigen, aufgrund welcher Beliefs die modallogischen Beliefs abgeleitet wurden.

In der fünften Spur (*Joint Attention*) wird gezeigt, wann die Bedingungen der Definition von *Joint Attention* (siehe Abschnitt 5.2.8, Def. 20) jeweils in Kraft treten. Dabei sind der *initiate-act* des Interlokutors, der *respond-act* von *cMax* und der Zustand der *Joint Attention* extra markiert.

Abbildung 6.6: Ablaufskizze *Joint Attention*

Abbildung 6.6 zeigt eine Ablaufskizze einer *Joint Attention*-Interaktion. Der zeitliche Verlauf der Interaktion ist dabei nur qualitativ erfasst. Es werden die relevanten Handlungen und Wahrnehmungen sowie der mentale Zustand aus Sicht eines der Interaktionspartner präsentiert.

Für das Erreichen von *Joint Attention* werden im *cMax* drei Hauptpläne eingesetzt. Der erste Plan besteht aus einem *Achieve*-Plan, der *Joint Attention* mit dem Interaktionspartner in Bezug auf ein Objekt herbeiführen soll (siehe Pseudocode 7). Dieser Plan instantiiert das Ziel (*ACHIEVE detect „JOINT\_ATT“ „self“ \$interlocutor \$obj*). Dieses Ziel wird als *Top-level* Ziel auf den Intentionsstack geschoben und operiert mit einer sehr hohen Priorität parallel zu dem *Achieve*-Plan (siehe Pseudocode 6). Der Detektionsplan enthält als komplexe Vorbedingungen die Bedingungen an den mentalen Zustand des Agenten, die nötig sind, damit dieser an *Joint Attention* glaubt (siehe Definition 18, Abschnitt 5.2.10). Der Agent überprüft immer wieder, ob die Bedingungen für *Joint Attention* gelten. So bald dies der Fall ist, assertiert der Plan automatisch den *Belief*, dass *Joint Attention* zwischen dem Agenten und seinem Interlokutor detektiert wurde (*detect JOINT\_ATT „self“ \$interlocutor \$obj*).

Der Detektionsplan stellt einen passiven Plan dar. Um *Joint Attention* herbeizuführen, wird neben dem Detektionsplan ein Handlungsziel als Teilziel instantiiert. Dieses besteht aus dem *Perform*-Ziel (*PERFORM JOINT\_ATT „self“ \$interlocutor \$obj*). Ein dazu passender Handlungsplan (siehe Pseudocode 8) überprüft, welche Bedingungen für *Joint Attention* schon gültig sind, und instantiiert bei Bedarf entsprechende Handlungspläne (*initiate-acts*). Waren die *initiate-acts* erfolgreich und wurden die geforderten *respond-acts* abgesetzt, so stößt der Agent als letztes noch eine Blickfokussierung (wahrnehmbare *Test*-Aktion, siehe Definition 24 in Abschnitt A.1) in Bezug auf das Objekt *\$obj* an. Damit sollte die letzte fehlende Bedingung für den Detektionsplan erfüllt sein. Der *Perform*-Plan beendet sich, im *Achieve*-Plan ist die Wartebedingung erfüllt (*WAIT detect „JOINT\_ATT“ „self“ \$interlocutor \$obj*) und somit wird automatisch das *Belief* (*JOINT\_ATT „self“ \$interlocutor \$obj*) in das Arbeitsgedächtnis des Agenten eingetragen. Damit sind alle Bedingungen an den mentalen Zustand des Agenten erfüllt, beide Interaktionspartner fokussieren das Objekt - *Joint Attention* wurde erreicht.

---

**Algorithm 6** Detektionsplan Joint Attention

---

```

1: procedure ACHIEVE DETECT „JOINT_ATT“ „SELF“ $AGENTJ $OBJ
    ▷ PRECONDITION: ( (FACT ATT $agentJ $obj
        ▷ ∧ (FACT ATT „self“ $obj)
    ▷ ∧ (FACT GOAL $agentJ [ATT $agentJ $obj])
        ▷ ∧ (FACT GOAL $agentJ [ATT „self“ $obj])
            ▷ ∧ (GOAL [ATT „self“ $obj])
        ▷ ∧ (GOAL [ATT „agentJ“ $obj]) )

2:     PRIORITY: 50
3: end procedure

```

---



---

**Algorithm 7** Achieve-Plan Joint Attention

---

```

1: procedure ACHIEVE „JOINT_ATT“ „SELF“ $INTERLOCUTOR $OBJ

2:     POST ACHIEVE detect „JOINT_ATT“ „self“ $interlocutor $obj
3:     PERFORM JOINT_ATT „self“ $interlocutor $obj
4:     WAIT detect „JOINT_ATT“ „self“ $interlocutor $obj
5:     PRIORITY: 10
6: end procedure

```

---

**Algorithm 8** Handlungsplan Joint Attention

---

```

1: procedure PERFORM „JOINT_ATT“ „SELF“ $AGENTJ $OBJ
2:   $now ← get_current_time()
3:   while ((¬ FACT_NEWER $now GOAL $agentJ [ATT $agentJ $obj] )
   ∧ (¬ FACT_NEWER $now ATT $agentJ $obj)
   ∧ (¬ FACT_NEWER $now GOAL $agentJ [ATT $agentJ $obj])
   ∧ (¬ FACT_NEWER $now GOAL $agentJ [ATT „self“ $obj])
   ∧ (¬ FACT_NEWER $now ATT „self“ $obj) ) do
4:     if ((FACT_NEWER $now ATT $agentJ $x) ∧ ($x ≠ $obj)) then
5:       activateDetectors
6:     end if
7:     if ((¬ FACT_NEWER $now GOAL $agentJ [ATT $agentJ $obj] ) ∨
   (¬ FACT_NEWER $now GOAL $agentJ [ATT „self“ $obj])) then
8:       PERFORM initiate_act $agentJ $obj
9:     end if
10:  end while
11:  if ( ¬(FACT_NEWER $now ATT „self“ $obj) ) then
12:    PERFORM focus_on $obj
13:  end if
14:  PRIORITY: 12
15: end procedure

```

---

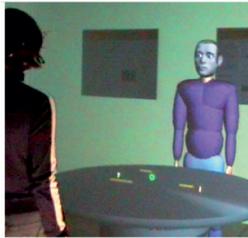
## 6.2 Conversation Layer

### 6.2.1 Kontaktmanagement

Damit der Agent mit seinem Kooperationspartner in eine Interaktion treten kann, benötigt er die im Rahmen des *BCTC*-Modells aufgestellten Regeln des Kontaktmanagements (**k1**) - (**k3**) (siehe Abschnitt 3.3.2). Im *cMax* sind diese durch Obligationen umgesetzt.

**(k1)** Wird ein neuer Konversationspartner wahrgenommen, so assertiert die Wahrnehmung ein entsprechendes Belief. Die Wahrnehmung des menschlichen Gegenübers geschieht bei *cMax* über den Eyetracker (siehe 6.1.2). Der Basisplan (*CONCLUDE conversationalPartner \$agent*) registriert den potentiellen Konversationsteilnehmer und lenkt den Blick auf ihn. Anschließend wirft der Plan die Obligation auf, den Agenten zu begrüßen (*PERFORM greet \$agent*) (**contact-rule 1a**). Für die Ausführung gibt es verschiedene Handlungsoptionen; so kann

*cMax* seinem Gegenüber zunicken oder ihm zuwinken, er kann auch freundlich „hallo“ oder „guten Tag“ sagen (siehe Abbildung 6.7). Wird *cMax* angesprochen, so besteht die Obligation darauf zu antworten (**contact-rule 1c**). Dies wird durch Diskursregeln (siehe Abschnitt 6.2.3) umgesetzt.



Durch Detektion der mit Markern versehenen Brille erkennt *cMax* einen Konversationspartner.



*cMax* begrüßt den Interlokutor mit „Hallo“ und einem freundlichen Winken.

Abbildung 6.7: Kontaktmanagement: Begrüßen eines potentiellen Konversationspartners

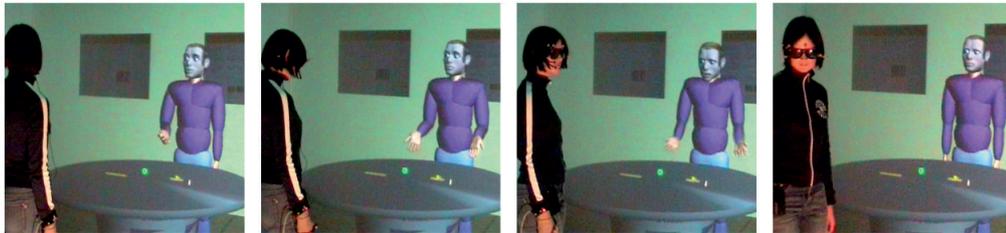
**(k2)** Das Kontaktverhalten des Agenten während einer Interaktion wird zum einen durch *Turn-Taking*-Pläne realisiert, die das Verhalten des Agenten in Bezug auf das Abwechseln der Redebeiträge organisieren. Sie werden im anschließenden Abschnitt über *Turn-Taking* 6.2.2 diskutiert. Andererseits existiert zusätzlich ein Plan, der den Aufmerksamkeitsfokus des Agenten auf Handlungen des Gegenübers lenkt (**contact-rule 2c**). Alle wahrgenommenen Ereignisse und damit auch die Handlungen des Gegenübers werden als Belief assertiert und durch *Conclude*-Pläne verarbeitet, z.B. (*CONCLUDE event \$actor \$val1 \$val2*). Befindet sich der Agent in einer Konversation mit einem Interlokutor, so wird ein *Conclude*-Plan aktiviert, der die Ereignisse mit einer höheren Priorität behandelt, die den Interlokutor involvieren. Dafür dient ein weiterer *Conclude*-Plan mit höherer Priorität, der als Vorbedingungen Plan-Bedingungen der Form (*FACT conversationWith \$interlocutor*) und (*== \$actor \$interlocutor*) enthält. Im *Plan-Body* wird die Behandlung des Ereignisses angestoßen. Außerdem wird dafür gesorgt, dass der Agent seinen Aufmerksamkeitsfokus sichtbar dem Ereignis zuwendet (Fokussierung  $\geq 2s$ ), damit der Interlokutor wissen kann, dass der Agent das Ereignis wahrgenommen hat.

**(k3)** Die Regeln (**contact-rule 3a**) - (**contact-rule 3d**) dienen der Beendigung der Interaktion, wenn der Interlokutor sie abbrechen möchte. Dafür muss der Agent erkennen, wann der Interlokutor nicht mehr richtig in die Interaktion involviert ist (**contact-rule 3a**), (**contact-rule 3b**). Umgesetzt wird dies durch Erwartungen in Bezug auf Diskursregeln; hat der Agent dem Interlokutor den Turn zugewiesen, so erwartet er eine Reaktion (siehe Abschnitt 6.2.3). Geht der Interlokutor nicht darauf ein, so wird der Agent sein Ziel noch einmal mit deutlicheren Signalen verfolgen. Glaubt der Agent, dass der Interlokutor die Interaktion möglicherweise beenden möchte, so wird die Obligation aufgeworfen, explizit nachzufragen, ob der Interlokutor die Interaktion wirklich beenden möchte (**contact-rule 3c**). Das Belief (*POSSIBLY (GOAL \$interlocutor (convState „noConv“))*) führt zu der Instantiierung des Ziels (*PERFORM verify (GOAL \$interlocutor (convState „noConv“))*). Glaubt der Agent, dass der Interlokutor die Interaktion wirklich beenden möchte, beendet er die Interaktion explizit (**contact-rule 3d**) (*PERFORM endConversation \$interlocutor*).

Die Regeln zur aktiven Beendigung eines Kontakts durch den Agenten (**contact-rule 3e**) werden durch die folgenden Pläne umgesetzt. Damit der Agent erkennt, ob er selbst die Konversation beenden sollte, wird ein Plan eingesetzt, der überwacht, ob der Agent glaubt, sich in einer Konversation mit einem anderen Agenten (*BEL conversationWith \$agent*) und zugleich im konversationalen Zustand *no-conversation* zu befinden. Der Agent setzt dafür einen *Conclude*-Plan ein mit der Vorbedingung (*BEL conversationWith \$agent*) und der zu überwachenden Bedingung (*BEL conversationalState „noConv“*). Im *Plan-Body* überprüft der Agent, ob er konversationale Ziele in Bezug auf den Interaktionspartner *\$agent* verfolgt. Ist dies der Fall, wird die Einschätzung des konversationalen Zustands auf *gap* gesetzt, d.h. es wird zwar gerade nichts gesagt, aber die Agenten befinden sich noch immer in einer Gesprächssituation. Der Agent kann dies nutzen, um seine konversationalen Ziele weiter zu verfolgen, es kann aber auch sein, dass er gerade noch auf die Ausführung einer Handlung oder Ähnliches wartet. Dann bleibt die Gesprächssituation *gap* bestehen, bis einer der Agenten wieder den Turn ergreift. Bestehen von Seiten des Agenten keine Obligationen oder Ziele in Bezug auf den Interlokutor, so wird eine Obligation aufgeworfen, das Gespräch explizit zu beenden (*PERFORM endConversation \$agent*). Der Agent kann diesen Plan auch einfach direkt einsetzen, wenn er selbst den Kontakt bzw. die Interaktion mit seinem Gegenüber beenden möchte. Der Plan löscht das Belief (*BEL conversationWith \$agent*) und wickelt die Beendigung durch eine Verabschiedung ab.

Abbildung 6.8 präsentiert ein Beispiel für das Beenden einer Interaktion. In diesem hält sich der menschliche Interlokutor nicht an die Kontaktmanagementregeln. Er beendet die Konversation nicht, sondern reagiert einfach nicht auf die Interaktionsversuche von *cMax* und wendet sich ab. Das Beispiel illustriert sehr gut, wie *cMax* zunächst erkennt, dass der Interlokutor den Turn nicht ergreift,

obwohl er das müsste (siehe Diskursregeln Tabelle 3.1), da er ihn gerade explizit zugewiesen bekam. *cMax* versucht zunächst, durch eine weitere *Giving-Turn* Geste seinen Interlokutor zu einer Reaktion aufzufordern (a). Als dieser nicht reagiert, versucht er es erneut mit einer deutlicheren Handlung (b). Nachdem *cMax* davon ausgeht, dass sein Interlokutor den Kontakt abbrechen möchte, übernimmt er die Abwicklung und erklärt die Konversation für beendet („Dann belassen wir es dabei.“)(c). Die nicht erfüllten Erwartungen und Verstöße gegen die Kontaktmanagementregeln, wirken sich auf den emotionalen Zustand von *cMax* aus; er ist zunehmend verärgert (d). Zur Simulation des emotionalen Zustands des *cMax* siehe (Becker et al., 2004).



(a) Interlokutor ergreift den zugewiesenen Turn nicht. *cMax* macht eine *Giving-Turn* Geste.

(b) *cMax* versucht erneut Interaktion fortsetzen. Interlokutor geht nicht darauf ein.

(c) *cMax* versichert sich, ob der Interlokutor die Konversation beenden möchte.

(d) Interlokutor geht darauf nicht ein. Die Konversation ist beendet. *cMax* ist verärgert.

Abbildung 6.8: Beenden einer Konversation unter Berücksichtigung der Kontaktmanagementregeln

## 6.2.2 Turn-Taking

*Turn-Taking* spielt im Rahmen kooperativer Szenarien eine entscheidende Rolle. Die Umsetzung der *Turn-Taking*-Fähigkeiten des *cMax* folgt in ihrer Struktur dem Schichtenmodell der Wahrnehmungs- und Handlungsgenerierung von Thorisson (siehe 2.2.2). Die multimodale Wahrnehmung wird durch ein kontextsensitives Detektornetz umgesetzt. Die daraus abgeleiteten und erkannten *konversationalen Funktionen* stellen zusammen mit dem *konversationalen Zustand*, der Verteilung der *konversationellen Rollen* sowie dem Diskurskontext die relevante Wissensrepräsentation dar. Die Entscheidungsfindung geschieht in *cMax* im Gegensatz zu anderen Systemen (2.3) nicht in einem einzigen zentralen Schritt, sondern setzt sich aus verschiedenen Entscheidungsprozessen zusammen, welche auf

unterschiedlichen Zeitskalen und unter Berücksichtigung verschiedener Kontextfaktoren operieren. Auch die Handlungsgenerierung wird durch unterschiedliche Prozesse auf verschiedenen Ebenen angestoßen und äußert sich durch Blickverhalten, Gesten und das Aushandeln des Turns (siehe (Leßmann et al., 2004)).

### Erkennen von *Turn-Taking*-Signalen - Detektoren

Für die Erkennung von *Turn-Taking*-Signalen des menschlichen Interlokutors werden ein *Infrarot-Kamera-Tracking*-System (Pfeiffer, 2008), Datenhandschuhe, sowie ein Spracherkennungssystem (Fink et al., 1998) eingesetzt. Während ein Parser versucht, die Ergebnisse des Spracherkenners zu interpretieren, wird das **Sprachsignal** auch direkt an einen *is-speaking*-Detektor geleitet. Die Aufgabe dieses Detektors besteht darin, Sprechakte zu erkennen. Er reagiert erst ab Äußerungen einer gewissen Länge, sodass Feedbacksignale wie „hmm“ und „okay“ nicht als Sprechakt im Sinne eines Turns gewertet werden.

Zusätzlich zur Sprachverarbeitung werden Detektoren für **Gestenerkennung** und multimodale Integration eingesetzt. Die Detektoren sind als Berechnungsknoten hierarchisch in Netzen organisiert und durch das PrOSA-Framework umgesetzt (Latoschik, 2001) (Biermann & Wachsmuth, 2003). Die Detektoren beziehen den aktuellen Konversationszustand in ihre Berechnungen mit ein und operieren kontextsensitiv. So wird beispielsweise eine *TakingTurn*-Geste nicht als solche erkannt, wenn das Gegenüber bereits im Besitz des Turns ist; umgekehrt ist es nicht sinnvoll, eine *GivingTurn*-Geste während des eigenen Turns zu klassifizieren. Die erkannten Gesten, Blickrichtungen und sprachlichen Signale werden durch die Detektoren auf *Funktionen* abgebildet. Innerhalb der kognitiven Komponente spielen die konkreten Ausprägungen der Signale keine Rolle, es geht vielmehr darum, die Funktion und Intention hinter den Signalen und Handlungen zu erkennen. Die Ausdrucksformen der Signale können die *Dringlichkeit* ihrer Funktion verdeutlichen. Die Dringlichkeit geht als *Urgency*-Faktor in den Entscheidungsprozess ein.

Abbildung 6.9 zeigt eine ausgeprägte Form einer *WantingTurn*- bzw. *TakingTurn*-Geste. Die flache Hand ist erhoben und die Handfläche zeigt nach vorne. Diese Geste wird als Signal für eine große Dringlichkeit interpretiert. Eine Form einer detektierbaren *GivingTurn*-Geste gleicht einer Gebe-Geste. Die flache Hand wird mit der Handfläche nach oben und leicht nach links geneigt, wobei die Fingerspitzen nach vorne zeigen (siehe Abbildung 6.10).

Die Detektoren werden in Abhängigkeit des konversationalen Zustands aktiviert und deaktiviert. Wurde eine Geste und eine damit verbundene konversationale Funktion erkannt, so wird diese Information an die Kognitionskomponente übermittelt, welche die neue Information in ihr dynamisches Arbeitsgedächtnis aufnimmt und im Rahmen der kognitiven Verarbeitung auf das Ereignis eingeht.

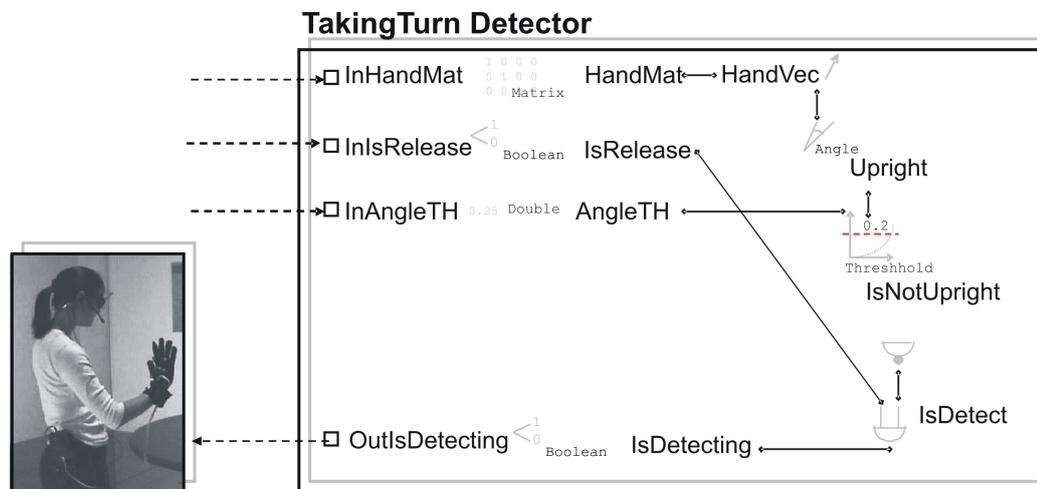
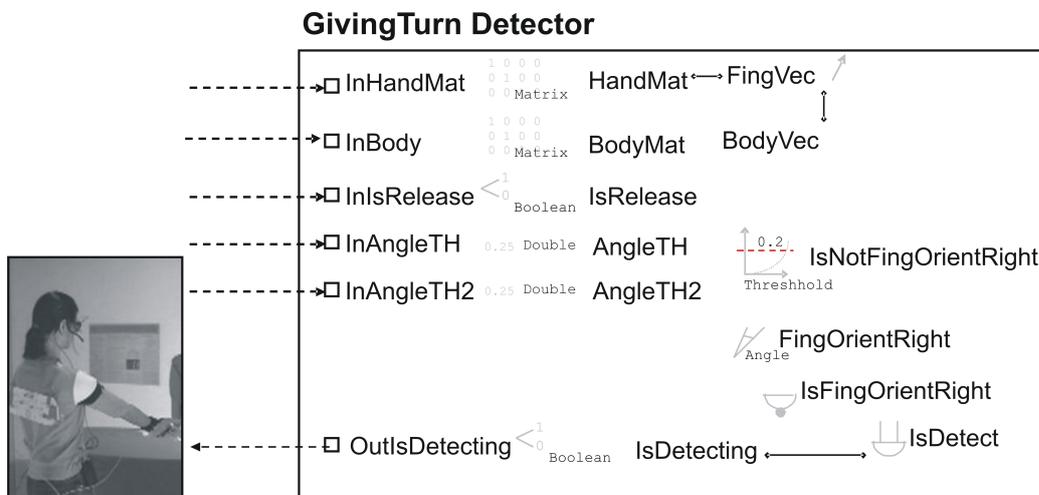


Abbildung 6.9: Taking-Turn-Detektor

### Verarbeitung von Turn-Taking-Signalen

Um dem regelbasierten Charakter der *Turn-Taking*-Mechanismen gerecht zu werden, wird in *cMax* auf unterer Ebene eine Menge an *Conclude*-Plänen eingesetzt, welche aktiviert werden, wenn der Agent in eine Konversation eintritt. Diese Pläne lauern auf die Signale *Taking-Turn*, *Wanting-Turn*, *Giving-Turn* oder *Yielding-Turn*. Sie sind jeweils durch Vorbedingungen so eingeschränkt, dass ein kontextsensitives Verhalten in Abhängigkeit des konversationalen Zustands und der konversationalen Funktion des eingehenden Signals generiert wird.

Zunächst wird ein eingehendes *Turn-Taking*-Signal von einem Plan verarbeitet, der Verhaltensweisen (z.B. Blicke und Gesten) anstößt, die dem Gegenüber vermitteln, dass der Agent den Interlokutor und sein Anliegen wahrgenommen hat (siehe PseudoCode 9). Die Tatsache, dass es sich bei dem Plan um einen *Conclude*-Plan mit sehr hoher Priorität handelt, sorgt dafür, dass der Plan direkt ausgewählt und durchgeführt wird. Dies trägt dem Aspekt Rechnung, dass der Agent die Signale „bewusst“ wahrnimmt, jedoch seine erste Reaktion nicht in einer lange geplanten, wohl überlegten Handlung besteht, sondern vielmehr eine direkte Reaktion auf das Wahrgenommene und Interpretierte darstellt. Der Plan stößt ein Blickverhalten des Agenten mit mittlerer Priorität an, welches sich auf den Interlokutor richtet. Dies Verhalten erfüllt Anforderungen des *gegenseitigen Aufeinandereingehens* (siehe Definition 2). Dann wird durch den *Conclude*-Plan ein neues Ziel für die Verarbeitung des Signals instantiiert (*PERFORM handle\_conversationalFunction \$turn\_signal \$urgency*). Dabei stellt der Ursprung des Ziels kein selbst aufgeworfenes Ziel des Agenten dar, sondern eine *Obligation*,

Abbildung 6.10: *Giving-Turn*-Detektor

sich an konversationale Interaktionsregeln zu halten und auf die Signale des Gegenübers einzugehen.

---

#### Algorithm 9 Conclude-Plan für konversationale Funktionen

---

- 1: **procedure** CONCLUDE CONVFUNCTION \$INTERLOCUTORR \$CONVFKT \$URGENCY  
     ▷ PRECONDITION: FACT conversationWith „self“ \$interlocutor
  - 2:     TRY(lookAt \$interlocutor middle-Utility)
  - 3:     PERFORM handle\_convFunction \$interlocutor \$convFkt \$urgency
  - 4:     PRIORITY: 100
  - 5: **end procedure**
- 

#### Eingehen auf *Turn-Taking*-Signale - Handlungsoptionen

Die Obligation, auf das *Turn-Taking*-Signal angemessen einzugehen, wird durch die Generierung verschiedener Handlungsoptionen bearbeitet (siehe *Means-Ends-Reasoning* Abschnitt 5.2.4). Hierfür kommen die **kontextfreien** Regeln des *Turn-Taking* aus Tabelle 3.1 (siehe Kapitel BCTC 3.3.1) zum Einsatz. Für jedes Ziel, eine konversationale Funktion zu behandeln, existieren entsprechende Pläne, die aufgrund der Regelmenge als zulässige Handlungsoptionen deklariert sind. Die Entscheidung, welcher der Pläne ausgewählt wird, geschieht über den normalen *Means-Ends-Reasoning*-Prozess. Das bedeutet, dass die Entscheidung aufgrund

der *Utility*-Funktionswerte sowie der Aktivierung und der spezifizierten und gewichteten Attribute getroffen wird. Es sind drei Zeitpunkte im Verlaufe der Interaktion vorgesehen, zu denen der Agent entscheiden kann, auf das Signal, z.B. eine Zwischenfrage, einzugehen. Der inkrementelle Entscheidungsprozess ermöglicht schnelle Reaktionszeiten bei gleichzeitiger Berücksichtigung aller relevanten Entscheidungsfaktoren.

Der **erste Entscheidungszeitpunkt** ist unabhängig vom Inhalt der Äußerung des Gegenübers. Zu dem Zeitpunkt, wo die *Turn-Taking*-Funktion detektiert wird, hat der Agent den Inhalt der Äußerung noch nicht interpretiert. Die Entscheidung, sich zu diesem frühen Zeitpunkt unterbrechen zu lassen, ist damit allein abhängig von der *Dominanzbeziehung* zwischen den Konversationspartnern und den Dringlichkeiten, mit welchen die Interlokutoren jeweils ihr Rederecht verfolgen. Die Tabellen Abbildung 6.11 und Abbildung 6.12 zeigen die Entscheidungsgrundlage des Agenten für das Halten (*h-hold*) bzw. das Abgeben (*g-give*) des Turns. Als Entscheidungsgrundlage werden der *Utility*-Wert der eigenen Intention, die Dringlichkeit der Signale des Interlokutors (*Urgency*) sowie die Dominanzbeziehung (*Dominance*) jeweils in die Wertebereiche *low*, *middle* und *high* klassifiziert. Für die Klassifizierung sind verschiedene Funktionen einsetzbar, so kann beispielsweise zu Beginn einer Interaktion der Dominanzwert zwischen Agent und menschlichem Interaktionspartner direkt gesetzt werden, es ist aber auch möglich, diesen dynamisch aus der Interaktion abzuleiten.

Die klassifizierten Werte werden in Funktionen verrechnet (siehe Funktion 6.3 und Funktion 6.4), deren Ergebnisse sich dann auf Prioritätswerte der Handlungsoptionen auswirken und damit das Verhalten des Agenten entsprechend beeinflussen. 0 steht dafür, dass der Agent den Turn abgeben sollte und 1 dafür, den Turn zu behalten.

$$tt\_Fkt(dom, util, urg) = \begin{cases} (low(\$dom) \wedge low(\$urg)), & 1 \\ (mid(\$dom) \wedge low(\$util) \wedge \neg low(\$urg)) & 1 \\ (mid(\$dom) \wedge low(\$urg)) & 1 \\ (mid(\$dom) \wedge (middle(\$urg) \wedge (\neg high(\$urg)))) & 1 \\ (high(\$dom) \wedge \neg(high(\$urg))) & 1 \\ \text{sonst} & 0 \end{cases} \quad (6.3)$$

		urgency low	urgency middle	urgency high
dominance low	utility low	g	g	g
	utility middle	g	g	g
	utility high	h	g	g

		urgency low	urgency middle	urgency high
dominance middle	utility low	h	g	g
	utility middle	h	g	g
	utility high	h	h	g

		urgency low	urgency middle	urgency high
dominance high	utility low	h	h	g
	utility middle	h	h	g
	utility high	h	h	g

Abbildung 6.11: Entscheidungsgrundlage für *TakingTurn*-Signale, (g für *give turn*, h für *hold turn*)

$$wt\_Fkt(dom, util, urg) = \begin{cases} (low(\$dom) \wedge low(\$util)), & 0 \\ (low(\$dom) \wedge mid(\$util) \wedge \neg low(\$util)), & 0 \\ (low(\$dom) \wedge high(\$util) \wedge high(\$urg)), & 0 \\ (mid(\$dom) \wedge low(\$util) \wedge \neg high(\$urg)), & 0 \\ (low(\$dom) \wedge high(\$util)), & 0 \\ \text{sonst} & 1 \end{cases} \quad (6.4)$$

Pseudocode (10, 11, 12) zeigen exemplarisch die Pläne für das Verarbeiten eines *WantingTurn* Signals auf. Hierfür bestehen laut Tabelle 3.1 die Handlungsoptionen: *GivingTurn*, *YieldingTurn* und  *HoldingTurn*. Die *Utility*-Werte der Pläne sind jeweils so gewählt, dass die Entscheidungsfunktion (6.4) sich gemäß der Tabelle auf die *Utility*-Funktion auswirkt.

		urgency low	urgency middle	urgency high
dominance low	utility low	g	g	g
	utility middle	h	g	g
	utility high	h	h	g

		urgency low	urgency middle	urgency high
dominance middle	utility low	h	g	g
	utility middle	h	h	g
	utility high	h	h	h

		urgency low	urgency middle	urgency high
dominance high	utility low	h	h	g
	utility middle	h	h	h
	utility high	h	h	h

Abbildung 6.12: Entscheidungsgrundlage für *WantingTurn*-Signale, (g für *give turn*, h für *hold turn*)

---

**Algorithm 10** Handle conversational Function - Decision Give

---

- 1: **procedure** PERFORM HANDLE\_CONVFKT „WANTINGTURN“ \$URGENCY
    - ▷ PRECONDITION: FACT convState „myTurn“
      - ▷  $timeNow \leftarrow get\_time\_now()$
      - ▷  $current\_goal \leftarrow get\_current\_goal()$
      - ▷  $current\_utility \leftarrow get\_utility(current\_goal)$
      - ▷  $dominance \leftarrow get\_current\_dominance\_value()$
    - ▷  $decisionFactor = \$wtDecisionFKT(\$current\_utility\$dominance\$urgency)$
  - 2: PERFORM givingTurn \$interlocutor ;
  - 3: PRIORITY: \$current\_utility + \$decisionFactor
  - 4: **end procedure**
-

---

**Algorithm 11** Handle conversational function - Decision Yield

---

```

1: procedure PERFORM HANDLE_CONVFKT „WANTINGTURN“ $INTERLO-
   CUTOR $URGENCY
   ▷ PRECONDITION: FACT convState „myTurn“
     ▷ timeNow ← get_time_now()
     ▷ current_goal ← get_current_goal()
     ▷ current_utility ← get_utility(current_goal)
     ▷ dominance ← get_current_dominance_value()
   ▷ decisionFactor = $wtDecisionFKT($current_utility$dominance$urgency)

2:   PERFORM yieldingTurn

3:   ATTRIBUTES: polite, subordinate
4:   PRIORITY: $current_utility + $decisionFactor
5: end procedure

```

---



---

**Algorithm 12** handle conversational function - Decision Hold

---

```

1: procedure PERFORM HANDLE_CONVFKT „WANTINGTURN“ $INTERLO-
   CUTOR $URGENCY
   ▷ PRECONDITION: FACT convState „myTurn“
     ▷ timeNow ← get_time_now()
     ▷ current_goal ← get_current_goal()
     ▷ current_utility ← get_utility(current_goal)
     ▷ dominance ← get_current_dominance_value()
   ▷ decisionFactor = $wtDecisionFKT($current_utility$dominance$urgency)

2:   PERFORM holdingTurn

3:   ATTRIBUTES: dominant
4:   PRIORITY: $current_utility + $decisionFactor
5: end procedure

```

---

Nachdem der eingehende Sprechakt interpretiert wurde, besteht erneut die Möglichkeit, dass der Agent sich unterbrechen lässt. Zu diesem **zweiten Entscheidungspunkt** beruht die Entscheidung allein auf dem bis dahin erkannten Performativ des *Interaction-Moves*. Es greift die folgende Regel:

- Handelt es sich bei dem *Interaction-Move* um eine Frage (*query*) oder eine direkte Anweisung (*request*), so sollte der Agent direkt seine Äußerung unterbrechen, um sich seinem Konversationspartner zuzuwenden.

- Im Falle der Performative *inform* und *propose* fährt der Agent mit seiner Äußerung fort.

Um dieses umzusetzen gibt es einen weiteren *Hold*-Plan (siehe Pseudocode 13), der in seinem *Plan-Body* auf das Ergebnis der *Performativ-Analyse* wartet und entsprechend der spezifizierten Regel darauf eingeht.

---

**Algorithm 13** handle conversational function - Decision Hold
 

---

```

1: procedure PERFORM HANDLE_CONVFKT „WANTINGTURN“ $URGENCY
    ▷ PRECONDITION: FACT convState „myTurn“
        ▷ timeNow ← get_time_now()
        ▷ current_goal ← get_current_goal()
        ▷ current_utility ← get_utility(current_goal)
        ▷ dominance ← get_current_dominance_value()/10
    ▷ decisionFactor = $ttDecisionFkt($current_utility$dominance$urgency)

2:   WAIT: ( $\forall$  (get_time_now() – timeNow ≤ waitTime)
   GOAL (PERFORM $timeNow handleInteractionMove $performative))
3:   if ($performative = „query“) or ($performative = „request“) then
4:     PERFORM givingTurn
5:   end if

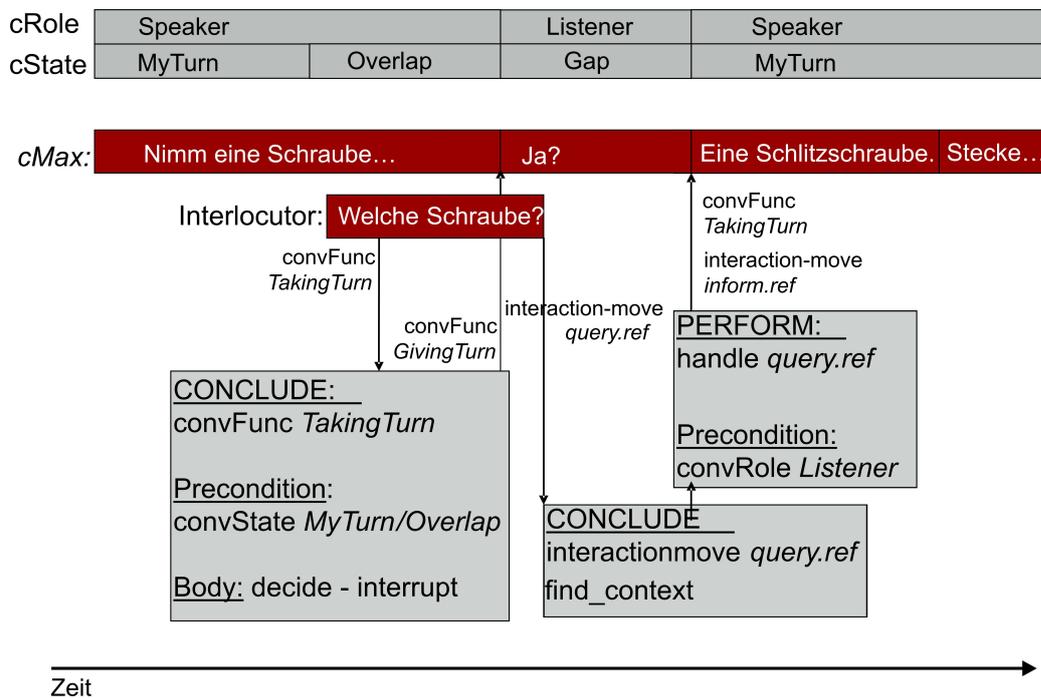
6:   PRIORITY: $current_utility + $decisionFactor
7:   ATTRIBUTES: dominant
8: end procedure
  
```

---

Wurde die Äußerung in den intentionalen Kontext des Agenten eingeordnet, so hängt die Entscheidung, ob der Agent direkt auf den Sprechakt eingeht, zu diesem **dritten Zeitpunkt** von der Relevanz des eingegangenen Sprechakts ab. Führt dieser zu einem offensichtlichen Konflikt mit bestehenden Plänen und Intentionen, so gilt es für den Agenten, direkt darauf einzugehen. Die letzte Möglichkeit des Agenten, auf eine Unterbrechung einzugehen, liegt besteht nach Beendigung seines *Turns*.

Die *handle\_ConvFkt*-Pläne regeln, wie die Übergabe des Turns angemessen zu vollziehen ist, oder wie durch Abwenden des Blicks dem Gegenüber vermitteln wird, dass der Agent nicht gewillt ist, den Turn abzugeben. Im Falle eines *Interaction-Moves*, der ausschließlich aus einem *Turn-Taking*-Signal besteht, ist die Verarbeitung und Entscheidungsfindung an dieser Stelle abgeschlossen.

Abbildung 6.13 zeigt eine Beispielsinteraktion und beschreibt die Verarbeitung des *cMax* in zeitlicher Abfolge. Es werden die relevanten Zustände sowie aktiven Pläne des Agenten präsentiert. Oben in dem Bild wird der Zustand so-

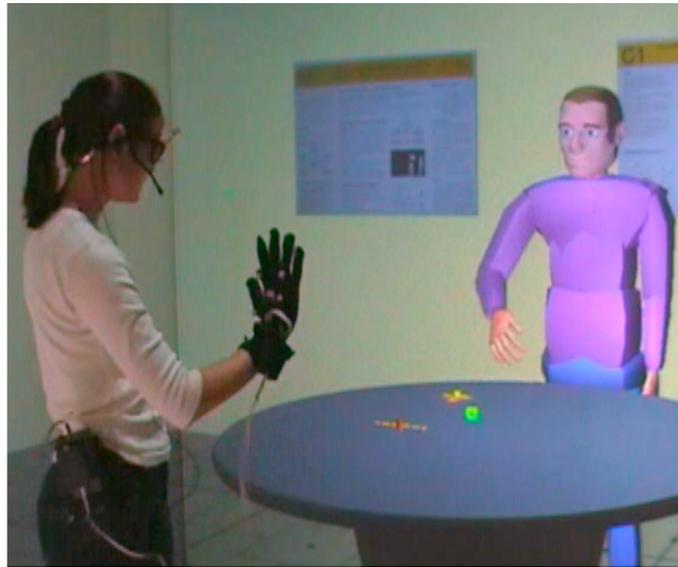
Abbildung 6.13: *Taking-Turn*-Verarbeitung: Zeitlicher Ablauf

wie der Wechsel des konversationalen Zustands und der konversationalen Rollen angezeigt. In den darunter dargestellten Spuren sieht man die Sprechhandlungen der Agenten. Man sieht, wie der Interlokutor *cMax* mit einer Zwischenfrage unterbricht. Dies führt zur Detektion der konversationalen Funktion *TakingTurn*. Sie wird durch einen *Conclude*-Plan verarbeitet. *cMax* entscheidet sich seinen Turn zu unterbrechen und macht dies durch eine *GivingTurn*-Handlung deutlich („Ja?“). Mittlerweile wurde der Inhalt der Zwischenfrage analysiert (*Interaction-Move query.ref*) und ein passender Kontext für die Frage gefunden. *cMax* geht auf den *Interaction-Move* ein, indem er eine Antwort generiert (*InteractionMove inform.ref*) und seinen Interlokutor darüber aufklärt, dass er eine Schlitzschraube meinte. Danach führt *cMax* seinen ursprünglichen Turn fort.

Abbildung 6.14 und 6.15 präsentieren zwei Beispiele, in denen *cMax* einmal auf eine Unterbrechung eingeht und das andere Mal im konversationalen Zustand *Gap* den Turn aushandelt.

### Aktives Ergreifen des Turns

Möchte der Agent eine Äußerung machen, so muss er zunächst die Rolle des Sprechers annehmen. Da in einer sozialen Konversation die Rolle des Sprechers jedoch ausgehandelt wird, berücksichtigt der Agent dafür *konversationale Obli-*



**cMax:** Stecke diese Schraube in die Mitte einer Dreierleiste.

**Interlocutor:** [unterbricht] Halt!

**cMax:** [fokussiert Interlokutor] Ja bitte?

**Interlocutor:** [guckt sich die Bauteile genau an, gibt dann den Turn zurück] Okay.

Abbildung 6.14: Interlokutor unterbricht *cMax*

*gationen* und Regeln. Der Agent modelliert intern den konversationalen Zustand sowie die Rollenverteilung der Interlokutoren während der Interaktion. Die Rollenzuschreibung besteht aus den Werten *uninvolved*, *speaker* und *listener* (siehe Kapitel BCTC 3.3.1). Die konversationale Rolle spiegelt die Überzeugung und Sichtweise des Agenten wider. Sie dient dazu, den *konversationalen Zustand* bestimmen zu können. Die angenommene Rolle des Konversationspartners wird aufgrund seines overtten Verhaltens beurteilt. Spricht der Interlokutor eine längere Äußerung, so wird für ihn die Rolle des Sprechers eingetragen. Die Rolle des Agenten selbst lässt sich jedoch nicht daran ablesen, ob er gerade spricht oder nicht, sondern sie wird anhand seiner internen Einstellung abgeleitet. Diese wird explizit gesetzt:

*ACHIEVE convRole \$self \$role*

Durch diese Modellierung wird es ermöglicht, dass der Agent konversationales Feedback geben kann, ohne sich dafür in der Rolle des Sprechers zu sehen. Die



**cMax:** Jetzt müssen wir die Leiste quer zur Leiste drehen.  
 [Wartet auf den Interlokutor, den Turn zu ergreifen, oder die Bauteile zu drehen]  
**Interlocutor:** [reagiert nicht; Konversationaler Zustand *Gap*]  
**cMax:** [nach einer Weile, *cMax* macht *Giving-Turn* Geste]  
**Interlocutor:** [nimmt den Turn nicht an, macht auch *Giving-Turn* Geste]  
**cMax:** [nimmt den Turn] Okay, dann mach ich das.

Abbildung 6.15: Konversationaler Zustand *Gap*: Aushandeln des Turns

Annahme verschiedener Rollen kodiert damit die Intention des Agenten in Bezug auf den konversationalen Zustand. Der Agent wird in der Regel nur dann die Rolle des Sprechers annehmen, wenn er zuvor durch *Turn-Taking*-Pläne den entsprechenden konversationalen Zustand *MyTurn* herbeigeführt hat. Um die Rolle des Sprechers anzunehmen und den Zustand *MyTurn* zu erreichen, wirft der Agent das Teilziel auf:

*ACHIEVE convState „MyTurn“*

Aufgrund der Eigenschaften eines *Achieve*-Ziels, wird das Ziel nur dann aktiv verfolgt, wenn es noch nicht gilt. Ist der Agent also gerade der Sprecher und verfügt über den Turn, so kann er mit der nächsten Äußerung direkt fortfahren. Besteht der konversationale Zustand jedoch aus *OthersTurn*, *Gap* oder *Overlap*, so verfügt der Agent über verschiedene Handlungsoptionen. Auch diese Handlungsoptionen werden durch kontext-sensitive Pläne mit den entsprechenden Vor- bzw. Kontextbedingungen modelliert.

Im Zustand *no-conversation*, regelt ein Plan, dass der Agent sein Gegenüber fokussiert und mit einer deutlichen *WantingTurn*- bzw. *TakingTurn*-Geste auf sich aufmerksam macht und eine Konversation beginnt. Besteht der Zustand aus *Gap*, kann der Agent den Turn einfach ergreifen. Die Zustände *Overlap* und *Other-sTurn* gestalten sich komplizierter. In der Regel wird der Agent versuchen, durch *WantingTurn*-Signale auf sich und seine Intention, etwas zu sagen, aufmerksam zu machen. Ein Plan regelt, dass der Agent dieses Anliegen zum Ausdruck bringt:

*PERFORM* „self“ *wantingTurnBehavior* \$urgency

Es gibt verschiedene Formen der Signale, die sich in ihrer Ausdrucksstärke unterscheiden. In Abhängigkeit des Dominanzverhältnisses zwischen den Interlokutoren und der Dringlichkeit stoßen die Pläne die Produktion von Gesten an, deren Ausdrucksstärke gesteigert wird, bis der Agent sein Ziel erreicht und den Turn bekommen hat oder der Plan als gescheitert gilt und der Agent über Alternativen nachdenken muss. Abbildungen 6.16 und 6.17 zeigen die Gesten in ihren verschiedenen Ausdrucksformen.

Möchte der Agent bewusst sein Gegenüber unterbrechen und mit großer Dringlichkeit etwas äußern, so findet ein gesonderter *utter-interrupt*-Plan Einsatz. Während der normale *utter*-Plan zunächst das Teilziel aufwirft, den konversationalen Zustand *MyTurn* herzustellen, wird der *utter-interrupt*-Plan unabhängig von der Rollenverteilung in der Interaktion und des konversationalen Zustands ausgeführt.

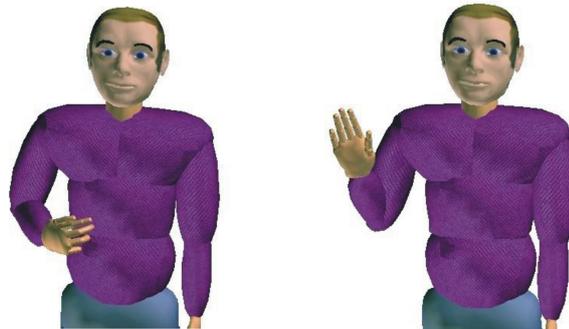


Abbildung 6.16: *cMax* zeigt *WantingTurn* und *TakingTurn* Geste

### 6.2.3 Dialogkompetenz

Situierte Kommunikation im Handlungskontext geht über reine Instruierbarkeit hinaus; sie impliziert Dialogfähigkeiten des künstlichen Systems, das in der La-

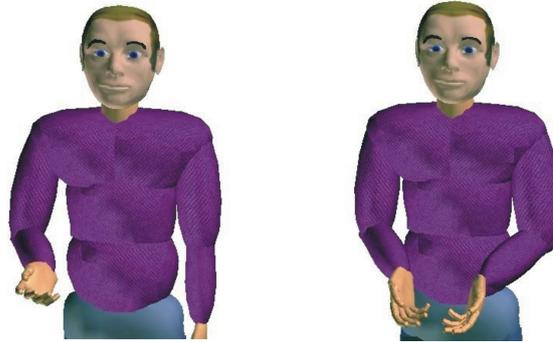


Abbildung 6.17: *cMax* zeigt *GivingTurn* Geste Stufe 1 und 2

ge sein muss, Sprache situationssensitiv zu verstehen und zu generieren. Im vorherigen Abschnitt standen interaktionale Aspekte im Vordergrund und es wurde präsentiert, wie *cMax* auf eine Unterbrechung reagiert. Im Folgenden geht es um die inhaltliche Ebene.

Wie in dem TRAINS-System (Traum, 1996) wird in der CASEC-Architektur von Sprechakten ausgegangen, die dann durch *Obligationen* abgehandelt werden. Im Gegensatz zum TRAINS-System ist der Ansatz in der CASEC-Architektur auf *Interaction-Moves* erweitert, sodass nicht nur Dialogbeiträge berücksichtigt werden, sondern auch manipulative Handlungen.

### Behandlung von Fragen

Am Beispiel der Behandlung einer Zwischenfrage sollen die Dialogkompetenzen des *cMax* vorgestellt werden. Das Bearbeiten einer Frage geschieht zunächst über den Mechanismus der allgemeinen sozialen Dialogregeln, die zur *Obligation* führen, auf eine Frage angemessen zu reagieren. Die Verarbeitung und das Zusammenspiel der Ziele und Pläne im Rahmen des BDI-Interpreters werden an dem Beispiel eines *Interaction-Move* in Form einer referentiellen Frage (*query.ref*) erläutert. Wird der *Interaction-Move* in die *Beliefs* des Agenten eingetragen, so greift als erstes ein *Conclude*-Plan der folgenden Form:

```
CONCLUDE interactionMove „query.ref“ $actor $var1;
```

Dieser Plan instantiiert eine *Obligation*, auf den *Interaction-Move* einzugehen, die folgende Form annimmt:

```
PERFORM handleInteractionMove „query.ref“ $actor $var1;
```

Zur Bearbeitung einer solchen *Obligation* ist im *cMax* ein Plan vorgesehen, der zunächst einen Impuls aussendet, den Sprecher anzuschauen. Dann werden Interpretationsalgorithmen eingesetzt, um den Kontext und die zugrunde liegende Bedeutung des *Interaction-Moves* zu ermitteln. Es wird versucht, die passende Intention in der Intensionsstruktur des Agenten zu ermitteln, auf die sich die Frage bezieht.

Handelt es sich um eine allgemeine Frage, zu der keine kontextrelevante Intention vorliegt, so wird versucht, die Information aus dem Langzeitgedächtnis bzw. der Wahrnehmung abzuleiten. Bei der Suche der passenden Intention müssen die Attribute der Referenz und des potentiell referenzierten Objekts nicht genau übereinstimmen. Es wird ein Korrelationswert berechnet, welcher *Constraints*, Attribute und Typangaben berücksichtigt (siehe CASEC Abschnitt 5.2.7).

Ist der Agent der Adressat der Frage und konnte das Objekt ermittelt werden, so besteht nun die *Obligation* für den Agenten, auf die Frage angemessen zu reagieren. Dazu stehen ihm vier Möglichkeiten zur Auswahl, um die *Obligation* zu erfüllen.

$$\text{handle query.ref} = \begin{cases} \text{inform.ref,} & \text{falls Antwort bekannt} \\ \text{inform.not\_known,} & \text{falls Antwort unbekannt} \\ \text{inform.not\_understood,} & \text{falls Frage unverständlich} \\ \text{inform.refuse,} & \text{falls Verweigern der Aussage} \end{cases} \quad (6.5)$$

Der Agent kann eine direkte Antwort geben, welche dann kontextsensitiv generiert wird. Anderenfalls kann er sein Gegenüber informieren, dass er die Antwort nicht kennt (er konnte den Kontext sowie die Variable ermitteln, weiss aber selber keinen passenden Wert für die Variable) oder die Anfrage nicht verstanden hat (er konnte die Referenz nicht auflösen und keinen Kandidaten ermitteln). Des Weiteren kann er auch die Aussage verweigern.

Das Abarbeiten der *Obligation* geschieht über eine OR-Verschachtelung, welche die vier Optionen umfasst. Die Entscheidung wird über den normalen *Means-Ends-Reasoning*-Prozess vorgenommen, wobei die einzelnen Optionen durch Vorbedingungen so eingeschränkt sind, dass sie die Entscheidung des Agenten lenken. So kann er beispielsweise nur dann eine konkrete Antwort generieren, wenn er die Referenz auflösen konnte.

### Äußerungsgenerierung

Um für die Dialogfähigkeit des Agenten Sprache situationssensitiv zu generieren, wird ein Äußerungsgenerator (*Utterance-Generator*) eingesetzt. Dieser ermöglicht es, aufbauend auf einem Lexikon und dem Gedächtnis des Agenten,

multimodale Äußerungen zu generieren (Kopp & Wachsmuth, 2000). Er geht immer von einem kommunikativen Ziel sowie dem vorhandenen Wissen über den Partner und den bisherigen Diskurs aus. Das kommunikative Ziel leitet sich direkt aus den Intentionen des Agenten ab und führt zur Auswahl eines geeigneten Aktionstyps (*Performativ*). Die folgenden Schritte entsprechen dem Aufbau nach den verbreiteten Ansätzen zur Sprachgenerierung (Reiter & Dale, 2000):

**(Inhaltsplanung)** Auswahl der zur übermittelnden Information,

**(Diskursplanung)** Strukturierung der Information und Bestimmung der Anknüpfung zum vorherigen Diskurs,

**(Satzplanung und Realisierung)** Wahl geeigneter Wörter, Sätze oder Gesten zur Realisierung einer multimodalen Äußerung.

Der Ablauf der Äußerungsplanung in *cMax* ist in Abbildung 6.18 dargestellt. In der Äußerungsgenerierung von *cMax* betrifft die Inhaltsplanung sowohl die Konkretisierung des Performativs als auch die Ableitung der notwendig zu kommunizierenden Fakten aus den zugehörigen *Constraints* (siehe oben). Dieser Schritt ist nicht nur durch das kommunikative Ziel bestimmt, sondern kann auch durch den emotionalen Zustand des Agenten sowie die soziale Beziehung zum Konversationspartner beeinflusst werden. Satzplanung und Realisierung betreffen dann die Bestimmung konkreter verbaler und coverbaler Aktionen, die den illokutionären Akt und den propositionalen Gehalt möglichst erfolgsversprechend und effizient für den Adressaten deutlich machen. Klassische Ansätze der Sprachgenerierung bauen auf komplexen lexikalischen und grammatikalischen Formalismen auf. Der Ablauf der Äußerungsplanung in *cMax* ist in Abbildung 6.18 dargestellt und geht von einem kommunikativen Ziel aus, das sich direkt aus der aktuellen Intentionsstruktur des Agenten ergibt (Inhaltsplanung) (**Content Selection**). Die folgenden Schritte werden von einem speziellen Planer vorgenommen, der sich in seiner Vorgehensweise an *systemischen Grammatiken* (Halliday, 1985) orientiert. Er wendet verschiedene Entscheidungsmodule an, die den passenden Dialogakttyp auswählen und einzelne Konzepte gemäß ihrer semantischen Rolle schrittweise expandieren. Er operiert auf einer Repräsentation, die sich aus einem zentralen *Template*-Knoten für die Aktion (das Performativ) sowie weiteren Knoten (*MessageNodes*) für die involvierten Konzepte zusammensetzt, aus denen schlussendlich mögliche Konstituenten des Satzes (z.B. Subjekt, Prädikat, Objekt, Attribut) werden können.

In der Satzplanung (**Sentence Planning**) werden die *MessageNodes* mit den abstrakten inhaltlichen Konzepten gefüllt, die sie repräsentieren sollen und die durch Merkmalswissen in Form von Attribut-Wert-Listen angereichert sind. Diese Knoten dienen als Container für Realisierungsanweisungen (*realization state*-

ments), die die lexiko-grammatikalischen Konsequenzen einzelner Generierungsentscheidungen repräsentieren und die syntagmatische Struktur der Äußerung Schritt für Schritt einschränken. Solche Anweisungen können sich beispielsweise auf das Einfügen von Konstituenten (+Verb, +Subjekt), deren Reihenfolge (*Subjekt* > *Verb*) oder die Realisierung von grammatikalischen Beugungen beziehen. Dadurch können Subjekt-Verb-Kongruenz oder Attribut-Nomen-Kongruenz realisiert werden, auch kann der Typ eines Objektes die Auswahl des lexikalischen Eintrags des mit dem Objekt assoziierten Verbs beeinflussen.

Während der Satzplanung baut der Agent multimodale, referierende Ausdrücke aus sprachlichen und gestischen Anteilen auf. Insbesondere deiktische Gesten, mit denen direkt auf Objekte in der Umwelt gezeigt werden kann, sowie ikonische Gesten, die räumlich-visuelle Informationen darstellen können, spielen hierfür eine entscheidende Rolle. *cMax* kann Gesten explizit durch die Formeigenschaften der bedeutungstragenden Phase definieren (wie z.B. Handform oder Bewegung) oder indirekt durch die Angabe einer kommunikativen Funktion spezifizieren, die es zu erfüllen gilt (Kopp & Wachsmuth, 2004).

In Abbildung 6.18 wird im *MessageNode* der gelben, definit bestimmten Schraube (*SCHRAUBE-23*) eine zusätzliche deiktische Geste vorgesehen. Als letzten Unterschritt der Satzplanung leitet *cMax* die Funktion der Äußerung im Diskurs sowie das Diskurssegment, zu dem sie beiträgt, aus der Struktur seiner aktuellen Intentionen ab. Ist die Äußerung Teil einer bereits vorgeplanten längeren Sequenz, so fügt er passende Konnektoren ein, wie zum Beispiel „als erstes“ zu Beginn einer Sequenz „und dann“, „als nächstes“ für Zwischenschritte und „als letztes“ für den Abschluss der Sequenz.

In der Realisierungsphase (**Realization**) wird für jeden Knoten ein sprachlicher Ausdruck aus einem Lexikon ausgewählt und entsprechend der jeweiligen lexikalisch grammatikalischen Informationen nach den Realisierungsanweisungen angepasst. Für multimodale Äußerungen mit sprachbegleitender Gestik muss sichergestellt werden, dass die Gesten zeitlich synchron zu denjenigen verbalen Elementen auftreten, zu denen sie semantisch und pragmatisch coexpressiv sind (McNeill, 1992). Der Realisierer leitet diese Information aus dem multimodalen *MessageNode* ab und fügt entsprechende Zeitmarken in die sprachliche Äußerung ein, die dann als Ankerpunkte für den Gestenverlauf dienen. So geplante Äußerungen werden in **MURML**, einer Auszeichnungssprache auf XML-Basis (Kopp & Wachsmuth, 2004), formuliert und dann von der Physis des Agenten in synthetische Sprache und simultane Gesten- bzw. Gesichtsanimation umgesetzt.

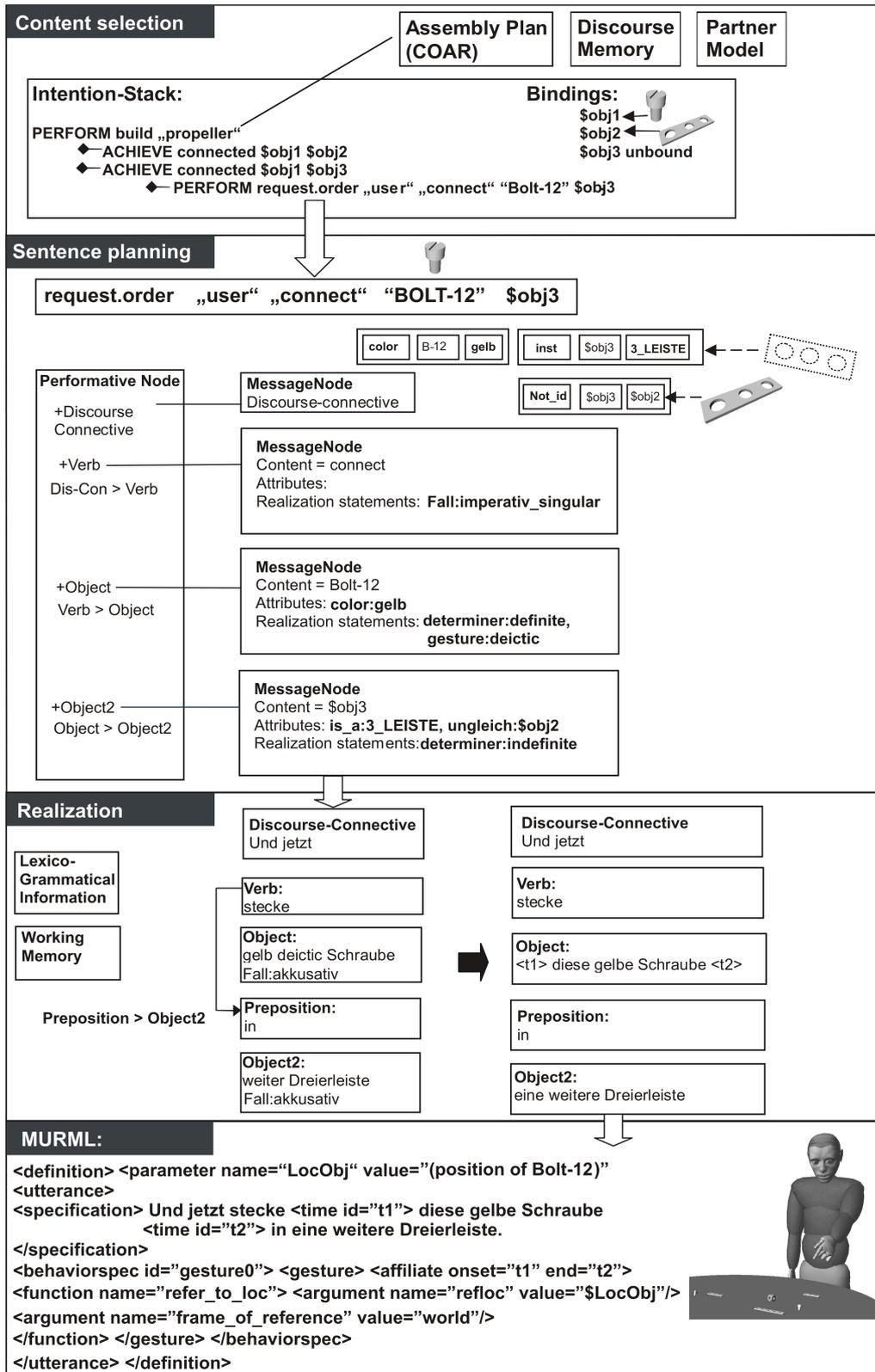


Abbildung 6.18: Äußerungsgenerator (Leßmann et al., 2006, S.323)

## 6.3 *Task Layer*

### 6.3.1 Einsatz externer Planer

Die CASEC-Architektur unterstützt den Einsatz externer Planer, welche über prozedurale Pläne eingebunden werden. Dies ermöglicht es, angepasst an die jeweilige Aufgabe verschiedene Heuristiken und Planer einzusetzen und dennoch von der Grundstruktur der CASEC-Architektur zu profitieren.

Im Rahmen des SFB360-Szenarios greift *cMax* auf konzeptuelles Wissen der Baufixdomäne zurück und nutzt einen *Assembly-Planer* zur Aufstellung eines abstrakten Konstruktionsplans. Dieser greift auf Domänenwissen zurück, welches mittels der *Frame*-basierten Repräsentationsprache *Concepts for Objects, Assemblies, and Roles (COAR)* (Jung & Wachsmuth, 1998) dargestellt wird. In dieser aktiven, dynamischen Wissensbasis wird spezifiziert, welche Baufixteile in einem bestimmten Aggregat enthalten sind und welche Bedingungen bzw. Einschränkungen hierbei gelten. Auch die Rollen der einzelnen Baufixteile werden angegeben.

Die Pläne des *Assembly-Planers* können als verschachtelte Sammlung von Zielen in das Arbeitsgedächtnis des Agenten überführt werden. Damit wird sich der Agent der einzelnen Planungsschritte „bewusst“. Das deklarative Wissen über die nötigen Planschritte wird je nach Aufgabenstellung in konkrete *Perform/Achieve*-Pläne überführt und in den *Deliberations*- sowie *Means-Ends-Reasoning*-Prozess des Agenten einbezogen. Durch die deklarative Wissensstruktur des Plans wird es dem Agenten ermöglicht, auch Pläne aufzustellen und darüber zu kommunizieren, ohne dass zwangsläufig eine konkrete Absicht vorliegen muss, den Plan auch auszuführen. Dies ermöglicht eine Verschachtelung von Kommunikation, Planen und manipulativen Handlungen. *cMax* kann die deklarative Wissensstruktur auch einsetzen um als Tutor aufzutreten (Kopp et al., 2003).

### 6.3.2 Dynamisch situiertes Problemlösen

Unter dynamisch situiertem Problemlösen wird der Prozess des Problemlösens unter Einsatz partieller Pläne verstanden. Um die Handlungsplanung dynamisch an die Konstruktion des entstehenden Aggregats anpassen zu können, werden einmal gefasste Pläne mit Hilfe des Kategorienwissens so allgemein wie möglich formuliert. Wenn *cMax* einen Plan für den Bau eines Propellers entwirft, so gibt dieser Plan allgemein die Art der dafür zu verwendenden Bauteile an, legt sich aber nicht auf konkrete Objekte im Situationskontext fest. Dadurch ist der Agent in der Lage, sich auf die Absichten und Aktionen seines Interaktionspartners einzustellen. Eine *Constraint*-basierte Repräsentation erlaubt es, möglichst viele Optionen

offen zu lassen und macht gleichzeitig die geltenden und ausgehandelten Einschränkungen explizit. Diese *Constraints* gelten in dem gesamten Handlungskontext, werden an Teilpläne vererbt und können versprachlicht werden. Beziehungen zwischen einzelnen Planungsschritten (ob zwei Teilziele beide erreicht werden müssen oder nur eines von beiden (AND, OR Beziehung), sowie zeitliche Beziehungen (ein Ziel muss vor einem bestimmten Teilziel ausgeführt werden) können durch typisierte Zielverknüpfungen realisiert werden (siehe Abschnitt 5.2.1).

Die *Constraint*-basierte Repräsentation partieller Pläne ermöglicht das einfache Zusammenführen und Verknüpfen von Plänen und Teilplänen (*meshing of plans*, siehe Abschnitt 2.1.1). Die *Constraints* können Objekte, Aktionen oder Akteure einschränken. Hierbei können die *Constraints* auch verschachtelt aufgestellt werden und so spezifische Eigenschaften von Objekten zueinander in Beziehung setzen.

**Beispiel** Abbildung 6.21 zeigt den Ablauf einer Konstruktionsinteraktion (siehe auch (Leßmann et al., 2006)). In Tabelle 6.1 werden exemplarisch die ersten *Interaction-Moves* der Interaktion vorgestellt. Nachdem sich die beiden Interaktanten darauf geeinigt haben einen Propeller zu bauen (*Interaction-Move (1) - (3)*), erstellt *cMax* mit Hilfe des *Assembly-Planners* einen Konstruktionsplan. In den *Interaction-Moves (3) und (5)* beschreibt *cMax* den ersten erforderlichen Konstruktionschritt. Abbildung 6.19 zeigt die Zielrepräsentation des unterspezifizierten ersten Konstruktionschrittes. Eine beliebige Schraube (\$obj1) soll in das mittlere Loch (\$port2) einer nicht weiter spezifizierten Leiste (\$obj2) gesteckt werden. Wenn der menschliche Interaktionspartner eine Aktion ausführt, welche mit der

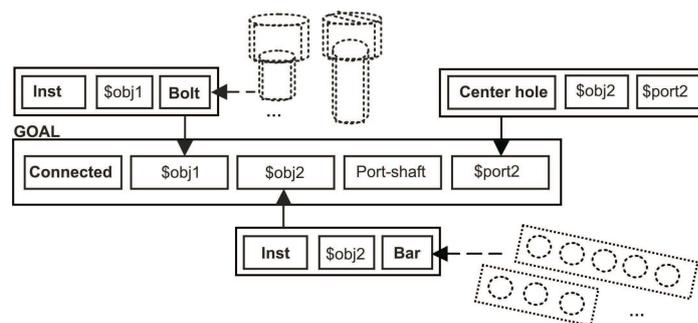


Abbildung 6.19: Einschränkungsrepräsentation eines Konstruktionschrittes (Leßmann et al., 2006, S.320)

Zielrepräsentation nicht kompatibel ist, nämlich die Schraube in das falsche Loch steckt (*Interaction-Move (7)*), ist der Agent in der Lage, den verletzten *Constraint* zu erkennen und dies zu verbalisieren (Abbildung 6.21 *Interaction-Move*

(8)). Eine Konkretisierung der Zielrepräsentation, die auftritt, nachdem der erste Konstruktionsschritt erfolgreich ausgeführt wurde und eine gelbe Schraube mit der Mitte einer Dreilochleiste verbunden wurde (*Interaction-Move (11)*), zeigt die Variablenbindungen in Abbildung 6.20. In *Interaction-Move (12)* verbalisiert der Agent den nächsten Konstruktionsschritt. Die Zielrepräsentation, die einige Variablenbindungen beinhaltet, ist in Abbildung 6.20 aufgezeigt. Da die erste Objekt-

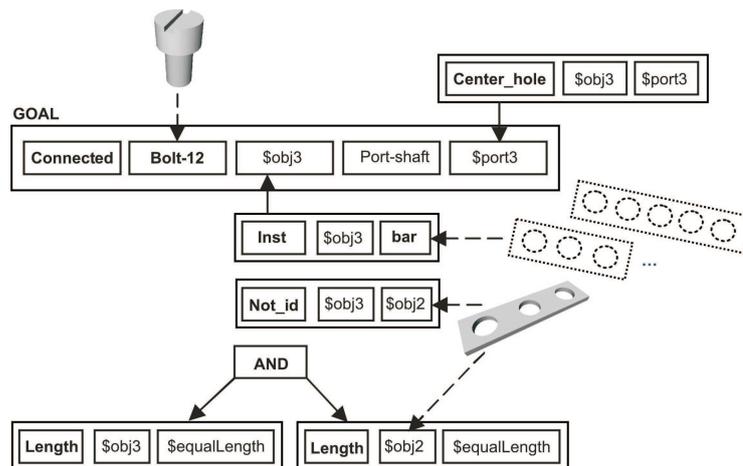


Abbildung 6.20: Einschränkungsrepräsentation eines komplexen Konstruktionsschritts (Leßmann et al., 2006, S.321)

variable ( $\$obj1$ ) an einen Wert gebunden ist, in diesem Fall an eine spezifische gelbe Schlitzschraube (Bolt-12), benutzt der Agent eine Zeigegeste, um auf sie zu referenzieren. Das zweite Objekt ( $\$obj3$ ) besitzt die Einschränkung, zwar eine Leiste zu sein, aber nicht identisch mit der Leiste zu sein, die zuvor benutzt wurde (der an die Variable  $\$obj2$  gebundene Wert). Diesen Aspekt betont *cMax*, indem er sagt, dass es sich um eine *andere* Leiste handeln muss. Zusätzlich werden Einschränkungen spezifiziert, die aus dem Langzeitwissen des Agenten stammen und festlegen, dass die Längen der Rotorblätter gleich lang sein müssen, um einen gültigen Propeller zu bauen. Dies führt zu den zusätzlichen *Constraints*, dass die Länge der Leiste  $\$obj3$  mit derjenigen der Leiste  $\$obj2$  übereinstimmen muss. Das komplexe Netz von *Constraints* zeigt den Einsatz logischer Verknüpfungen und negierter *Constraints*. Die *Constraint*-Repräsentation ist ausdrucksstark und ermöglicht die Repräsentation komplexer Ziele, die noch soviel Spielraum wie möglich lassen. Wenn dem Agenten aufgetragen wird, die Assemblierung selbst vorzunehmen (Abbildung 6.21 *Interaction-Move (13)*), ist er in der Lage, die passende Leiste als zweites Rotorblatt zu wählen und die gelbe Schraube in die Mitte der Leiste zu stecken.

*cMax* nimmt das Montagegeschehen in der Szene wahr und aktualisiert schritt haltend eine COAR-Beschreibung, die Informationen über die Objekte und deren eingegangenen Verbindungen enthält. Durch die Erkennung von Baugruppen (*Aggregatkonzeptualisierung*) werden in dieser Beschreibung unstrukturierte Aggregate als sinnvolle Teilbaugruppen erkannt (z.B. als Fahrwerk) (Jung & Wachsmuth, 1998). Durch die Zuschreibung von Rollen entsprechend ihrer Verwendung als Komponenten werden bestimmte Baugruppen zudem mit kontextabhängigen Merkmalsätzen ausgestattet (z.B. wird eine Schraube zu einer Achse). Das Konzeptwissen über Aggregate und die Rollen einzelner Komponenten in der Langzeitwissensbasis ist über assoziative Verknüpfungen an die zentrale Deliberation angebunden. Wenn der Agent darüber nachdenkt, welches Objekt sich für den Bau eines Aggregats eignet, wird so relevantes Langzeitwissen aktiv und für weitere Inferenzen verfügbar.

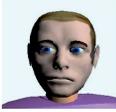
		1 Interlokutor: <i>Lass uns einen Propeller bauen.</i>
		2 cMax: <i>Okay.</i>
		3 cMax: <i>Lass uns eine Schraube...</i>
		4 Interlokutor: <i>Welche Schraube?</i>
		5 cMax: <i>... in das mittlere Loch einer Leiste stecken.</i>
		6 cMax: <i>Eine Schlitzschraube.</i>
		7 Interlokutor [steckt die Schraube in das erste Loch einer Dreierleiste]
		8 cMax: <i>Nein. Das war das falsche Loch.</i>
		9 cMax [macht die Handlung rückgängig]
		10 cMax: <i>Nochmal eine Schraube muss mit dem mittleren Loch einer Leiste verbunden werden.</i>
		11 Interlokutor [steckt die gelbe Schraube in die Mitte einer Dreierleiste]
		12 cMax: <i>Gut. Jetzt stecke diese &lt;Zeigegeste&gt; Schraube in eine andere Dreierleiste.</i>
		13 Interlokutor: <i>Mach Du das.</i>
		14 cMax: <i>Okay, dann mach ich das.</i>
		15 cMax [verbindet die Teile]
		16 cMax: <i>Und jetzt muss diese Leiste quer zu der Leiste gedreht werden.</i>
		17 [Eine lange Pause tritt auf. Der Interlokutor reagiert nicht.]
		18 cMax: <i>Okay, dann übernehme ich das.</i>
		19 cMax [dreht die Bauteile]
		20 cMax: <i>Super. Wir haben so eben einen Propeller richtig zusammen gebaut.</i>

Abbildung 6.21: *cMax* baut zusammen mit seinem Interlokutor einen Baufixpropeller. Die Bilder links zeigen den künstlichen Agenten, seinen Interaktionspartner und die Bauteile in verschiedenen Momentaufnahmen der Interaktion. (nach (Leßmann et al., 2006, S.298)

<b>IM</b>	<b>Interaction Move</b>	Lass uns einen Propeller bauen.	Okay.	Lass uns eine Schraube in das mittlere Loch einer Leiste stecken.
<b>AC</b>	<b>Action</b>	propose.action	inform.agree	request.order
<b>GO</b>	<b>Goal</b>	(ACHIEVE exists prop)	(PERFORM inform. agree)	(ACHIEVE connected \$s \$b \$p1 \$p2)
<b>CO</b>	<b>Content</b>	(Build prop team) (member team self) (member team interlocutor)	(Build prop team)	(Connect \$s \$b \$p1 \$p2) (inst \$s bolt) (inst \$b bar) (center_hole \$b \$p2)
<b>SF</b>	<b>Surface form</b>	< words > <sub>t</sub>	< words > <sub>t</sub>	< words > <sub>t</sub>
<b>TT</b>	<b>Turn-Taking</b>	take   give	take   keep	give
<b>DF</b>	<b>Discourse function</b>	startsegment (DSP=prop)	contribute	startsegment (DSP=props1)
<b>AG</b>	<b>Agent</b>	Human	cMax	cMax
<b>AD</b>	<b>Adressee</b>	cMax	Human	Human

<b>IM</b>	Welche Schraube?	Eine Schlitzschraube	[steckt gelbe Schraube in das erste Loch einer Dreierleiste]	Nein. Das war das falsche Loch.
<b>AC</b>	query.ref	inform.ref	connect	inform.disagree
<b>GO</b>	(PERFORM query.ref \$s)	(PERFORM inform.ref \$s)	(ACHIEVE connected bolt-2 bar-1 port-1 port-3)	(PERFORM (inform.disagree (Connect...))) \$s \$b \$p1 \$p2
<b>CO</b>	(inst \$s bolt)	(inst \$s bolt)	(Connect bolt-2 bar-1 port-1 port-3)	(Not (Center-hole bar-1 port-3))
<b>SF</b>	< words > <sub>t</sub>	< words > <sub>t</sub>	< manipulation > <sub>t</sub>	< words > <sub>t</sub>
<b>TT</b>	take   give	take   give	take   yield	take   yield
<b>DF</b>	contribute	contribute	contribute	contribute
<b>AG</b>	Human	cMax	Human	cMax
<b>AD</b>	cMax	Human	-	Human

Tabelle 6.1: Exemplarische Darstellung der *Interaction-Moves* des Interaktionsbeispiels aus Abbildung 6.21, (DSP = *Discourse Segment Purpose*, prop = *proposal*, inst = *instance*)

## 6.4 Collaboration Layer

Die allgemeinen Prinzipien, welchen der Kooperationsansatz des *cMax* folgt, sind die folgenden:

- Der *SharedPlans*-Ansatz von B. Grosz & Hunsberger (2006) (siehe Abschnitt 2.1.3, (SP1)-(SP5)) dient als Ausgangspunkt für die Modellierung gemeinsamer Ziele und Pläne.
- Gegenseitiges Aufeinandereingehen wird auf allen Ebenen berücksichtigt
- Die Kooperationsmechanismen werden direkt in CASEC integriert, sodass allgemeine Basismechanismen für Kooperationsmechanismen nutzbar gemacht werden können (Aktivierungsmechanismen, Zielinteraktion, Obligationen, explizite Modellierung von Intentionen).

### 6.4.1 Gemeinsame Pläne - (*SharedPlans*)

#### (SP1) Individuelle Intention, dass ein Team *a* ausführt

In der CASEC-Architektur wird der Ansatz verfolgt, gemeinsame Intentionen auf eigene Intentionen zurückzuführen. Dies steht in Einklang mit der ersten Bedingung (**SP 1**) des *SharedPlans*-Ansatzes (siehe Definition 1, Abschnitt 2.1.3) und dem Modell Bratmans (siehe Abschnitt 2.1.1).



Abbildung 6.22: Individuelles Ziel, ein gemeinsames Ziel zu erreichen

Abbildung 6.22 zeigt die Repräsentation eines individuellen Ziels, ein gemeinsames Ziel zu erreichen. Das *Shared-Flag* markiert das Ziel als Instanz eines gemeinsamen Ziels. Das **Team-Feld** beinhaltet eine Liste der Mitglieder des Teams, die sich dem gemeinsamen Ziel verpflichtet haben. Dabei lassen sich die Mitglieder des Teams aus der Spezifikation der *\$actor*-Variablen ablesen. Ein Team kann aus zwei oder mehr Mitgliedern bestehen.

Die Repräsentation der Handlung besteht aus einer **holistischen Repräsentation** (siehe Abschnitt *Gemeinsame Intentionalität* 2.1.4), die ein Feld *\$actor* enthält, das vermerkt, welcher Agent für die Ausführung des Plans verantwortlich

ist. Diese Repräsentation ermöglicht es dem Agenten, sich in die Rolle seines Interaktionspartners hinein versetzen zu können. So kann der Agent leichter Handlungen des Kooperationspartners übernehmen oder bessere Hilfestellung leisten. Durch die explizite Repräsentation der *SharedPlans* wird der Agent in die Lage versetzt, Auskünfte über gemeinsame Ziele zu erteilen. Die Markierung vererbt sich an die Teilziele, die im Dienste des gemeinsamen Ziels aufgeworfen werden und zudem *critical actions* (siehe Abschnitt CASEC 5.2.8) darstellen.

### **(SP 2) Gemeinsamer Belief eines (eventuell unvollständigen) Rezepts für $a$**

Die zweite Forderung der *SharedPlans* (**SP 2**) (siehe Definition 1, Abschnitt 2.1.3) bezieht sich auf das gemeinsame Wissen eines Rezepts. Gemeinsames Wissen entsteht inkrementell durch das Diskutieren eines möglichen Rezepts. Dafür müssen die Agenten in ihren Planbibliotheken ein passendes Rezept suchen und dieses mit ihrem Kooperationspartner abstimmen. Dies geschieht durch die regulären *Means-Ends-Reasoning*-Mechanismen, welche *Obligationen* berücksichtigen, die aufgrund der *CCR* entstehen.

### **(SP 3) Individuelle oder Gruppenpläne für die Aktionen eines Rezepts**

Die Basispläne (*elaborate\_plan\_step \$plan*), (*elaborate \$actor*), (*elaborate \$parameter*) regeln die Bestimmung potentieller Objektkandidaten für die Variablen des Plans. Sie greifen auf die *Unify/Match*-Mechanismen der CASEC-Architektur zurück (siehe Abschnitt 5.1.7). In diesem Rahmen können Variablen an konkrete Werte gebunden sowie Einschränkungen über bestimmte Aspekte des Plans spezifiziert werden. Diese Informationen werden in einem *Mutual*-Feld über gemeinsames Wissen lokal im Kontext der Intention vermerkt.

Die Wissensstruktur ist kontextsensitiv und lokal ausgelegt. Im Gegensatz zu einer allgemeinen *Belief*-Menge, in der sämtliche Informationen des Agenten vorgehalten werden, hat sie den entscheidenden Vorteil, dass die Informationen gemeinsam mit der Intention vorgehalten werden. So können Wissen und Entscheidungen lokal vererbt und propagiert werden.

Das *identified*-Feld enthält die Werte, die *cMax* für die Variablen eines Plans vorsieht. Es ermöglicht, Entscheidungen des Agenten im Kontext seiner Intention zu vermerken, bevor er diese mit seinem Kooperationspartner abgestimmt hat. Dadurch wird der Entscheidungsprozess des Agenten explizit modelliert (siehe Abschnitt 2.2.4). Das *identified*-Feld dient zudem dazu, eine Ausgangsbasis für Deliberationsprozesse zu schaffen, bevor sich die Agenten auf einen Wert geeinigt haben. Wurde ein Planschritt mit allen beteiligten Kooperationspartnern abgestimmt, so wird er als *mutual* markiert.

Abbildung 6.23 zeigt, wie durch die *Interaction-Moves propose.ref* und *inform.agree* gemeinsames Wissen (*mutual belief*) über einen auszuführenden Konstruktionsschritt erreicht wird. Zunächst wird in der Repräsentation des gemeinsamen Ziels spezifiziert, dass jemand (*\$actor* ist noch nicht spezifiziert) die Schraube *\$s1* mit einer Leiste verbinden sollte. Dabei hat der Agent die Schraube *s-23* für die Rolle der *\$s1* als *identified* in seinem Plan vermerkt. Damit ein gemeinsamer Belief über der Einsatz entsteht, setzt der Agent einen *Interaction-Move* mit dem Performativ *propose* ein und schlägt vor, die Schraube *s-23* zu nehmen. Nachdem der Interlokutor dem Vorschlag zustimmt (*inform.agree*), wird der Einsatz der Schraube *s-23* als gemeinsamer Belief (*mutual*) in der Zielrepräsentation vermerkt.

Durch das Protokollieren gemeinsamer Überzeugungen und Ziele wird die Form eines *psychologischen Commonground* erreicht (siehe 2.1.4).

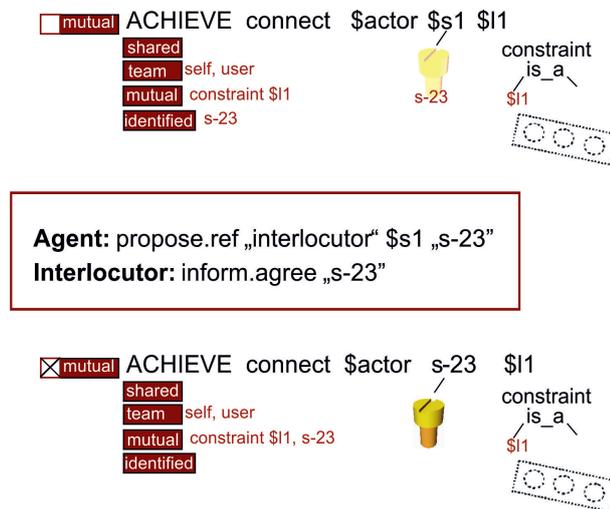


Abbildung 6.23: Entstehung gemeinsamer *Beliefs*

#### (SP4) Intentionen, erfolgreich zu sein

Die Bedingung (SP4) der *SharedPlans* über die Existenz von Intentionen, dass die ausgewählten Agenten oder Untergruppen erfolgreich die ihnen zugewiesenen Aktionen ausführen, wird durch das normale *Commitment*-Verhalten des *cMax* umgesetzt. Im Falle der gemeinsamen Ziele bzw. Pläne wird durch den *Achieve*-Zieltyp erreicht, dass *cMax* überwacht, wann sein Gegenüber das gemeinsame Ziel erreicht hat.

Der **Zieltyp** von Fremdhandlungen besteht grundsätzlich aus *Achieve*-Zielen. Hierbei besteht das Ziel darin, einen Zustand zu erreichen, in welchem der In-

teraktionspartner die Handlung auf eine spezifizierte Art und Weise ausgeführt hat. Die Deklaration des Ziels als *Achieve*-Ziel sorgt dafür, dass der Agent immer wieder überprüft, ob das Ziel schon erreicht ist. Der Agent selbst kann das Ziel nicht allein erreichen und ist somit auf die Handlung seines Kooperationspartners angewiesen. In der Architektur sind verschiedene Handlungspläne in der Planbibliothek vorgesehen, die dazu dienen, einen Kooperationspartner zur Ausführung einer Handlung zu bewegen.

- Der einfachste Plan besteht darin, zu **warten**. Der Plan verfügt über einen sehr niedrigen *Effort*-Wert. Führt der Interaktionspartner jedoch keine Handlung aus, so scheitert der Plan. Nach und nach (in Abhängigkeit der Aktivierungswerte der anderen Pläne, der Attribute, der Dominanz, möglicher Zeitbeschränkungen, *Saliency*-Wert, *Desirability*-Wert) verliert der Plan an Attraktivität, und ein anderer Plan wird dem Agenten als beste Handlungsoption erscheinen.
- Eine weitere Option besteht darin, eine Aufgabe an einen Interaktionspartner zu delegieren und einen **Anweisungsplan** (*request*) zu instantiieren. Dieser wirft das Teilziel *PERFORM request \$interlocutor \$task* auf, welches eine Anweisung durch den Äußerungsgenerator (siehe Abschnitt 6.2.3) erzeugt.
- Eine weitere Möglichkeit besteht darin, das gemeinsame Ziel genauer zu spezifizieren und **Handlungsoptionen zu diskutieren**, um den Lösungsprozess voranzutreiben. Dies kann durch informative Äußerungen geschehen, in welchen der Agent zusätzliche Informationen spezifiziert oder durch *Propose*-Pläne, durch welche der Agent seinem Gegenüber Vorschläge dazu unterbreitet, auf welche Weise das gemeinsame Ziel erreicht werden könnte.

Abbildung 6.24 beschreibt exemplarisch die verschiedenen Handlungsoptionen des Agenten in Abhängigkeit davon, an welchen Wert die \$actor-Variable gebunden ist.

#### (SP5) Intention, den Plan zu vervollständigen

Die Forderung der Bedingung **(SP5)** besteht im Falle eines unvollständigen *SharedPlan* aus individuellen Intentionen, dass die Gruppe den Plan vollendet. Möchte der Agent nur einen konkreten Plan für ein (gemeinsames) Ziel aufstellen, so wirft er dafür explizit ein Ziel auf:

*ACHIEVE shared\_plan \$task \$plan* bzw. *ACHIEVE plan \$task \$plan*

```

ACHIEVE connect "self" $obj1 $obj2
$sactor = "self"
PEFORM connect $obj1 $obj2
PEFORM inform $adresse "connect" "self" $obj1 $obj2

ACHIEVE connect "interlocutor" $obj1 $obj2
$sactor = "interlocutor"
PEFORM request.order $adresse "connect" $obj1 $obj2
PEFORM request.beg $adresse "connect" $obj1 $obj2
PEFORM propose $adresse "connect" "interlocutor" $obj1 $obj2
PEFORM query.if $adresse "connect" "interlocutor" $obj1 $obj2
PEFORM wait $time

ACHIEVE connect $sactor $obj1 $obj2
$sactor = unbound
PEFORM query.who $adresse "connect" $sactor $obj1 $obj2
+ option for action of $sactor=self and $sactor=interlocutor...

```

Abbildung 6.24: Handlungsoptionen in Bezug auf eine referentielle Frage

Geht es dem Agenten aber sowohl um das Planen als auch um die Ausführung eines Plans, so besteht das Ziel beispielsweise aus dem Bau eines konkreten Aggregats:

*ACHIEVE build \$sactor \$propeller*

Dieses Ziel lässt sich in die Teilziele *ACHIEVE shared\_plan \$plan* und das Teilziel des Bauens unterteilen. Durch eine AND-Verknüpfung der Pläne wird repräsentiert, dass der Agent beide Teilziele erreichen muss, um das Gesamtziel zu erreichen (siehe Abbildung 6.25).

Bei einer Verschachtelung von Planen und Handeln ist die Möglichkeit wichtig, Einschränkungen in Bezug auf AND-Verknüpfungen von Zielen spezifizieren zu können. Dies wird durch die typisierten Zielverknüpfungen der CASEC-Architektur ermöglicht (siehe CASEC Abschnitt 5.2.1). Ist die Reihenfolge beliebig, so kann der Agent an dem Teilziel arbeiten, das für ihn gerade die größte Relevanz besitzt; d.h. die Entscheidung fällt aufgrund der *Utility*- und Aktivierungswerte sowie der Interaktion mit dem Kooperationspartner.

Ein auszuführender *Plan* eines gemeinsamen Ziels besteht aus einem expliziten Teilziel, einen *SharedPlan* aufzustellen, und einem weiteren Teilziel, die entsprechenden Planschritte auszuführen. Dabei sorgt das Teilziel *ACHIEVE shared\_plan \$plan* dafür, dass der Agent die einzelnen Planschritte soweit plant, dass aufgrund der spezifizierten *Effekte* der einzelnen Planschritte zu erwarten ist, dass bei einer erfolgreichen Ausführung der einzelnen Schritte das Hauptziel erreicht

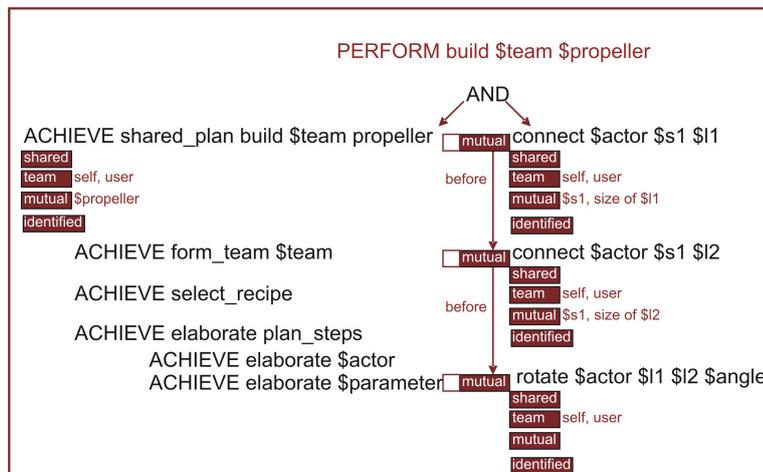


Abbildung 6.25: Intention der Planvervollständigung

wird. Die einzelnen Planschritte müssen so konkret spezifiziert sein, dass eine Ausführung vorgenommen werden kann.

Zu entscheiden, bis zu welcher Detaillierungsebene der Plan spezifiziert sein muss, stellt ein kontext- bzw. domänenabhängiges Problem dar. Eine Möglichkeit zur Problemlösung besteht darin, sich mit seinem Kooperationspartner abzustimmen, ob der Plan jetzt genau genug spezifiziert ist. Dabei versucht der Agent beispielsweise einen Zustand herbeizuführen, in welchem zumindest für jede Variable eine Einschränkung definiert oder der Wert der Variablen an einen Wert gebunden ist.

### *Coordinated Cultivation Requirement (CCR)*

Beim Aufstellen eines gemeinsamen Plans stimmt der Agent die einzelnen Schritte des Plans mit seinem Kooperationspartnern ab. Für jeden Teilplan wird vermerkt, dass es sich um einen gemeinsamen Plan handelt und wer dafür vorgesehen ist, den Plan auszuführen. Es kann zunächst offen bleiben, wer den Plan ausführen soll.

Im Rahmen der *SharedPlans* besteht die Obligation, seinem Kooperationspartner den neuen Zustand gemeinsamer Ziele und der damit verbundenen Pläne mitzuteilen, sollte sich an deren Zustand etwas Signifikantes ändern. Auch das Aufgeben eines gemeinsamen Ziels kann nur unter Berücksichtigung der **Mitteilungsverpflichtung** (siehe Abschnitt 2.1.2) erfolgen. Der Agent ist damit verpflichtet, seine Kooperationspartner über das Aufgeben eines Ziel zu informieren. Für *cMax* gelten folgende Regeln:

- Geht es um das Binden eines Parameters, werden zunächst Kandidaten bestimmt. Existieren mehrere potentielle Kandidaten, kommuniziert der Agent, welche Kandidaten er als sinnvoll erachtet. Stimmt sein Kooperationspartner mit ihm überein, so wird die Variable an den Wert gebunden.
- Gibt es bei der Auswahl eines Rezept mehrere Optionen, wie das Ziel erreicht werden kann, so informiert *cMax* sein Gegenüber über die Optionen.
- Ändert sich der Zustand eines gemeinsamen Ziels, so informiert *cMax* seinen Interaktionspartner auch darüber.

Der Agent ist verpflichtet, seinen Kooperationspartner über den Fortschritt des gemeinsamen Plans zu informieren. Allerdings wirkt es sehr unnatürlich, wenn der Agent seinen Kooperationspartner über Aktionen informiert, die dieser selbst gerade gesehen oder gehört hat. Daher arbeitet der Plan, seinen Kooperationspartner über relevante Ereignisse zu informieren, eng mit der *Behavior*-Ebene zusammen, welche unter anderem den Aufmerksamkeitsfokus des Gegenübers verfolgt. Hat der Kooperationspartner während der Ausführung einer manipulativen Handlung direkt auf die beteiligten Objekte fokussiert, so wird der Agent ihn nicht durch eine komplexe Äußerung darauf hinweisen. War der Kooperationspartner jedoch abgelenkt oder mit etwas anderem beschäftigt, wird er ihn über den Erfolg des Teilziels informieren.

## 6.4.2 Repräsentation des Kooperationspartners

Eine Grundannahme gemeinsamer Pläne und gemeinsamer Intentionalität besteht darin, dass es sich bei dem Interaktionspartner um einen *intentionalen* Agenten handelt (siehe Abschnitt 2.1.4). Für *cMax* ist diese Annahme explizit umgesetzt. *cMax* geht davon aus, dass jeder Agent, der ihn mit Sprache und auf ihn ausgerichteten Handlungen kontaktiert, einen intentionalen Agenten darstellt.

Durch die **explizite Repräsentation** der delegierten Handlungsziele in der Intensionsstruktur des Agenten wird eine zentrale Bedingung für *Gemeinsame Intentionalität* erfüllt (siehe Abschnitt 2.1.4). Diese besagt, dass die Ziele und Intentionen des Gegenübers Inhalt der Repräsentation der Ziele und Intentionen des Agenten sein müssen.

Zusätzlich soll in zukünftiger Arbeit ein *Partnermodell* aufgebaut werden, welches einen leichteren Zugriff auf die Überzeugungen und den Wissensstand des Gegenübers ermöglicht. Dabei kann auf die verschachtelten Beliefs des dynamischen Gedächtnismodells zurückgegriffen werden, die den mentalen Zustand des Partners repräsentieren.

## 6.5 Zusammenfassung

In diesem Realisierungskapitel wurde an ausgewählten Beispielen gezeigt, wie *cMax* konsequent das entwickelte BCTC-Modell (Kapitel 3) umsetzt. Die Implementation stützt sich dabei auf die CASEC-Architektur (Kapitel 5). *cMax* wurde gezielt entwickelt, um umfangreiche Kompetenzen in allen fünf Hauptmechanismen der Kooperation zu realisieren. Dies wird im Folgenden anhand des Modells der interaktiven Kooperation (Abbildung 2.9 hier noch mal als 6.26 abgedruckt) zusammenfassend deutlich gemacht, indem in Abbildung 6.27 die realisierten Fähigkeiten in das Modell eingeordnet werden.



Abbildung 6.26: Modell der interaktiven Kooperation

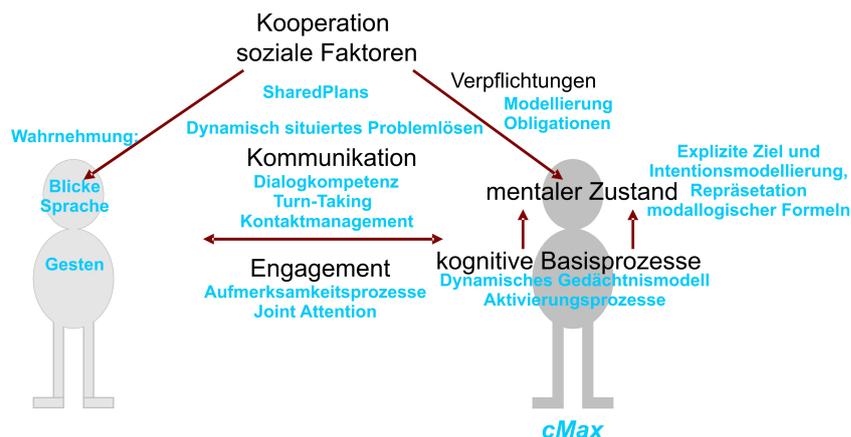


Abbildung 6.27: Beiträge der vorliegenden Arbeit für *cMax*

**Kooperation, soziale Faktoren** Das Kooperationsmodell der *SharedPlans* (siehe Abschnitt 2.1.3) stellt konkrete Anforderungen an den mentalen Zustand eines kooperierenden Agenten auf. *cMax* erfüllt alle 5 Anforderungen (**SP1**)-(**SP5**) des Modells. Durch die Realisierung von *SharedPlans* innerhalb der CASEC-Architektur ist *cMax* in der Lage, zu gegenseitiger Unterstützung auf der *Collaboration-Ebene*. Darüber hinausgehend ist *cMax* aufgrund der modallogischen Repräsentation der Beliefs und Intentionen seines Kooperationspartners befähigt, einen Zustand **Gemeinsamer Intentionalität** einzugehen. Damit werden Basismechanismen zu komplexeren Mechanismen erweitert, die auf einer höheren Ebene angesiedelt sind: Durch das Hinzukommen gemeinsamer Intentionalität kann aus Blickverfolgung *Joint Attention*, aus sozialer Manipulation kooperative Kommunikation sowie aus Gruppenaktivitäten Kooperation werden (Tomasello & Carpenter, 2007)(siehe Abschnitt 2.1.4).

Aufgrund der Repräsentation partieller Pläne und *Constraints* kann *cMax* auf der *Task-Ebene* koordiniert planen und handeln. Er verfügt über Kompetenzen des dynamisch situierten Problemlösens. Ebenso ist er in der Lage, Ziel- und Planinteraktionen zu erkennen. Er kann in Form eines Tutors seinen Interaktionspartner unterstützen und ihn auf Fehlhandlungen hinweisen.

**Kommunikation** Situierte Kommunikation wird auf der *Conversation-Ebene* umgesetzt. *cMax* verfügt über Kompetenzen des Kontaktmanagements, des *Turn-Taking* und der Dialogführung. Er setzt die Kontaktmanagementregeln des *BCTC*-Modells durch Obligationen um. Des Weiteren sind Verarbeitungsmechanismen für die Wahrnehmung und Generierung von *Turn-Taking*-Signalen realisiert. Ein mehrstufiger Entscheidungsprozess gewährleistet eine schnelle kontextsensitive Verarbeitung.

**Mentaler Zustand** Die explizite Repräsentation des mentalen Zustands erlaubt es *cMax*, Auskunft über seine Ziele, Intentionen und Überzeugungen zu erteilen. Durch die situierte Anpassung des Wissenstands über kognitive Basisprozesse kann *cMax* jederzeit Änderungen in seine Planung einbeziehen.

**Engagement** Auf der *Behavior-Ebene* geht es um das gegenseitige Aufeinandereingehen der Kooperationspartner. *cMax* erscheint durch Engagementsignale als freundlich, aufmerksam und interessiert. Er kann durch seine Aufmerksamkeitsprozesse auf seinen Partner eingehen.

*cMax* erfüllt alle 5 Anforderungen (**Att1**)-(**Att5**) des *BCTC*-Modells in Bezug auf Aufmerksamkeitsmechanismen (siehe Abschnitt 3.2.3). Aufbauend auf den kognitiven Basisprozessen richtet *cMax* seinen Aufmerksamkeitsfokus und Blick auf die Objekte, welche für ihn die größte Signifikanz besitzen (**Att1**). Die

Signifikanz lässt sich an den Aktivierungswerten der Entitäten in seinem Arbeitsgedächtnis sowie an Valenzwerten ablesen. Eine weitere Heuristik ist durch die *Turn-Taking*-Fähigkeiten umgesetzt (**Att2**). Das Lenken der Aufmerksamkeit bei signifikanten Ereignissen (**Att4**) ist durch die Umsetzung der *SharedPlans* (siehe Abschnitt 6.4.1) realisiert. Durch die Signalgenerierung des *Cue-Processor* zeigt *cMax* sein Interesse und Engagement in der Interaktion (**Att5**). Der Prozess der *Joint Attention* (**Att3**) wird durch die Verhaltensgenerierung und die Modellierung des mentalen Zustands von *cMax* erreicht. *cMax* ist sowohl in der Lage, auf die Einladung zur *Joint Attention* einzugehen, als auch selbst *Joint Attention* herbeizuführen. Dabei geht er während des Interaktionsprozesses durch Feedbacksignale kontextsensitiv auf seinen Interaktionspartner ein. Die Modellierung der *Joint Attention*-Prozesse erfüllt die in dieser Arbeit entwickelte formale Definition der *Joint Attention* (siehe Abschnitt 5.2.10).

*cMax* kann sich auf allen Ebenen der Kooperation einbringen und erfüllt die Anforderungen an einen Kooperationspartner. Dies gelingt durch die Realisierung der vorgestellten interaktionalen Fähigkeiten aufbauend auf der kognitiven CASEC-Architektur.



# Kapitel 7

## Resümee

### 7.1 Diskussion und Ergebnisse

**BCTC** Um die komplexe Aufgabe der Modellierung eines künstlichen Kooperationspartners zu bewältigen, wurde aus der Analyse von Kooperationsmodellen verschiedener Disziplinen (Bratman, 1992; P. Cohen & Levesque, 1991; B. Grosz & Hunsberger, 2006; Tomasello et al., 2005) ein *Modell der interaktiven Kooperation* (siehe Abbildung 7.1) entwickelt.

Während die vorgestellten Kooperationsmodelle Anforderungen auf einer abstrakten Basis aufstellen, werden im BCTC-Modell aufbauend auf einer komplexen Analyse der Mechanismen einer Kooperation unter Einbezug von Dialogmodellen (Bunt, 1996a; Traum & Larsson, 2003), Turn-Taking-Modellen (Thorisson, 2002; Cassell, Bickmore, Campbell et al., 2000b) sowie Engagement-Modellen (Sidner et al., 2003b; Nakano & Nishida, 2005a) konkrete Regeln definiert und Anforderungen an die Architektur eines Kooperationspartners abgeleitet.

Das BCTC-Modell beschreibt vier Ebenen der Interaktion. Wie in den verbreiteten Dialogsystemen wird auf der *Conversation*-Ebene von Sprechakten ausgegangen. Das BCTC-Modell geht aber über die beschriebenen Ansätze hinaus, indem nicht nur Dialogbeiträge sondern auch manipulative Handlungen berücksichtigt werden. Aus diesem Grund wird im BCTC-Modell zusätzlich zu den Ebenen des MRE-Modells (Traum & Rickel, 2002) eine Ebene der Intentionen berücksichtigt; es geht nicht nur darum, die Funktionen der Interaktionsbeiträge zu erkennen, sondern insbesondere darum, die Intentionen bzw. Ziele, die hinter der Handlung stehen, zu erkennen und zu verarbeiten. Gefasst wird dies in dem neuen Konzept des *Interaction-Moves*.

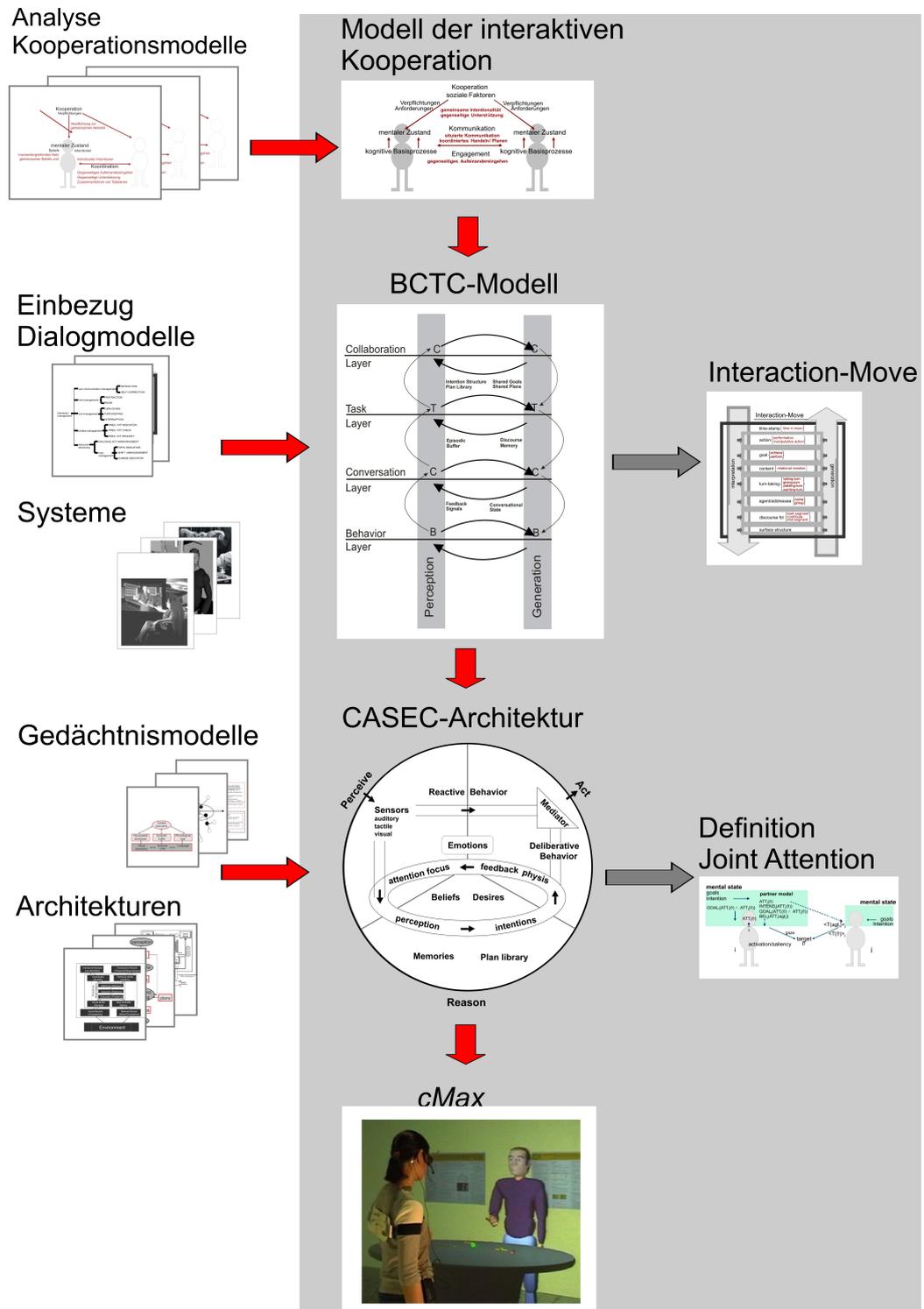


Abbildung 7.1: Veranschaulichung der Struktur der vorliegenden Arbeit, eigene Beiträge sind grau unterlegt

**CASEC** Da das Zusammenwirken sämtlicher Mechanismen der Kooperation zu einem stimmigen Gesamtverhalten des Agenten führen muss, erscheint ein kognitiver Ansatz unumgänglich, der sich in seiner Struktur und seinen Mechanismen am Menschen orientiert. Um diesen Anforderungen gerecht zu werden, wurde im Rahmen dieser Arbeit die CASEC-Architektur entwickelt. Diese vereint die Modellierung kognitiver Basismechanismen mit der Modellierung eines expliziten mentalen Zustands. Im Gegensatz zu kognitiven Architekturen wie Act-R (Anderson et al., 2004), Soar (Rosenbloom et al., 1993) und Icarus (Langley & Choi, 2006), die zumeist für die kognitive Modellierung einzelner Module bzw. Phänomene eingesetzt werden (z.B. Autofahren, Textlesen), muss die CASEC-Architektur einen ganzheitlichen Ansatz verfolgen, der eine Verarbeitung von Engagementsignalen über die Analyse von Interaktionsbeiträgen bis hin zur Intentionserkennung ermöglicht.

Die CASEC-Architektur erfüllt dies, indem sie Aspekte der klassischen BDI-Modellierung mit kognitiven Basisprozessen zusammenführt. Dabei werden bei der Konzeption der Architektur auch Designkriterien erfolgreicher Dialogsysteme und ECAs (z.B. Steve (Rickel & Johnson, 2000), FXPAL-Architektur (Cassell, Bickmore, Campbell et al., 2000a)) berücksichtigt. Im Gegensatz zu diesen Systemen sind die Strukturen und Basismechanismen in der CASEC-Architektur jedoch in stärkerem Maße an kognitionswissenschaftlichen Modellen ausgerichtet. Die CASEC-Architektur ist auf situierte Interaktion zugeschnitten; dafür werden Deliberationsprozesse in eine Hybridarchitektur eingebettet, um schnelle reaktive Basisprozesse zu ermöglichen und die Körperlichkeit des Agenten zu berücksichtigen.

Um die Kluft zwischen modalen BDI-Logiken und realen Implementierungen (Móra et al., 1999) zu verringern, erweitert die CASEC-Architektur die Komplexität ihres Repräsentationsformats im Vergleich zu den klassischen BDI-Implementierungen entscheidend. In CASEC können verschachtelte *Beliefs* verarbeitet werden, welche es ermöglichen, modallogische Formeln über den Interaktionspartner darzustellen. Dies ist in keiner der anderen BDI-Architekturen (UMPRS (Lee et al., 1994), JAM (Huber, 1999), Jadex (Pokahr et al., 2005b)) vorgesehen, ist aber für die Erstellung eines Partnermodells und für gemeinsame Intentionalität unabdingbar. Des Weiteren werden in der CASEC-Architektur im Gegensatz zu den anderen Systemen zeitliche Aspekte integriert und *Belief-Chunks*, Ziele und Intentionen mit einem Zeitstempel versehen, sodass zeitliche Bedingungen spezifiziert und überprüft werden können.

Darüber hinaus aktualisiert CASEC das klassische BDI-Paradigma mit neuen Ergebnissen der Kognitionswissenschaft. Dafür wird die starre Menge der *Beliefs* klassischer BDI-Systeme durch ein dynamisches Arbeitsgedächtnis ersetzt. Die Struktur des Arbeitsgedächtnisses folgt dabei den psychologischen Modellen von Cowan (1999) und Oberauer (2002) und ermöglicht die Modellierung verschie-

dener Zugriffsstrukturen, die einen effizienten kontextsensitiven Zugang zu den *Beliefs* ermöglichen. Zentral dafür sind Aktivierungs- und Primingprozesse, die unterschiedliche Ursprünge besitzen können und so Mechanismen des *Retrieval*, der *Spreading Activation* und des perzeptuellen sowie des assoziativen *Primings* umsetzen.

Die Ziel- und Intentionsmodellierung in CASEC liefert sowohl komplexe Strukturen für die Diskurs- als auch für die Interaktionsmodellierung. Die Intensionsstruktur wird in CASEC auch für die Intensions- und Planerkennung genutzt. Durch die Integration von *Constraints* wird es möglich, komplexe Zielstrukturen verarbeiten zu können. Die Zielmodellierung geht durch typisierte Zielverknüpfungen sowie die Integration von *Constraints* über andere Architekturen hinaus. Die *Constraints* dienen auch als Basis für die Repräsentation partieller Pläne, die eine Form des dynamisch situativen Problemlösens erlauben.

Während in der Basisstruktur von *Utility*-basierten Systemen wie JAM (Huber, 1999) lediglich ein Prioritätswert bzw. eine Prioritätsfunktion vorgesehen ist, besteht in der CASEC-Architektur die Möglichkeit, den gesamten Zielkontext des Agenten zu berücksichtigen und Ziel- und Planinteraktionen in den Entscheidungsprozess mit einzubeziehen. Dadurch dass mögliche Konflikte detektiert werden können, kann in CASEC zudem eine parallele Zielverfolgung unterstützt werden.

Auf der höchsten Ebene der Kooperation ermöglicht die CASEC-Architektur schließlich die Modellierung von *SharedPlans* (B. Grosz & Hunsberger, 2006), indem Repräsentationen der *Beliefs* und Intentionen des Kooperationspartners im Arbeitsgedächtnis und der Intensionsstruktur vermerkt werden können.

**cMax** Die CASEC-Architektur wurde eingesetzt, um den virtuellen Kooperationspartner *cMax* zu realisieren, der sich auf allen Ebenen des BCTC-Modells in eine Kooperation einbringen kann.

Betrachtet man den Stand der Forschung im Bereich der Engagementmodellierung, so lassen sich die Aufmerksamkeits- und Kontaktmanagement-Fähigkeiten des *cMax* mit denen des IPOC-Systems (Nakano & Nishida, 2005a) und Mel (Sidner & Lee, 2003) vergleichen. Während es jedoch in IPOC und Mel primär um die Produktion von Engagement-Signalen geht, sind im *cMax* die Engagement-Fähigkeiten in ein ganzheitliches System integriert, d.h. die Verhaltensweisen beruhen nicht auf Detektierte-Funktion-Reaktion-Strukturen sondern werden vielmehr durch Ziele und Obligationen umgesetzt. *cMax* ist sich dadurch seiner Verhaltensweisen „bewusst“ und kann sie kontextsensitiv anpassen. Durch den Einsatz eines Eyetrackers und der Modellierung kontextsensitiver Regeln können in *cMax* Engagement-Signale schnell und in sehr kleinen Verarbeitungsschleifen verarbeitet werden. *cMax* geht damit über die anderen Systeme hinaus, in denen ledig-

lich ein *Head-Tracking-System* eingesetzt wird. Mit diesem sind Blickfixationen nicht messbar, sie stellen aber wichtige Engagement-Signale dar. Während *cMax* in *mixed-initiative* Interaktionen agieren kann, werden diese weder in Mel noch in IPOC unterstützt.

Obwohl im Bereich der Robotik ein steigendes Interesse an der Modellierung von *Joint Attention* zu beobachten ist (Breazeal et al., 2004; Ogasawara et al., 2005), konzentrieren sich die meisten Modelle auf einzelne, isolierte Elemente des *Joint Attention*-Phänomens. Sie decken oberflächliche Verhaltensweisen wie gleichzeitige Blickfokussierung (Kim et al., 2005; Gu & Badler, 2006) oder einfache koordinierte Verhaltensweisen ab (Triesch, 2006), aber befassen sich nicht mit den kognitiven Herausforderungen von *Joint Attention*.

In der vorliegenden Arbeit dagegen wird eine logische Definition von *Joint Attention* vorgenommen, die aufbauend auf Aktivierungsprozessen und mentalen Zuständen durch formale Konzepte in der CASEC-Architektur verankert ist. Der Prozess der *Joint Attention* ist formalisiert sowie in *cMax* umgesetzt. Um einen Zustand der *Joint Attention* zu erreichen, ist sich *cMax* des Aufmerksamkeitsfokus seines Gegenübers und des Prozesses des Teilens der Aufmerksamkeit bewusst. Er ist sowohl in der Lage, auf die Einladung zur *Joint Attention* einzugehen, als auch selbst *Joint Attention* herbeizuführen.

*cMax* verfügt über komplexe *Turn-Taking*-Fähigkeiten. Die CASEC-Architektur erfüllt die Designrichtlinien, die für *Turn-Taking* verarbeitende Systeme gefordert werden (Cassell, Bickmore, Campbell et al., 2000b; Thorisson, 2002). Für die Modellierung des *Turn-Takings* wird ein sowohl Regel- als auch Signal-basierter Ansatz verfolgt. Ein mehrstufiger Entscheidungsprozess gewährleistet eine schnelle kontextsensitive Verarbeitung. Darüber hinaus wird nicht allein die Funktion eines *Turn-Taking*-Signals sondern auch der intentionale Beitrag sowie die Dringlichkeit und das Dominanzverhältnis der Interlokutoren im Entscheidungsprozess berücksichtigt.

Bisher bestehen wenige Systeme, die in einer gemeinsamen Umwelt als direkter Kooperationspartner des Menschen agieren. In SASO-ST geht es zwar um Aushandlungsprozesse, aber es gibt keine gemeinsame Aufgabe, die die Umwelt der Interaktionspartner einbezieht (Traum et al., 2005). Leonardo erfüllt viele der Anforderungen an einen Kooperationspartner (Breazeal et al., 2004). Aufgrund fehlender Sprachkompetenz ist er aber nicht in der Lage, gemeinsame Pläne aufzustellen und Handeln und Planen in einer verschachtelten Weise auszuführen. Das TRIPS-System (Allen et al., 2001) liefert wichtige Beiträge in Bezug auf das verschachtelte Auftreten von Planen und Handeln, allerdings stellt TRIPS ein Dialogsystem und keinen verkörperten Agenten dar. *cMax* vereint die Fähigkeiten, als verkörperter Agent in einer gemeinsamen Umwelt mit einem Partner in einer Kooperation zu interagieren, die sowohl Planen als auch Handeln in verschachtelter Form umfasst.

## 7.2 Zusammenfassung

Natürliche Kooperation bezieht sich auf eine Interaktion von Angesicht-zu-Angesicht, d.h. auf die Zusammenarbeit zwischen Kommunikationspartnern, die sich in einer gemeinsamen Situation befinden und sich in ihrer körperlichen Erscheinung wahrnehmen können. Dem Ausdruck von Information im und durch den Körper, ob gewollt oder nicht, kommt dabei eine große Bedeutung zu. Künstliche Systeme, die die Rolle eines solchen kooperativen Gegenübers einnehmen wollen, müssen über natürliche Modalitäten wie Sprache, Mimik und Gestik verfügen und mit einer anthropomorphen Hülle ausgestattet sein. Dabei muss der körperliche Ausdruck durch eine Menge kognitiver Basisprozesse gesteuert werden, um ein konsistentes, kontextsensitives Verhalten zu produzieren.

In der vorliegenden Arbeit wird von der Konzeption bis hin zur Implementation die Entwicklung eines künstlichen kognitiven Kooperationspartner dargestellt, der befähigt ist zum:

- Fällen von selbstständigen, kontextsensitiven Entscheidungen
- Abwägen von Plänen und Handlungsoptionen
- Erkennen und Behandeln von Zielkonflikten
- Herbeiführen gemeinsamer Aufmerksamkeit
- Führen von *mixed-initiative* Dialogen
- Aufstellen und Verfolgen gemeinsamer Pläne
- Dynamischen Adaptieren an den Kooperationspartner und die Umwelt

Zu seiner Entwicklung liefert der Mensch mit seinen besonderen Fähigkeiten das Leitbild, indem das Gedächtnis und das prozedurale Verhalten des Agenten den menschlichen Entsprechungen weitgehend nachgebildet werden. Damit wird dem Menschen ein künstliches Gegenüber geschaffen, das - fokussiert auf einen beschränkten Ausschnitt - menschenähnliches Verhalten produziert.

In dieser Arbeit dient die Konzeption des kooperationsfähigen Agenten, der sich - eingebettet in eine dynamisch wechselnde Weltsituation - mit Menschen sprachlich, gestisch, mimisch sowie durch manipulative Handlungen dynamisch in den Lösungsprozess einer gemeinsamen Aufgabe einbringen kann, als Anlass, einen ganzheitlichen Blick auf die Realisierung eines Künstlichen Kooperationspartners zu nehmen. Das Gewicht liegt somit auf den Fähigkeiten natürlicher Interaktion, gepaart mit Fähigkeiten eines kognitiven Systems, sich auf die Anforderungen aus wechselnden Aufgabenzusammenhängen situationsgerecht einzustellen. Vorgestellt wird dies am Beispiel des virtuellen Kooperationspartners *cMax*.

Leitend ist dabei das Ziel, Methoden der modernen Kooperationsmodellierung sowie Erkenntnisse über kognitive Architekturen und heutige Realisierungsmöglichkeiten von Perzeptions- und Aktionskomponenten in einem integrierten Demonstrator zusammenzuführen.

Ein charakteristisches Merkmal intelligenter Systeme ist das Ineinandergreifen zahlreicher Teilfunktionen. Einhergehend mit der Ausdifferenzierung der KI in zahlreiche Teilgebiete mit jeweils eigenen, häufig sehr speziellen Forschungsfragen und Methoden beschränkten sich in der Vergangenheit viele Projekte darauf, die Realisierung eines geeigneten Umfangs von tragfähigen Teilfunktionalitäten anzustreben. Beispiele sind perzeptive Komponenten im Bildverstehen, Sprachdialogsysteme, Planer, u.v.m. Mit dem wieder erwachten Interesse an integrierten handlungsfähigen Systemen, beispielsweise im Kontext von humanoiden Robotern oder *Embodied Conversational Agents*, verschieben die Fortschritte auf diesem Feld die Herausforderung mehr und mehr zur Frage einer übergreifenden Architektur, die eine große Anzahl von Teilfunktionen integrieren und zum intelligenten Zusammenwirken führen kann.

Mit der CASEC-Architektur wird ein Schritt in die Richtung einer solchen komplexen Architektur unternommen. Dabei werden Interaktionskompetenzen auf verschiedenen Ebenen konzipiert und umgesetzt. Die dabei berücksichtigten Fähigkeiten reichen von einem dynamischen Anpassen des Aufmerksamkeitsfokus im Rahmen dyadischer Engagementprozesse über Dialogkompetenzen bis hin zu dynamisch-situativem Problemlösen sowie der Fähigkeit zur gemeinsamen Intentionalität. Als Berechnungsgrundlage der kooperativen Fähigkeiten verfolgt die CASEC-Architektur einen hybriden Ansatz. Durch symbolbasierte Repräsentationen von Fakten und Algorithmen werden feste Strukturen bereitgestellt, in welchen subsymbolische Prozesse die potentiellen Kandidaten, die Berechnungsreihenfolge und die Berechnungstiefe beeinflussen können. Einerseits sorgen die subsymbolischen Prozesse dafür, den aktuellen Kontext in den Entscheidungsprozess mit einzubeziehen und dem Aspekt eines situierten Agenten Rechnung zu tragen. Des Weiteren sind die subsymbolischen Prozesse dafür verantwortlich, dass sich der Agent an den situationalen Kontext anpasst. Andererseits sorgt die symbolische Struktur dafür, dass der Agent rational agiert und die subsymbolischen Prozesse nicht außer Kontrolle geraten. Sie kontrolliert ein intelligentes, angemessenes Commitmentverhalten und bringt Stabilität und Zuverlässigkeit in die Interaktion. Diese Aspekte sind gerade im Rahmen einer Kooperation von entscheidender Bedeutung.

Wichtiger als eine perfekte Simulation einer menschenähnlichen Erscheinung ist im Rahmen einer solchen Arbeit *interaktiver Realismus* und ein schlüssiges Zusammenspiel der Verhaltensweisen. *cMax* zeigt natürliches Dialogverhalten: Er schaut seinen Interlokutor beim Sprechen und Zuhören an und setzt ein kontextsensitives Blickverhalten zur Verdeutlichung seines Aufmerksamkeitsfokus ein.

Er bedient sich einer natürlich wirkenden Gestik und zeigt elementare Feedback-signale. Im Rahmen des *Turn-Taking* wartet er, bis sein Gegenüber ausgedet hat, bevor er selbst spricht, kann aber auch mit Unterbrechungen natürlich umgehen. Er ist in der Lage, sich an wechselnde Initiativesituationen im Rahmen dynamisch-situativen Problemlösens anzupassen. Darüber hinaus verfügt er über Kompetenzen echter Kooperation im Rahmen gemeinsamer Pläne und gemeinsamer Intentionalität.

Wenn auch nicht verwechselbar menschenähnlich so zeigt *cMax* dennoch die dem Menschen vertrauten Formen der Kommunikation und Kooperation und kann als angenehmer Kooperationspartner agieren.

# Literaturverzeichnis

- Allen, J. (1999). Mixed-initiative interaction. *Trends and Controversies, IEEE Intelligent Systems*, 15 (4), 14–23.
- Allen, J., Byron, D., Dzikovska, M., Ferguson, G. & Galescu, L. (2001). Towards Conversational Human-Computer Interaction. *AI Magazine*, 22 (4), 27–37.
- Allwood, J. (1976). Linguistic communication in action and co-operation: A study in pragmatics. In *Gothenberg Monographs in Linguistics* (Bd. 2, S. 637–663). Gothenberg: Department of Linguistics, University of Gothenberg.
- Allwood, J. (2001). Cooperation and Flexibility in Multimodal Communication. *Lecture Notes in Computer Science*, 2155, 113–124.
- Anderson, J. R. (1983). A Spreading Activation Theory of Memory. *Journal of Verbal Learning and Verbal Behavior*, 22, 261–295.
- Anderson, J. R. (1993). *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum.
- Anderson, J. R. (2001). *Kognitive Psychologie*. Heidelberg: Spektrum Akad. Verl.
- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C. & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111 (4), 1036–1060.
- Anderson, J. R. & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.
- Argyle, M. (1973). *Social Interaction*. London: Tavistock Pubns.
- Argyle, M. & Cook, M. (1976). *Gaze and mutual gaze*. Cambridge, UK: Cambridge University Press.
- Austin, J. L. (1962). *How to do things with words*. Oxford, UK: Oxford University Press.

- Axelrod, R. (1984). *Evolution of Cooperation*. New York: Basic Books.
- Baars, B. J. (2002). The conscious access hypothesis: origins and recent evidence. *Trends Cognitive Science*, 6, 47-52.
- Baars, B. J. & Franklin, S. (2003). How conscious experience and working memory interact. *Trends in Cognitive Sciences*, 7 (4), 166–172.
- Baddeley, A. (1993). Working memory and conscious awareness. In A. F. Collins, S. E. Gathercole, M. A. Conway & B. E. Morris (Hrsg.), *Theories of Memory* (S. 11–28). Hove, UK: Lawrence Erlbaum Associates.
- Baddeley, A. (1997). *Human memory. theory and practice*. Hove, UK: Psychology Press.
- Baddeley, A. (2000). The Episodic Buffer: A New Component of Working Memory? *Trends in Cognitive Sciences*, 4 (11), 417-423.
- Baddeley, A. (2001). Is Working Memory Still Working? *American Psychologist*, 56, 849-864.
- Baddeley, A. (2002). Fractionating the Central Executive. In D. T. Stuss & R. T. Knight (Hrsg.), *Principles of Frontal Lobe Function* (S. 246–260). New York: Oxford University Press.
- Baddeley, A. (2003). Working Memory: Looking Back and Looking Forward. *Nature Reviews Neuroscience*, 4 (10), 829–839.
- Baddeley, A. (2007). *Working Memory, Thought, and Action*. New York: Oxford University Press.
- Baddeley, A., Allen, R. & Hitch, G. J. (2010). Investigating the episodic buffer. *Psychologica Belgica*, 50, 223–243.
- Baddeley, A. & Andrade, J. (1998). Working memory and consciousness: An empirical approach. In M. Conway, S. Gathercole & C. Cornoldi (Hrsg.), *Theories of MemoryII* (S. 1–24). Psychology Press.
- Baddeley, A. & Hitch, G. (1974). Working memory. In G. Bower (Hrsg.), *The psychology of learning and motivation: Advances in research and theory* (Bd. 8, S. 47–89). New York: Academic Press.
- Baddeley, A. & Logie, R. H. (1999). Working memory: The multi-component model. In A. Miyake & P. Shah (Hrsg.), *Models of Working Memory, Mechanisms of Active Maintenance and Executive Control* (S. 28–62). Cambridge, UK: Cambridge University Press.

- Balkanski, C. T. & Hurault-Plantet, M. (2000). Cooperative requests and replies in a collaborative dialogue model. *Int. J. Hum.-Comput. Stud.*, 53 (6), 915-968.
- Barber, K. S. & Liu, T. H. (2000). Conflict Detection during Plan Integration Based on E-PERT Diagrams. In C. Sierra, M. Gini & J. S. Rosenschein (Hrsg.), *Proceedings of the Fourth International Conference on Autonomous Agents* (S. 106–107). Barcelona, Catalonia, Spain: ACM Press.
- Baron-Cohen, S. (1994). The Eye-Direction Detector (EDD) and the Shared Attention Mechanism (SAM): Two Cases for Evolutionary Psychology. In C. Moore & P. Dunham (Hrsg.), *Joint Attention: Its origins and role in development* (S. 41-61). L. Erlbaum.
- Beaudoin, L. (1994). *Goal processing in autonomous agents*. PhD thesis, School of Computer Science, The University of Birmingham.
- Bechara, A. (2004). The role of emotion in decision-making: Evidence from neurological patients with orbitofrontal damage. *Brain and Cognition*, 55, 30-40.
- Becker, C., Kopp, S. & Wachsmuth, I. (2004). Simulating the emotion dynamics of a multimodal conversational agent. In *Proceedings of affective dialogue systems: Tutorial and research workshop (ads 2004)* (S. 154-165). Berlin Heidelberg: Springer.
- Becker, C., Leßmann, N., Kopp, S. & Wachsmuth, I. (2006). Connecting feelings and thoughts - modeling the interaction of emotion and cognition in embodied agents. In *Proceedings of the Seventh International Conference on Cognitive Modeling (ICCM-06)* (S. 32-37). Trieste: Edizioni Goliardiche.
- Bickmore, T. & Cassell, J. (2005). Social dialogue with embodied conversational agents. In J. van Kuppevelt, L. Dybkjaer & N. Bernsen (Hrsg.), *Advances in Natural, Multimodal Dialogue Systems* (S. 1–27). New York: Kluwer Academic Publishers.
- Biermann, P. & Wachsmuth, I. (2003). An implemented approach for a visual programming environment in VR. In *Proceedings of the Fifth Virtual Reality Interaction Conference (VRIC 2003), Laval, France* (S. 229–234).
- Biron. (2011). *Biron – (Bielefeld ROBot CompanioN) published on website:* (<http://www.hofemann.de/nils/wiss.html>)
- Blaylock, N. (2005). *Towards Tractable Agent-based Dialogue*. PhD thesis, Department of Computer Science, The College Arts and Sciences, University of Rochester, Rochester, New York.

- Blaylock, N. & Allen, J. (2005). A collaborative problem-solving model of dialogue. In L. Dybkjær & W. Minker (Hrsg.), *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue* (S. 200–211). East Stroudsburg, PA, USA: Association for Computational Linguistics.
- Blaylock, N., Allen, J. & Ferguson, G. (2003). Managing communicative intentions with collaborative problem solving. In J. van Kuppevelt & R. Smith (Hrsg.), *Current and New Directions in Discourse and Dialogue* (Bd. 22, S. 63–84). The Netherlands: Kluwer Academic Publishers.
- Boutilier, C. & Brafman, R. I. (1997). Planning with concurrent interacting actions. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97)* (S. 720–726). Cambridge, USA: AAAI Press / MIT Press.
- Bratman, M. (1987). *Intention, Plans, and Practical Reason* (Bd. 250). Cambridge, MA, USA: Harvard University Press.
- Bratman, M. (1992). Shared Cooperative Activity. *The Philosophical Review*, 101(2), 327-341.
- Bratman, M. (1999). *Faces of Intention: Selected Essays on Intention and Agency*. Cambridge, UK: Cambridge University Press.
- Bratman, M., Israel, D. & Pollack, M. (1991). Plans and Resource-Bounded Practical Reasoning. In R. Cummins & J. L. Pollock (Hrsg.), *Philosophy and AI: Essays at the Interface* (S. 1–22). Cambridge, Massachusetts: The MIT Press.
- Braubach, L., Pokahr, A., Lamersdorf, W. & Moldt, D. (2004, 7). Goal Representation for BDI Agent Systems. In R. H. Bordini, M. Dastani, J. Dix & A. E. Fallah-Seghrouchni (Hrsg.), *Second International Workshop on Programming Multiagent Systems: Languages and Tools* (S. 9–20). Heidelberg: Springer-Verlag.
- Braver, T. S., Barch, D. M. & Cohen, J. (2002). The role of prefrontal cortex in normal and disordered cognitive control: A cognitive neuroscience perspective. In D. T. Stuss & R. T. Knight (Hrsg.), *Principles of Frontal Lobe Function* (S. 428-447). Oxford, UK: Oxford University Press.
- Braver, T. S. & Cohen, J. (2000). On the control of control: The role of dopamine in regulating prefrontal function and working memory. In S. Monsell & J. Driver (Hrsg.), *Attention and Performance XVIII; Control of cognitive processes* (S. 713-737).

- Breazeal, C., Brooks, A., Gray, J., Hoffman, G., Kidd, C., Lee, H. et al. (2004). Humanoid robots as cooperative partners for people. *Int. Journal of Humanoid Robots*, 1–34.
- Brinck, I. (2003). The objects of attention. In *Proc. of ESPP2003, Torino* (S. 1–4).
- Broersen, J., Dastani, M. & Torre, L. van der. (2005). Beliefs, Obligations, Intentions and Desires as Components in an Agent Architecture. *International Journal of Intelligent Systems*, 20 (9), 893–920.
- Bryson, J. & Thorisson, K. R. (2000). Bats, Dragons & Evil Knights: A Three-Layer Design Approach to Character-Based Creative Play. In *Virtual Reality, Special Issue on Intelligent Virtual Agents*, 5 (S. 57-71). Heidelberg: Springer-Verlag.
- Bunt, H. (1996a). *Dynamic Interpretation and Dialogue Theory* (Bericht). Amsterdam: Laboratoire LIMSI-CNRS, CHM/96/37) Orsay/Paris Sud.
- Bunt, H. (1996b). Interaction management functions and context representation requirements. In S. LuperFoy, A. Nijholt & G. V. van Zanten (Hrsg.), *Dialogue Management in Natural Language Systems* (S. 187-198). Enschede: Universiteit Twente.
- Busetta, P., Howden, N., Rönquist, R. & Hodgson, A. (2000). Structuring BDI Agents in Functional Clusters. In N. Jennings & Y. Lespérance (Hrsg.), *Intelligent Agents VI. Agent Theories Architectures, and Languages* (Bd. 1757, S. 277–289). Berlin / Heidelberg: Springer-Verlag.
- Byrne, M. (2003). Cognitive architecture. In J. Jacko & A. Sears (Hrsg.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications* (S. 97–117). Hillsdale, NJ: Erlbaum.
- Carberry, S. (1990). *Plan Recognition in Natural Language Dialogue*. Cambridge, MA, USA: MIT Press.
- Carletta, J., Isard, A., Isard, S., Kowtko, J. C., Doherty-Sneddon, G. & Anderson, A. H. (1997, 1997). The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23 (1), 13–31.
- Cassell, J. (2011). *Justine Cassell - Rea published on website:* ([http://www.justinecassell.com/jc\\_research](http://www.justinecassell.com/jc_research))

- Cassell, J. & Bickmore, T. (2003). Negotiated Collusion: Modeling Social Language and its Relationship Effects in Intelligent Agents. *User Modeling and User-Adapted Interaction*, 13 (1-2), 89-132.
- Cassell, J., Bickmore, T., Campbell, L., Chang, K., Vilhjálmsón, H. & Yan, H. (1999). Requirements for an Architecture for Embodied Conversational Characters. In *Proceedings of Computer Animation and Simulation* (S. 109–120). Springer-Verlag.
- Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsón, H. & Yan, H. (2000a). Conversation as a system framework: Designing embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost & E. Churchill (Hrsg.), *Embodied Conversational Agents*. Cambridge, MA, USA: MIT Press.
- Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsón, H. & Yan, H. (2000b). Human conversation as a system framework: Designing embodied conversational agents. In *Embodied Conversational Agents* (S. 29–63). Cambridge, MA, USA: MIT Press.
- Cassell, J., Bickmore, T., Vilhjálmsón, H. & Yan, H. (2000). More than just a pretty face: affordances of embodiment. In *Proceedings of the 5th International Conference on Intelligent User Interfaces* (S. 52–59). New York, NY, USA: ACM.
- Cassell, J., Sullivan, J., Prevost, S. & Churchill, E. (2000). *Embodied Conversational Agents*. Cambridge (MA): MIT Press.
- Chalmers, S. & Gray, P. (2001). BDI Agents and Constraint Logic. *AISB Journal Special Issue on Agent Technology*, 1 (1), 21–40.
- Choi, D., Kaufman, M., Langley, P., Nejati, N. & Shapiro, D. (2004). An Architecture for Persistent Reactive Behavior. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* (S. 988–995). Washington, DC, USA: IEEE Computer Society.
- Choi, D. & Langley, P. (2006). *A unified cognitive architecture for physical agents* (Bericht). Computational Learning Laboratory, CSLI, Stanford University, CA.
- Chu-Carroll, J. & Brown, M. K. (1997). Tracking initiative in collaborative dialogue interactions. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics* (S. 262–270). Morristown, NJ, USA: Association for Computational Linguistics.

- Churchland, P. (1981). Eliminative Materialism and the Propositional Attitudes. *Journal of Philosophy*, 78, 67-90.
- Clark, H. H. (1996). *Using Language*. Cambridge, UK: Cambridge University Press.
- Clark, H. H. & Schaefer, E. F. (1989). Contributing to Discourse. *Cognitive Science*, 13, 259-294.
- Clement, B. J. & Durfee, E. H. (1999). *Identifying and resolving conflicts among agents with hierarchical plans* (Bericht). Menlo Park, California, USA: In AAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities.
- Cochran, R. E., Lee, F. J. & Chown, E. (2006). Modeling emotion: Arousal's impact on memory. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society, Vancouver, British Columbia, Canada* (S. 1133–1138).
- Cohen, J., Aston-Jones, G. & Gilzenrat, M. (2004). A systems-level perspective on attention and cognitive control: Guided activation, adaptive gating, conflict monitoring, and exploitation vs. exploration. In M. Posner (Hrsg.), *Cognitive Neuroscience of Attention* (S. 71-90). New York: Guilford Press.
- Cohen, P. & Levesque, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42, 213–261.
- Cohen, P. & Levesque, H. (1991). Teamwork. *Special Issue on Cognitive Science and AI*, 25, 4, 487-512.
- Cohen, P., Levesque, H. & Smith, I. (1998). On team formation. In J. Hintikka & R. Tuomela (Hrsg.), *Contemporary action theory* (S. 87–114). Kluwer Academic Publisher.
- Conway, M. (2001). Sensory perceptual episodic memory and its context: autobiographical memory. *Philosophical Transactions B*, 356 (1413), 53-71.
- Conway, M. & Pleydell-Pearce, C. (2000). The construction of autobiographical memories in the self-memory system. *Psychological Review*, 107 (2), 261-288.
- Cooper, R. & Larsson, S. (1999). Dialogue moves and information states. In *Proceedings of the Third International Workshop on Computational Semantics* (S. 398–400).
- Cooper, R. & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological review*, 113 (4), 887–916.

- Courtney, S. M. (2004). Attention and cognitive control as emergent properties of information representation in working memory. *Cognitive, Affective, & Behavioral Neuroscience*, 4 (4), 501-516.
- Cowan, N. (1995). *Attention and memory: An integrated framework*. (Nr. 26). New York: Oxford University Press.
- Cowan, N. (1999). An embedded-processes model of working memory. In A. Miyake & P. Shah (Hrsg.), *Models of Working Memory, Mechanisms of Active Maintenance and Executive Control* (S. 62-102). Cambridge, UK: Cambridge University Press.
- Cowan, N., Fristoe, N. M., Elliott, W., Brunner, R. & Sauls, J. (2006). Scope of attention, control of attention, and intelligence in children and adults. *Memory & Cognition*, 34 (8), 1754–1768.
- Damasio, A. (1994). *Descartes' Error: Emotion, Reason, and the Human Brain*. New York: Putnam.
- Dastani, M., Dignum, F. & Meyer, J. (2003). Autonomy and Agent Deliberation. In M. Rovatsos & M. Nickles (Hrsg.), *Proceedings of The First International Workshop on Computational Autonomy - Potential, Risks, Solutions (Autonomous 2003)* (S. 23-35). Heidelberg: Springer-Verlag.
- Della, S. S., Baddeley, A. & Spinnler, C. P. H. (1995). Dual task paradigm: A means to examine the central executive. In F. B. J. Grafman K. J. Holyoak (Hrsg.), *Structure and functions of the human prefrontal cortex* (Bd. 769, S. 161–171). New York: Annals of the New York Academy of Sciences.
- Dennett, D. (1987). *The Intentional Stance*. Cambridge, MA: MIT Press.
- Dignum, F., Kinny, D. & Sonenberg, L. (2002). From desires, obligations and norms to goals. *Cognitive Science Quarterly*, 2 (3-4), 407–430.
- d'Inverno, M., Kinny, D., Luck, M. & Wooldridge, M. (1998). A Formal Specification of dMARS. In *ATAL '97: Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages* (S. 155–176). London, UK: Springer-Verlag.
- Duncan, S. (1972). Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23(2), 283-292.
- Duncan, S. (1974). On the structure of speaker-auditor interaction during speaking turns. *Language in Society*, 3, 161-180.

- Elio, R. (2002). Belief-desire-intention agency in a general cognitive architecture. *Cognitive Science Quarterly*, 2, 321–340.
- Engle, R. W., Kane, M. J. & Tuchsolski, S. W. (1999). Individual differences in working memory capacity and what they tell us about controlled attention, general fluid intelligence, and the functions of the prefrontal cortex. In A. Miyake & P. Shah (Hrsg.), *Models of Working Memory, Mechanisms of Active Maintenance and Executive Control* (S. 102-135). Cambridge, UK: Cambridge University Press.
- Ericsson, K. A. & Kintsch, W. (1995). Long-term working memory. *Psychological Review*, 102, 211-245.
- Ferguson, G. & Allen, J. (2007). Mixed Initiative Systems for Collaborative Problem Solving. *AI*, 23-32.
- Fikes, R. & Nilsson, N. (1971). Strips: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2, 189–208.
- Fink, G., Fritsch, J., Leßmann, N., Ritter, H., Sagerer, G., Steil, J. et al. (2003). Architekturen situierter Kommunikatoren: Von Perception Über Kognition zum Lernen. In K. Dittrich, W. König, A. Oberweis, K. Rannenber & W. Wahlster (Hrsg.), *Informatik 2003, Innovative Informatikanwendungen, Beiträge der 33. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 29. September - 2. Oktober 2003, Frankfurt am Main* (Bd. P-35, S. 29-44). Bonn, Germany: German Informatics Society (GI), Köllen Verlag.
- Fink, G., Schillo, C., Kummert, F. & Sagerer, G. (1998). Incremental speech recognition for multimodal interfaces. In *24th Annual Conference of the IEEE Industrial Electronics Society* (Bd. 4, S. 2012-2017). Washington, DC, USA: IEEE Computing Society.
- Fodor, J. (1988). Psychosemantics. *Journal of Philosophy*, LXXXV, 384–389.
- Franklin, S. & Ferkin, M. (2006). An ontology for comparative cognition: A functional approach. *Comparative Cognition and Behavior Reviews*, 1, 36-52.
- Franklin, S., Ramamurthy, U., Mello, S. K. D., McCauley, L., Negatu, A., L, R. S. et al. (2007). LIDA: A computational model of global workspace theory and developmental learning. In *AAAI Fall Symposium on AI and Consciousness: Theoretical Foundations and Current Approaches* (S. 61–66). Menlo Park, California, USA: AAAI Press.

- George, M. & Rao, A. (1996). A Profile of the Australian AI Institute. *IEEE Expert*, 11 (6).
- Georgeff, M. & Lansky, A. (1987). Reactive reasoning and planning. In *Proceedings of the Sixth National Conference of Artificial Intelligence* (S. 677-682). Menlo Park, California, USA: AAAI Press/MIT Press.
- Georgeff, M., Pell, B., Pollack, M., Tambe, M. & Wooldridge, M. (1999). The Belief-Desire-Intention Model of Agency. In J. Müller, M. P. Singh & A. S. Rao (Hrsg.), *Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages (ATAL-98)* (Bd. 1555, S. 1–10). Heidelberg, Germany: Springer-Verlag.
- Gilbert, M. (1989). *On social facts*. New York: Routledge.
- Goodwin, C. (1981a). Achieving Mutual Orientation at Turn Beginning. *Conversational Organization: Interaction between speakers and hearers*, 55-89.
- Goodwin, C. (1981b). *Conversational organization: Interaction between speakers and hearers*. Academic Press New York.
- Gratch, J. & Marsella, S. (2004). A Domain-independent Framework for Modeling Emotion. *Journal of Cognitive Science Research*, 5, 269-306.
- Gray, J. R. (2004). Integration of Emotion and Cognitive Control. *Current Directions in Psychological Science*, 13 (2), 46-48.
- Gray, J. R. & Braver, T. S. (2002). Integration of emotion and cognitive control - A neurocomputational hypothesis of dynamic goal regulation. In S. C. Moore & M. Oaksford (Hrsg.), *Emotional Cognition - From Brain to Behavior* (S. 289–316). Amsterdam, Netherlands: John Benjamins.
- Gray, P. (1999). *Psychology, Fourth Edition*. New York: Worth Publishers.
- Grice, H. P. (1969). Utterer's meaning and intentions. *Philosophical Review*, 78, 147-177.
- Grosz, B. (1996). Collaborative Systems. *AI magazin, AAAI-94 Presidential Address*, 17 (2), 67–85.
- Grosz, B. & Hunsberger, L. (2006). The dynamics of intention in collaborative activity. *Cognitive Systems Research*, 7 (2-3), 259–272.
- Grosz, B. & Kraus, S. (1993, 1993). Collaborative Plans for Group Activities. In *IJCAI 1993* (S. 367–375). San Francisco, USA: Morgan Kaufmann Publishers.

- Grosz, B. & Sidner, C. (1986). Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12:3.
- Grosz, B. J. & Kraus, S. (1998). The Evolution of SharedPlans. In A. Rao & M. Wooldridge (Hrsg.), *Foundations and theories of rational agency* (S. 227–262). Kluwer Academic Publishers.
- Gu, E. & Badler, N. (2006). Visual attention and eye gaze during multiparty conversations with distractions. *LNCS Intelligent Virtual Agents*, 4133, 193–204.
- Gunzelmann, G. & Anderson, J. R. (2003). Problem solving: Increased planning with practice. *Cognitive Systems Research*, 4, 57-76.
- Haller, S. & Fossum, T. (1999). Using protocols to model mixed initiative interaction. In *AAAI Workshop on Mixed Initiative Intelligence*. AAAI Press.
- Halliday, M. (1985). *An Introduction to Functional Grammar*. Arnold.
- Hobson, R. P. (2005). What Puts the Jointness into Joint Attention? In N. Eilan, C. Hoerl, T. McCormack & J. Roessler (Hrsg.), *Joint attention: communication and other minds* (S. 185-204). Oxford, UK: Oxford University Press.
- Hoffman, G. & Breazeal, C. (2004). Collaboration in Human-Robot Teams. In *Proceedings of AIAA First Intelligent Systems Technical Conference, Chicago, IL*. AIAA.
- Hommel, B. (2000). Intentional control of automatic stimulus-response translation. In Y. Rossetti & A. Revonsuo (Hrsg.), *Interaction between dissociable conscious and nonconscious processes* (S. 223-244). Amsterdam, Netherlands: John Benjamins Publishing Company.
- Horvitz, E. (2007). Reflections on Challenges and Promises of Mixed-initiative Interaction. *AI Magazine*, 19-22.
- Howden, N., Rönquist, R., Hodgson, A. & Lucas, A. (2001). JACK Intelligent Agents-Summary of an Agent Infrastructure. In *Proceedings of the 5th ACM International Conference on Autonomous Agents*.
- Huber, M. J. (1999). JAM: A BDI-theoretic Mobile Agent Architecture. In *Proceedings of the International Conference on Autonomous Agents* (S. 236–243). Seattle, WA.

- Hummel, J. (1999). The Binding Problem. In R. A. Wilson & F. C. Keil (Hrsg.), *The MIT Encyclopedia of Cognitive Sciences*. Cambridge, MA, USA: MIT Press.
- ISO/IEC. (2002). *Z Formal Specification Notation – Syntax, Type System and Semantics*. Information Technology (ISO/IEC 13568:2002 ed.).
- Jennings, N. R. (1993). Specification and Implementation of a Belief-Desire-Joint-Intention Architecture for Collaborative Problem Solving. *Intl. Journal of Intelligent and Cooperative Information Systems*, 2 (3), 289–318.
- Jones, R. M., Lebiere, C. & Crossman, J. A. (2007). Comparing modeling idioms in ACT-R and Soar. In A. Arbor (Hrsg.), *Proceedings of the 8th International Conference on Cognitive Modeling*. New York, NY, USA: Taylor & Francis.
- Jung, B. & Wachsmuth, I. (1995). Situierete Instruktionsverarbeitung im CODY Virtuellen Konstrukteur. In *KI-95 Activities: Workshops, Posters, Demos* (S. 332–334). Bonn: Gesellschaft für Informatik.
- Jung, B. & Wachsmuth, I. (1998). Integration of Geometric and Conceptual Reasoning for Interacting with Virtual Environments. In *Proc. AAAI'98 Spring Symposium on Multimodal Reasoning* (S. 22–27).
- Kane, M. J., Conway, A. R. A., Hambrick, D. Z. & Engle, R. W. (2007). Variation in working-memory capacity as variation in executive attention and control. In A. R. A. Conway, C. Jarrold, M. J. Kane, A. Miyake & J. Towse (Hrsg.), *Variation in Working Memory* (S. 21–48). USA: Oxford University Press.
- Kaplan, F. & Hafner, V. (2006). The challenges of joint attention. *Interaction Studies*, 7(2), 135-169.
- Kautz, H. & Allen, J. (1986). Generalized plan recognition. In *Proceedings of the Fifth National Conference on Artificial Intelligence* (S. 32-37). Menlo Park, California, USA: AAAI Press.
- Kendon, A. (1967). Some functions of gaze direction in social interaction. *Acta Psychologica*, 26, 1-47.
- Kieras, D. E., Meyer, D. E., Mueller, S. & Seymour, T. (1999). Insights into working memory from the perspectives of the EPIC architecture for modeling skilled perceptual-motor and cognitive human performance. In A. Miyake & P. Shah (Hrsg.), *Models of Working Memory, Mechanisms of Active Maintenance and Executive Control* (S. 183–223). Cambridge University Press.

- Kim, Y., Hill, R. & Traum, D. (2005). Controlling the focus of perceptual attention in embodied conversational agents. In *Proceedings AAMAS 2005* (S. 1097–1098). Utrecht, The Netherlands.
- Kinny, D., Sonenberg, E., Ljungberg, M., Tidhar, G., Rao, A. & Werner, E. (1994). Planned team activity. *Artificial Social Systems*, 227–256.
- Kluwe, R. H. (2000). Steuerung des Denkens und Handelns. *Zeitschrift für Psychologie*, 208 (1-2), 1-31.
- Kopp, S., Allwood, J., Ahlsen, E., Grammer, K. & Stocksmeier, T. (2008). Modeling embodied feedback in a virtual human. In I. Wachsmuth & G. Knoblich (Hrsg.), *Modeling Communication With Robots And Virtual Humans* (S. 18–37). Berlin: Springer-Verlag.
- Kopp, S., Jung, B., Leßmann, N. & Wachsmuth, I. (2003). Max - A Multimodal Assistant in Virtual Reality Construction. *KI-Künstliche Intelligenz*, 4/03, 11-17.
- Kopp, S. & Wachsmuth, I. (2000). A Knowledge-based Approach for Lifelike Gesture Animation. In W. Horn (Hrsg.), *ECAI 2000 Proceedings of the 14th European Conference on Artificial Intelligence* (S. 661-667). IOS Press, Amsterdam.
- Kopp, S. & Wachsmuth, I. (2004). Synthesizing multimodal utterances for conversational agents. *Computer Animation and Virtual Worlds*, 15 (1), 39-52.
- Langley, P. (2006). Cognitive architectures and general intelligent systems. *AI Magazin*, 27, 33–44.
- Langley, P. & Choi, D. (2006). A unified cognitive architecture for physical agents. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence* (S. 1469–1474). Boston, Massachusetts: AAAI Press.
- Larsson, S., Ljunglöf, P., Cooper, R., Engdahl, E. & Ericsson, S. (2000). GoDiS - An accommodating dialogue system. In *Proceedings of NAACL Workshop on Conversational Systems* (S. 7-10). Seattle: ACL.
- Larsson, S. & Traum, D. (2000). Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. In *Natural Language Engineering Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering* (S. 323–340). U.K: Cambridge University Press.

- Latoschik, M. E. (2001). A gesture processing framework for multimodal interaction in virtual reality. In *Proceedings of the 1st International Conference on Computer Graphics, Virtual Reality and Visualisation* (S. 95–100). New York, NY, USA: ACM.
- Lee, J., Huber, M. J., Kenny, P. G. & Durfee, E. H. (1994). UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications. In *Conference on Intelligent Robotics in Field, Factory, Service, and Space (CIRFFSS)* (S. 842–849). Reston, VA, USA: American Institute of Aeronautics and Astronautics, Inc.
- Lehman, J. F., Laird, J. & Rosenbloom, P. (1996). A gentle introduction to Soar, an architecture for human cognition. In D. Scarborough & S. Sternberg (Hrsg.), *Invitation to Cognitive Science*. MIT Press.
- Leßmann, N. & Kopp, S. (2005). Engagement in Collaborative Construction Tasks with Max. In *AAMAS 2005 Workshop Proceedings: Creating Bonds with ECAs*. New York, NY, USA: ACM Press.
- Leßmann, N., Kopp, S. & Wachsmuth, I. (2006). Situated interaction with a virtual human - perception, action, and cognition. In G. Rickheit & I. Wachsmuth (Hrsg.), *Situated Communication* (S. 287-323). Berlin: Mouton de Gruyter.
- Leßmann, N., Kranstedt, A. & Wachsmuth, I. (2004). Towards a Cognitively Motivated Processing of Turn-Taking Signals for the Embodied Conversational Agent Max. In *AAMAS 2004 Workshop Proceedings: Embodied Conversational Agents: Balanced Perception and Action*. New York, NY, USA: ACM Press.
- Leßmann, N. & Wachsmuth, I. (2003). A Cognitively Motivated Architecture for an Anthropomorphic Artificial Communicator. In *Proceedings of the Fifth International Conference on Cognitive Modeling (ICCM-5 2003), Bamberg, April 2003* (S. 277–278).
- Lemon, O., Bracy, A., Gruenstein, A. & Peters, S. (2001). Information states in a multi-modal dialogue system for human-robot conversation. In *Proceedings of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue (Bi-Dialog), Bielefeld: ZiF* (S. 57-67).
- Lesh, N., Rich, C. & Sidner, C. L. (1999). Using plan recognition in human-computer collaboration. In *Proceedings of the Seventh International Conference on User Modeling* (S. 23–32). Secaucus, NJ, USA: Springer-Verlag.
- Lochbaum, K. E. (1995). *Using collaborative plans to model the intentional structure of discourse*. PhD thesis, Harvard University, Cambridge, MA, USA.

- Lochbaum, K. E. (1998). A collaborative planning model of intentional structure. *Comput. Linguist.*, 24 (4), 525–572.
- Lochbaum, K. E., Grosz, B. J. & Sidner, C. (2000). Discourse structure and intention recognition. In R. Dale, G. Moisl & H. Somers (Hrsg.), *Handbook of Natural Language Processing* (S. 123–146). New York, Marcel Dekker.
- Logie, R. H. (1995). *Visuo-Spatial Working Memory*. Lawrence Erlbaum, Hove, Sussex.
- Lorini, E., Tummolini, L. & Herzig, A. (2005). Establishing Mutual Beliefs by Joint Attention: Towards a Formal Model of Public Events. In *Proceedings of CogSci05, Stresa 21-23 July, 2005*.
- Luu, P. & Tucker, D. (2004). Self-regulation by the medial frontal cortex: limbic representation of motive set-points. In M. Beauregard (Hrsg.), *Consciousness, Emotional Selfregulation and the Brain* (S. 123–161). Amsterdam: John Benjamin.
- Maes, P. (1989). How to do the right thing. *Connection Science*, 1, 291-323.
- Marsella, S. & Gratch, J. (2001). Modeling the Interplay of Emotions and Plans in Multi-Agent Simulations. In *Proceedings of 23rd Annual Conf. of the Cognitive Science Society*. Edinburgh, Scotland.
- McKoon, G. & Ratcliff, R. (1992). Spreading Activation Versus Compound Cue Accounts of Priming: Mediated Priming Revisited. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18 (6), 1155-1172.
- McNeill, D. (1992). *Hand and Mind: What Gestures Reveal about Thought*. Chicago: University of Chicago Press.
- McTear, M. (2002). Spoken dialogue technology: Enabling the conversational user interface. *ACM Computing Surveys*, 34 (1), 90-169.
- Metzinger, T. (1999). *Subjekt und Selbstmodell. Die Perspektivität phänomenalen Bewußtseins vor dem Hintergrund einer naturalistischen Theorie mentaler Repräsentation*. Paderborn: mentis.
- Miller, E. & Cohen, J. D. (2001). An Integrative Theory of Prefrontal Cortex Function. *Annual Reviews Neuroscience*, 24, 167-202.
- Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 101 (2), 343-352.

- Minsky, M. (1985). *The Society of Mind*. New York, NY, USA: Simon & Schuster Paperbacks.
- Miyake, A. & Shah, P. (Hrsg.). (1999). *Models of Working Memory, Mechanisms of Active Maintenance and Executive Control*. Cambridge, UK: Cambridge University Press.
- Móra, M. C., Lopes, J. G., Viccari, R. M. & Coelho, H. (1999). BDI models and systems: Reducing the gap. In *Proc. of the 5th International Workshop of Intelligent Agents* (S. 11–27). Heidelberg: Springer-Verlag.
- Muller, T. J., Heuvelink, A. & Both, F. (2008). Implementing a Cognitive Model in Soar and ACT-R: A Comparison. In Jung, Michel, Ricci & Petta (Hrsg.), *AT2AI-6 Working Notes, From Agent Theory to Agent Implementation, 6th Int. Workshop, May 13, 2008, AAMAS 2008, Estoril, Portugal, EU*. New York, NY, USA: ACM Press.
- Myers, K. L. & Wilkins, D. E. (1997). *The Act Formalism, Version 2.2*. (Bericht). Menlo Park, CA, USA: SRI International Artificial Intelligence Center.
- Nadel, J. & Camaioni, L. (1993). *New perspectives in early communicative development*. London: Routledge.
- Nakano, Y. I. & Nishida, T. (2005a). Awareness of Perceived World and Conversational Engagement by Conversational Agents. In *Proceedings of the AISB 2005 Symposium on Conversational Informatics for Supporting Social Intelligence & Interaction – Situational and environmental information enforcing involvement in conversation*. Hatfield, UK: AISB.
- Nakano, Y. I. & Nishida, T. (2005b). Interpreting User's Attention as Engagement Cues in Multiparty Immersive Conversations. In *CHI 2005 Workshop: CHI Virtuality 2005*.
- Nakano, Y. I., Reinstein, G., Stocky, T. & Cassell, J. (2003). Towards a Model of Face-to-Face Grounding. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL03), (Sapporo, Japan)*. ACL.
- Nason, S. & Laird, J. (2005). Soar-RL: Integrating reinforcement learning with Soar. *Cognitive Systems Research*, 6 (1), 51–59.
- Newell, A. (1982). The knowledge level. *Artificial Intelligence*, 18 (1), 1-20.
- Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA, USA: Harvard University Press.

- Newell, A. & Simon, H. A. (1961). GPS, A Program that Simulates Human Thought. In H. Billing (Hrsg.), *Lernende Automaten* (S. 109-124). München: Oldenbourg.
- Nguyen, M. & Wobcke, W. (2005). A Flexible Framework for SharedPlans. *Proceedings of the Australian Undergraduate Students' Computing Conference 2005*.
- Novick, D. & Sutton, S. (1997, March 24-26). What is mixed-initiative interaction? In *Papers from the 1997 AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction, Technical Report SS-97-04*. AAAI Press.
- Nuku, P. & Bekkering, H. (2008). Joint attention: Inferring what others perceive (and don't perceive). *Conscious and Cognition*, 17 (1), 339-349.
- Nuxoll, A. M. & Laird, J. E. (2007). Extending Cognitive Architecture with Episodic Memory. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI)* (S. 1560–1565). Vancouver, British Columbia, Canada: AAAI Press.
- Oberauer, K. (2001). Removing Irrelevant Information from Working Memory: A Cognitive Aging Study with the Modified Sternberg Task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27, 948-957.
- Oberauer, K. (2002). Access to Information in Working Memory: Exploring the Focus of Attention. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28 (3), 411–421.
- Oberauer, K. (2003). The Multiple Faces of Working Memory: Storage, Processing, Supervision, and Coordination. *Intelligence*, 31, 167-193.
- Oberauer, K., Süß, H. M., Schulze, R., Wilhelm, O. & Wittmann, W. (2000). Working memory capacity - facets of a cognitive ability construct. *Personality and Individual Differences*, 29, 1017–1045.
- Ogasawara, Y., Okamoto, M., Nakano, Y. & Nishida, T. (2005). Establishing natural communication environment between a human and a listener robot. *Proc. Symposium on Conversational Informatics for Supporting Social Intelligence and Interaction, AISB*, 42-51.
- Oliveira, E. C. & Mouta, F. A. (1993). A distributed AI architecture enabling multi-agent cooperation. In *IEA/AIE'93: Proceedings of the 6th international conference on Industrial and engineering applications of artificial intelligence and expert systems* (S. 136–143). Gordon & Breach Science Publishers.

- Oliveira, W. & Barros, R. (1998). The real numbers in Z. In A. Evans (Hrsg.), *2nd BCS-FACS Northern Formal Methods Workshop. Electronic Workshop in Computing*. Berlin: Springer-Verlag.
- O'Reilly, R. C., Braver, T. S. & Cohen, J. (1999). A biologically based computational model of working memory. In A. Miyake & P. Shah (Hrsg.), *Models of Working Memory, Mechanisms of Active Maintenance and Executive Control* (S. 375–411). Cambridge University Press.
- Pfeiffer, T. (2008). Towards Gaze Interaction in Immersive Virtual Reality. In *Virtuelle und Erweiterte Realität - Fünfter VR/AR Workshop* (S. 81–92). Shaker Verlag.
- Pfeiffer, T. & Latoschik, M. E. (2004). Resolving Object References in Multimodal Dialogues for Immersive Virtual Environments. In Y. I. et al. (Hrsg.), *Proceedings of the IEEE Virtual Reality 2004*. Illinois: Chicago.
- Pfeiffer-Leßmann, N. & Wachsmuth, I. (2008). Toward alignment with a virtual human - Achieving joint attention. In A. Dengel, K. Berns & T. Breuel (Hrsg.), *KI 2008: Advances in Artificial Intelligence*. Berlin: Springer-Verlag.
- Pfeiffer-Leßmann, N. & Wachsmuth, I. (2009). Formalizing joint attention in cooperative interaction with a virtual human. In B. Mertsching, M. Hund & Z. Aziz (Hrsg.), *KI 2009: Advances in Artificial Intelligence* (S. 540–547). Berlin: Springer.
- Pickering, M. J. & Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Science*, 27, 169-226.
- Poggi, I. & Pelachaud, C. (2000). Performative facial expressions in animated faces. In J. Cassell, J. Sullivan, S. Prevost & E. Churchill (Hrsg.), *Embodied Conversational Agents* (S. 155–188). Cambridge, MA, USA: MIT Press.
- Pokahr, A., Braubach, L. & Lamersdorf, W. (2005a). A Goal Deliberation Strategy for BDI Agent Systems. In T. Eymann, F. Klügel, W. Lamersdorf, M. Klusch & M. N. Huhns (Hrsg.), *Third German conference on Multi-Agent System TEchnologieS (MATES-2005)* (S. 82–93). Berlin Heidelberg: Springer-Verlag.
- Pokahr, A., Braubach, L. & Lamersdorf, W. (2005b). Jadex: A BDI Reasoning Engine. In R. Bordini, M. Dastani, J. Dix & A. E. F. Seghrouchni (Hrsg.), *Multi-Agent Programming* (S. 149-174). USA: Springer Science+Business Media Inc.

- Pollack, M. (1990). Plans as Complex Mental Attitudes. In P. R. Cohen, J. Morgan & M. Pollack (Hrsg.), *Intentions in Communication* (S. 77–103). Cambridge, Massachusetts: MIT Press.
- Pollack, M. (1992). The Uses of Plans. *Artificial Intelligence*, 57 (1), 43-68.
- Posner, M. & Petersen, S. (1990). The attention system of the human brain. *Annu. Rev. Neurosci.*, 13, 25-42.
- Rao, A. & Georgeff, M. (1991). Modeling rational behavior within a BDI-architecture. In *Proceedings Intl. Conference on Principles of Knowledge Representation and Planning* (S. 473-484).
- Rao, A. & Georgeff, M. (1995). BDI agents: from theory to practice. In *Proceedings of the First Intl. Congress on Multi-Agent Systems (ICMAS-95)* (S. 312-319). San Francisco, California, USA: MIT Press.
- Rao, A. & Georgeff, M. (1998). Decision procedures of BDI logics. In *Journal of Logic and Computation* (Bd. 8, S. 293-344).
- Reiter, E. & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.
- Rich, C. & Sidner, C. L. (1998). COLLAGEN: A Collaboration Manager for Software Interface Agents. *User Modeling and User-Adapted Interaction*, 8 (3-4), 315–350.
- Rich, C., Sidner, C. L. & Lesh, N. (2001). COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. *AI Magazine, Special Issue on Intelligent User Interfaces*, 22 (4), 15-25.
- Rickel, J. & Johnson, W. (2000). Task-Oriented Collaboration with Embodied Agents in Virtual Worlds. In J. Cassell, J. Sullivan, S. Prevost & E. Churchill (Hrsg.), *Embodied Conversational Agents*. Boston: MIT Press.
- Rickel, J., Marsella, S., Gratch, J., Hill, R., Traum, D. & Swartout, W. (2002). Toward a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems*, 17.
- Rickheit, G. & Wachsmuth, I. (1996). Collaborative Research Centre Situated Artificial Communicators at the University of Bielefeld. *Artificial Intelligence Review*, 10, 165-170.
- Rosenbloom, P., Laird, J. & Newell, A. (1993). *The Soar Papers: Research on Integrated Intelligence*. Cambridge, MA, USA: MIT Press.

- Russell, S. & Norvig, P. (2003). *Artificial Intelligence - A Modern Approach (second edition)*. Upper Saddle River, NJ, USA: Prentice Halls Series in Artificial Intelligence.
- Sacks, H., Schegloff, E. & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language*, 50, 696-735.
- Salovey, P., Mayer, J. D. & Caruso, D. (2002). The positive psychology of emotional intelligence. In C. R. Snyder & S. J. Lopez (Hrsg.), *Handbook of Positive Psychology* (S. 159-171). Oxford, UK: Oxford University Press.
- Sardina, S., Silva, L. P. de & Padgham, L. (2006, May). Hierarchical Planning in BDI Agent Programming Languages: A Formal Approach. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)* (S. 1001–1008). Hakodate, Japan: ACM Press.
- Schuette, M. (2004). Representation (mental). In *Encyclopedic Reference of Neuroscience*. Berlin: Springer-Verlag.
- Schultheis, H., Barkowsky, T. & Bertel, S. (2006). LTM-C — An Improved Long-Term Memory for Cognitive Architectures. In D. Fum, F. del Missier & A. Stocco (Hrsg.), *Proceedings of the 7th International Conference on Cognitive Modeling, Trieste (ICCM 2006)* (S. 274–279). Trieste: Edizioni Goliardiche.
- Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. New York: Cambridge University Press.
- Searle, J. (1987). *Intentionalität*. Berlin: Suhrkamp.
- Searle, J. (1990). Collective intentions and actions. In P. R. Cohen, J. Morgan & M. E. Pollak (Hrsg.), *Intentions in Communication* (S. 401-415). Cambridge, MA, USA: MIT Press.
- Searle, J. (1995). *The construction of social reality*. New York: Free Press.
- Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.
- Sidner, C. & Lee, C. (2003). *An Architecture Engagement in Collaborative Conversations between a Robot and Humans* (Bericht). Cambridge, MA, USA: Mitsubishi Electric Research Labs TR2003-13.
- Sidner, C., Lee, C. & Kidd, C. (2005). Engagement During Dialogues with Robots. In *AAAI Spring Symposia 2005*.

- Sidner, C., Lee, C. & Lesh, N. (2003a). Engagement By Looking: Behaviors for Robots When Collaborating with People. In Kruiff-Korbayova & Kosny (Hrsg.), *DiaBruck: The Proceedings of the Seventh Workshop on the Semantics and Pragmatics of Dialogue* (S. 123-130). Germany: University of Saarland.
- Sidner, C., Lee, C. & Lesh, N. (2003b). Engagement Rules for Human-Robot Collaborative Interactions. In *IEEE International Conference on Systems, Man & Cybernetics (CSMC)* (Bd. 4, S. 3957-3962). IEEE Press.
- Singh, M., Rao, A. & George, M. (1999). Formal Methods in DAI: Logic-based representations and reasoning. In G. Weiss (Hrsg.), *Multiagent Systems, a Modern Approach to Distributed Artificial Intelligence* (S. 331-377). MIT Press, Cambridge MA., USA.
- Slovan, A. (1987). Motives, mechanisms and emotions. *Cognition and Emotion*, 1, 217-234.
- Smith, R. G. & Davis, R. (1988). Frameworks for cooperation in distributed problem solving. In (S. 61-70). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Spencer, P. (2000). Looking without listing: Is audition a prerequisite for normal development of visual attention during infancy? *Journal of Deaf Studies and Deaf Education*, 5, 291-302.
- Stanford Encyclopedia of Philosophy*. (2009). Stanford, CA, USA: The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University.
- Strayer, S. & Heeman, P. (2001). Reconciling initiative and discourse structure. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue* (S. 153-161). Stroudsburg, PA, USA: ACL.
- Strayer, S., Heeman, P. & Yang, F. (2003). Reconciling control and discourse structure. In J. van Kuppevelt & R. Smith (Hrsg.), *Current and New Directions in Discourse and Dialogue* (S. 305-323). The Netherlands: Kluwer Academic Publishers.
- Strube, G., Becker, B., Freksa, C., Hahn, U., Opwis, K. & Palm, G. (Hrsg.). (1996). *Wörterbuch der Kognitionswissenschaft*. Stuttgart: Klett-Cotta.
- Sun, R. (2004). Desiderata for cognitive architectures. *Philosophical Psychology*, 17.

- Swartout, W. R., Gratch, J., Jr., R. W. H., Hovy, E. H., Marsella, S., Rickel, J. et al. (2006). Toward Virtual Humans. *AI Magazine*, 27 (2), 96-108.
- Tambe, M. (1997). Towards Flexible Teamwork. *Journal of Artificial Intelligence Research*, 7, 83–124.
- Tasker, S. L. & Schmidt, L. A. (2008). The dual usage problem in the explanations of joint attention and children's socioemotional development: A reconceptualization. *Developmental Review*, 28 (3), 263–288.
- Terveen, L. (1995). Overview of human-computer collaboration. *Knowledge-Based Systems*, 8 (2-3), 67–81.
- Thangarajah, J., Padgham, L. & Harland, J. (2002). Representation and reasoning for goals in BDI agents. *Aust. Comput. Sci. Commun.*, 24 (1), 259–265.
- Thangarajah, J., Padgham, L. & Winikoff, M. (2003a). Detecting & Avoiding Interference Between Goals in Intelligent Agents. In *IJCAI* (S. 721-726).
- Thangarajah, J., Padgham, L. & Winikoff, M. (2003b). Detecting & exploiting positive goal interaction in intelligent agents. In *AAMAS '03: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems* (S. 401–408). New York, NY, USA: ACM Press.
- Thangarajah, J., Winikoff, M., Padgham, L. & Fischer, K. (2002). Avoiding Resource Conflicts in Intelligent Agents. In van Harmelen (Hrsg.), *Proceedings of the 15th European Conference on Artificial Intelligence 2002 (ECAI 2002), Lyon, France*.
- Thomason, R. H. (2000). Desires and Defaults: A Framework for Planning with Inferred Goals. In A. G. Cohn, F. Giunchiglia & B. Selman (Hrsg.), *KR2000: Principles of Knowledge Representation and Reasoning* (S. 702–713). San Francisco: Morgan Kaufmann.
- Thomason, R. H. & Horty, J. F. (1996). Nondeterministic action and dominance: foundations for planning and qualitative decision. In *TARK '96: Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge* (S. 229–250). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Thorisson, K. R. (1997). Gandalf: An Embodied Humanoid Capable of Real-Time Multimodal Dialogue with People. In *First ACM International Conference on Autonomous Agents* (S. 536-537). New York, NY, USA: ACM Press.

- Thorisson, K. R. (2002). Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perception to Action. In I. K. B. Granström D. House (Hrsg.), *Multimodality in Language and Speech Systems* (S. 173-207). The Netherlands: Kluwer Academic Publishers.
- Tomasello, M. (1995). Joint attention as social cognition. In C. Moore & P. Dunham (Hrsg.), *Joint Attention: Its origin and role in development* (S. 103-128). L. Erlbaum.
- Tomasello, M. & Carpenter, M. (2007). Shared intentionality. *Developmental Science*, 10 (1), 121-125.
- Tomasello, M., Carpenter, M., Call, J., Behne, T. & Moll, H. (2005). Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, 28, 675-691.
- Traum, D. (1996). Conversational agency: the TRAINS-93 dialogue manager. In *Proceedings of the 11th Workshop on Language Technology: Dialogue Management in Natural Language Systems*. Enschede, The Netherlands: Universiteit Twente.
- Traum, D. & Allen, J. (1994). Discourse obligations in dialogue processing. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (S. 1-8). Morristown, NJ, USA: Association for Computational Linguistics.
- Traum, D. & Larsson, S. (2003). The Information State Approach to Dialogue Management. In R. Smith & J. Kuppevelt (Hrsg.), *Current and New Directions in Dialogue* (S. 325-354). The Netherlands: Kluwer Academic Press.
- Traum, D. & Rickel, J. (2002). Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2* (S. 766-773). New York, NY, USA: ACM.
- Traum, D., Swartout, W., Gratch, J. & Marsella, S. (2005). Virtual Humans for Non-team Interaction Training. In *Autonomous Agents and Multi-Agent Systems (AAMAS) Workshop on Creating Bonds with Humanoids*.
- Triesch, J. (2006). Modeling the emergence of gaze following. *KI - Künstliche Intelligenz*, 3, 27-30.
- Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Hrsg.), *Organisation of memory* (S. 381-403). New York: Academic Press.

- Tulving, E. & Donaldson, W. (1972). *Organization of memory*. New York: Academic Press.
- Tulving, E. & Schacter, D. (1990). Priming and human memory systems. *Science*, 247 (4940), 301–306.
- Tuomela, R. (1995). *The importance of us*. Stanford, CA, USA: Stanford University Press.
- Tuomela, R. (2000). Collective and Joint Intention. *Mind and Society*, 1 (2).
- Wachsmuth, I. (1985). LAKOS - Ein Modell der Wissensrepräsentation zur Erklärung kognitiven Verhaltens. In H. Mandl & P. M. Fischer (Hrsg.), *Lernen im Dialog mit dem Computer* (S. 24-39). München: Urban und Schwarzenberg.
- Wachsmuth, I. (1987). *On Structuring Domain-Specific Knowledge* (Bericht). LILOG-Report 12, IBM Deutschland GmbH, WT LILOG, Dept 3504.
- Wachsmuth, I. (1988). Modeling the knowledge base of mathematics learners: Situation-specific and situation-nonspecific knowledge. In H. Mandl & A. Lesgold (Hrsg.), *Learning Issues for Intelligent Tutoring Systems* (S. 63-79). New York: Springer-Verlag.
- Wachsmuth, I. & Gängler, B. (1991). Knowledge packets and knowledge packet structures. In O. Herzog & C.-R. Rollinger (Hrsg.), *Text Understanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence* (S. 380-393). Berlin Heidelberg: Springer (LNAI 546).
- Wachsmuth, I. & Leßmann, N. (2002). Eine kognitiv motivierte Architektur für einen anthropomorphen Künstlichen Kommunikator. In *Tagungsbeiträge: Human Centered Robotic Systems 2002* (S. 141-148).
- Walker, M. A. & Whittaker, S. (1990). Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics* (S. 70-78). Stroudsburg, PA, USA: ACL.
- Whittaker, S. & Stenton, P. (1988). Cues and Control in Expert Client Dialogues. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics* (S. 123-130). Stroudsburg, PA, USA: ACL.
- Wilkins, D. E. & Myers, K. L. (1995). A Common Knowledge Representation for Plan Generation and Reactive Execution. *Journal of Logic and Computation*, 5 (6), 731 - 761.

- Winikoff, M., Padgham, L., Harland, J. & Thangarajah, J. (2002). Declarative and Procedural Goals in Intelligent Agent Systems. In *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002)*. Boston, MA, USA: Morgan Kaufmann.
- Wooldridge, M. (2000). *Reasoning about Rational Agents*. Cambridge, MA, USA: MIT Press.
- Wooldridge, M. (2002). *An Introduction to Multiagent Systems*. Chichester, UK: John Wiley & Sons.
- Wooldridge, M. & Jennings, N. R. (1994). *Formalizing the Cooperative Problem Solving Process*. Lake Quinhalt, WA, USA.
- Wooldridge, M. & Jennings, N. R. (1999). The Cooperative Problem-solving Process. *Journal of Logic and Computation*, 9 (4), 563-592.
- Zimbardo, P. G. (1995). *Psychologie* (S. Hoppe-Graff, B. Keller & I. Engel, Hrsg.). Springer-Verlag.



# Anhang A

## Formale Spezifikationen

### A.1 Modallogik

Der Formalismus, der in der vorliegenden Arbeit eingesetzt wird, basiert auf der doxastischen Modallogik KD45. Um Ziele und Annahmen spezifizieren zu können, werden im Einklang mit Rao & Georgeff (1991) drei modale Operatoren eingesetzt: BEL, GOAL und INTEND.

**Definition 19** *Jede Formel der Prädiaktenlogik erster Stufe ist eine Zustandsformel. Wenn  $\varphi_1$  und  $\varphi_2$  Zustandsformeln sind, so auch  $\neg\varphi_1$  und  $\varphi_1 \vee \varphi_2$  und  $\varphi_1 \wedge \varphi_2$ . Wenn  $\varphi$  eine Formel ist, dann ist auch  $BEL(\varphi)$  eine Zustandsformel. Diese Formelbezeichnungen bezeichnen wir als Belief-Formel. Sei  $i$  ein Agent, dann ist  $(BEL_i \varphi)$  eine Abkürzung, welche besagt, dass Agent  $i$  eine Formel  $\varphi$  glaubt (Lorini et al., 2005).*

Um Aussagen über das Verhalten und die daraus resultierenden Beliefs eines Agenten machen zu können, wird folgende Handlungsdefinition in der CASEC Architektur verwendet:

**Definition 20** *Sei  $ACT$  die Menge aller atomaren, pragmatischen Handlungen.  $\{Act(\varphi)\}_i$  steht dafür, dass Agent  $i$  die Handlung  $Act$  mit den Parametern, welche durch  $\varphi$  spezifiziert werden ausführt. Die geschweiften Klammern zeigen an, dass es sich bei  $Act$  weder um ein Prädikat noch um einen Modaloperator handelt sondern um eine Handlung. Sei  $ACT^+$  die Menge aller Handlungen eines Agenten bestehend aus  $ACT$  und Sequenzen der Elemente aus  $ACT$  ( $ACT \subseteq ACT^+$ ).*

Zusätzlich zu den bereits beschriebenen Modaloperatoren, folgt die vorliegende Arbeit P. Cohen & Levesque (1990) indem HAPPENS und DONE zu den atomaren Modaloperatoren hinzugefügt werden.

**Definition 21** Sei  $\alpha$  eine Handlung ( $\alpha \in ACT$ ) dann bedeutet  $(HAPPENS\alpha)$  dass die Handlung  $\alpha$  als nächstes passieren wird.  $(DONE\alpha)$  bedeutet dass Handlung  $\alpha$  gerade passiert ist. Diese zeitlichen Basisoperatoren werden durch den Operator  $;$  für Sequenzen von Handlungen erweitert.  $(\alpha; \beta)$  bedeutet, dass erst Handlung  $\alpha$  und dann Handlung  $\beta$  ausgeführt wird.

**Definition 22** Sei  $\vartheta \in PROPOSITION$  oder  $\vartheta \in OBJECTS$  dann bedeutet  $(ATT_i(\vartheta))$ , dass der Aufmerksamkeitsfokus des Agenten  $i$  auf die Proposition  $\vartheta$  oder das Objekt  $\vartheta$  ausgerichtet ist.  $ATT$  wird zu den Modaloperatoren hinzugefügt.

In der CASEC Architektur sind epistemische perzeptuelle Handlungen anwendbar auf alle Formeln (sowohl Propositionen als auch Handlungen). Sie erlauben aber keinen direkten Zugriff auf mentale Zustände. Ein Agent kann jedoch overt Verhaltenweisen seines Gegenübers sowie Propositionen von Objekten wahrnehmen (Lorini et al., 2005). In der vorliegenden Arbeit werden die epistemischen Handlungen *Perceive-that* und *Test-If* unterstützt:

**Definition 23 *Perceive-that*:** Sei  $P \in ACT, \vartheta$  eine Zustandsformel.  $P$  wird als Akt des Wahrnehmens von  $\vartheta$  in der Umwelt definiert.  $\{P(\vartheta)\}_j$  (der Agent  $j$  hat  $\vartheta$  wahrgenommen). Auswirkung auf die Beliefs des Agenten:  $\{P(\vartheta)\}_j \Rightarrow (BEL_j \vartheta)$

Der Agent glaubt zunächst alles, was er wahrgenommen hat. In einem weiteren Schritt kann ein solcher Belief jedoch auch revidiert werden.  $\vartheta$  muss nicht zwangsläufig in der Welt wahr sein.

**Definition 24 *Test-if*:** Sei  $T \in ACT, \vartheta$  eine Zustandsformel.  $\{T(\vartheta)\}_j$  wird als Akt des Testens/Überprüfens eines Zustands  $\vartheta$  ausgeführt von Agent  $j$  definiert. *Test-if* Akte zeichnen sich durch ein overt, wahrnehmbares Verhalten aus und können von anderen Agenten wahrgenommen und überprüft werden.

*Test-if* Handlungen sind Wegbereiter von *Perceive-that* Handlungen.

**Definition 25** Sei  $\varphi$  ein Belief-Chunk, dann ist auch  $\varphi_{[t1]}$ , ( $t1 = \text{Point in time}$ ) ein Belief-Chunk. Die Funktion  $Time(\varphi) = t1$  liefert den Entstehungszeitpunkt des Belief-Chunks zurück.

Neben der Erweiterung der Belief-Chunks um Zeitlichkeit werden auch Handlungen um einen Zeitstempel erweitert, aus welchem sich dann zeitliche Beziehungen zwischen Handlungen ableiten lassen.

**Definition 26** Sei  $\alpha \in ACT$  eine Handlung, dann bezeichnet  $Begin(\alpha) := t_{begin}$ , ( $t_{begin} = \text{Point in Time}$ ) den Zeitpunkt zu dem die Handlung  $\alpha$  beginnt,  $End(\alpha) := t_{end}$ , ( $t_{end} = \text{Point in Time}$ ) bezeichnet den Zeitpunkt zu dem die Handlung  $\alpha$  endet. Die Dauer der Handlung besteht aus  $Dur(\alpha) := End(\alpha) - Begin(\alpha)$ .  $(\alpha)_{j[t_{begin}, t_{end}]}$  dient als Schreibweise, um die Zeitpunkte des Beginns und Endes der Handlung  $\alpha$  ausgeführt von Agent  $j$  zu beschreiben.

Proposition	Bedeutung
(facing $j$ )	Belief des Agenten, dass er dem Agenten $j$ gegenüber steht, wird assertiert, wenn der Agent das erste Mal den Interlokutor $j$ bewusst wahrnimmt
(conversationWith $j$ )	Belief des Agenten, welcher assertiert wird, wenn der Agent in eine Konversation mit Agent $j$ involviert ist (es wurde mindestens jeweils ein kommunikativer Akt ausgetauscht)
(convState $j$ $s$ )	gibt den angenommenen Konversationalen Zustand $s$ in Bezug auf Agent $j$ an,
$s \in [noConv, myTurn, othersTurn, gap, overlap]$	

Tabelle A.1: Beschreibung Prädikate im Rahmen des Kontaktmanagements

Handlung	Bedeutung
$\{Greet(j)\}_i$	Begrüßung des Agenten $i$ gerichtet an Agent $j$ („Hallo“ sagen, Winken oder Ähnliches)
$\{SeeOf(j)\}_i$	Verabschiedung des Agenten $i$ gerichtet an Agent $j$ („Tschüss“ sagen, Winken oder Ähnliches)
$\{BeginConv(j)\}_i$	Beginnen einer Konversation (Begrüßen des Gegenübers, Einladen zur Konversation)
$\{EndConv(j)\}_i$	Beenden einer Konversation (Verabschieden des Gegenübers, Konversation abschließen)
$\{TakeTurn\}_i$	Turn-Taking Signal, Ergreifen des Turns (siehe 2.2.2 ausgeführt von Agent $i$ )
$\{GiveTurn\}_i$	Turn-Taking Signal, Zuweisen des Turns (siehe 2.2.2 ausgeführt von Agent $i$ )

Tabelle A.2: Beschreibung Handlungen im Rahmen des Kontaktmanagements

## A.2 Algorithmen

### A.2.1 Unifikationsalgorithmus

Der Unifikationsalgorithmus der CASEC-Architektur wird hier in Pseudocode konkret vorgestellt. Er ähnelt den Unifikationsalgorithmen von JAM und UMPRS, wurde jedoch beispielsweise in Bezug auf Constraints und verschaltete Beliefs entscheidend erweitert (siehe auch Abschnitt 5.1.7).

---

#### Algorithm 14 unify

```

1: procedure UNIFY(Relation* dstRel, Binding* dstB, Relation* srcRel, Binding* srcB)
   ▷ Unify
   Relations
2:   if (dstRel==0  $\wedge$  srcRel==0) then
3:     return 0
4:   else if (getRelName(dstRel)  $\neq$  getRelName(srcRel)) then
5:     return 0
6:   else if getArity(dstRel) == 0  $\wedge$  getArity(srcRel) == 0 then
7:     return 1
8:   else if (getArity(dstRel)  $\neq$  getArity(srcRel)) then
9:     return 0
10:  else if negated(dstRel)  $\neq$  negated(srcRel) then
11:    return 0
12:  end if
13:  while (dstRel and srcRel have further argument) do
14:    evaluate(srcRel, srcB)
15:    evaluate(dstRel, dstBcopy)
16:    dstArg  $\leftarrow$  nextElement(dstRel)
17:    srcArg  $\leftarrow$  nextElement(srcRel)
18:    if (dstArg.isValue()  $\wedge$  srcArg.isValue()) then
19:      if (dstArg == srcArg) then
20:        continue
21:      else
22:        return 0
23:      end if
24:    end if

```

---

---

```
25:     if (dstArg.isVariable()  $\wedge$  srcArg.isVariable()) then
26:         if (checkConstraints(dstArg, srcArg) == true) then
27:             link variables in bindings, adapt variable constraints
28:         continue
29:     else
30:         return 0
31:     end if
32: end if
33: if (dstArg.isVariable()  $\wedge$  srcArg.isValue()) then
34:     if (checkConstraints(dstArg, srcArg) == true) then
35:         set value in binding
36:     continue
37:     else
38:         return 0
39:     end if
40: end if
41: if (dstArg.isNested()  $\wedge$  srcArg.isNested()) then
42:     if (unify(dstArg, dstB, srcArg, srcB)) then
43:         continue
44:     else
45:         return 0
46:     end if
47: end if
48: end while
49: adapt bindings with unification
50: return 1
51: end procedure
```

---

## A.2.2 Match-Algorithmus

Der *Match*-Algorithmus setzt den zuvor vorgestellten Unifikationsalgorithmus ein. In Abschnitt 5.1.7 wird er anhand eines Beispiels vorgestellt.

---

**Algorithm 15** get\_matching\_belief()

---

```

1: procedure GET_MATCHING_BELIEF(Relation* rel, Binding* b)
2:   relation_list  $\leftarrow$  candidate set of working memory
3:   while more candidates in relation_list do
4:     candidate  $\leftarrow$  relation with next highest activation
5:     if candidate matches relation with binding then
6:       set last_retrieved in candidate
7:       activate candidate with  $\delta_{SMALL}$ 
8:       activate attributes of candidate with  $\delta_{SMALL}$ 
9:       return candidate
10:    else
11:      activate attributes of candidate with  $\delta_{SMALL}/10$ 
12:      continue
13:    end if
14:  end while
15:  return 0
16: end procedure

```

---

### A.2.3 Applicable-Plan-List Algorithmus

---

**Algorithm 16** APL

---

```

1: procedure APPLICABLEPLANLIST(KaTable* plans, WmTable* wm, Inten-
   tionStructure* is, int meta-level)
2:   generateWMbased_APL
3:   generateGoalBased_APL
4: end procedure

```

---



---

**Algorithm 17** genWMbasedSoak

---

```

1: procedure GENWMBASEDSOAK(KaTable* plans, WmTable* wm, Intenti-
   onStructure* is)
2:   if not anything new then
3:     return
4:   else
5:     check all Plans for Conclude-Plans
6:     loop through all WM entries matching the relation
7:     get all matching relations
8:     check whether they are new (flag)

```

---

---

```

9:         if there has been a match then
10:             if precondition and context condition are valid then
11:                 iterate through bindings in list
12:                 create temporary goal (add-unique)
13:             end if
14:         end if
15:     end if
16: end procedure

```

---



---

**Algorithm 18** genGoalbasedSoak

```

1: procedure GENGOALBASEDSOAK(KaTable* plans, WmTable* wm, Intenti-
   onStructure* is, int meta-level)
2:     go through each possible Stack
3:     move down to leaf node
4:     if  $\neg$  (generate_soak()) or goal is a Conclude-Goal then
5:         continue
6:     else
7:         if goal is a Maintain-Goal then
8:             if  $\neg$  (state of goal is true) then
9:                 post goal
10:            end if
11:        end if
12:        try to find possible plan
13:        if no plan found then
14:            try to activate plan from LTM
15:        else
16:            go through each possible plan
17:            filter according to :by and :not-by
18:            if  $\neg$  goal is_eligible then
19:                continue
20:            else
21:                go through each possible plan variable binding and unify goal
                and plan relations
22:                instantiate plan
23:            end if
24:        end if
25:    end if
26: end procedure

```

---