

Universität der Bundeswehr München  
Fakultät für Luft- und Raumfahrttechnik  
Institut für Systemdynamik und Flugmechanik

# **Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung**

Dipl.-Phys. Martin Pellkofer

Vollständiger Abdruck der von  
der Fakultät für Luft- und Raumfahrttechnik  
der Universität der Bundeswehr München  
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation

Vorsitzender:	Prof. Dr.-Ing. B. Katzy
1. Berichterstatter:	Prof. Dr.-Ing. E.-D. Dickmanns
2. Berichterstatter:	Prof. Dr.-Ing. A. Schulte

Diese Dissertation wurde am 8. Januar 2003 bei der Universität  
der Bundeswehr München, 85577 Neubiberg, Werner-Heisenberg-Weg 39  
eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik  
am 15. Januar 2003 angenommen.

Tag der Promotion: 6. Mai 2003



# Danke

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Systemdynamik und Flugmechanik (ISF) der Universität der Bundeswehr München.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. E.-D. Dickmanns, welcher mir die Möglichkeit gab, in einem faszinierenden Themengebiet wissenschaftlich zu arbeiten. Seine zahlreichen Anregungen und Hinweise und die ausgezeichneten Rahmenbedingungen am Institut waren die Grundlage für den Erfolg dieser Arbeit.

Des weiteren möchte ich Herrn Prof. Dr.-Ing. A. Schulte herzlich für die Übernahme des Korreferats und Herrn Prof. Dr.-Ing. B. Katzy für die freundliche Übernahme des Vorsitzes der Prüfungskommission danken.

Diese Arbeit wäre ohne die gute Zusammenarbeit meiner zahlreichen Kollegen nicht möglich gewesen. Deshalb schulde ich den Herrn Dr. Rudolf Gregor, Dr. Michael Lützeler, Ulrich Hofmann, Dr. Markus Maurer, Simon Fürst, Dr. Andreas Schubert, Stefan Baten und Volker von Holt recht herzlichen Dank. Insbesondere möchte ich meinen einstigen Zimmergenossen Herrn Dr. André Rieder nennen, von dem ich gut strukturierte und damit leicht erweiterbare Software ‚erben‘ durfte, und der sich die Zeit nahm, mir diese auch zu erklären. Ferner möchte ich Herrn Karl-Heinz Siedersberger für die Kooperation bei der Realisierung des Fähigkeitenkonzepts und für die hervorragende Ausstattung der Versuchsfahrzeuge danken.

Den Mitgliedern der Werkstatt, namentlich Herrn Josef Hollmayer, Eduard Oestereicher, Michael Bold und Richard Stömmer, sei für die schnelle und fachmännische Ausführung der Arbeiten an der Hardware mein Dank ausgesprochen. Ferner möchte ich Frau Madeleine Gabler nennen. Sie war die Quelle des guten Klimas am Institut und unterstützte mich bei der Beschaffung zahlreicher Literatur. Herrn Henrik Putzer danke ich für seine Hilfe am Videoschneideplatz, sowie allen anderen Mitarbeitern des Instituts für die kameradschaftliche Zusammenarbeit.

Schließlich möchte ich meinen Eltern danken, welche mir meinen Ausbildungsweg und damit auch diese Promotion ermöglichten. Meiner Schwester Andrea danke ich für die gewissenhafte Durchsicht der Arbeit. Mein ganz besonderer Dank geht an meine liebe Lebensgefährtin Melanie für ihre Unterstützung und ihr Verständnis, denn die Anfertigung dieser Arbeit ging oft zu Lasten der gemeinsamen Freizeit.

für Melanie und meine Eltern

# Inhalt

<b>Danke</b>	<b>III</b>
<b>Inhalt</b>	<b>V</b>
<b>Liste der verwendeten Abkürzungen</b>	<b>XI</b>
<b>Kurzfassung</b>	<b>XIII</b>
<b>Abstract</b>	<b>XIV</b>

## TEIL A: EINFÜHRUNG

<b>Kapitel 1: Einleitung</b>	<b>1</b>
<b>1.1 Terminologie</b>	<b>2</b>
1.1.1 Objekt – Subjekt	2
1.1.2 Fähigkeit – Aktion – und davon abgeleitete Begriffe	3
1.1.3 Szene – Situation – Situationsanalyse	4
1.1.4 Intention	4
1.1.5 Domäne	5
<b>1.2 Entwicklungshintergrund am ISF</b>	<b>5</b>
1.2.1 Der 4D-Ansatz	5
1.2.2 Das Sehsystem der 3. Generation: EMS-VISION	7
<b>1.3 Beitrag der Arbeit</b>	<b>8</b>
<b>1.4 Aufbau der Arbeit</b>	<b>10</b>
<b>Kapitel 2: Stand der Technik</b>	<b>11</b>
<b>2.1 Was ist ‚Aktives Sehen‘?</b>	<b>11</b>
2.1.1 Entstehung des Paradigmas ‚Aktives Sehen‘	11
2.1.2 Das Vorbild in der Natur: Das Sehverhalten der Wirbeltiere	12
2.1.3 Eigenschaften des Aktiven Sehens	16
2.1.4 Entwicklungsziele für aktiv sehende Systeme	19
<b>2.2 Arbeiten im Bereich ‚Aktives Sehen‘</b>	<b>21</b>

---

2.2.1	Blickfixierung	21
2.2.1.1	Universität Erlangen-Nürnberg	21
2.2.1.2	Universität Tübingen	21
2.2.1.3	Royal Institute of Technology, Stockholm, Schweden	22
2.2.1.4	University of Coimbra, Portugal	22
2.2.1.5	Vanderbilt University, Nashville, USA	23
2.2.2	Blickrichtungssteuerung	23
2.2.2.1	Universität Hamburg	23
2.2.2.2	Osaka University, Japan	24
2.2.3	Foveales Sehen	25
2.2.3.1	University of Oxford, Großbritannien	25
2.2.3.2	Simon Fraser University, Burnaby, Kanada	26
2.2.3.3	Computer Vision Centre, Bellaterra, Spanien	26
2.2.4	Blickrichtungssteuerung zur lokalen Trajektorienplanung	27
2.2.4.1	Institut für Neuroinformatik, Ruhr-Universität Bochum	27
2.2.4.2	University of Oxford, Großbritannien	27
2.2.4.3	Carnegie Mellon University, Pittsburgh, USA	28
2.2.4.4	Osaka University, Japan	28
2.2.5	Aufmerksamkeitssteuerung	29
2.2.5.1	Universität Erlangen-Nürnberg	29
2.2.5.2	Carnegie Mellon University, Pittsburgh, USA	29
2.2.6	Steuerung der Wahrnehmungsprozesse	30
2.2.6.1	Örebro University, Schweden	30
2.2.6.2	Simon Fraser University, Burnaby, Canada	30
2.2.7	Verhaltensbasiertes Sehen	31
2.2.7.1	NASA Johnson Space Center, Houston, USA	31
2.2.7.2	University of Virginia, Charlottesville, USA	31
2.2.7.3	Applied AI Systems Inc., Ontario, Kanada	32
2.2.7.4	Instituto de Automática Industrial, Madrid, Spanien	32
<b>2.3</b>	<b>Modellierung von Wissen und Verhalten</b>	<b>34</b>
2.3.1	Die Informationsverarbeitung durch den Menschen	34
2.3.2	Klassische Beschreibungsmethoden	35
2.3.2.1	Rahmen und Skripte	35
2.3.2.2	Semantische Netze	35
2.3.2.3	Endliche Automaten	36
2.3.2.4	Zustandskarten	36

2.3.2.5	Entscheidungsbäume und -tabellen	36
2.3.2.6	Produktionssysteme	36
2.3.2.7	Petrinetze	37
2.3.2.8	Fallbasiertes Schließen	37
2.3.3	Beschreibung durch unscharfe Mengen und Regeln	38
2.3.4	Architekturen für autonome Systeme	40
2.3.4.1	Funktionsbasierte, deliberative Architekturen	41
2.3.4.2	Verhaltensbasierte, reaktive Architekturen	42
2.3.4.3	Hybride Architekturen	43
2.3.5	Weiterführende Literatur	44

## TEIL B: KONZEPTE UND REALISIERUNGEN

<b>Kapitel 3: Verhaltensentscheidung</b>	<b>45</b>
<b>3.1 Überblick</b>	<b>46</b>
<b>3.2 Situationsanalyse</b>	<b>50</b>
3.2.1 Grundlegendes zur Analyse von Situationen	50
3.2.2 Der Situationsaspekt als linguistische Variable	51
3.2.2.1 Name	52
3.2.2.2 Liste von Objektklassen für Zuordnung	53
3.2.2.3 Liste von Objektklassen für Referenzen	53
3.2.2.4 Datenvorverarbeitung	53
3.2.2.5 Datenlogik	53
3.2.2.6 Linguistische Werte	53
3.2.2.7 Zugehörigkeitsfunktionen	54
3.2.2.8 Gültigkeitsdauer	54
3.2.2.9 Berechnungslogik	55
3.2.2.10 Zugehörigkeitsgrade	55
3.2.3 Die Objektbezogenheit von Situationsaspekten	55
3.2.3.1 Objektbezogenheit 0. Grades	55
3.2.3.2 Objektbezogenheit 1. Grades	55
3.2.3.3 Objektbezogenheit 2. Grades	56
3.2.4 Beispiel für die rekursive Ausführung der Situationsanalyse	58
3.2.5 Realisierung der Situationsanalyse	60
<b>3.3 Fähigkeiten im verteilten System</b>	<b>62</b>
3.3.1 Grundlegendes zur Repräsentation von Fähigkeiten	62

3.3.2	Das Fähigkeitenkonzept	63
3.3.3	Die Fähigkeiten der Systemkomponenten	65
3.3.4	Fortschrittsmaße für Aktionen	67
3.3.5	Repräsentation der Fähigkeiten durch das Fähigkeitsnetz	69
3.3.6	Die internen Ablaufmodelle der Fähigkeiten	71
3.3.6.1	Das Ablaufmodell der Fähigkeit <i>Überholen</i>	73
3.3.6.2	Das Ablaufmodell der Fähigkeit <i>Anhalten</i>	75
3.3.7	Die Steuerung kooperierender Aktionen im System über das Fähigkeitsnetz	76
3.3.8	Implementierte Fähigkeiten	79
<b>3.4</b>	<b>Unscharfe Regelbasis</b>	<b>81</b>
3.4.1	Grundlegendes zu unscharfen Regeln	81
3.4.2	Die effiziente Verwaltung der Regelbasis	83
3.4.2.1	Dynamisches Laden der Fallbasis	84
3.4.2.2	Explizite Repräsentation von Ähnlichkeit	84
3.4.3	Die sinnvolle Gestaltung der Regelbasis	86
3.4.4	Implementierte Regelbasen	89
<b>3.5</b>	<b>Realisierung der Verhaltensentscheidung</b>	<b>92</b>
3.5.1	Die Einbettung der Verhaltensentscheidung ins Gesamtsystem	92
3.5.1.1	Die organisatorische Hierarchie von Entscheidungsinstanzen	92
3.5.1.2	Die zentrale Entscheidungsinstanz Central Decision (CD)	93
3.5.1.3	Die unterlagerten Entscheidungsinstanzen BDGA und BDL	94
3.5.1.4	Die zentrale Situationsrepräsentation	95
3.5.1.5	Das Fähigkeitsnetz als Schnittstelle zwischen d. Entscheidungsinstanzen	95
3.5.1.6	Die hybride Architektur des EMS-VISION Systems	96
3.5.2	Die prozedurale Ausführung der Situationsanalyse u. Verhaltensentscheidung	96
3.5.3	Die Auswertung beim Regelwechsel	97
<b>3.6</b>	<b>Informationsfluss im Gesamtsystem</b>	<b>99</b>
<b>Kapitel 4:</b>	<b>Blickrichtungssteuerung</b>	<b>101</b>
<b>4.1</b>	<b>Das Fahrzeugauge MARVEYE</b>	<b>101</b>
<b>4.2</b>	<b>Das eingebettete System zur Blickrichtungssteuerung</b>	<b>102</b>
<b>4.3</b>	<b>Der Aufbau der Blickrichtungssteuerung</b>	<b>104</b>
<b>4.4</b>	<b>Die Schnittstelle zur Blickrichtungssteuerung</b>	<b>105</b>
<b>4.5</b>	<b>Der Algorithmus zur Blickrichtungssteuerung</b>	<b>108</b>
4.5.1	Die Planungsphase	109

---

4.5.2	Die Ausführungsphase	115
<b>4.6</b>	<b>Die Fertigkeiten der Blickrichtungssteuerung</b>	<b>115</b>
<b>4.7</b>	<b>Die Fähigkeiten der Blickrichtungssteuerung</b>	<b>117</b>
<b>TEIL C: EXPERIMENTELLE ERGEBNISSE</b>		
<b>Kapitel 5:</b>	<b>Das EMS-VISION System</b>	<b>119</b>
5.1	Testfahrzeuge	119
5.2	Hardware	120
5.3	Prozesse	121
<b>Kapitel 6:</b>	<b>Resultate der Verhaltensentscheidung</b>	<b>125</b>
6.1	Resultate in der Situationsanalyse	125
6.1.1	Die verwendeten Situationsaspekte	126
6.1.2	Fahren auf Feldwegen mit Anhalten vor positiven Hindernissen	126
6.1.3	Querfeldeinfahren mit Anhalten vor negativen Hindernissen	128
6.2	Ausführung einer komplexen Mission	129
6.3	Beispiel einer Kompensation systematischer Abweichungen	133
<b>Kapitel 7:</b>	<b>Resultate der Blickrichtungssteuerung</b>	<b>137</b>
7.1	Abbiegen	137
7.2	Überqueren eines Abzweigs	140
7.3	Querfeldeinfahren mit Ausweichen vor einem Hindernis	141
<b>Kapitel 8:</b>	<b>Zusammenfassung und Ausblick</b>	<b>149</b>
<b>Literatur</b>		<b>153</b>
<b>Anhang A:</b>	<b>Basisklassen der Situationsanalyse</b>	<b>163</b>
A.1	Die Klasse CSituationAspect	163
A.2	Die Klasse CMembershipMatrixEntry	163
A.3	Die Klasse CValency	164
A.4	Die Klasse CMembershipFunction	164
A.5	Die Klasse CDataProcessing	165
A.6	Die Klassen CSubject und CEgo	165

---

<b>A.7</b>	<b>Die Behandlung von neuen und verschwindenden Szenenobjekten</b>	<b>166</b>
<b>Anhang B: Basisklassen des Fähigkeitenkonzepts</b>		
<b>B.1</b>	<b>Die Basisklasse CNetNode</b>	<b>167</b>
<b>B.2</b>	<b>Die Basisklasse CAbilityNode für alle Fähigkeitenknoten</b>	<b>167</b>
<b>B.3</b>	<b>Der Prozessrahmen durch die Klasse CClientScene</b>	<b>169</b>
<b>B.4</b>	<b>Die Basisklasse CAbility für alle Fähigkeitenklassen</b>	<b>170</b>
<b>Anhang C: Basisklassen des Regelbasis</b>		
<b>C.1</b>	<b>Die Klasse CFuzzyRule</b>	<b>171</b>
<b>C.2</b>	<b>Die Klasse CFuzzyRuleTerm</b>	<b>172</b>
<b>C.3</b>	<b>Die Klasse CIntentionTerm</b>	<b>173</b>
<b>C.4</b>	<b>Die Klasse CRuleCmd</b>	<b>173</b>
<b>Anhang D: Der Prozessrahmen CCentralDecisionScene</b>		
<b>175</b>		
<b>Anhang E: Die Basisklassen zur Verwaltung der Aktionen</b>		
<b>177</b>		
<b>E.1</b>	<b>Die Basisklasse CAbilityModel</b>	<b>177</b>
<b>E.2</b>	<b>Die Klasse CAbilityInstance</b>	<b>177</b>
<b>E.3</b>	<b>Die Klasse CInstanceState</b>	<b>177</b>
<b>E.4</b>	<b>Die Klasse CAbilityCtrl</b>	<b>178</b>

## Liste der verwendeten Abkürzungen

AUTONAV	<u>A</u> utonomous <u>N</u> avigation (Projekt)
BDGA	<u>B</u> ehavior <u>D</u> ecision for <u>G</u> aze & <u>A</u> ttention
BDL	<u>B</u> ehavior <u>D</u> ecision for <u>L</u> ocomotion
BF	betrachtetes <u>F</u> ahrzeug (Bezugsfahrzeug)
BV	<u>B</u> ild <u>v</u> erarbeitung
CCD	<u>C</u> harge <u>C</u> oupled <u>D</u> evice
CD	<u>C</u> entral <u>D</u> ecision
DB	<u>D</u> etektions <u>b</u> ereich
DKR	<u>D</u> ynamic <u>K</u> nowledge <u>R</u> epresentation (zentrale Wissensrepräsentation)
DOB	<u>D</u> ynamic <u>O</u> bject Data <u>B</u> ase (dynamische Objektdatenbasis)
DOF	<u>d</u> egree <u>o</u> f <u>f</u> reedom
EMS-VISION	<u>E</u> xpected-based <u>M</u> ulti-focal <u>S</u> accadic Vision (Projekt)
GC	<u>G</u> aze <u>C</u> ontrol
GPS	<u>G</u> lobal <u>P</u> ositioning <u>S</u> ystem
HKT	<u>h</u> omogene <u>K</u> oordinat <u>e</u> n <u>t</u> ransformation
i. A.	<u>i</u> m <u>A</u> llgemeinen
ID	<u>I</u> dentifizierungsnummer
ISF	<u>I</u> nstitut für <u>S</u> ystemdynamik und <u>F</u> lugmechanik
KI	<u>K</u> ünstliche <u>I</u> ntelligenz (Forschungsgebiet)
MARVEYE	<u>M</u> ulti-Focal <u>A</u> ctive- <u>R</u> eactive <u>V</u> ehicle <u>E</u> ye
ME	<u>M</u> ission <u>e</u> lement
ODfT	<u>O</u> bstacle <u>D</u> etection for <u>T</u> errain (Wahrnehmungsprozess)
OVB	<u>O</u> ptimization of <u>V</u> iewing <u>B</u> ehavior (Blickrichtungsfähigkeit)
PC	<u>P</u> ersonal <u>C</u> omputer
PercG	<u>P</u> erceive <u>G</u> round (Wahrnehmungsprozess)
PNO	<u>P</u> erceive <u>N</u> egative <u>O</u> bstacles (Wahrnehmungsfähigkeit)
RDT <sub>L/D</sub>	<u>R</u> oad <u>D</u> etection and <u>T</u> racking for <u>L</u> ocal/ <u>D</u> istant Road Segment
ROA	<u>r</u> egion <u>o</u> f <u>a</u> ttention (Aufmerksamkeitsbereich)
ROI	<u>r</u> egion <u>o</u> f <u>i</u> nterest
SA	<u>S</u> ituations <u>a</u> spekt
SAGA	<u>S</u> ituation <u>A</u> nalysis for <u>G</u> aze & <u>A</u> ttention
SB	<u>S</u> ichtbarkeits <u>b</u> ereich

SCI	<u>Scalable Coherent Interface</u> (Kommunikationsmedium)
SW	<u>Software</u>
UniBwM	<u>Universität der Bundeswehr München</u>
UML	<u>Unified Modeling Language</u>
VAGA	<u>Visibility Analysis for Gaze &amp; Attention</u>
VAMoRS	<u>Versuchsfahrzeug für autonome Mobilität und Rechnersehen</u>
VAMP	<u>Versuchsfahrzeug für autonome Mobilität und Rechnersehen im PKW</u>
VC	<u>Vehicle Control</u>
VFE	<u>Vision Front End 200</u> (Spezialhardware zur Bildverarbeitung)
u. U.	<u>unter Umständen</u>

## Kurzfassung

Autonome Fahrzeuge stellen hochkomplexe, dynamische Systeme dar, für deren Verwirklichung die Forschungsergebnisse vieler Wissenschaftszweige zusammengeführt werden müssen. So sind für die Fahrzeugführung – insbesondere bei hohen Geschwindigkeiten – die Methoden der Regelungstechnik und Systemdynamik unverzichtbar. Zur Messung des Fahrzeugzustands und für die Wahrnehmung von Objekten in der Umgebung müssen die Signale unterschiedlichster Sensoren verarbeitet und fusioniert werden. Inertialsensoren, GPS-Sensoren, Radar und Echtzeitbildverarbeitung sind nur einige mögliche Informationsquellen für ein autonomes Fahrzeug. Mit zunehmender Anzahl von Fähigkeiten des autonomen Fahrzeugs erhöht sich auch die Notwendigkeit einer flexiblen, effizienten Situationsanalyse und Verhaltensentscheidung. Hier benötigt man geeignete Methoden zur Wissensrepräsentation, Verhaltensmodellierung und künstlichen Intelligenz.

Die vorliegende Arbeit möchte eine Brücke zwischen den oben genannten Wissenschaftsbereichen sein. Sie stellt ein Gesamtsystem vor, welches es erlaubt, Bildverarbeitung, Fahrzeugführung, Situationsanalyse, Verhaltensentscheidung, Blickrichtungssteuerung und Missionsplanung im „geschlossenen Kreis“ – d. h. mit Rückkopplung über die reale Welt – zu betreiben und zu testen, so dass alle Komponenten des Systems parallel verbessert werden können.

Zu diesem Zweck entwickelte der Autor eine Verhaltensentscheidung für ein autonomes Bodenfahrzeug, welches aus mehreren Systemkomponenten besteht. So repräsentiert das sogenannte „Fähigkeitsnetz“ die einzelnen Fähigkeiten der Systemkomponenten mit ihren unterschiedlichen Abstraktionsgraden und deren Abhängigkeiten untereinander. Das Modul zur Situationsanalyse und Verhaltensentscheidung generiert situationsbeschreibende, linguistische Aussagen, wählt die repräsentierten Fähigkeiten situations- und zielgerecht aus und kontrolliert und koordiniert die Aktivitäten im System. Dabei beschreibt eine Bibliothek unscharfer Regeln das Sollverhalten des Fahrzeugs.

Das in der Arbeit vorgestellte autonome Fahrzeug ist kein passiver Beobachter, sondern kann mit seinem multifokalen Fahrzeugauge (MARVEYE) und seiner aktiven Kameraplattform bestimmen, welche Information es über die Umwelt mit welcher Auflösung aufnimmt. Die Arbeit beschreibt den aktuellen Stand der Technik bezüglich der Entwicklung von aktiv sehenden Fahrzeugen und mobilen Robotern. Daraus geht hervor, dass das vorgestellte Kraftfahrzeug das erste seiner Art ist, welches über eine aktive Blickrichtungssteuerung verfügt, die die Bedürfnisse und Eigenschaften der verschiedenen Messprozesse und die Geometrien der wahrzunehmenden Objekte berücksichtigt.

Der Autor versucht, die Implementierung der vorgestellten Konzepte möglichst erschöpfend dazustellen. Er bedient sich dazu der inzwischen allgemein anerkannten *Unified Modeling Language* (UML). Die Verifikation der Konzepte zur Situationsanalyse, Verhaltensentscheidung und Blickrichtungssteuerung geschieht im Rahmen von komplexen, autonomen Missionen, die mit dem Versuchsfahrzeug VAMORS in natürlicher Umgebung durchgeführt wurden. In diesen Missionen navigiert das Fahrzeug auf einem engen Wegenetz und auf unbefestigtem Gelände. Wichtige Elemente sind das autonome Abbiegen und das Ausweichen vor einem Graben.

## Abstract

Autonomous vehicles have to interact with their environment in a very complex and dynamic manner. For the realization of such systems, the outcomes of many fields of research must be combined. For locomotion control – especially at high speed – the methods of control theory and system dynamics must be applied. Measuring the vehicle state and perceiving the objects in the environment require the fusion of signals generated by sensors with very different characteristics. Inertial sensors, GPS sensors, radar and real time image processing are only some sources of information for an autonomous vehicle. With an increasing number of capabilities of the autonomous vehicle, flexible and efficient modules for situation analysis and behavior decision become more and more relevant. It is here where suitable methods for knowledge representation, behavior modeling and artificial intelligence must be applied.

This thesis wants to be a bridge between the fields of research listed above. A system is presented, which allows to run and test image processing, locomotion control, situation analysis, behavior decision and mission planing in a closed loop – this means with feedback from the real world – so that all system components can be improved in parallel.

For this purpose, the author developed a behavior generation module for an autonomous ground vehicle. This module uses the so-called capability net, which represents the functionality of the different system components with their abstraction levels and dependencies among each other. The module for situation analysis and behavior decision generates linguistic statements describing relevant situation aspects, selects capabilities being suitable for the current situation and mission and controls and coordinates the activities in the system. The vehicle's behavior is determined by a library of fuzzy rules.

The autonomous vehicle is not a passive observer. Instead, it can determine which environmental information is perceived with suitable resolution using a multi-focal vehicle eye – called MARVEYE – mounted on an active camera head. The thesis describes the state-of-the-art regarding the development of vehicles and mobile robots using 'Active Vision'. This survey shows, that the vehicle presented is the first one using active gaze control by evaluating the requirements and properties of different perception modules and the geometry of the objects to perceive.

The author attempts to describe the concepts and implementations as completely as possible using the generally accepted *Unified Modeling Language* (UML). The verification of the concepts for situation analysis, behavior decision and viewing direction control is done within the scope of complex driving missions. In this missions, the experimental vehicle VAMORS navigates autonomously in natural environment on narrow, unmarked roads and off-road. Important elements are turn-off and evasion maneuvers.

# **TEIL A**

## **EINFÜHRUNG**



# Kapitel 1

## Einleitung

Nach LORENZ [1973] ist es die Erkenntnis vieler Philosophen und Naturforscher, dass der Fortschritt im organischen Werden fast immer dadurch erzielt wird, indem eine Anzahl verschiedener und bis dahin unabhängig voneinander funktionierender Systeme zu einer Einheit höherer Ordnung integriert werden. Im Verlauf dieser Integration treten an den Komponenten Veränderungen auf, die sie zur Mitarbeit in dem neu entstehenden übergeordneten Ganzen geeigneter machen. Die Vereinigung von Verschiedenem zur funktionellen Ganzheit bedeutet zunächst eine größere Komplexität des lebenden Systems. Im Laufe der Evolution vereinfacht sich jedoch oft das neue System dadurch, dass jedes dieser in ihm vereinigten Untersysteme sich spezialisiert und es Funktionen, die es zur Zeit seiner Selbständigkeit ebenfalls erfüllen musste, anderen Gliedern des Ganzen überlässt.

Diese bei biologischen Systemen gemachte Beobachtung gilt nach der Überzeugung von LORENZ [1973] auch bei vom Menschen konstruierten Maschinen. Diese Maschinen sind in ihren ersten, versuchsweise hergestellten Exemplaren komplizierter gebaut, als in ihrer endgültigen, ausgereiften Version. Wechselwirkungen, insbesondere der Austausch von Information zwischen Untersystemen, werden vereinfacht oder in direktere Bahnen gelenkt, und unnötige historische Reste abgebaut.

Die parallele Entwicklung und Verbesserung von Systemkomponenten im Rahmen eines – zumindest partiell – funktionsfähigen Gesamtsystems sind umso wichtiger, je komplexer das Gesamtsystem ist, und je weniger quantitativ beschreibbar die Rahmenbedingungen sind, mit denen es fertig werden muss. Ein autonomes Landfahrzeug, welches in natürlicher Umgebung navigiert, ist ein solches komplexes System.

Wie aus der einschlägigen Fachliteratur hervorgeht<sup>1</sup>, wurden bereits viele Einzelfunktionalitäten für autonome Fahrzeuge entwickelt und in ihrem eingeschränkten Anwendungsbereich verifiziert. Viele Arbeiten konzentrieren sich auf Probleme der Sensordatenverarbeitung, insbesondere der Verarbeitung von Bildfolgen von Kameras. Andere Arbeiten beschreiben robuste Regelalgorithmen, um ein Kraftfahrzeug in der Fahrspur zu halten. Dagegen beschränken sich die Veröffentlichungen im Forschungsgebiet der „Künstlichen Intelligenz“ bezüglich Fahrzeugführung und Fahrerassistenzsysteme fast immer auf Simulationen. Zu stark ist die Versuchung, sich in quantitativ beschreibbare Umgebungen zurückzuziehen.

So kommt es, dass in der Literatur nur sehr wenig Bodenfahrzeuge beschrieben werden, welche gestützt auf ihren Gesichtssinn und auf andere Sensoren komplexe Missionen in einer natürlichen Umgebung vollständig autonom ausführen können, obwohl die Grundlagen für ein solches Fahrzeug bereits zur Verfügung stehen. Ein Pionier, der sich schon früh dieser

---

<sup>1</sup> Hier seien als Beispiele die Bände der Symposien „*Intelligent Vehicles*“ und „*Intelligent Transportation Systems*“ der letzten 10 Jahre genannt.

Problematik annahm, ist das Institut für Systemdynamik und Flugmechanik (ISF) der Universität der Bundeswehr München. Seit über 20 Jahren versucht man an diesem Institut, bei der Entwicklung dynamischer Systeme mit der aktuell verfügbaren Rechenleistung auszukommen<sup>2</sup>. Das Prinzip, am funktionsfähigen, dynamischen System zu arbeiten und dieses in seiner Zielumgebung weiterzuentwickeln, zwingt zum effizienten Einsatz der Rechenleistung und der Systemressourcen und schließlich zur Entwicklung des sogenannten 4D-Ansatzes [DICKMANN, 1987].

Nach den Ausführungen von LORENZ [1973] profitieren alle Komponenten eines Systems von den Erfahrungen und Erkenntnissen, die durch den Betrieb des funktionsfähigen Ganzen gewonnen werden. So ist es nur konsequent, wenn sich ein autonomes Fahrzeug eines mächtigen Hilfsmittels bedient, welches sich in der Evolutionsgeschichte der Wirbeltiere als gewinnbringend herausgestellt hat: die aktive Steuerung der Blickrichtung. Auch hier führt der zusätzliche Freiheitsgrad in der Aktion zunächst zu einer größeren Komplexität des Systems. Wie jedoch die vorliegende Arbeit zeigt, ist die Lösung der Gesamtaufgabe durch eine intelligente Wechselwirkung und Anpassung der Systemkomponenten aufeinander leichter möglich.

Auch BERNASCH [1997] stellt heraus, dass weder biologische noch rechnergestützte Sehsysteme in der Lage sind, die gesamte potentiell erfassbare visuelle Information intensiv zu verarbeiten, da jedes dieser Systeme natürlichen Ressourcenbeschränkungen unterliegt. Aufgrund dieser Tatsache kann man qualitativ bessere Ergebnisse und eine größere Robustheit erzielen, wenn die vorhandenen Verarbeitungsressourcen statt auf alle visuellen Daten nur auf wesentliche angesetzt werden. Diese Aussage gilt unabhängig von Größe und Umfang der verfügbaren Rechenleistung und Sensorik.

Zusammenfassend lässt sich feststellen, dass die vorliegende Arbeit – sowie auch frühere Arbeiten am ISF – sich an folgenden Prinzipien orientieren:

- Biologisch motivierte Auswahl der Sensorik mit Inkaufnahme der dadurch erhöhten Komplexität des Systems.
- Test und Verifikation der Module im funktionsfähigen Gesamtsystem in der natürlichen (Ziel-)Umgebung. Die während der Entwicklungsarbeit angefertigten, einfach gehaltenen Simulationen gelten nicht als Funktionsnachweis.
- Parallele Weiterentwicklung und Verbesserung der Module auf allen Systemebenen. Die Komplexität der Algorithmen wächst mit der Komplexität der auszuführenden Missionen („*das System wächst mit seinen Aufgaben*“).

## 1.1 Terminologie

Um Missverständnisse bezüglich des Inhalts von Aussagen zu vermeiden, beschreibt dieser Abschnitt zunächst die in dieser Arbeit benutzten Begriffe. Anschließend wird auf den Entwicklungshintergrund am ISF – insbesondere den 4D-Ansatz – eingegangen.

### 1.1.1 Objekt – Subjekt

Der wohl in dieser Arbeit am häufigsten verwendete Begriff heißt *Objekt*. Dieser Begriff wird in zweierlei Hinsicht gebraucht. Zum einen bezeichnet er ein physikalisches Objekt, welches sich dadurch auszeichnet, dass dessen Bewegungszustand unverändert bleibt, solange

---

<sup>2</sup> Eine Historie der Entwicklungen am ISF findet sich z. B. in [GREGOR, 2002] und [DICKMANN, 2001].

von außen keine Kräfte einwirken. Dabei ist das Bewegungsverhalten auf der Zeitachse ebenso charakteristisch für das Objekt, wie seine Form im Raum. Zum zweiten sind Objekte im Kontext der objektorientierten Programmierung Instanzen von Klassen. Die objektorientierte Programmierung ist gekennzeichnet durch die Aspekte Identität, Klassifikation, Polymorphismus und Vererbung<sup>3</sup>. Der 4D-Ansatz von DICKMANN [1987] ist in seiner aktuellen Implementierung in beiderlei Hinsicht *objektorientiert*: Er arbeitet mit Repräsentationen physikalischer Objekte und nutzt intelligent die Möglichkeiten der objektorientierten Programmierung [GREGOR et al., 2000].

Als besondere physikalische Objekte sieht DICKMANN [1989, 1992] die sogenannten *Subjekte*. Subjekte sind handlungsfähige Individuen, welche ihre internen Freiheitsgrade dazu nutzen können, ihren eigenen Zustand und den ihrer Umgebung durch die Vorgabe von Steuergrößen entsprechend ihren Zielen zu beeinflussen. Den Bewegungszustand eines Subjekts kann man nur unvollständig aus einer äußeren Beobachtung des Subjekts und seiner Umgebung ableiten. Die Bewegungen des Subjekts werden vielmehr in Abhängigkeit von internen, „mental“ Zuständen gesteuert, welche von außen nicht direkt beobachtbar sind. Die „mental“ Zustände können sich ändern, ohne dass damit Änderungen in der Umgebung einhergehen. Dies ist z. B. bei einem Wechsel der Intention des Subjekts oder aufgrund einer andersartigen Interpretation der Sensordaten der Fall. Um das Verhalten und damit den Bewegungszustand eines Subjekts vorhersagen zu können, ist es notwendig, neben den systemdynamischen Zuständen auch die internen, „mental“ Zustände des Subjekts und seine Verhaltensregeln zu kennen.

Der Begriff des Subjekts kann mit dem von KRÜGER [1991] definierten Begriff des *Akteurs* gleichgesetzt werden. Der Autor bezeichnet damit ein System mit internen Freiheitsgraden, die es zur Auswahl alternativer Handlungsmöglichkeiten nutzen kann.

Die Trajektorien von Objekten sagen etwas über die Natur der aktuell in der Umwelt auftretenden Störungen aus, genauso wie aus den Trajektorien der Subjekte etwas über deren Intentionen abgeleitet werden kann. Ferner zeigen Subjekte meist ein charakteristisches, (stereo)typisches Bewegungsverhalten, welches man – neben der räumlichen Form – zur Erkennung und Identifikation des Subjekts heranziehen kann. Dieses indirekt ermittelte Wissen ist für die Interpretation der Situation von entscheidender Bedeutung.

### 1.1.2 Fähigkeit – Aktion – und davon abgeleitete Begriffe

In der vorliegenden Arbeit werden die von einem autonomen System prinzipiell ausführbaren Aktivitäten und Funktionalitäten die *Fähigkeiten* des Systems genannt. Je nach Situation nutzt das autonome System einige dieser Fähigkeiten. Die Nutzung einer Fähigkeit ist i. A. mit Ressourcenverbrauch verbunden. Das Modul oder die Systemkomponente, welche eine Fähigkeit anbietet, heißt *Experte* der Fähigkeit. Bei der Ausführung einer Fähigkeit wird eine Instanz dieser Fähigkeit vom Experten angelegt. Eine Instanz einer Fähigkeit heißt *Aktion*. Um auf die Qualität und den Fortschritt einer Aktion reagieren zu können, sind diese durch geeignete Gütemaße zu messen.

Die Fähigkeiten eines autonomen Systems sind i. A. voneinander abhängig. Aufgrund dieser Abhängigkeiten und aufgrund ihres Abstraktionsgrades kann man die Fähigkeiten hierarchisch ordnen. Für die Fähigkeiten und Aktionen der verschiedenen Hierarchiestufen werden in dieser Arbeit spezielle Begriffe verwendet. *Fertigkeiten* sind elementare Fähigkeiten und

---

<sup>3</sup> Zur Beschreibung und Darstellung der verschiedenen objektorientierten Entwurfsmethoden wurde eine ‚vereinheitlichte Modellierungssprache‘ (*Unified Modeling Language*, UML) entwickelt [OESTEREICH, 1997].

stehen in der Hierarchie an unterster Stelle. Instanzen dieser Fertigkeiten heißen *Handlungsprimitive*. *Schematische Fähigkeiten* kombinieren mehrere Fertigkeiten, um ein komplexeres Verhalten zu generieren. Eine Aktion auf dieser mittleren Hierarchiestufe heißt *Handlung*, *Aktivität* oder *Tätigkeit*. An oberster Stelle in der Hierarchie stehen die abstrakten *Verhaltensfähigkeiten*. Die Aktionen dieser abstrakten Fähigkeiten heißen *Verhaltensweisen*. Näheres zu diesen Begriffen und ihren Unterscheidungsmerkmalen findet sich in Abschnitt 3.3.2.

### 1.1.3 Szene – Situation – Situationsanalyse

Die räumlich-zeitliche Anordnung von physikalischen Objekten aus der Sicht eines Betrachters wird beim maschinellen Sehen als *Szene* bezeichnet. Oft nennt man die Gesamtheit der systemdynamischen Zustände aller Objekte und Subjekte einer Fahrzeugumgebung kurz den Zustand des Systems [HAAG, 1998]. Der in dieser Arbeit verwendete Begriff der Szene beinhaltet eine solche Zustandsbeschreibung der Umgebung, trägt aber zusätzlich der Egozentriertheit der Beobachtung Rechnung. Die Szene wird im 4D-Ansatz baumartig modelliert (vgl. z. B. [DICKMANN D., 1997]).

In der Literatur existieren hinsichtlich des Begriffs *Situation* sehr viele verschiedene Definitionen (vgl. z. B. [MAURER, 2000]). Nach HAAG [1998] unterscheidet sich eine Situation vom Zustand (Szene) um den Handlungsaspekt. Auch KRÜGER [1991] versteht unter dem Begriff der Situation einen erweiterten Zustand, bei dem Akteure nicht nur als physikalische Gegenstände (Objekte) betrachtet werden, sondern zusätzlich deren Handlungen und Handlungsalternativen, um so die zeitliche Entwicklung der Situation abzuschätzen. Folgerichtig sehen Subjekte, die sich im selben Zustand befinden, aber unterschiedliche Aktionsmöglichkeiten zur Verfügung haben, unterschiedliche Situationen. Daneben führt bei Subjekten mit unterschiedlichen Zielen und Fähigkeiten die gleiche Szene zu unterschiedlichen Situationsbeschreibungen [DICKMANN S. & FÜRST, 1998]. In dieser Arbeit steht der Begriff der Situation für die Summe der für die Verhaltensentscheidungen relevanten Sachverhalte. Relevant für die Verhaltensentscheidung sind die aktuelle Szene, die Intentionen und Aktionen aller Subjekte in der Szene (einschließlich des Eigenfahrzeugs) und die Fähigkeiten des Eigenfahrzeugs, welche die Entscheidungsalternativen darstellen.

Nach HAAG [1998] beschäftigt sich die Situationsanalyse eines handlungsfähigen Individuums mit der Einschätzung der Situation unter Einbeziehung dessen Zustands, des Zustands der Umgebung sowie der internen und externen Freiheitsgrade, die das Individuum nutzen kann, um den eigenen Zustand und den Zustand der Umwelt zu ändern. Charakteristisch für die Situationsanalyse ist nach MAURER [2000] der Übergang „vom Signal zum Sinn“. Nach SCHÄFER [1996] hat die Situationsanalyse zwei Aufgaben zu erfüllen. Die erste Analyseaufgabe ist die Ablaufferkennung, welche komplexe und zeitlich ausgedehnte Abläufe mit natürlichsprachlichen Begriffen zusammenfasst. Definierte Abläufe können dabei rekursiv zur Bildung höherer, begrifflicher Abstraktionen dienen. Die zweite Analyseaufgabe hat die Erkennung von Situationen zum Ziel, in denen sich die Subjekte zu bestimmten Zeitpunkten befinden. Weitere Aspekte bezüglich Situation und Situationsanalyse sind in Abschnitt 3.2.1 zusammengetragen.

### 1.1.4 Intention

Der Begriff der Intention steht in der vorliegenden Arbeit für die Handlungsabsicht eines Subjekts. Um die Intention eines Subjekts zu erkennen, muss sich der Beobachter in das Subjekt hineinversetzen und in seiner Vorstellung dessen Perspektive einnehmen. Nach RETZSCHMIDT [1991] lassen sich Intentionen in eine hierarchische Ordnung bringen. Die Autorin

zitiert als Beispiel die folgenden intentionalen Behandlungen einer Person namens Sue, welche nach aufsteigender Hierarchie geordnet sind:

„zu Sue *‘sehr gut‘* sagen“ — „Sue loben“ — „Sue glücklich machen“

Anhand dieses Beispiels ist zu erkennen, dass die Intentionen einer Hierarchieebene durch die Intentionen der tieferen Ebenen realisiert werden. Der Beobachter muss für die vollständige Erkennung der Absicht eines Subjekts versuchen, diese Hierarchie von Intentionen zu rekonstruieren.

Im Bereich der Fahrzeugführung ordnet man die Intentionen entweder der Navigationsebene, der Führungsebene oder der Stabilisierungsebene zu. Ebenfalls oft findet man die inhaltlich äquivalente Einteilung in die strategische, taktische und operationale Ebene (vgl. [MOCKHECKER, 1994]). Die Hierarchie von Intentionen führt zu einer analogen hierarchischen Gliederung der sich aus den Intentionen ergebenden Aktionen (vgl. Abschnitt 3.3.2). Nach MOCKHECKER [1994] müssen zur Erkennung kritischer Verkehrssituationen nicht die operationalen Intentionen der Verkehrsteilnehmer ermittelt werden, sondern ihre taktischen. Konflikte zwischen den Intentionen von Subjekten bezeichnet man in der Literatur auch als negative Interaktion.

### 1.1.5 Domäne

Ein (autonomes) Fahrzeug kann sich in unterschiedlichen Klassen von Umgebungen befinden, in denen ganze Gruppen von Aktionen erlaubt, notwendig oder verboten sind. Zum Beispiel ist auf Autobahnen Rückwärtsfahren und Wenden verboten, was beim Querfeldeinfahren ein ganz normaler Vorgang ist. Ferner ist in gewissen Umgebungen das Auftreten bestimmter Ereignisse oder Objekte auszuschließen. So muss man auf Autobahnen nicht mit entgegenkommenden Fahrzeugen rechnen. Auf Landstraßen wiederum gilt diese Annahme nicht. Klassen von Umgebungen werden in dieser Arbeit als *Domänen* bezeichnet.

## 1.2 Entwicklungshintergrund am ISF<sup>4</sup>

Das Institut für Systemdynamik und Flugmechanik (ISF) der Universität der Bundeswehr München (UniBwM) beschäftigt sich seit mehr als zwei Jahrzehnten mit dem dynamischen, maschinellen Sehen und hat in diesem Zeitraum sieben autonome Fahrzeug mit Sehsystemen ausgestattet. Während dieses Zeitraums wurden Anwendungen für verschiedene Domänen und Aufgabenstellungen realisiert und getestet. Den Kern der Wahrnehmungsprozesse bildete dabei stets der 4D-Ansatz von DICKMANN [1987, 1992].

### 1.2.1 Der 4D-Ansatz

Im Forschungsbereich der „Künstlichen Intelligenz“ (KI) wurde anfangs maschinelles Sehen als quasi-statisches Problem aufgefasst. So verwendete man viel Energie darauf, Mittel und Wege zu finden, die perspektivische Abbildung durch die Verarbeitung von Bildfolgen zu invertieren. Diese Ansätze ziehen jedoch keinen Vorteil aus der zeitlichen Kontinuität, der alle Objekte und Subjekte in der physikalischen Welt unterworfen sind. Die Bewegungen der Objektmerkmale im Bild, welche aus einer räumlichen Relativbewegung resultieren, sind aufgrund der nicht linearen perspektivischen Abbildung nicht durch einfache Bewegungsmodelle beschreibbar. Auch die Integrität der Form eines Objekts kann durch die perspektivische Ab-

<sup>4</sup> Der Inhalt dieses Abschnitts basiert auf Veröffentlichungen von E. D. Dickmanns und Mitarbeitern des ISF.

bildung verloren gehen (z. B. durch Verdeckung). Aus diesem Grund muss die Form und die Bewegung der Objekte im 3D-Raum repräsentiert werden.

Der 4D-Ansatz vermeidet es, zeitaufwendige Bildoperationen (z. B. optischer Fluss) auszuführen. Vielmehr wird unter Ausnutzung der zeitlichen und räumlichen Kontinuität der physikalischen Objekte ein frühzeitiger Übergang von der Merkmalebene auf die Objektebene durchgeführt. Dieses Vorgehen reduziert einerseits die Daten, welche für die Szenenbeschreibung notwendig sind; zum anderen können Hintergrundwissen wie Form- und Dynamikmodelle und die Wirkungen der Aktionen des Beobachters frühzeitig berücksichtigt werden. Die dabei verwendete Repräsentation hat die gleichen Dimensionen wie die repräsentierte Welt: 3 räumliche und 1 zeitliche (deshalb 4D-Ansatz). Der 4D-Ansatz beruht auf der gleichzeitigen Nutzung folgender Modelle und Methoden [DICKMANNNS, 1998]:

- Dynamische Bewegungsmodelle beschreiben die Bewegung eines Objekts und dessen Schwerpunkts unter Berücksichtigung der Steuergrößen des beobachtenden Subjekts und der bei der Wahrnehmung auftretenden Totzeiten.
- Räumliche 3D-Formmodelle der Objekte dienen zur Spezifizierung visuell vermessbarer Merkmale und deren Lage.
- Unter Zuhilfenahme eines perspektivischen Abbildungsmodells werden die vermessbaren 3D-Objektmerkmale ins Bild projiziert.
- Für die Bestimmung der wirklichen Lage der vermessbaren Bildmerkmale setzt man selektiv Messprozesse auf<sup>5</sup>.
- Durch die Rückkopplung der Vorhersage-Fehler bezüglich Form und Lage werden die 4D-Objektzustände geschätzt und die Vorstellungen über die Bewegungsabläufe in Raum und Zeit nachgeführt. Dabei kommen Erweiterte Kalman-Filter zur Anwendung, welche an die Verarbeitung von Bildfolgen angepasst sind.

Die rekursiven Erweiterten Kalman-Filter erlauben es, unter Ausnutzung des Wissens über die Statistiken des Mess- und Systemrauschens und durch die Verwendung dynamischer Bewegungsmodelle, Rauschen von systematischen Zustandsänderungen zu trennen und eine optimale<sup>6</sup> Zustandsschätzung aus einer reduzierten Auswahl von Messgrößen zu ermitteln. Damit wird eine Inversion der perspektivischen Abbildung umgangen.

Bei der Schätzung der Parameter der Form- und Dynamikmodelle durch selektive Messungen im Bild ist das sogenannte Korrespondenzproblem zu lösen: Merkmale im letzten und im aktuellen Kamerabild müssen einander zugeordnet werden. Tastet man eine Szene mit einer ausreichend hohen Rate durch eine Kamera ab, so verschwindet das Korrespondenzproblem, da sich zwischen den Bildern nur geringfügige Veränderungen ergeben. Die Suchpfade zum Wiederfinden der Merkmale können dann sehr klein sein. Nach DICKMANNNS [1998, 1992] wächst der Suchaufwand zum Wiederauffinden eines prädierten Merkmals in der 2D-Bildebene mit dem Quadrat der Unsicherheit. Ferner erlaubt eine hohe Abtastrate die Linearisierung der nichtlinearen Bewegungsgleichungen um den Ort des aktuell eingenommenen Zustands im Zustandsraum. Diese lineare Approximation eines nichtlinearen Systems ist insbesondere dann gültig, wenn schnelle Rückkoppelkreise die Abweichungen von den Sollzuständen klein halten.

---

<sup>5</sup> Diese Art der Bildverarbeitung steht im starken Kontrast zu der extensiven Analyse einzelner (Stand-)Bilder (*pictorial vision*) und der detaillierten Inversion der perspektivischen Abbildung (*computational vision*) [DICKMANNNS, 1998].

<sup>6</sup> optimal hinsichtlich der quadratischen Abweichung zwischen den Soll- und Istwerten.

Durch die physikalisch fundierte, zeitlich-räumliche Beschreibung der Szenenobjekte kann man die Auswirkungen der Handlungen von Subjekten auf die Szene auf direkte Art und Weise voraussimulieren und die Ergebnisse dieser Voraussimulationen als Entscheidungsgrundlage zur Auswahl und Adaption von Verhalten heranziehen.

Der 4D-Ansatz erlaubt die Realisierung eines technischen Sehens, welches Psychologen die *Gestalt-Idee* beim menschlichen Sehen bezeichnen [DICKMANNNS, 1994]. Unter Zuhilfenahme von Form- und Bewegungsmodellen von Objekten, welche man im Missionskontext und aufgrund von Hintergrundwissen in der Umwelt erwartet, können die im Kamerabild gemessenen Merkmale gruppiert werden. Diese Gruppierung erlaubt die Erkennung von Objekten auch in komplexen, gestörten Umgebungen, in denen eine rein *bottom-up* basierende Merkmalgruppierung zu einer kombinatorischen Explosion und damit zum Scheitern führt. Die Gestalt-Idee ermöglicht ferner eine erwartungsbasierte Steuerung der Wahrnehmungsprozesse.

DICKMANNNS D. [1997] sieht den 4D-Ansatz komplementär zu den klassischen KI-Methoden. Der 4D-Ansatz erlaubt die automatische Wissensaufnahme durch (visuelle) Wahrnehmung und verbindet damit symbolisches Wissen mit einer Beschreibung der Außenwelt durch systemdynamische Größen – was auch als *symbol grounding* bezeichnet wird. Im Gegensatz zu den in der KI verwendeten symbolischen Modellen stehen die raum-zeitlichen Modelle des 4D-Ansatzes im direkten Zusammenhang zur dynamischen Außenwelt und gelten für den augenblicklichen Zeitpunkt. Aus den raum-zeitlichen Modellen lassen sich integrale, symbolische Aussagen generieren, welche sich dann wiederum mit den Methoden und Repräsentationen der KI verarbeiten lassen. Dadurch ist jedem abstrakten Symbol stets eine Interpretation in Raum und Zeit zugeordnet. Die auf der symbolischen Ebene vollzogenen Schlussfolgerungen sind dann nicht auf diese Ebene beschränkt, sondern können mittels der raum-zeitlichen Modelle durch Vorabsimulationen konkretisiert und überprüft werden.

### 1.2.2 Das Sehsystem der 3. Generation: EMS-VISION

Die am ISF bis zum Jahr 1997 implementierten Systeme waren eingebettet in statische Konfigurationen und optimiert für spezielle Anwendungen. Die Module waren nur lose gekoppelt und liefen auf teurer, unflexibler Spezialhardware. So konnte man mit den Systemen nur einzelne Wahrnehmungs- und Fortbewegungsfähigkeiten zeigen. Komplexe Missionen, für die eine Kombination einer Vielzahl von Fähigkeiten erforderlich sind, waren nicht möglich. Um alle diese Einschränkungen zu überwinden, begann man im Jahr 1997 mit der Entwicklung einer völlig neuen Systemarchitektur: EMS-VISION.

EMS-VISION ist ein maschinelles Echtzeitsehsystem für die Steuerung realer, autonomer Fahrzeuge. Die Initialen E-M-S stehen für die Schlüsseleigenschaften des Systems:

- **Erwartungsbasiert:** Die Erwartung, welche Klassen von Objekten sich in der Umgebung des Fahrzeugs befinden können, und die dem System bereits bekannten Objektinformationen werden dazu benutzt, die von mehreren Kameras gelieferten Bildströme mit Hilfe rekursiver Schätzverfahren in Echtzeit zielgerichtet auszuwerten. Dabei kommen die Form- und Bewegungsmodelle des 4D-Ansatzes zur Anwendung, die eine Prädiktion der Zustandsgrößen der Objekte erlauben. Ausfälle einzelner Messwerte oder Totzeiten im System werden durch die Prädiktion überbrückt.
- **Multifokal:** Das Sehsystem verfügt über ein multifokales Fahrzeugauge namens MARVEYE (vgl. Abschnitt 4.1). Die unterschiedlichen Auflösungen im peripheren und fo-

vealen Gesichtsbereich ermöglichen eine Imitation des in Abschnitt 2.1.2 beschriebenen Wirbeltiersehens.

- **Sakkadisch:** Durch eine ereignisgetriebene aktive/reaktive Blickrichtungssteuerung lassen sich die fovealen Bereiche mit großer Auflösung auf jeden interessanten Bereich innerhalb des Gesichtsfeldes ausrichten (vgl. Kapitel 4). Zusätzlich führt die Blickrichtungssteuerung zu einer Vergrößerung des Gesichtsfeldes.

Mit dem EMS-VISION System verfolgte man folgende Entwicklungsziele: Es sollten allgemeine Erkennungsprozesse geschaffen werden, die eine möglichst große Anzahl verschiedener Objektklassen wahrnehmen und handhaben können. Alle Objekte waren innerhalb einer dynamischen Objektdatenbasis auf einheitliche Weise zu repräsentieren. Um das autonome System flexibel und erweiterbar zu gestalten, wurde eine explizite Repräsentation der Situation und der Fähigkeiten des Systems als notwendig erachtet. Für die Handhabung komplexer Situationen konzipierte man Entscheidungsinstanzen für die unterschiedlichen Aspekte von Verhalten. Im Rahmen des erwartungsbasierten Ansatzes sollten diese Entscheidungsinstanzen zu einem zielgerichteten Verhalten des Fahrzeugs und zur wissensbasierten Steuerung der Wahrnehmung führen. Das Ziel war ein bordautonomes Fahrzeug, welches ohne zusätzliche Infrastruktur<sup>7</sup> auskommt und sich ausschließlich der an Bord befindlichen Sensorik bedient.

Die Blickrichtungssteuerung eines Fahrzeugs erlaubt es, bei der Ausrüstung mit Sensoren auf ökonomische Gesichtspunkte und technische Handhabbarkeit Rücksicht zu nehmen und trotzdem eine Vielzahl von Situationen und Domänen abzudecken. Neben Bremse, Gas und Lenkung bildet die aktive Kameraplattform einen weiteren Aktuator des Fahrzeugs.

Um robuste autonome mobile Plattformen zu entwickeln, ist die gemeinsame Nutzung von visuellen und inertialen Signalen zwingend notwendig. Bei Automobilen kann dabei auf die bereits in den Serienfahrzeugen eingebaute Sensorik zurückgegriffen werden. Der 4D-Ansatz bietet die Mechanismen und Repräsentationsschemata an, um die Daten verschiedener Sensoren wie Kameras, Odometrie- oder Inertialsensoren zu einer einheitlichen Szenenbeschreibung zu fusionieren.

### 1.3 Beitrag der Arbeit

Funktionales Ziel dieser Arbeit ist es, das Verhalten eines autonomen, aktiv sehenden Fahrzeugs derart zu steuern, dass ein vorgegebener Missionsplan zur Ausführung gelangt. Bei den dabei zu treffenden Verhaltensentscheidungen sind die Fähigkeiten des autonomen Systems zu berücksichtigen. Aus diesem Grund wurde im Rahmen des Projekts EMS-VISION das sogenannte Fähigkeitenkonzept entwickelt [VERHALTEN, 2000]. Der Kern dieses Fähigkeitenkonzepts ist ein gerichteter, zyklensfreier Graph (Fähigkeitennetz), der neben den prinzipiell (statisch) verfügbaren Fähigkeiten und deren Abhängigkeiten auch die aktuelle dynamische Verfügbarkeit anzeigt. Die Realisierung der Basisfunktionalität des Fähigkeitenkonzepts erfolgte im Rahmen dieser Arbeit. Seitens der Fahrzeugführung wurde das Fähigkeitenkonzept von SIEDERSBERGER [2003] entwickelt.

Bei der Realisierung des Fähigkeitennetzes wurde darauf geachtet, der Verhaltensentscheidung eine Schnittstelle zur Verfügung zu stellen, mit der sie die repräsentierten Fähigkeiten nutzen und koordinieren kann. Zu diesem Zweck entwickelte der Autor Methoden und Kommunikationsmittel, um die Steuerbefehle der Verhaltensentscheidung und den Status und

---

<sup>7</sup> Von bordautonomen Fahrzeugen wird lediglich diejenige Infrastruktur vorausgesetzt, die auch der menschliche Fahrer benötigt.

Fortschritt der aktuell im verteilten System ablaufenden Aktionen über das Fähigkeitsnetz zu kommunizieren. Die Funktionalität des Fähigkeitsnetzes wurde in zahlreichen autonomen Testfahrten nachgewiesen.

Durch die Konzeption und Implementierung einer Situationsanalyse und Verhaltensentscheidung wird es erstmals ermöglicht, die im Fähigkeitsnetz repräsentierten Fortbewegungs-, Blickrichtungs- und Wahrnehmungsfähigkeiten zu nutzen, um dynamisch erstellte, abstrakte Missionspläne von einem realen, autonomen Fahrzeug ausführen zu lassen. Der Autor entwickelte dazu eine Bibliothek von Situationsaspekten, welche implizit in der Szenenbeschreibung enthaltenes Wissen herausarbeiten und es explizit in Form von linguistischen Aussagen repräsentieren. Durch die Implementierung in Form einer Bibliothek ist die Situationsanalyse einfach um weitere Situationsaspekte erweiterbar.

Ferner wurde in der Literatur nach geeigneten Methoden gesucht, um das Verhalten eines autonomen Fahrzeugs zu modellieren. Kriterien bei der Auswahl waren, dass die Verhaltensregeln nicht implizit sondern explizit repräsentiert sind und dass das Verhalten des Fahrzeugs im Testbetrieb jederzeit nachvollziehbar bleibt. Die Wahl fiel auf eine Modellierung mit unscharfen Regeln; in der Literatur auch als *fuzzy reasoning* bezeichnet. Die unscharfen Regeln umschreiben mehr oder weniger detailliert Situationen. Eine exakte Beschreibung von Situationen wird dabei vermieden. Die Implementation des Regelwerks geschah ebenfalls in Form einer Bibliothek und ist daher leicht erweiterbar. Der Autor entwickelte Regelbasen für das Wahrnehmen und Navigieren in den Domänen „*Fahren auf Straßen niederer Ordnung*“ und „*Querfeldeinfahren*“.

Die Situationsanalyse und Verhaltensentscheidung sind in einem komfortablen Prozessrahmen eingebettet, welcher die Grundlage für die zentrale Entscheidungsinstanz *Central Decision*<sup>8</sup> des EMS-VISION Systems bildet. Der Prozessrahmen unterstützt alle Aspekte der Verhaltensentscheidung. So ist es z. B. durch eine softwaretechnische Repräsentation der Subjekte möglich, die Bibliothek von Verhaltensregeln sowohl auf das Eigenfahrzeug als auch auf andere Subjekte in der Szene anzuwenden. Ferner entwickelte der Autor einen iterativen Ablauf, durch den ausschließlich die bei der Verhaltensentscheidung relevanten Situationsaspekte berechnet werden.

Das zweite Hauptthema dieser Arbeit bildet eine aktive Blickrichtungssteuerung, welche die Wünsche und Eigenschaften der Wahrnehmungsprozesse im System hinsichtlich der Blickrichtung berücksichtigt. Die von der Blickrichtungssteuerung angebotene Schnittstelle erlaubt es, für jedes wahrzunehmende Objekt sogenannte Aufmerksamkeitsbereiche zu definieren und diese mit einer zweistufigen Logik zu verknüpfen. Die Aufmerksamkeitsbereiche erstrecken sich über Objektteile mit wahrnehmbaren Merkmalen. Mit der Logik gibt der Wahrnehmungsexperte bekannt, mit welcher Priorität er welche Kombinationen von Aufmerksamkeitsbereichen abgebildet haben will.

Eine weitere Eingangsgröße für die Blickrichtungssteuerung ist der Aufmerksamkeitsbedarf jedes Objekts. Der Aufmerksamkeitsbedarf wird von der Verhaltensentscheidung situationsabhängig festgelegt und beschreibt, wie viel Aufmerksamkeit jedem Objekt von der Blickrichtungssteuerung zugestanden werden soll. Der Optimierungsprozess in der Blickrichtungssteuerung berücksichtigt den Informationsverlust durch Sakkaden, den Aufmerksamkeitsbedarf der um die Ressource Blickrichtung konkurrierenden Objekte und den Informationsgewinn durch die alternativen Blickrichtungssequenzen. Das Resultat ist eine optimale Sequenz von Sakkaden und Blickverfolgungen. Das entwickelte Konzept ermöglicht eine dynamische

---

<sup>8</sup> Die Namensgebung geht auf MAURER [2000] zurück.

Interaktion zwischen Wahrnehmung und Blickrichtungssteuerung in Echtzeit, welche während vieler Testfahrten mit einer variierenden Anzahl wahrzunehmender Objekte und Objektklassen im geschlossenen Kreis demonstriert wurde.

## 1.4 Aufbau der Arbeit

Nach dieser Einführung in die Thematik der vorliegenden Arbeit beginnt Kapitel 2 mit dem Paradigma ‚Aktives Sehen‘ bei der Bildverarbeitung und beschreibt den Stand der Technik bezüglich aktiv sehender, mobiler Roboter und Fahrzeuge. Im Anschluss daran werden die für die Repräsentation von Wissen und Verhalten in der Literatur vorgeschlagenen Methoden kurz gegenübergestellt und weiterführende Literatur angegeben. Die in dieser Arbeit verwendete Modellierungsmethode durch unscharfe Mengen und Regeln nimmt dabei einen breiteren Raum ein. Ferner werden alternative Architekturen für autonome Systeme diskutiert.

Kapitel 3 beschäftigt sich mit dem entwickelten Konzept zur Verhaltensentscheidung für autonome Fahrzeuge. Abschnitt 3.1 gibt einen Überblick über die an der Verhaltensentscheidung beteiligten Komponenten. Abschnitt 3.2 beschreibt das Vorgehen bei der Situationsanalyse und Abschnitt 3.3 stellt das Fähigkeitsnetz vor, durch welches die Fähigkeiten des autonomen Systems explizit repräsentiert und die Aktionen im System koordiniert werden. Die unscharfe Regelbasis wird in Abschnitt 3.4 behandelt. Abschnitt 3.5 beschreibt einige wichtige Aspekte bezüglich der Realisierung des vorgestellten Konzepts, z. B. dessen Einbettung ins Gesamtsystem und die iterative Ausführung von Situationsanalyse und Verhaltensentscheidung. Abschließend zeigt Abschnitt 3.6 den Informationsfluss im Gesamtsystem.

Kapitel 4 behandelt die vom Autor entwickelte Blickrichtungssteuerung für autonome Fahrzeuge. Dabei wird in den Abschnitten 4.1 und 4.2 zunächst das Fahrzeugauge MARVEYE und das eingebettete System zur Blickrichtungssteuerung vorgestellt. Daran anschließend beschreibt Abschnitt 4.3 den Aufbau der Blickrichtungssteuerung und Abschnitt 4.4 dessen Schnittstelle zu den Wahrnehmungsprozessen. Der Algorithmus zur Optimierung des Blickverhaltens ist Thema des Abschnitts 4.5. Die abschließenden Unterkapitel 4.6 und 4.7 stellen die Fertigkeiten und Fähigkeiten der Blickrichtungssteuerung heraus.

Kapitel 5 beschreibt das EMS-VISION System, welches die experimentelle Grundlage dieser Arbeit darstellt. Im Anschluss daran werden in den Kapiteln 6 und 7 die experimentellen Ergebnisse diskutiert. Kapitel 8 beinhaltet eine Zusammenfassung der Arbeit und zählt mögliche Weiterentwicklungen auf.

# Kapitel 2

## Stand der Technik

Dieser Abschnitt beschreibt und benennt Literatur, welche für die Thematik und die Ergebnisse dieser Arbeit relevant sind. Im Abschnitt 2.1 wird zunächst erörtert, was unter dem Begriff ‚Aktives Sehen‘ zu verstehen ist und welche Ziele damit verfolgt werden können. Abschnitt 2.2 stellt eine Auswahl neuerer Arbeiten vor, die das Paradigma ‚Aktives Sehen‘ für die Steuerung und Regelung von Fahrzeugen und mobilen Robotern benutzen. Abschnitt 2.3 geht auf die Modellierung von Wissen und Verhalten ein. Dabei wird in Unterabschnitt 2.3.1 zunächst die Informationsverarbeitung des Menschen betrachtet, bevor Unterabschnitt 2.3.2 Modellierungsmethoden für Wissen, Abläufe und Verhalten in technischen Systemen vorstellt. Die in dieser Arbeit aufgegriffene Modellierungsmethode durch unscharfe Mengen und Regeln wird in Abschnitt 2.3.3 ausführlicher behandelt. Abschnitt 2.3.4 beschreibt alternative Architekturen zur Steuerung autonomer Systeme. Zum Abschluss gibt Kapitel 2.3.5 relevante, weiterführende Literatur zum Thema autonome Fahrzeugführung an.

### 2.1 Was ist ‚Aktives Sehen‘?

#### 2.1.1 Entstehung des Paradigmas ‚Aktives Sehen‘

Seit über 30 Jahren versucht der Mensch, Maschinen mit einem visuellen Sensor auszustatten. Kurz gesagt: „*Maschinen sollen Sehen lernen*“. Nach der anfänglichen Euphorie in den 60er Jahren tauchten beim Versuch, dieses Ziel zu erreichen, viele unerwartete Schwierigkeiten auf. Bis dahin waren mit Rechnern im Wesentlichen arithmetische Aufgaben gelöst worden. Verglichen damit erwies sich die Gewinnung von Informationen aus Bildern als erheblich schwieriger [BERNASCH, 1997].

Zu Beginn der 60er Jahre begann man damit, einfache Szenen mit geometrisch geformten Körpern mit homogenen Farbflächen und sauberen Kanten mittels Bildverarbeitung zu erkennen und zu lokalisieren. Dabei wurden Ansätze verfolgt, die mittels Wissen über die Reflexionseigenschaften und die Beleuchtung der Szene und den Standpunkt des Beobachters aus den Kanten und Grauwerten eines Bildes die visuelle Welt rekonstruieren. Ein prominenter Vertreter dieses *Rekonstruktionsansatzes* ist MARR [1982]. Der Rekonstruktionsansatz ist nach DENZLER [1997] folgendermaßen motiviert: Hat man eine vollständige Rekonstruktion und somit eine genaue Repräsentation der 3D-Welt, so kann mittels dieser Information jede visuelle Aufgabe gelöst werden. Diese Sichtweise spiegelt sich auch in der Aufgabenbeschreibung für das Rechnersehen von CHARNIAK & MCDERMOTT [1985] wieder:

„*Give a two-dimensional image, infer to object that produced it, including their shapes, positions, colors, and sizes.*“

Viele Projekte der 60er Jahre lösten solche Probleme sehr erfolgreich [BERNASCH, 1997]. Daraufhin sah man das prinzipielle Problem des Sehens schon fast als gelöst an. Die Erweiterung der Anwendungen auf komplexere, natürliche Welten schien durch eine konsequente Weiterentwicklung der Algorithmen möglich zu sein. So versuchte man auch in komplexen Umgebungen Objekte mit ihrer Form und Orientierung zu erkennen.

Die starke Konzentration auf das allgemeine Rekonstruktionsproblem führte zu vielen speziellen Lösungen (z. B. *stereo motion*, *shape from shading*, *shape from contour*, *edge detection*, *regularization*, *optical flow*, *structure from motion*)<sup>1</sup>, die zwar in stark eingeschränkten Umgebungen funktionieren, aber Eigenschaften besitzen, die sie in natürlichen Umgebungen scheitern lassen. Nach BERNASCH [1997] sind dies die zu wenig allgemeingültigen Voraussetzungen an die Umgebung, die Mehrdeutigkeiten in den Ergebnissen mancher Methoden (ohne einschränkende Annahmen) und die Instabilität gegenüber Messrauschen. DENZLER [1997] sieht die Ursachen darin, dass es sich bei der visuellen Rekonstruktion um ein schlecht konditioniertes und stark unterbestimmtes Problem handelt. Der Autor zeigt, dass die oben genannten Ansätze wie z. B. *shape from shading* erst durch einen aktiven Beobachter besser konditioniert und eindeutig lösbar werden. In anderen Arbeiten versuchte man, die Mehrdeutigkeiten durch die Kombination mehrerer bildbezogener Informationen aufzulösen, z. B. beim *shape from photometric-stereo* oder *shape from shading and optical flow* (vgl. [PAULI, 1993]).

Aus allen diesen Gründen war ein Übertrag der in den künstlichen Umgebungen so erfolgreichen Ansätze auf komplexe Szenen unter Beibehaltung des Rekonstruktionsansatzes nicht möglich. So kam es, dass der Fortschritt im Bereich Bildverarbeitung für natürliche Umgebungen im Vergleich zur Leistungssteigerung im Software- und Hardwarebereich eher gering war. Dieser geringe Fortschritt motivierte dazu, das funktionale Verständnis des *menschlichen* Sehens bei der Entwicklung von Algorithmen und Methoden des *maschinellen* Sehens mit einzubeziehen. Man erkannte im zunehmenden Maße die gegenseitige Abhängigkeit von visueller Aufgabe und aktivem Handeln des Beobachters. Auf der anderen Seite ermöglichte erst die im vergangenen Jahrzehnt gestiegene Rechenleistung eine Rückkopplung der Bilddaten über die Verarbeitungsebenen eines Robotersystems und eine Interaktion zwischen sehendem Roboter und Umwelt. Der Begriff *Aktives Sehen* (*active vision*) war geboren [BAJCSY, 1988].

### 2.1.2 Das Vorbild in der Natur: Das Sehverhalten der Wirbeltiere

Die Natur liefert viele Argumente für das Aktive Sehen: So dient die visuelle Wahrnehmung von Lebewesen immer einem konkreten Zweck. Dies spiegelt sich in der Tatsache wieder, dass Lebewesen in unterschiedlichen Lebensräumen und mit unterschiedlichen Fress- und Jagdverhalten verschiedene visuelle Sensoren besitzen. Bei jagenden Tieren (z. B. Katze und Eule) sind die Augen parallel angeordnet, um ein möglichst gutes räumliches Sehen zu erreichen. Dagegen sind bei Beutetieren (z. B. Hase und Reh) die Augen fast entgegengesetzt ausgerichtet, um in einem Gesichtsfeld von annähernd 360° Feinde aus allen Richtungen erkennen zu können. Nach TARR & BLACK [1994] ist im visuellen System des Frosches ein Detektor für Nahrung direkt realisiert. Auch der Mensch besitzt kein allgemeines Sehsystem. So kann er nicht Licht aller Wellenlängen wahrnehmen, hat unter Wasser nur eingeschränkte Sicht und ein Farbempfinden, welches auf kein objektives Maß zurückzuführen ist [DENZLER, 1997].

---

<sup>1</sup> Eine gute Darstellung der Rekonstruktionsansätze „*shape-from-X*“ befindet sich in [PAULI, 1993].

Beim menschlichen Auge ist die Dichte der lichtempfindlichen Rezeptoren (Stäbchen und Zapfen) in der Netzhaut (Retina) ortsabhängig. Die Rezeptordichte ist nahe der optischen Achse ca. 147 000 Rezeptoren/mm<sup>2</sup>. Dieser zentrale Bereich wird *fovea centralis* oder kurz Fovea genannt und deckt einen Winkelbereich von ca. 2° ab [STARK & CHOI, 1996]. In der Fovea findet mittels der Zapfen das photopische Sehen statt, mit dem Helligkeit und Farbe wahrgenommen werden kann [SCHIEHLEN, 1995]. Außerhalb der Fovea fällt die Dichte der Rezeptoren bereits nach wenigen Millimetern auf weniger als 10 000 Rezeptoren/mm<sup>2</sup> ab [WENZEL, 1995]. Das periphere Gesichtsfeld des menschlichen Auges deckt einen Winkelbereich von ca. 180° ab. Im peripheren Gesichtsfeld findet mit den Stäbchen skotopisches Sehen statt, wobei nur die Helligkeit wahrgenommen wird, dies aber auch bei schlechter Beleuchtung (Dämmerung). Das skotopische Sehen ist ferner besonders empfindlich gegenüber Bewegungen in der Außenwelt. Nicht nur der Mensch besitzt Augen mit ortsabhängiger Auflösung, sondern dies ist ein Charakteristikum aller Wirbeltiere [PAHLAVAN, 1996].

Zum Zweck der Objekterkennung muss der Mensch die Objekte fixieren, d. h. diese auf den Bereich des schärfsten Sehens mit maximaler Rezeptordichte abbilden (foveales Sehen). Dies macht eine ständige Augenbewegung notwendig. Der Mensch führt beim natürlichen Umherschauen etwa 3 - 5 Sakkaden pro Sekunde aus [FISCHER, 1996]. Zwischen den Blicksprüngen fixiert und fokussiert das Auge. Während der Fixation eines Objekts mit der Fovea können parallel dazu weitere bekannte Objekte im Gesichtsfeld mit stark reduzierter Auflösung verfolgt werden. Gleichzeitig ziehen plötzliche Änderungen im Gesichtsfeld hinsichtlich Helligkeit und Bewegung die Aufmerksamkeit auf sich. Diese Aufmerksamkeit führt zu einer kurzzeitigen Fixation und intensiven Verarbeitung des betreffenden Bereichs, bis klar ist, wodurch die Aufmerksamkeit erregt wurde. Die Latenzzeit (Reaktionszeit) zwischen dem auslösenden Reiz auf der Netzhaut und dem Auslösen der Sakkade beträgt beim Menschen 150 - 350 ms [PAHLAVAN, 1996]. Die Geschwindigkeit von Sakkaden liegt meist im Bereich 200 - 400 °/s und kann bei großen Amplituden 700 °/s erreichen. Die Dauer einer Sakkade wächst annähernd proportional mit der Amplitude und liegt zwischen 20 - 100 ms [SCHIEHLEN, 1995]. Trifft eine Sakkade den Zielpunkt nicht genau, so findet anschließend eine kleine Korrektursakkade statt (Mikrosakkade). Zusätzlich wird durch schnelle Augenbewegungen mit kleinen Amplituden (Augentremor) eine Reizauswaschung durch ein zu konstantes Bild auf der Netzhaut vermieden [FLEMISCH, 2001].

Den ständigen Wechsel zwischen Fixation und Sakkade leistet ein optomotorischer Zyklus, der im Allgemeinen automatisch und ohne bewusstes Zutun im Gehirn abläuft [FISCHER, 1996]. Das Blickverhalten des Menschen ist so perfekt gestaltet, dass sich dieser über die abnehmende Auflösung seiner Augen abseits der optischen Achsen und seiner schnellen Augenbewegungen gar nicht bewusst ist. Dies liegt nach PAHLAVAN [1996] daran, dass die Sakkadendauer kleiner ist, als die Totzeit im visuellen Rückkopplungskreis. Der Mensch hat einen Eindruck von Klarheit, Vollständigkeit und Kontinuität, als sehe er in kontinuierlicher Weise das gesamte Gesichtsfeld mit der hohen Auflösung der Fovea [STARK & CHOI, 1996].

Im Gegensatz dazu ist sich das menschliche Gehirn der ortsabhängigen Auflösung der Augen und dem begrenzten Öffnungswinkel der Fovea wohl bewusst, so dass es Schlüsseigenschaften der Szene, welche aus früheren Wahrnehmungen oder aus der Erwartung heraus bekannt sind, kontinuierlich abtastet. 10 % der Zeit nimmt dabei die Ausführung der Sakkaden ein; in dieser Zeit findet keine Informationsaufnahme statt (sakkadische Suppression)<sup>2</sup>. In den restlichen 90 % der Zeit fixiert er Objekte oder Merkmale in der Umgebung. Die maximale Geschwindigkeit der Folgebewegung beträgt 80 - 160 °/s [PAHLAVAN, 1996]. Wird während

---

<sup>2</sup> Aus diesem Grund handelt es sich bei der Sakkadenbewegung um eine reine Steuerung.

der Fixation die Lageabweichung zu groß, so dass das fixierte Objekt die Fovea zu verlassen droht, so führt dies zu einer Korrektursakkade.

Psychologische Untersuchungen des menschlichen Blickverhaltens haben zur Entwicklung der sogenannten *scanpath*-Theorie geführt, nach der ein internes kognitives System die Suchpfade des Auges kontrolliert und nicht die externe Welt. STARK & CHOI [1996] unterscheiden zwischen ‚*searchpaths*‘ und ‚*scanpaths*‘. *Searchpaths* sind strukturell ähnlich den *scanpaths* und in gleicher Weise definiert als „*repetitive, idiosyncratic sequences of saccades alternating with fixation*“. Der Unterschied zwischen beiden besteht darin, dass *searchpaths* durch räumliche Modelle generiert und kontrolliert werden und nicht wie *scanpaths* von symbolischen, kognitiven Modellen. Nach den Ausführungen von STARK & CHOI [1996] hat sich in Experimenten gezeigt, dass der Mensch mit räumlichen Modellen in der Lage ist, sich an bis zu 6 - 7 Orte mit interessanten Objekten oder Merkmalen (*region of interest* (ROI)) zu erinnern. Wenn die Anzahl der ROI größer als 7 wird, kann das räumliche Modell diese Komplexität nicht mehr handhaben. Sie führt dazu, dass sich damit regelmäßig wiederholende Sprungsequenzen nicht mehr ausführen lassen. Das Gehirn des Menschen löst dieses Problem dadurch, dass die ROI Objekten, Merkmalmustern oder hierarchischen Merkmalgruppen zugeordnet werden, so dass ein kognitiver Prozess sie symbolisch repräsentieren und handhaben kann. Das Gehirn des Menschen operiert in diesem symbolischen Schauplatz und generiert eine Sequenz von Abtastungen, um Wissen aus der Umwelt aufzunehmen und in regelmäßiger Art und Weise immer wieder neu zu bestätigen. STARK & CHOI [1996] geben weiterführende Literatur an, welche diese kognitiven Modelle des menschlichen Sehens beschreibt.

Neben der Steuerung der Wahrnehmung durch Reize aus der Umgebung (*bottom-up*) kann der Mensch seine Blickrichtung auch willentlich festlegen (*top-down*), indem er z. B. bestimmte Dinge in der Umgebung bewusst anschaut und seinen Blick darauf fixiert und andere wiederum völlig ignoriert.

Nach BROCKMANN et al. [1999] müssen Veränderungen des Aufmerksamkeitsfokus von den Sakkaden der Augenbewegung klar getrennt werden. Die Autoren zitieren tierexperimentelle und neuropsychologische Untersuchungen, die eine komplexe Wechselwirkung zwischen diesen beiden Arten aufzeigen. Diese Wechselwirkung hängt von der zu lösenden Aufgabe und der Art der betrachteten Szene ab. Laut BROCKMANN et al. [1999] zeigen psychophysische Experimente, dass die menschliche Aufmerksamkeit eine parallele und eine serielle Verarbeitungsstufe besitzt. Die parallele Stufe detektiert auf schnelle und effiziente Weise unerwartete oder herausragende Teile der Eingabe (z. B. Farbe, Orientierung, Symmetrie) und richtet unwillkürlich die Aufmerksamkeit darauf (*bottom-up* Aktivierung). Daneben kann eine serielle Stufe Erwartungen über die Stimuli mit einbeziehen (*top-down* Aktivierung). Die Parallelität der Aktivierungsmechanismen (*bottom-up* und *top-down*) verbessert die visuelle Verarbeitungsleistung erheblich. Die serielle Stufe wird von der parallelen Stufe geführt und verarbeitet deren Information weiter. Die serielle Stufe zeichnet sich auch dadurch aus, dass sie nach Kombinationen von Merkmalen und subtileren Unterschieden suchen kann.

Aufmerksamkeitswechsel dienen beim Menschen der Vorbereitung einer sakkadischen Augenbewegung (offenkundiger Aufmerksamkeitswechsel), können aber auch rein mental, d. h. ohne nachfolgende Sakkade ablaufen (verdeckter Aufmerksamkeitswechsel). Die sogenannten Reflexsakkaden werden extern durch ein plötzlich auftauchendes Blickziel getriggert. Ihnen geht eine reflexartige Ausrichtung des Aufmerksamkeitsfokus voraus. Davon zu unterscheiden sind nach BROCKMANN et al. [1999] die Willkürsakkaden, die intern getriggert werden und denen eine willkürlich initiierte Verlagerung des Aufmerksamkeitsfokus auf ein schon längere Zeit sichtbares oder erinnertes Blickziel vorausgeht. Willkürsakkaden werden z. B. bei einer Exploration der Szene ausgeführt. HAMKER & GROß [1996] nennen die reflexarti-

ge Zuwendung der Aufmerksamkeit auch *passive Aufmerksamkeit*, im Gegensatz zur intentionalen oder selektiven Aufmerksamkeit, die als *aktive Aufmerksamkeit* bezeichnet wird.

Nach FISCHER [1996] besteht das Sehsystem von Primaten aus einem Sakkaden- und einem Fixationssystem, welche sich gegenseitig hemmen. Ein hierarchisch höher angeordnetes Aufmerksamkeitssystem mit einer räumlichen Organisation hemmt ebenfalls das Sakkadensystem. Dies verhindert, dass jeder visuelle Reiz, der auf die Netzhaut fällt, eine Sakkade auslöst. Ein ebenfalls hoch angeordnetes System für bewusste Entscheidungen greift zusätzlich in die Blickrichtungssteuerung ein, damit ein ausgewählter Punkt fixiert werden kann. Der Autor leitet diese Erkenntnisse aus Untersuchungen mit gesunden und kranken Menschen ab.

FISCHER [1996] untersucht ferner den sogenannten optomotorischen Reflex, mit dessen Hilfe ein plötzlich erscheinender Lichtreiz durch eine sogenannte Express-Sakkade extrem schnell fovealisiert werden kann. Express-Sakkaden lösen beim Menschen nach einer sehr kurzen Reaktionszeit von etwa 100 ms aus. Der optomotorische Reflex wird durch einen Fixationsprozess gehemmt, mit dessen Hilfe der Blick auf einen bestimmten Reiz gerichtet und stabilisiert bleibt. Der Blick wird erst wieder für andere Objekte und Merkmale freigegeben, wenn der Fixationsprozess aufgehoben ist. Sehr kleine Sakkaden, deren Ziele zu nahe bei der Fovea erscheinen, können keine Express-Sakkaden sein, weil der Reiz selbst das Fixationssystem und damit eine Hemmung des Sakkadensystems anregt. Die Expressreaktion wird ferner durch Aufmerksamkeit gehemmt, die willkürlich auf einen Reiz im peripheren Gesichtsfeld gerichtet ist. Auch hier muss vor der Sakkade die Loslösung der Aufmerksamkeit stattfinden.

Weit verbreitet ist die sogenannte *Spotlight*-Annahme bezüglich der menschlichen Aufmerksamkeitsverteilung. Diese Metapher besagt, dass der menschliche Aufmerksamkeitsfokus (*focus of attention*) das Gesichtsfeld durchstreift, wie ein Schweinwerferkegel eine Bühne. Das „*Spotlight*“ ist in sich homogen, besitzt eine kompakte Form und durchleuchtet die Szenen auf unterschiedlichen Abstraktionsebenen. Eine Folge der *Spotlight*-Annahme ist es, dass Aufmerksamkeit nicht gleichzeitig für verschiedene, räumlich voneinander getrennte Regionen des Gesichtsfeldes hergestellt werden kann. MÜSSELER & ASCHERSLEBEN [1996] widersprechen der *Spotlight*-Annahme und zeigen eine unstetige räumlich/zeitliche Spezifizierung des menschlichen Zuwendungsprozesses.

HUMPHREYS et al. [1996] untersuchen die menschliche Verarbeitung visueller Reize anhand psychologischer Experimente. Diese Experimente belegen, dass einfache visuelle Merkmale wie Kanten bestimmter Ortsfrequenzen, Farben, Farbflecke, Inhomogenitäten in der Textur oder andere von der Orientierung des Objekts unabhängige beobachtbare Eigenschaften (Invariante) als Reiz wirken und die Aufmerksamkeit auf sich ziehen (*preattentive saccades*). Diese einfachen visuellen Reize werden durch spezielle Detektoren aufgenommen, welche parallel über das gesamte Gesichtsfeld arbeiten. Daneben können aber auch komplexere Eigenschaften, wie z. B. der räumliche Zusammenhang einer Gruppe von Merkmalen, ebenfalls als Reiz wirken (*attentive saccades*). Für diese Detektion von 3D-Information muss nach HUMPHREYS et al. [1996] eine Beschreibung kompletter Objekte auf einer höheren Repräsentationsebene existieren. Die Reizverarbeitung mit 3D-Information führt z. B. dazu, dass einem Menschen falsche 3D-Effekte in Skizzen sofort „ins Auge stechen“. Diese 3D-Objektinformation kann nur durch kognitiv aufwendigere und langsamere Detektionsprozesse gewonnen werden, welche sich nicht parallel über das gesamte Gesichtsfeld erstrecken, sondern eine sequentielle Suche erfordern. Nach HUMPHREYS et al. [1996] wirkt sich die Anzahl der visuellen Reize, die auf den Menschen einwirken, auf dessen Detektionsstrategie aus.

MÜSSELER & ASCHERSLEBEN [1996] und FISCHER [1996] zeigen in Experimenten, dass die Aufmerksamkeit des Menschen durch Hinweisreize gelenkt werden kann. Die durch den Hinweisreiz erzielte vorzeitige Aufmerksamkeitsausrichtung ist sowohl in Reaktionszeitverkürzungen, als auch in Identifikationserleichterungen nachweisbar. Der Mensch nimmt also einen Reiz besser wahr, wenn er zuvor Informationen darüber erhält, wo er erscheinen wird. Dies beweist, dass die Aufmerksamkeitssteuerung des Menschen auch auf räumliche Information zurückgreift.

Neben der Augenbewegung, die eine Rotationsamplitude von ca. 20° abdeckt, besitzt der Mensch die Möglichkeit der Kopfbewegung. Durch die Kopfbewegung wird sein Gesichtsfeld nochmals vergrößert. Verfolgt der Mensch mit seinem Augenpaar ein sich bewegendes Objekt, so führt die Bewegung zu einer Asymmetrie in der Vergenz [PAHLAVAN, 1996]. Durch die der Augenbewegung niederfrequent unterlagerten Kopfbewegung kann während der Fixation diese Asymmetrie ausgeglichen werden, so dass die Bilder des linken und rechten Auges möglichst identisch sind. Die Augen-Kopf-Koordination erhöht auch die Schnelligkeit von Blickrichtungsänderungen (Summe von Augenbewegung und Kopfbewegung). So unterstützt der Mensch Augensakkaden mit einer Amplitude größer als 15° durch Kopfwendungen [SCHIEHLEN, 1995]. Die Kopfbewegung kann auch der Stabilisierung des Bildes auf der Netzhaut dienen. Diese Art der Stabilisierung ist besonders bei Kamelen (vertikale Kopfbewegungen) und Tauben (longitudinale Kopfbewegungen) ausgeprägt [PAHLAVAN, 1996].

Die sogenannten Nystagmen beseitigen die Bewegungsunschärfe der Bilder auf der Netzhaut. Der optokinetische Nystagmus tritt auf, wenn sich alle Objekte bezüglich eines statischen Beobachters bewegen, wie z. B. bei einem Blick aus einem fahrenden Zug. Läuft ein Objekt aus dem Sichtbereich heraus, so wird durch eine Sakkade ein neuer Punkt fixiert. Diese reflexartigen Augenbewegungen laufen völlig unbewusst ab. So ist es nicht möglich, die Augen bewusst still zu halten, auch wenn die sichtbaren Gegenstände völlig uninteressant sind [SCHIEHLEN, 1995]. Der vestibulär-okulare Nystagmus kompensiert schnelle Kopfbewegungen und nutzt dazu die Gleichgewichtsorgane im Ohr. Die *Bogengangorgane* werden von Winkelbeschleunigungen angeregt und die *Maculaorgane* von translatorischen Beschleunigungen. Auch der vestibulär-okulare Nystagmus ist reflexartig, was man daran erkennt, dass dieser auch bei völliger Dunkelheit ausgeführt wird.

### 2.1.3 Eigenschaften des Aktiven Sehens

Wie die vorangegangenen Kapitel zeigen, ist Sehen nichts Statisches und immer an eine bestimmte Aktion geknüpft. Viele Aufgaben lassen sich nur oder zumindest einfacher lösen, wenn man mit der Umwelt interagiert. Als Beispiel führt DENZLER [1997] das Einparken an: Ein Autofahrer, der in eine Parklücke einfahren möchte, versucht nicht durch passives Wahrnehmen die externe Welt vollständig zu rekonstruieren und erst dann zu agieren. Vielmehr wendet er eine wesentlich aktivere Strategie an: Er schätzt während des Einparkens ständig die Relativposition seines Fahrzeugs zur Parklücke und zu den anderen parkenden Autos. Damit korrigiert und konkretisiert der Fahrer die fehlenden oder fehlerbehafteten Informationen vom Beginn des Einparkmanövers (nicht „sehen, dann fahren“, sondern „sehen und fahren“). Dies hat zur Folge, dass anfänglich schlecht oder gar nicht abschätzbare Größen (wie z. B. die Breite und Tiefe der Parkbucht) erst während der Aktion ermittelt werden.

ALOIMONOS [1993] betont, dass bei der Konzeption von intelligenten Sehsystemen die Tatsache zu berücksichtigen ist, dass die Physiologie des Betrachters den Aufgaben angepasst ist, die dieser zu erfüllen hat. Ein allgemeines Sehsystem (*general observer*) ist nach Ansicht des Autors ein theoretisches Konzept, welches in der Natur nicht existiert und aus vielerlei Grün-

den auch nicht künstlich realisiert werden kann. So gibt es nach Meinung des Autors einfach zu viel über die Welt zu erfahren, um eine allgemeingültige und aufgabenunabhängige Beschreibung aufzubauen. Bezüglich Bildverarbeitung bedeutet dies, dass es zu viele Möglichkeiten gibt, Bildinformation auszuwerten, um eine aufgabenunabhängige Beschreibung der Welt zu rekonstruieren [SCHIEHLEN, 1995]. Das eigentliche Problem ist die Entscheidung, welche Bildinformation zu berechnen ist, um eine gewünschte Interaktion zwischen dem sehenden System und der Umwelt zu ermöglichen. Sehsysteme kann man damit nicht ohne Berücksichtigung der Umwelt konzipieren und betrachten.

TSOTSOS [1989] beschreibt ebenfalls die inhärente, exponentiell wachsende Komplexität des maschinellen Sehens, die eine reine *bottom-up* Problemlösung verhindert. Nach der Ansicht des Autors kämpfen sowohl biologische als auch maschinelle Sehsysteme mit dieser Komplexität, welche nur durch ein hierarchisch organisiertes, abstrahiertes Hintergrundwissen, durch eine quantitative Begrenzung der Sensordaten und durch eine selektive Berechnung von Information zu begegnen ist. Der Autor betont auch, dass bei einer aufgabenorientierten Wahrnehmung vorhandene Mehrdeutigkeiten in der Interpretation der visuellen Information aufgehoben werden. Diese Mehrdeutigkeiten äußern sich z. B. beim Menschen in Form von Fehlinterpretationen bei der Wahrnehmung von völlig Unerwartetem.

BALLARD & BROWN [1982] sind der Meinung, dass es beim Rechnersehen von elementarer Bedeutung ist, sowohl low-level (*early processing*) als auch kognitives (*high-level*) Wissen über die Welt zu nutzen. Nur die Repräsentation und Kooperation beider Arten von Wissen könne zu einem ausgefeilten Sehsystem führen. Je höher die Ebene der Repräsentation ist, desto ausgefeilter ist der Kontrollfluss hinunter zu den tieferen Ebenen (Aufmerksamkeitssteuerung, Allokierung von Verarbeitungsleistung) und umso größer ist die Tendenz der Algorithmen zur seriellen Verarbeitung. Nach Ansicht der Autoren kann das Sehen nicht von anderen kognitiven Fähigkeiten wie das logische Denken oder von der Motivation und Intention des Subjekts getrennt werden.

Aktives Sehen wird oft als ein neues Paradigma in der Bildverarbeitung angesehen, dessen Ziel nicht mehr die vollständige Rekonstruktion der visuellen Welt durch die Invertierung des bildgebenden Prozesses ist, sondern von einem aktiven Beobachter ausgeht, der anhand seiner Ziele und Erfordernisse die Analyse des Bildstromes über die Steuerung der Bewegung des Bildsensors oder des Bildausschnittes beeinflusst. VIÉVILLE [1997] vertritt den Standpunkt, dass durch reaktive Mechanismen die Komplexität des maschinellen Sehens reduziert werden kann und dass sich Aktives Sehen durch eine Rückkopplung der visuellen Information auszeichnet. Diese Rückkopplung kann die Lokalisierung, Orientierung und Bewegung des visuellen Sensors, die Eigenschaften der Bildaufnahme (Zoom, Schärfe, Blende und Beleuchtung) und/oder den internen Zustand der Bildverarbeitung beeinflussen. RIMEY & BROWN [1992] stellen die wesentlichen Unterschiede zwischen dem Rekonstruktionsansatz nach MARR [1982] – oft auch als ‚passives Sehen‘ bezeichnet – und einem ‚aufgabenbezogenen Sehen‘ (*task-oriented vision*) wie folgt gegenüber:

passives Sehen	aufgabenbezogenes Sehen
setzt immer alle Sehfähigkeiten ein	setzt nur passende Sehfähigkeiten ein
es wird immer das gesamte Bild ausgewertet	es werden nur Bildausschnitte ausgewertet
arbeitet mit maximaler Auflösung	arbeitet mit ausreichender Auflösung (dynamisch)
zuerst wird Repräsentation berechnet	zuerst wird nach der Aufgabe gefragt
die Repräsentation dient zur Lösung der Aufgabe	ermittelte Szenendaten dienen zur Lösung der Aufgabe
die Begrenzung der Ressourcen geht nicht ein	es wird von begrenzten Ressourcen ausgegangen

Tab. 2.1: Passives und aufgabenbezogenes Sehen nach [RIMEY & BROWN, 1992]

Passives Sehen sollte aber nicht als Gegenteil des Aktiven Sehens betrachtet werden. Dies äußert sich z. B. in der Tatsache, dass der aktiv sehende Mensch auch ohne Augenbewegungen Objekte in seiner Umgebung erkennen kann. BALLARD & BROWN [1993] betrachten beim Aktiven Sehen die Außenwelt als Quelle aller Information, welche jederzeit abrufbar ist. Aus diesem Grund müsse die Außenwelt nicht intern repräsentiert werden. Dahinter steckt die Idee, die Welt als ihre eigene Repräsentation zu benutzen, auf welche man durch Echtzeitsehen zugreifen kann.

BARTOFF et al. [1999] deuten Aktives Sehen auf zwei komplementäre Arten: als Sehen zur Unterstützung von Handlungen („*vision for action*“) oder als kontrollierte Handlung zur Stabilisierung des Sehens („*action for vision*“). DENZLER [1997] stellt eine Vielzahl von Definitionen des Begriffs ‚Aktives Sehen‘ gegenüber, wie man sie in der Literatur findet. Er betont jedoch, dass sich bisher keine allgemein anerkannte, vom Anwendungsgebiet unabhängige Definition herauskristallisiert hat. Andere, ebenfalls im Zusammenhang des Aktiven Sehens genannte Begriffe sind z. B. ‚zielgerichtetes Sehen‘ (*purposive vision*) oder *animate vision*.

Der Begriff des Aktiven Sehens ist eng mit dem Begriff der Aufmerksamkeitssteuerung verknüpft. Allgemein spricht man bei einer Aufmerksamkeitssteuerung von der Fähigkeit zur aufgabenbezogenen Selektion bestimmter Information aus einem (großen) Informationsangebot. Die Information muss dabei nicht visueller Natur sein. Nach HAMKER & GROß [1996] ist Aufmerksamkeit keine Frage von „begrenzter Kapazität“, sondern eine Frage von gewollter und nicht gewollter Information für weitere Handlungen. In der Kognitionspsychologie sieht man Aufmerksamkeit als selektive Wahrnehmung oder die selektive Antwort auf einen bestimmten Reiz. Psychologische Theorien betrachten Aufmerksamkeit als Fähigkeit und/oder Ressource mit Information umgehen zu können [ULTES, 2002]. Bei allen diesen Auffassungen wird durch eine Steuerung der Aufmerksamkeit die angebotene Information mit einer Handlung verknüpft. Bei den Begriffen ‚selektive Wahrnehmung‘ (*selective perception*), ‚intelligente Wahrnehmung‘ (*smart sensing*) und ‚aktive/erkundende Wahrnehmung‘ (*active/exploratory perception*) wird z. B. neben dem Sehen auch das Hören, Greifen, Berühren oder Tasten in die Aktivität miteinbezogen [DENZLER, 1997].

Eine Blickrichtungssteuerung ist eine spezielle Form von Aufmerksamkeitssteuerung, welche ausschließlich aus visueller Information auswählt und deren Aktionen auf die Ausrichtung des visuellen Sensors und dessen inertielle Stabilisierung beschränkt sind. Daneben gibt es noch weitere Möglichkeiten, den Bildaufnahmeprozess zu beeinflussen: die Steuerung von Blende, Kontrast, Fokus und der Position der Kamera. In Anlehnung an den menschlichen Sehapparat werden oft Stereokameraköpfe mit steuerbarem Fokus und Vergenzwinkel als wesentliche Bestandteile eines aktiven Sehsystems genannt.

### 2.1.4 Entwicklungsziele für aktiv sehende Systeme

Ein aktiver Beobachter hat mehrere Möglichkeiten, seinen Sensorinput zu seinem Vorteil zu beeinflussen. Durch die Wahl seines Standortes und seines Blickwinkels kann er zielgerichtet die gewünschte perspektivische Ansicht eines Objekts herstellen. Zum anderen werden nur die für die spezielle Aufgabe notwendigen Sensordaten verarbeitet, was die Datenmenge stark reduziert und einen in Echtzeit arbeitenden, maschinellen Beobachter ermöglicht. BERNASCH [1997] sieht folgende Aufgaben und Möglichkeiten des Aktiven Sehens:

- Es sind Strategien zur Sensornutzung zu entwickeln, die es ermöglichen, auf der Basis der bereits visuell gewonnenen Information durch inkrementelle Bewegungen des Sensors in den 6 Freiheitsgraden (in Echtzeit) neue, relevante Information zu gewinnen.
- Durch die gesteuerte Bewegung des Sensors und durch die Analyse von Bildfolgen können Objektinformationen gewonnen werden, die sich aus Einzelbildern nicht extrahieren lassen (z. B. Relativbewegung zwischen Objekt und Beobachter). Das Verfolgen (*tracking*) von einmal detektierten Merkmalen reduziert die Komplexität des Problems. Durch Kalman-Filter ist es möglich, die Verfolgung von Objekten durch die Vorhersage der Objektmerkmale und durch die Einbeziehung der Sicherheit des Wissens über die Merkmale zu vereinfachen [DICKMANN, 1987].
- Die Steuerung der Aufmerksamkeit soll es höheren Bildverarbeitungsschichten ermöglichen, sich auf die Verarbeitung weniger aber wesentlicher Daten zu beschränken. Dabei ist es notwendig, die Selektion und Lokalisation wesentlicher Information durch eine einfache Analyse der gesamten Eingabe durchzuführen, so dass bei der Datenreduktion keine wesentliche Information verloren geht. Die Aufmerksamkeitssteuerung detektiert und identifiziert ständig neue relevante Objekte und verbindet die aufeinanderfolgenden Bildauswertungen zu einem schlüssigen Ganzen.

Nach DENZLER [1997] führt das Aktive Sehen auf der Ebene der signalnahen Verarbeitung eine Selektion in Raum, Zeit und Auflösung durch. Die Selektion im Raum bewirkt, dass nur gewisse für die Lösung der Aufgabe relevante Bildteile verarbeitet werden. Die Selektion in der Auflösung ermöglicht es, aufgabenangepasst mit unterschiedlichen Auflösungen zu arbeiten<sup>3</sup>. Auch die Selektion über der Zeit folgt dem Prinzip, nur das Notwendige zu berechnen: Falls sich die aufeinanderfolgenden Bilder einer Bildfolge nur wenig unterscheiden, so ist es nicht notwendig, jedes Bild vollständig neu zu bearbeiten. Damit aber dabei keine neue, relevante Bildinformation verloren geht, müssen parallel dazu Mechanismen zur Aufmerksamkeitssteuerung vorgesehen werden. Ansätze zur Aufmerksamkeitssteuerung arbeiten z. B. mit einem „*Hypothese und Test*“-Mechanismus [TSOTSOS, 1989], assoziativen Speichern, neuronalen Netzen, Bewegungsinformation (Differenzbilder, optische Flussfelder) oder Strategien zur perzeptuellen Gruppierung (Objekt-Hintergrund-Trennung) [WETZEL, 1995].

Nach HAMKER & GROB [1996] ist die Merkmalextraktion ein unverzichtbarer Teil des Aktiven Sehens und dürfe niemals passiv ablaufen. So sind je nach Situation und Intention verschiedene Merkmale unterschiedlich bedeutsam und deshalb müsse die Handlung und Situation bereits bei der Merkmalextraktion Berücksichtigung finden. Nach Ansicht der Autoren ist die Erkennung eine erlernbare Handlung, durch die selektiv die Umwelt nach neuer Information abgefragt wird.

---

<sup>3</sup> Hätte die Netzhaut des menschlichen Auges die konstant hohe Auflösung ihres fovealen Bereichs, so müsste nach ALOIMONOS [1993] das menschliche Gehirn für die Verarbeitung der Datenmenge 30 000 Pfund wiegen.

BERNASCH [1997] sieht die selektive *Fixationssteuerung* und das *Verfolgen von Objekten* als wesentliche Bestandteile eines aktiven und effizienten Bildverarbeitungssystems. Höhere Systemebenen haben auf die Fixationssteuerung nur geringen Einfluss und beschränken sich auf die Auswahl des Objekts bzw. der Merkmalgruppe. Weitere Vorteile der Fixation von Objekten sind nach BALLARD & BROWN [1993]:

- Viele Rechengesetze vereinfachen sich dadurch, dass sich das Objekt nahe der optischen Achse(n) der Kamera(s) befindet (z. B. Gültigkeit der orthographischen Projektion).
- Die Verfolgung von Bildmerkmalen vereinfacht die Objekterkennung und die Objekt-Hintergrund-Trennung, da der Hintergrund und andere Objekte, die sich nicht mit der gleichen Geschwindigkeit wie das fixierte Objekt bewegen, verschmieren (*motion blur*).
- Wird ein Objekt von zwei Stereokameras mittels aktiver Steuerung des Vergenzwinkels<sup>4</sup> fixiert, so ist die Stereodisparität<sup>5</sup> in der Bildmitte Null und in der Umgebung klein, was die Berechnung der Disparität vereinfacht.
- Kamera und Fixationspunkt definieren zusammen ein Koordinatensystem, bezüglich dem Verhaltensregeln und Bewegungen ausgeführt werden können. Ein Beispiel dafür ist die Annäherung an ein fixiertes, sich bewegendes Objekt. Mit dieser ‚*fixation point*‘-Strategie lassen sich viele Probleme vermeiden, wie z. B. Verdeckungen oder Messungenauigkeiten in der Abstandsschätzung oder in der Abschätzung der Kamerabewegung. Die Strategie findet insbesondere bei sogenannten „*eye in hand*“-Systemen<sup>6</sup> Anwendung.
- Das Gesichtsfeld des visuellen Sensors kann unabhängig von seinem Öffnungswinkel beliebig vergrößert werden.

Nach TSOTSOS [1989] muss ein aktiv sehendes System sowohl eine *bottom-up* als auch eine *top-down* Komponente besitzen. Die *bottom-up* Komponente hat zum Ziel, unbekannte und unerwartete Objekte zu detektieren. Im Gegensatz dazu kann der *top-down* aktivierbare Teil Vorwissen über bekannte oder erwartete Objekte in die Verarbeitung mit einfließen lassen. BROWN [1994] ordnet den oben beschriebenen Rekonstruktionsansatz nach MARR [1982] dem *bottom-up* Sehen zu und das zielgerichtete, aktive Sehen dem *top-down* Sehen. Er vertritt die Ansicht, dass die visuellen Fähigkeiten des Menschen weder mit der einen noch der anderen Komponente modelliert werden können, sondern nur durch die Kombination beider.

Die zunehmende Rechenleistung wird in Zukunft neue Anwendungsfelder für aktive Sehsysteme erschließen. Durch aktiv sehende Roboter können z. B. in der Industrie die aufwendigen Umprogrammierungen für neue Anwendungen und die Anpassungen an neue Umgebungen entfallen [MESTER, 1996]. CLARK & FERRIER [1992] geben dem interessierten Leser einen weiterführenden Überblick über die Entwicklungsziele von aktiv sehenden Systemen mit typischen Anwendungsbeispielen.

---

<sup>4</sup> Der Vergenzwinkel ist der Winkel zwischen den optischen Achsen der Stereokameras.

<sup>5</sup> Die Disparität ist der Positionsunterschied in den beiden Stereobildern eines Merkmals aufgrund der Stereobasis und der Entfernung des Merkmals vom Stereokamerapaar (vgl. z. B. [RIEDER, 2000]).

<sup>6</sup> In solchen Systemen befindet sich die Kamera in einer bewegten Roboterhand.

## 2.2 Arbeiten im Bereich ‚Aktives Sehen‘

Dieser Abschnitt greift Arbeiten neueren Datums heraus, welche das Paradigma ‚Aktives Sehen‘ für die Steuerung und Regelung mobiler Roboter und autonomer Fahrzeuge nutzen. Die Auswahl erhebt keinen Anspruch auf Vollständigkeit. Die Arbeiten sind nach ihren Schwerpunkten geordnet, wobei aufgrund übergreifender Eigenschaften die Einordnung nicht immer eindeutig ist und deshalb nur als grobe Orientierung dienen kann. Eine Literaturanalyse zum Thema Aktives Sehen bis zum Jahr 1995 und eine Darstellung der Konstruktion und Modellierung aktiver Kameraplattformen finden sich in [SCHIEHLEN, 1995]. Dem an der Entwicklung von binokularen Kameraplattformen interessierten Leser sei [TRUONG, 2000] empfohlen. SCHMALZ [2001] nennt zahlreiche Literatur, in der Strategien aktiver Sehsysteme zur Objekterkennung herangezogen werden. BOLLMANN [1999] gibt einen Überblick über den Stand der Forschung auf dem Gebiet der visuellen Aufmerksamkeit und der Augenbewegungen in den Disziplinen Psychophysik und Neurobiologie und beschreibt einige sich daran anlehrende maschinelle Aufmerksamkeitsmodelle und Blickrichtungssteuerungen. In [BERNASCH, 1997] sind eine Vielzahl biologisch orientierter Modelle zur Aufmerksamkeitssteuerung und deren Anwendungen beschrieben. Ferner stellt ULTES [2002]<sup>7</sup> eine Reihe weiterer Arbeiten auf dem Gebiet des Aktiven Sehens vor.

### 2.2.1 Blickfixierung

#### 2.2.1.1 Universität Erlangen-Nürnberg

ZOBEL et al. [2001] stellen den autonomen, mobilen Serviceroboter namens MOBSY vor. MOBSY übernimmt die Aufgabe eines Rezeptionisten im Flur des Instituts der Autoren und interagiert mit ankommenden Besuchern durch Navigation, natürlicher Sprache und Blickzuwendung. Der Roboter ist mit dem binokularen Kameraträger BISIGHT der Firma HELPMATE ROBOTICS ausgestattet. Das Stereokamera paar des Roboters fixiert während des Dialogs mit dem Besucher dessen Gesicht, was diesem klar macht, dass die Aufmerksamkeit des Roboters auf ihn gerichtet ist. Die Detektion des Gesichts erfolgt in beiden Kamerabildern und basiert auf einer Farbbildsegmentierung. Der Schwerpunkt des Farbsegments in Hautfarbe wird als Mittelpunkt des Gesichts angenommen. Bei der Blickfixierung werden die Nick-, Gier- und Vergenzwinkel des Kameraträgers nachgeführt.

#### 2.2.1.2 Universität Tübingen

FEYRER & ZELL [1999] stellen einen mobilen Roboter vor, der einer Person in einer Büroumgebung folgt. Das System besteht aus einem Stereosehsystem und einem Navigationssystem, welche miteinander interagieren. Bei der Detektion extrahiert eine Komponente Bildbereiche (*region of interest*, ROI) mit hautähnlichen Farben und eine andere Komponente ROI mit signifikanter Bewegung. Die Bewegungserkennung erfolgt durch die Subtraktion von sukzessive aufgenommenen Bildern unter Berücksichtigung der relativ langsamen Roboterbewegung. Die Ermittlung der Kontur einer Person erfolgt durch einen Algorithmus, der auf *aktive Strahlen* beruht [DENZLER, 1997]. Die Standard-Stereoanordnung dient der Berechnung eines Tiefenbildes und wird von Zeit zu Zeit automatisch neu kalibriert. Zur Erhöhung der Robustheit wird versucht, die Augen und den Mund innerhalb der durch die Farbsegmentierung gewonnenen ROI zu lokalisieren. Dazu betrachtet man in diesem Bereich das Grauwertsignal und sucht nach lokalen Minima. Bei der Fusion werden die von den einzelnen Komponenten generierten ROI verglichen und ggf. zusammengefasst. Für die endgültige Auswahl

---

<sup>7</sup> Diese Studienarbeit wurde vom Autor der vorliegenden Arbeit betreut.

berechnet man für jeden in Frage kommenden ROI ein Konfidenzmaß, welches die Wahrscheinlichkeit angibt, dass sich in dem ROI ein Gesicht befindet. Nach der Erkennung des Gesichts fixiert der Roboter die Person mit seiner Kameraplattform und folgt dieser mit seinem Navigationssystem. Der Kameraträger besitzt zwei Freiheitsgrade für Nicken und Gieren. Die Navigation des Roboters beruht auf der Potentialfeldmethode. Das Potentialfeld setzt sich aus attraktiven und repulsiven Potentialen zusammen. Der attraktive Anteil zieht den Roboter in Richtung Zielposition (Person). Die repulsiven Anteile halten den Roboter auf Abstand zu Hindernissen, welche durch den 2D *laser-range-sensor* des Roboters wahrgenommen werden.

### 2.2.1.3 Royal Institute of Technology, Stockholm, Schweden

NORLUND & UHLIN [1995] stellen ein Objektverfolgungssystem für einen mobilen Roboter vor. Das Robotersystem ist mit einem Kameraträger mit 2 Freiheitsgraden für Gieren und Nicken ausgestattet. Das System verfügt u. a. über einen Bewegungsdetektor und einen  $\alpha$ - $\beta$ -Tracker zur Blickrichtungsnachführung. Der vorgestellte Algorithmus zur Bewegungsdetektion ist echtzeitfähig und nicht iterativ. Die Anzahl der Rechenoperationen steigt lediglich linear mit der Anzahl der Bildpixel. Die Autoren nennen zwei Randbedingungen: es darf sich lediglich ein bewegtes Objekt im Bildausschnitt befinden und dieses darf nur einen kleinen Teil des Bildes ausfüllen. Die Trennung der Eigenbewegung des Roboters von der Bewegung des Objekts geschieht durch eine Bildstabilisierung. Die Bewegung des Beobachters darf dabei translatorische und rotatorische Komponenten besitzen. Rechenzeit wird dadurch eingespart, dass man mittels eines affinen oder translatorischen Bewegungsmodells eine globale Geschwindigkeitsfunktion ermittelt. Dabei bringt man zwei aufeinanderfolgende Bilder unter Anwendung des Bewegungsmodells zur Deckung. In Experimenten folgt der Roboter Quadern, die sich ungleichmäßig im Raum bewegen. Der Hintergrund ist dabei unstrukturiert und der Kontrast stark wechselnd. Die Objektfixierung erreicht eine Winkelgeschwindigkeit von 10 °/s. Der Regelkreis zu Blickrichtungsnachführung arbeitet mit 25 Hz bei einer Bildgröße von  $366 \times 287$  Pixel. Die Totzeit des Systems beträgt  $85 \pm 15$  ms.

### 2.2.1.4 University of Coimbra, Portugal

BASTISTA et al. [1997] beschreiben eine Blickrichtungssteuerung für einen binokularen Roboterkopf mit sehr vielen Freiheitsgraden. Der Kopf ist in der Lage, ein Kopfnicken, -gieren und -rollen auszuführen. Unabhängig davon können die Kameras um eine gemeinsame Achse genickt und einzeln gegiert werden (Vergenz). Die Detektion eines optischen Flusses im peripheren Kamerabild löst eine Sakkade des Roboterkopfes aus. Im Fall einer Detektion in beiden Kameras erfolgt die sakkadische Bewegung unmittelbar in allen Freiheitsgraden, was die Sakkadendauer minimiert. Wird die Bewegung nur in einem Kamerabild erkannt, so sucht man zunächst gezielt nach der Bewegung im anderen Kamerabild. Erst wenn diese ebenfalls detektiert wurde, erfolgt die Auslösung der sakkadischen Bewegungen. Bei der anschließenden Fixierung gibt es infolge der vielen Freiheitsgrade des Roboterkopfes redundante Zustände. Diese Redundanzen sind durch Randbedingungen aufzulösen. So muss man die Bewegungen derart koordinieren, dass sich der Roboterkopf für die daran anschließende Aufgabe im bestmöglichen Zustand befindet. Die Autoren stimmen die Gierbewegung des Kopfes und die Vergenzbewegung der Kameras derart ab, dass die Kameras symmetrische Vergenzwinkel aufweisen. Die Fixierung geschieht durch einen Tracker, der auf Kreuzkorrelation basiert und die Geschwindigkeit des optischen Flusses ermittelt. Die Brennweite der Motorzoomkameras wird der Entfernung des fixierten Objekts angepasst. Die dabei benutzte *Lookup*-Tabelle resultiert aus einer Kalibrierung. Ein Zustandsautomat koppelt die einzelnen Blickrichtungsfertigkeiten des Roboterkopfes miteinander.

DIAS et al. [1998] beschreiben einen mobilen Roboter, der über ein aktiv steuerbares Stereokamerapaar und über Inertialsensoren verfügt. Der vorgestellte Algorithmus ermittelt aus

den Kamerabildern die Bewegungsrichtung des Roboters innerhalb der stationären Szene. Ein PID-Regler hält den Fluchtpunkt der Bewegung in den Bildzentren. Ferner ist der Roboter in der Lage, die Bodenebene zu identifizieren, indem er Daten von inertialen Sensoren mit denen aus der Bildverarbeitung fusioniert.

### 2.2.1.5 Vanderbilt University, Nashville, USA

PENG et al. [2000] beschreiben eine Blickrichtungssteuerung für einen humanoiden, mobilen Roboter. Der Roboter ist mit zwei Farbkameras ausgestattet, welche separat auf zwei unabhängig voneinander ansteuerbaren Kameraträgern befestigt sind. Jeder Kameraträger besitzt zwei Freiheitsgrade für Nicken und Gieren. Das Sehsystem ist dem menschlichen Sehen nachempfunden und in der Lage, Sakkaden und Objektverfolgungen auszuführen, die Vergenz des Stereokamera-paars zu steuern und die rotatorische Eigenbewegung des Roboters auszugleichen; ähnlich dem vestibulär-okularen Reflex beim Menschen. In Experimenten wird mittels Farbsegmentierung ein Objekt in der Umgebung ausgewählt, mit dem Stereokamera-paar fixiert und der Roboter auf das Objekt ausgerichtet. Bei einem sich bewegenden Objekt folgt der Roboter diesem.

In jedem Kamerabild ist ein fovealer Bereich durch einen Kreis um den Hauptpunkt<sup>8</sup> definiert. Ein weiterer, kleinerer Kreis mit gleichem Mittelpunkt definiert die sogenannte Totpunktzone (*dead zone*). Solange sich das Objekt in dieser Totpunktzone befindet, führt der Kameraträger keine Folgebewegung aus. Befindet sich das Objekt innerhalb des fovealen Bereichs aber außerhalb der Totpunktzone, so wird es fixiert. Die Objektfixierung verwendet einfache P-Regler und benutzt sowohl das Positionssignal als auch das Geschwindigkeitssignal aus der Bildverarbeitung. Verlässt das Objekt den fovealen Bereich oder wird ein neues Objekt detektiert, so erfolgt die Initiierung einer Sakkade. Für die nichtlineare Abbildung zwischen der Zielkoordinate der Sakkade im Bild und den Motorstellgrößen dient ein neuronales Netz (*multi-layer feed-forward network*). Die Vergenzsteuerung erfolgt derart, dass beide Kameras das gleiche Objekt fixieren und die Disparität minimal ist. Der Ausgleich der Eigenbewegung des Roboters zur Stabilisierung der Blickrichtung erfolgt durch ein negatives Aufschalten des Drehwinkels des Roboters auf die Regler der Kameraplattformen.

## 2.2.2 Blickrichtungssteuerung

### 2.2.2.1 Universität Hamburg

BOLLMANN [1999] entwickelte für das Sehsystem NAVIS eine Blickrichtungssteuerung, welche eine datengetriebene *bottom-up*-Komponente, eine modellgetriebene *top-down* Komponente und eine (bisher allerdings nur rudimentäre) Verhaltenssteuerung besitzt. Der Autor koppelte NAVIS über eine Funkübertragungsstrecke mit einem mobilen Roboter vom Typ PIONEER 1. Übertragen wurden das analoge Videosignal der Kamera und die seriellen Datenströme zur Steuerung der aktiven Zweiachsen-Kameraplattform des Roboters. Die Navigation des Roboters und die Steuerung seines Greifers lief auf dem Bordrechner des Roboters. Die Aufgabe des Roboters war es, beliebig im Raum verteilte Dominosteine zu einer geraden Kette anzuordnen, wobei die benachbarten Hälften zweier Dominosteine jeweils die gleiche Punktzahl aufweisen mussten.

Zu Beginn sucht der Roboter systematisch nach Clustern von Dominopunkten in seiner Umgebung, indem er eine festgelegte Folge von Kamerapositionen anfährt. Die Wahrnehmung der Punkte-Cluster geschieht durch Segmentierung und eine anschließende Überprüfung der Segmente hinsichtlich ihrer Orientierung und Exzentrizität. Sind die Punkte-Cluster

---

<sup>8</sup> Der Hauptpunkt ist der Schnittpunkt zwischen der optischen Achse und der Bildebene der Kamera.

detektiert und ihre Schwerpunkte berechnet, so werden sie jeweils einzeln fovealisiert. Die fovealisierten Punkte-Cluster segmentiert man ein weiteres Mal und untersucht sie hinsichtlich ihrer Ähnlichkeit zu horizontal ausgerichteten Ellipsen. Die Zuordnung eines Punkte-Clusters zu einer Zahl erfolgt durch einen Schablonenvergleich. Kann das aktuell fovealisierte Cluster keiner Zahl zugeordnet werden, so halbiert der Roboter sukzessive seine Distanz zum betreffenden Cluster, solange bis eine Zuordnung möglich ist. Aus den Schwerpunkten der Punkte-Cluster berechnen sich die Schwerpunkte und Orientierungen der Dominosteine. Anschließend startet das Einsammeln und Anlegen der Dominosteine. Der Roboter ist während der Bildanalyse in Ruhe, wogegen er während der Navigation blind ist. Der Roboter fährt einen Dominostein über einen sogenannten virtuellen Punkt an, an dem er kurz anhält, um mittels Wahrnehmung akkumulierte Positionsfehler zu korrigieren.

### 2.2.2.2 Osaka University, Japan

KIDONO et al. [2000] stellen eine intelligente Navigationsstrategie für einen mobilen Roboter vor, welche die benötigte Assistenz von außen minimiert. In einer Lernphase wird der Roboter über eine Fernbedienung – nicht notwendigerweise auf kürzestem Weg – zu seinem Ziel geführt. Während dieser Fahrt nimmt der Roboter mit einem Stereokamera paar seine Umgebung wahr und baut eine Karte auf. Die Stereokameras, deren optische Achsen horizontal zur Bodenebene und parallel zueinander ausgerichtet sind, befinden sich auf einem Kameraträger mit einem Gier-Freiheitsgrad. Während der Lernfahrt giert der Roboter seinen Kameraträger periodisch, um möglichst viele Objektpunkte in einem weiten Bereich zu erkennen. Ein Kalman-Filter fusioniert unter Berücksichtigung der Fahrtrajektorie die mehrfach aus unterschiedlichen Richtungen wahrgenommenen Objektpunkte. Die Unsicherheiten der Objektpositionen werden ebenfalls in die Karte eingetragen.

Nach der Generierung der Karte ist der Roboter in der Lage, den kürzesten Weg zum Ziel zu berechnen und diesem in kontinuierlicher Fahrt zu folgen. Aufbauend auf den Beobachtungen in der Lernphase stellt der Roboter einen Beobachtungsplan auf, der die Positionen der Bildaufnahmen und die dabei eingenommenen Blickrichtungen enthält. Zur Berechnung der Blickrichtungen führt man folgende Schritte durch: Aus dem Wissen, welche Objektpunkte bei welchen Positionen bei der Lernfahrt sichtbar waren, werden für jeden Objektpunkt die Bereiche abgeleitet, innerhalb derer er beobachtbar ist. Damit sind für jeden Ort die beobachtbaren Objektpunkte bekannt. Danach wird für jede Blickrichtung die Unsicherheit der Roboterposition prädiiziert, wobei die Positionsunsicherheit der Objektpunkte, die Unsicherheit ihrer Wahrnehmung und die Unsicherheit der Fahrtrajektorie eingehen. An jedem Beobachtungspunkt wählt man diejenige Blickrichtung aus, welche die Positionsunsicherheit des Roboters minimiert. Die Autoren zeigen in Experimenten, dass durch die beschriebene Blickrichtungsstrategie der Positionsfehler des Roboters im Vergleich zu Experimenten mit statischer Blickrichtung signifikant kleiner ist.

MIURA et al. [2000] stellen ein aktives Sehsystem zur Erkennung von Verkehrsschildern in Echtzeit vor. Das System ist in einen PKW integriert und verfügt über zwei Kameras, welche mit einem Weitwinkelobjektiv und einem Teleobjektiv bestückt sind. Die Blickrichtungen der Kameras sind mittels zweier aktiver Kameraträger separat steuerbar. Die Weitwinkelkamera blickt in die Fortbewegungsrichtung des Fahrzeugs und die Telekamera dient der Erkennung von Verkehrsschildern. Das System läuft in 5 Schritten ab: Zuerst werden Hypothesen für Verkehrsschilder generiert. Dabei findet eine Binarisierung des Bildes (im Farbraum) mit unterschiedlichen Schwellwerten und eine anschließende Segmentierung des Farbbildes statt. Als Hypothesen werden weiße Kreise (für Geschwindigkeitsbeschränkungen) und blaue Rechtecke (für Wegweiser) mit passender Größe ausgewählt. Im zweiten Schritt wertet man für eine Hypothesen-Hintergrund-Trennung das Kantenbild hinsichtlich der erwarteten, geo-

metrischen Formen aus. Dabei kommt eine auf Hough-Transformation oder Histogrammen basierende Technik zum Einsatz. Im dritten Schritt wird die Bewegung der Hypothese im Bild geschätzt und präzisiert und die Telekamera auf die präzisierte Position ausgerichtet. Dabei geht man von einer konstanten Fahrzeuggeschwindigkeit und einer geraden Fahrtrajektorie aus. Die Autoren zeigen, dass für die Prädiktion der Bildposition der Hypothese die Fahrzeuggeschwindigkeit und die Brennweite der Kamera nicht bekannt sein müssen. Solange die Hypothese im Telebild nicht die erforderliche Größe besitzt, wird diese während der Annäherungsphase des Fahrzeugs von der Telekamera fixiert. Im Falle von Wegweisern versucht man im 4. Schritt, innerhalb der Schildhypothese Textzeichen und Symbole zu lokalisieren. Hierzu baut man sukzessive lokale Histogramme für vertikale und horizontale Kanten auf. Im 5. und letzten Schritt erfolgt mittels Template-Matching die Identifikation des Verkehrsschildes bzw. der Textzeichen. Die dabei ausgeführte normalisierte Korrelation läuft auf einer kommerziellen Bildverarbeitungs-Hardware (IP-5000 von HITACHI) und greift auf eine Datenbank von Verkehrszeichen und Textzeichen zu.

### 2.2.3 Foveales Sehen

#### 2.2.3.1 University of Oxford, Großbritannien

MURRAY et al. [1995] stellen einen Ansatz vor, bei dem eine Bewegungsdetektion mittels optischen Fluss und eine Segmentierung von Bereichen mit ähnlicher Geschwindigkeit für eine Sakkadensteuerung und eine Objektverfolgung herangezogen werden. Die in den Experimenten benutzte Kamera besitzt nahe der optischen Achse einen fovealen Bereich mit hoher Auflösung und kleinem Öffnungswinkel ( $6^\circ$ ). Der Rest des Kamerabildes (Öffnungswinkel ca.  $50^\circ$ ) besitzt aufgrund einer Unterabtastung eine wesentlich geringere Auflösung. Die Kamera ist auf einer aktiven Kameraplattform montiert, welche mit einer Abtastfrequenz von 500 Hz geregelt wird. Das aktive Sehsystem weist im Wesentlichen 4 Blickrichtungsfertigkeiten auf: *waiting*, *saccading*, *pursuing* und *panicking*. Ein endlicher Zustandsautomat<sup>9</sup> kombiniert die Blickrichtungsfertigkeiten zu einem robusten Blickrichtungsverhalten.

Wird im peripheren Gesichtsfeld der Kamera ein sich bewegendes, vom statischen Hintergrund sich abhebendes Segment detektiert, so erfolgt die Initialisierung eines Kalman-Filters. Der Kameraträger zeigt zunächst noch keine Reaktion (*waiting*). Nach einer kleinen Anzahl von Zyklen liefert das Kalman-Filter die Position und die Geschwindigkeit des Segments, was zur Auslösung einer Sakkade führt (*saccading*). Die Initialisierung einer Sakkade berücksichtigt sowohl die Position als auch die Geschwindigkeit des Zielobjekts, so dass nach Beendigung der Sakkade (unter Annahme einer konstanten Objektgeschwindigkeit) auch der Geschwindigkeitsfehler abgebaut ist. Durch Prädiktion wird während einer Sakkade die Zeit vom Auslösen der Sakkade bis zum Wiedereinsetzen der Messung überbrückt. Nach Beendigung der Sakkade erfolgt eine Fixation des Zielobjekts (*pursuing*), wofür ausschließlich der foveale Bereich und das Geschwindigkeitssignal des Segments benutzt werden. Die Position und die Geschwindigkeit des Kameraträgers stehen durch eine Pufferung dem Kalman-Filter bei der Prädiktion zur Verfügung. Bei der Blickrichtungsfertigkeit *panicking* wird mit dem Geschwindigkeitsanteil eines Segments parallel zur optischen Achse eine Kollisionszeit (*time-to-contact*) berechnet. Unterschreitet diese Kollisionszeit einen Schwellwert, so löst das System eine „Paniksakkade“ aus, bei der sich die Kamera um  $90^\circ$  sowohl in der Gier- als auch in der Nickachse vom Objekt wegdreht.

---

<sup>9</sup> vgl. Abschnitt 2.3.2.3

### 2.2.3.2 Simon Fraser University, Burnaby, Kanada

WEI & LI [1998] stellen eine Blickrichtungssteuerung vor, die auf der sogenannten *Foveate Wavelet Transform* (FWT) basiert. Durch die FWT kann ein Kamerabild in ein Ergebnisbild mit ortsabhängiger Auflösung transformiert werden. So ist es möglich, in einem fovealen Bereich mit der originalen Auflösung zu arbeiten, während der periphere Bereich das Ergebnis einer Tiefpassfilterung und Datenreduktion ist. Der Übergang zwischen fovealem und peripherem Bereich ist fließend. Die Autoren benutzen die FWT zur Stereobildverarbeitung. Der Aufbau entspricht dabei der Standard-Stereoanordnung (d. h. die Bildebenen der Kameras liegen in der gleichen Ebene), wodurch sich die Disparitätenberechnung vereinfacht. Nach der Berechnung des Disparitätenbildes generiert ein Gruppierungsalgorithmus Regionen mit ähnlicher Disparität, welche als Indikatoren für Objekte dienen. Die durchschnittliche Disparität eines ausgewählten Objekts wird dazu benutzt, die Brennweite der Kameras zu steuern. Das Heranzoomen führt zu einer verbesserten Abbildung entfernter Objekte. Neben dem Zoom hat die Blickrichtungssteuerung die Freiheitsgrade Nicken und Gieren. In den Experimenten fixiert die Blickrichtungssteuerung dasjenige Objekt, für welches das Produkt aus überdeckter Bildfläche und Disparität maximal ist.

### 2.2.3.3 Computer Vision Centre, Bellaterra, Spanien

ROCA et al. [2000] stellen ein Stereosehsystem zum Zweck der lokalen Navigation vor. Die aktiv gesteuerten Größen sind Fokus, Vergenz und Blickrichtung des Stereokamerakopfes. Die Disparität der Stereobilder wird mittels der normalisierten Kreuzkorrelation (*normalised cross-correlation*) berechnet. Die Autoren stellen u. a. einen Autofokus-Algorithmus vor, der den Fokus der Kameras der Bewegung des fixierten Objekts nachführt. Aus der jeweiligen Fokuseinstellung lässt sich die Entfernung zum Objekt ableiten. Bewegt sich ein Objekt entlang der optischen Achse, so führt dies zu einer Vergenzbewegung der beiden Kameras mit gleicher Amplitude, aber entgegengesetzten Richtungen. Eine Objektbewegung senkrecht zur optischen Achse führt dagegen zu einer Rotation der Kameras in gleicher Richtung (Version). Die Blickrichtungsnachführung geht von einer permanenten Kompensation des Vergenzfehlers aus, so dass sich das fixierte Objekt in beiden Kamerabildern an der gleichen Position befindet und die Disparität Null ist.

Die Blickrichtungsnachführung arbeitet auf einem sogenannten Cyclop-Bild, welches sich aus der Addition der beiden Stereobilder und einer anschließenden Halbierung der Intensität ergibt. Zur Ermittlung der lateralen Objektbewegung wird der Inhalt eines kleinen Fensters im Mittelpunkt des Cyclop-Bildes zum Zyklus  $t$  mit dem Inhalt eines doppelt so großen Suchbereichs im Cyclop-Bild zum Zeitpunkt  $t-1$  korreliert<sup>10</sup>. Die Autoren kombinierten die Steuerungen für Vergenz, Fokus und Blickrichtung zu sogenannten Blickfertigkeiten. Bei der Blickfertigkeit *fixation* sind das Focus-, Vergenz- und Versionssystem parallel aktiv, wobei die ersten beiden Systeme Tiefeninformation liefern. Bei der Blickfertigkeit *shifting* führen beide Kameras zuerst die schnelle Sakkadenbewegung aus. Anschließend fixiert die rechte, dominante Kamera das neue Objekt in der Bildmitte. Die linke Kamera führt eine Vergenzbewegung aus, wobei auf die Tiefeninformation aus dem Fokussystem zugegriffen wird. Als Demonstrationsplattform für das aktive Sehsystem diente den Autoren der mobile Roboter GIDEBOT. In den vorgestellten Experimenten nutzt der Roboter die vom Sehsystem gelieferte räumliche Positionsinformation zur lokalen Navigation. Ein Beispiel ist das Annähern und Umfahren eines fixierten Objekts auf einem Kreis mit konstantem Radius und dem Objekt im Mittelpunkt.

<sup>10</sup> Der Algorithmus geht davon aus, dass zum Zeitpunkt  $t$  das Zielobjekt sich in beiden Bildzentren befindet.

## 2.2.4 Blickrichtungssteuerung zur lokalen Trajektorienplanung

### 2.2.4.1 Institut für Neuroinformatik, Ruhr-Universität Bochum

BRUCKHOFF & DAHM [1998] benutzen ein sogenanntes „neuronales Feld“ (NF) um eine lokale Trajektorienplanung für einen autonomen mobilen Roboter in unbekannter und sich dynamisch ändernden Umgebung zu verwirklichen. Das NF ist eine eindimensionale Schicht von Neuronen mit homogenen Verknüpfungen vom *global inhibition type*. Die Position eines Neurons im Feld kodiert eine feste Blickrichtung. Die Fahrt- und Blickrichtung des Roboters wird bestimmt durch die Position der maximalen Aktivierung im Feld und seine Geschwindigkeit durch die maximale Amplitude der Aktivierung. Die Amplitude der Aktivierung ist auch ein Maß für die Qualität der Trajektorienplanung. Laut den Autoren realisiert das NF eine stabile Trajektorienplanung auch bei verrauschten Sensorsignalen und sich ändernden Umgebungsbedingungen. Demonstriert wird die Funktionalität der Trajektorienplanung anhand einer Implementierung auf dem anthropomorphen, mobilen Serviceroboter ARNOLD, dessen einziger Sensor ein aktiver Stereokamerakopf ist. In Experimenten weicht ARNOLD statischen Objekten und einer plötzlich auftauchenden Person aus und verändert seine Geschwindigkeit und Fahrtrajektorie derart, dass trotzdem die Zielposition sicher erreicht wird. Die maximale Geschwindigkeit des Roboters ist 0,25 m/s und die Verarbeitung eines Bildes benötigt 300 ms.

### 2.2.4.2 University of Oxford, Großbritannien

In [DAVISON & MURRAY, 1998] wird ein autonomer Indoor-Roboter vorgestellt, der mit dem in Abschnitt 2.2.3.1 beschriebenen, aktiven Stereokamerakopf ausgestattet ist. Der Roboter detektiert und verfolgt während einer zielgerichteten Fahrt selbständig Merkmale (Landmarken) in seiner Umgebung, trägt sie in eine zweidimensionale Umgebungskarte ein und benutzt sie zur lokalen Navigation und zur Verbesserung der Eigenpositionsschätzung. Alle eingehenden Messungen werden in ein gemeinsames Erweitertes Kalman-Filter eingespeist, das die Eigenposition des Roboters schätzt und die voll besetzte Kovarianzmatrix liefert. Durch die Verwendung nur eines Kalman-Filters stützen die Vermessungen aktuell sichtbarer Merkmale auch die Prädiktion der nicht sichtbaren Merkmale. Kommt ein neues Merkmal bei der Positionsschätzung hinzu, so vergrößert sich die Dimension des Messvektors und der Kovarianzmatrix um eins. Die Größe des bei der Korrelation verwendeten Suchfensters zur (Wieder)Erkennung von Merkmalen ist abhängig von der Unsicherheit der Roboterposition. Die Unsicherheit und damit die Größe des Suchfensters korreliert mit dem Rechenaufwand und mit der Suchzeit. Bei großer Unsicherheit reduziert der Roboter seine Geschwindigkeit und die Frequenz seiner Bildverarbeitung.

Der Roboter versucht neue Merkmale zu detektieren, wenn weniger als 2 Merkmale sichtbar sind; sichtbare Merkmale werden verworfen, wenn von den letzten 10 Messversuchen 5 scheiterten. Der Roboter kann mit dem Stereokamerakopf gleichzeitig immer nur ein Merkmal fixieren, wobei er dasjenige auswählt, dessen Relativposition am unsichersten ist<sup>11</sup>. Dies hat zur Folge, dass nach mehrmaliger Vermessung und Reduzierung der Messunsicherheit eine andere Landmarke fixiert wird. Dieser oftmalige Wechsel der Blickrichtung trägt der Tatsache Rechnung, dass durch die optimale Vermessung einer punktförmigen Landmarke zwar die Relativposition in Richtung der Landmarke optimal bestimmt wird, jedoch die Unsicherheit in den anderen Richtungen anwächst. In einem Experiment fuhr der Roboter wiederholt den gleichen Weg in einer Laborumgebung hin und zurück. Durch das Eintragen detek-

---

<sup>11</sup> Die Messunsicherheit stellt im multidimensionalen Raum der Messwerte einen Ellipsoiden dar. Es wird immer dasjenige Merkmal von der Stereokameraplattform fixiert, in dessen Richtung der Ellipsoid die längste Achse besitzt.

tierter und vermessener Landmarken in eine Karte und das Wiedererkennen der Landmarken beim wiederholten Passieren, war nach 24 m Fahrt und mehreren Kehrtwendungen die Differenz zwischen geschätzter und realer Position und Ausrichtung kleiner als 7 cm bzw.  $0.01^\circ$ ! Der Roboter fuhr dabei mit einer Geschwindigkeit von 20 cm/s; die Bildverarbeitung und Zustandsschätzung arbeitete mit einer Frequenz von 5 Hz.

### 2.2.4.3 Carnegie Mellon University, Pittsburgh, USA

JOCHEM et al. [1996] benutzen das neuronale *Feed-Forward*-Netz ALVINN<sup>12</sup>, um mit einem Fahrzeug auf Straßen niedriger Ordnung zu navigieren. ALVINN wird mit einem stark unterabgetasteten Kamerabild gespeist und lernt durch Backpropagation. Jedes Bildpixel entspricht einer Eingangsgröße für das neuronale Netz und das Ausgangsneuron mit der größten Aktivierung bestimmt den Soll-Lenkswinkel des Fahrzeugs. Das neuronale Netz wurde mit Bildern trainiert, bei dessen Aufnahme die optische Achse der Kamera parallel zur Fahrzeuglängsachse ausgerichtet war. Die Autoren benutzen ALVINN, um zusätzlich Abzweige zu erkennen. Um ALVINN mit den ursprünglichen Trainingsdaten betreiben zu können, berechnen die Autoren aus dem realen Kamerabild das künstliche Bild einer virtuellen Kamera, welche sich vor dem Eigenfahrzeug befindet und parallel zur erwarteten Querstraße ausgerichtet ist. Die berechneten Bilder werden in ALVINN eingespeist. Befindet sich die Querstraße im virtuellen Kamerabild, so steigt ein Vertrauensmaß, welches den Grad der Übereinstimmung von Bilddaten und Trainingsdaten beschreibt. Ab einem bestimmten Schwellwert gilt der Abzweig als erkannt und das Abbiegemanöver beginnt. Während des Abbiegens wird kontinuierlich die Position und Ausrichtung des Fahrzeugs relativ zum Abzweig geschätzt und die virtuelle Kamera neu positioniert. Die reale Kamera ist auf einem aktiv ansteuerbaren Zweiachsen-Kameraträger montiert und der Kameraträger befindet sich auf dem Dach des Fahrzeugs. Durch eine geeignete Ansteuerung des Kameraträgers wird gewährleistet, dass die Eigenstraße und der größte Teil des Detektionsbereichs für die Querstraße sich stets im Kamerabild befinden. Durch das Gieren des Kameraträgers ist die optische Achse der realen Kamera nicht mehr parallel zur Fahrzeuglängsachse. Bei der Generierung des virtuellen Kamerabildes kann die Position der virtuellen Kamera unabhängig vom Gierwinkel der realen Kamera und entsprechend den Trainingsdaten des neuronalen Netzes festgelegt werden. Mit Hilfe des vorgestellten Ansatzes konnte das Experimentalfahrzeug NAVLAB2 Y- und T-Einmündungen robust durchfahren.

### 2.2.4.4 Osaka University, Japan

OKUBO et al. [1999] stellen eine Methode vor, mit der eine 3D-Szenenrekonstruktion durchgeführt werden kann, um für einen Roboter passierbare Wege in einer *Indoor*-Szene zu finden. Der mobile Roboter benötigt dazu ein Stereokamerapaar mit steuerbarem Vergrößerungswinkel auf einer Schwenk-Neige-Einheit. Der beschriebene Stereoalgorithmus erkennt Objekte in der Umgebung eines fixierten 3D-Punktes. Die Blickrichtungsstrategie beruht auf folgender Hypothese: Es sind für den Roboter diejenigen Richtungen am wahrscheinlichsten frei befahrbar, in denen möglichst ausgedehnte und möglichst weit entfernte Objektflächen wahrnehmbar sind. Im ersten Schritt stellt der Roboter deshalb die optischen Achsen der Kameras parallel und sucht nach großen, weit entfernten Objektflächen (Fixationspunkt im Unendlichen). Gefundene, ausgedehnte Objektflächen definieren Blickrichtungen, welche der Stereokamerakopf dann sukzessive anfährt. Im nächsten Schritt wird entlang dieser Blickrichtungen die Szene genauer rekonstruiert. Dazu verschiebt der Roboter stufenweise den Fixationspunkt entlang der Blickrichtung zu sich heran, solange bis eine minimale Distanz erreicht ist. Um jeden dieser Fixationspunkte erkennt der Stereoalgorithmus auch näher zum Roboter liegende Objekte und ermittelt deren 3D-Positionen. Innerhalb einer Fixation wird

<sup>12</sup> Autonomous Land Vehicle In a Neural Net

die Verlagerung des Kreises mit verschwindender binokularer Disparität durch pixelweises Verschieben des linken Kamerabildes erreicht. Diese Verschiebung wirkt sich nahe der optischen Achse wie eine Drehung der Kamera aus. Durch die räumliche Beschränkung der Stereobildverarbeitung um einen fixierten 3D-Punkt mit der Disparität Null lässt sich die Größe der korrelierten Bereiche und der untersuchte Disparitätenbereich stark eingrenzen.

## 2.2.5 Aufmerksamkeitssteuerung

### 2.2.5.1 Universität Erlangen-Nürnberg

SOBOTKA & WETZEL [1995] stellen eine Aufmerksamkeitssteuerung vor, die in dem System zur Verkehrsszenenanalyse MOSAIK eingebettet ist. MOSAIK verarbeitet monokulare Grauwertbildfolgen und verfügt über folgende Wahrnehmungszustände: *Recognition* (R), *Tracking* (T), *Simultaneous Tracking and Recognition* (STR) und *Attention Control* (A). Zu Beginn der Wahrnehmung befinden sich alle Bildregionen im Erkennungszustand R. Wird ein Fahrzeug erkannt, so wechselt der entsprechende Bildbereich in den Zustand T. Bildregionen in denen keine Fahrzeuge erkannt wurden, wechseln in den Erkennungszustand A. Die Initialisierungsphase ist beendet, wenn sich alle Objekte im Zustand T und alle objektfreien Bildbereiche im Zustand A befinden. In der anschließenden Ablaufphase werden die Zustandsgrößen der Fahrzeuge geschätzt. Weichen die geschätzten Zustandsgrößen zu stark von den prädierten Werten ab, so wechselt das Objekt in den Zustand STR. Nach einer erfolgreichen Korrektur der Zustandsgrößen kehrt das Objekt in den Erkennungszustand T zurück.

Während der Ablaufphase untersucht die Aufmerksamkeitssteuerung die objektfreien Bildregionen nach möglichen Hinweisen für Fahrzeuge. Sie unterteilt dazu das Kantenbild in Segmente mit einheitlicher Bewegung und entscheidet dann anhand heuristischer Gruppierungsstrategien, ob ein Bereich ein Fahrzeug beinhalten kann oder nicht. Bei der Suche nach Indizien innerhalb der Segmente betrachtet sie die räumliche Konsistenz und zeitliche Kontinuität von Gruppen (nahezu) vertikal und horizontal verlaufender Kanten. So werden im ersten Schritt Gruppen horizontaler, nahe beieinanderliegender und vertikal überlappender Linien gebildet. Im zweiten Schritt wird die analoge Gruppierung mit vertikalen Linien durchgeführt. Ein Fahrzeug gilt als detektiert, wenn in passender geometrischer Anordnung eine horizontale Gruppe zwischen zwei vertikalen Gruppen liegt. Die entsprechende Bildregion wird dann in den rechenzeitintensiveren Erkennungszustand T versetzt. Die Autoren betonen, dass die Interaktion zwischen Erkennung, Tracking und überwachender Aufmerksamkeitssteuerung durchschnittlich 30 % Rechenzeit einspart, ohne die Robustheit zu mindern.

### 2.2.5.2 Carnegie Mellon University, Pittsburgh, USA

KELLY & STENTZ [1998] beschreiben das autonome Navigationssystem RANGER für geländetaugliche Fahrzeuge, das mittels einer intelligenten Aufmerksamkeitssteuerung eine selektive Bildverarbeitung durchführt (verdeckte Aufmerksamkeit, vgl. Abschnitt 2.1.2). So beschränkt sich die Wahrnehmung auf einen Streifen (*region of interest*, ROI) vor dem Fahrzeug. Objekte die jenseits des ROI liegen, können oder müssen (noch) nicht wahrgenommen werden (*adaptive lookahead*). Nach unten wird der ROI durch die minimale Distanz beschränkt, bei der das Fahrzeug gerade noch rechtzeitig auf ein Hindernis gefährdungsvermindernd reagieren kann<sup>13</sup>. Die Anpassung der Wahrnehmung an die Geschwindigkeit und Position des Fahrzeugs und an die lokale Steigung des Untergrunds maximiert die Geschwindigkeit, die das Fahrzeug unter Einhaltung von Sicherheitsbedingungen fahren kann. Innerhalb des ROI erfolgt eine Korrelation der Stereobilder, wobei sich die Disparitätenberechnung auf

---

<sup>13</sup> Dies ist z. B. die Distanz, welche das Fahrzeug für die Ausführung einer 90° Kurve benötigt (unter Berücksichtigung der Totzeiten im System).

den erwarteten Disparitätenbereich beschränkt. Durch diese Einschränkung wird ggf. nicht das globale sondern ein lokales Maximum der Korrelationsfunktion gefunden und infolge ähnlicher Textur kann es zu Fehlzuordnungen kommen. Diese besitzen jedoch laut den Autoren eine geringe räumliche Ausdehnung und lassen sich durch eine anschließende Segmentierung, welche kleine Regionen unterdrückt, eliminieren. Die Autoren testeten das Navigationssystem RANGER u. a. in einem Militärfahrzeug (HMMWV). Das Fahrzeug war neben den Stereokameras u. a. mit einem *laser-range-finder* ausgestattet und konnte eine Durchschnittsgeschwindigkeit von 7 km/h und eine Höchstgeschwindigkeiten von 15 km/h sicher erreichen.

## 2.2.6 Steuerung der Wahrnehmungsprozesse

### 2.2.6.1 Örebro University, Schweden

In dem von SAFFIOTTI & LEBLANC [2000] vorgestellten autonomen Sehsystem werden die relevanten Objekte durch sogenannte Mikromodelle repräsentiert, welche im Wesentlichen zwei Aufgaben zu erfüllen haben. Die erste Aufgabe ist die Fusion von Hintergrundwissen mit bereits vorhandenen oder aktuell wahrgenommenen Objektdaten. Die Fusion ist so gestaltet, dass Steuerungsgesetze, welche ebenfalls in den Mikromodellen niedergelegt sind, ausgeführt werden können. Die zweite Aufgabe der Mikromodelle ist die Steuerung der Wahrnehmungsprozesse, um die für das Steuerungsgesetz notwendigen Informationen zu ermitteln. Das Wissen, welche Wahrnehmung für welche Steuerung notwendig ist, befindet sich in den Mikromodellen und damit auf der Wahrnehmungs-Aktions-Ebene und nicht auf höheren Entscheidungsebenen. Die Mikromodelle legen u. a. den Fixationspunkt einer aktiv steuerbaren Kamera fest und verfügen über ein Maß, welches beschreibt, wie wichtig das zugehörige Objekt für die Ausführung der aktuellen Aufgabe ist. Ein weiteres Maß beschreibt, wie stark das Mikromodell von kürzlich wahrgenommenen Daten gestützt wird. Dieses Maß nimmt bei einer reinen Prädiktion exponentiell mit der Zeit ab, wobei die Verfallskonstante von der Güte der Prädiktion und von der objekttypischen Eigenbewegung (statisch  $\leftrightarrow$  dynamisch) abhängt. Mittels dieser beiden Maße entscheidet man, welches Mikromodell wahrgenommen und welche Blickrichtung realisiert wird. Die Autoren haben den Ansatz dazu benutzt, um im Rahmen von ROBOCUP'99<sup>14</sup> ein Team von hundsähnlichen, fußballspielenden Robotern zu kontrollieren.

### 2.2.6.2 Simon Fraser University, Burnaby, Canada

Bei GVOZDIJAK & LI [1998]<sup>15</sup> werden Objekte durch das Finden und Erkennen verschiedener Objektansichten identifiziert. Die Bildverarbeitung läuft dabei auf einer speziell entworfenen Rechnerarchitektur (*SFU hybrid pyramid vision machine*). Das Robotersystem ist in der Lage, durch aktive Exploration modellierte Objekte im dreidimensionalen Raum wiederzufinden. Zur Anwendung kommen dabei eine hierarchische *top-down*-Suche (zur Verringerung des Suchraums), eine zielgerichtete Roboter- und Kamerabewegung zur Gewinnung verschiedener Ansichten und eine Mehrebenenauflösung für die Objekte zur Verringerung der Datenmenge und zur Erhöhung der Robustheit. Die Mehrebenenauflösung ist hierarchisch und baumartig strukturiert und beinhaltet Information darüber, welche Kamera- und Roboterbewegungen für eine Ansicht notwendig sind. Die Knoten des Baums repräsentieren Objektteile, wobei mit absteigender Baumhierarchie die Knoten immer feinere Objektstrukturen beschreiben. Mithilfe von Roboter- und Kamerabewegungen wird sukzessive versucht, neue Ansichten eines Objekts in der Umgebung zu gewinnen, um damit das Objekt zu identifizieren. Der Roboter verfährt dabei nach folgender Erkennungsstrategie: Die Kamera wird solange gegiert, bis das Objekt mit seiner größten, im Wurzelknoten repräsentierten Auflösung erkannt ist.

<sup>14</sup> ROBOCUP ist ein jährlich ausgetragener, internationaler Wettbewerb zwischen fußballspielenden Robotern.

<sup>15</sup> zitiert nach SCHMALZ [2001].

Besitzt das Objekt im Kamerabild nicht die erwartete Größe, so bewegt sich der Roboter vor oder zurück, wobei das Objekt von der Kamera fixiert bleibt. Die Baumstruktur absteigend, wird im Anschluss daran sukzessive nach Teilstrukturen gesucht, wobei man die dazu nötigen Kamera- und Roboterbewegungen den Knoten des Baums entnimmt. Das Objekt gilt als identifiziert, wenn eine ausreichend große Anzahl von Teilstrukturen wiedererkannt wurde. Die Autoren demonstrieren die Funktionsweise des aktiven Erkennungssystems anhand einer Suche und Identifikation von Tassen in einer Büroszene durch einen mobilen Roboter.

## 2.2.7 Verhaltensbasiertes Sehen<sup>16</sup>

### 2.2.7.1 NASA Johnson Space Center, Houston, USA

HUBER & KORTEKAMP [1998] beschreiben ein aktives Stereosehsystem, das seine Rechenzeit auf virtuelle, dreidimensionale Raumregionen – *proximity spaces* (PS) – konzentriert. Ein PS verknüpft eine 3D-Raumregion mit den von diesem überdeckten 2D-Bildregionen in der linken und rechten Stereokamera in zeitlich aufeinanderfolgenden Aufnahmen. In diesen (Kanten-)Bildbereichen werden Korrelationen durchgeführt, um Disparitäten und Bewegungsvektoren zu berechnen, welche wiederum als Eingangsgrößen für eine Menge von Blickfertigkeiten (*visual behaviors*) dienen. Die Korrelationswerte beschreiben ferner die Wahrscheinlichkeit, dass sich ein Objekt in der erwarteten Entfernung im 3D-Raumvolumen befindet. Die Blickfertigkeiten kontrollieren die Bewegungen und das Volumen der PS im 3D-Raum. Es gibt 9 verschiedene Blickfertigkeiten: *Follow*, *Cling*, *Avoid*, *Migrate*, *Pull*, *Search*, *Lead*, *Free path search* und *Resize*. Jede Blickfertigkeit erzeugt einen Vektor, in dessen Richtung und um dessen Betrag der PS aufgrund der Blickfertigkeit im nächsten Zyklus zu verschieben ist. Alle diese Vektoren addiert man vektoriell und der Gesamtvektor bestimmt die Verschiebung des PS im nächsten Bild. Die Gewichtung der Vektoren erfolgt so, dass ein robustes Verfolgen der sich vom Hintergrund abhebenden Bildmerkmale erreicht wird. Die Position des PS innerhalb des Gesichtsfeldes des Roboters bestimmt die Ausrichtung des Stereokamerakopfes. Der Kamerakopf besitzt drei Freiheitsgrade für Gieren, Nicken und Vergenz. Das Sehsystem wurde auf einem mobilen *Indoor*-Roboter in die sogenannte *3T-Kontrollarchitektur* integriert. Die Autoren koppelten mehrere PS über geometrische und kinematische Modelle und realisierten dadurch eine Gestenerkennung. In weiteren Experimenten folgt der Roboter Personen in einer nicht präparierten Büroumgebung.

### 2.2.7.2 University of Virginia, Charlottesville, USA

WASSON et al. [1999] führten auf der Fertigkeitenebene der bereits in Kapitel 2.2.7.1 erwähnten *3T-Kontrollarchitektur* das Konzept des Wahrnehmungsgedächtnis (*prereptual memory* (PM)) ein, welches als Schnittstelle zwischen den verschiedenen Wahrnehmungs- und Fortbewegungsfertigkeiten des Systems sowie zu höheren Systemebenen fungiert. Ein PM besteht aus einer Menge von Marken. Diese Marken repräsentieren ausschließlich diejenigen Objektattribute, welche für die vom autonomen Agenten auszuführende Aufgabe wichtig sind. So beschreibt eine Marke sowohl die Rolle des Objekts im Zusammenhang mit der Aufgabe also auch die Position des Objekts relativ zu einem egozentrierten Koordinatensystem. Ferner spezifiziert die Marke die geeigneten, dem System zur Verfügung stehenden Wahrnehmungsfähigkeiten, mit denen das repräsentierte Objekt identifiziert werden kann. Ferner verfügt eine Marke über ein Konfidenzmaß, das die Güte des Objektwissens beschreibt.

Der von den Autoren vorgestellte Roboter besitzt ein monochromes Stereokamerapaar und eine Farbkamera, welche zusammen auf einer aktiven Kameraplattform montiert sind. Die

---

<sup>16</sup> Die Eigenschaften verhaltensbasierter Systeme werden in Kapitel 2.3.4.2 dargestellt.

Aufgabe des Roboters ist es, Menschen in der Umgebung zu erkennen, mit den Kameras zu fixieren und sich diesen anzunähern. Jede Marke im PM repräsentiert dabei einen Menschen und dient sowohl der Blickrichtungssteuerung als auch der Fortbewegung als Zielobjekt. Marken können von höheren Systemkomponenten oder von der Farbbildverarbeitung als Hypothesen eingehängt werden. Die Farbbildverarbeitung sucht im Kamerabild nach sichtbaren, nicht instanziierten Marken und segmentiert lokal Bildbereiche mit Hauttönen, welche geometrischen Randbedingungen genügen (z. B. Relativlage zwischen Kopf und Arme). Da die monokulare Farbbildverarbeitung keine Tiefeninformation generiert, positioniert sie eine neue Marke unter der bekannten Raumrichtung mit einer minimalen Distanz zum Roboter. Nach der Ausrichtung der Kameras in diese Raumrichtung ermittelt das in Abschnitt 2.2.7.1 beschriebene Stereosystem die wirkliche Entfernung des Objekts und korrigiert die 3D-Position der Marke. Sind mehrere Personen erkannt, so fixiert der Roboter die entsprechenden Marken abwechselnd. Blickrichtungswechsel zu einer Person erfolgen dann, wenn das Konfidenzmaß in der zugehörigen Marke den Wert 0 erreicht. Das Konfidenzmaß verringert sich bei Nichtsichtbarkeit der Marke.

### 2.2.7.3 Applied AI Systems Inc., Ontario, Kanada

GOMI & DIE [1999] stellen das aktiv sehende Fahrzeug M-2 für den Außenbereich vor, welches mit extrem schlechten Bodenbedingungen zurechtkommt, wie z. B. Gras, Kies, kleine Büsche und Felsen. Eine der Navigationsaufgaben des Fahrzeugs ist es, auf einem Baustellen Gelände einer Person mit einer roten Sicherheitsweste zu folgen. Das Fahrzeug ist mit mehreren Sensoren ausgestattet: einem Tastsensor in der Stoßstange, Ultraschallsensoren, Infrarotsensoren und mit einer Videokamera. Jeder dieser Sensoren verfügt über ein unabhängiges Sensordatenverarbeitungsmodul, welches Steuergrößen für die Aktuatoren des Fahrzeugs generiert. Die Steuergrößen der Sensordatenverarbeitungsmodule sind priorisiert, wobei die Steuergrößen aufgrund der Bildverarbeitung die geringste Priorität besitzen. Entsprechend dem verhaltensbasierten Ansatz<sup>16</sup> entscheidet ein Schiedsrichter (*arbiter*) über die endgültig ausgeführte Verhaltensweise. Im vorliegenden Fall wird immer die Verhaltensweise mit der größten Priorität ausgewählt. So folgt das Fahrzeug einer durch die Videokamera wahrgenommenen Person nur, wenn von den Infrarotsensoren kein Hindernis wahrgenommen wird und damit von deren zugehörigen Sensordatenverarbeitung keine Steuergrößen (für ein Ausweichen) vorliegen.

Die Autoren betonen, dass trotz der einfachen Farbbildsegmentierung die bei Außenanwendungen typischen Schwierigkeiten auftreten (z. B. kein eindeutiger Farbeindruck und Farbkontrast aufgrund stark wechselnder Beleuchtung, Gegenlicht oder Morgenrot). Die Autoren statteten deshalb das Fahrzeug mit einem Kameraträger zur aktiven Steuerung des Gierwinkels der Kamera aus. Während der Navigation wird damit eine Objektfixierung durchgeführt. Zusätzlich beeinflusst eine automatische, zweistufige Motorzoomsteuerung die Objektgröße im Kamerabild. Die Autoren weisen nach, dass das aktiv sehende Fahrzeug die Navigationsaufgabe wesentlich besser löst.

### 2.2.7.4 Instituto de Automática Industrial, Madrid, Spanien

SCHNEIDER-FONTAN [1999] stellt eine Architektur für einen mobilen, aktiv sehenden Roboter vor. Die Robustheit der lokalen Navigation erreicht das System durch sogenannte *motor behaviors* (MB). Die MB besitzen aussagekräftige, symbolische Attribute, mit denen das Verhalten parametrisiert und adaptiert werden kann. Eine Art von MB sind die *vision motor behaviors* (VMB). Sie verarbeiten Kamerabilder und steuern die aktive Zweiachsen-Kameraplattform des Roboters. Es wurden VMB für die Suche, Detektion, Verfolgung und zur 3D-Positionsbestimmung von Landmarken in einer Büroumgebung entwickelt. Die sogenannten *navigation motor behaviors* (NMB) steuern die Räder des Roboters und verarbeiten die Sig-

nale von Sonarsensoren, odometrischen Sensoren und die von den VMB ermittelten 3D-Positionen der Landmarken. Die Mission für den Roboter wird von außen vorgegeben und besteht aus einer Reihe von (räumlichen) Zielen und symbolischen Hinweisen, wie die Ziele erreicht werden können (z. B. „*look to your right to find the bookshelf*“). Die MB und ihre Attribute werden als allgemeinverständliche Symbole für die Planung von Verhalten benutzt. Dabei sind die Planungsschritte an Beobachtungen in der realen Welt gekoppelt. In Experimenten zeigen die Autoren, dass durch die Spezifizierung von symbolischen Hinweisen Navigationsaufgaben schneller und direkter abgearbeitet werden, als ohne.

GARCIA-ALEGRE & RECIO [1998] entwickelten eine Hierarchie von Fähigkeiten zur sichtgestützten Roboternavigation<sup>17</sup>. Jede Hierarchieebene ist mit einem gewissen Abstraktionsgrad verbunden. Eine Fähigkeit wird durch die in der Hierarchie höher liegenden Fähigkeiten gesteuert. Auf der Fertigkeitenebene handhabt eine Fuzzy-Wissensbasis unsicheres und ungenaues Wissen. Die Autoren implementierten die Fortbewegungsfertigkeiten *Stop&Backward*, *Avoid* und *Forward*; welche durch die abstrakteren Fähigkeiten *Push* und *Move\_To* geführt werden. Der Roboter verfügt ferner über die Blickrichtungsfertigkeiten *Saccadic*, *Find\_Contour* und *Center*, deren dynamische Interaktionen ebenfalls von abstrakteren Blickrichtungsfähigkeiten namens *Search&Track* und *Search&Count* koordiniert werden. Die Autoren validierten ihren Ansatz mit einem sechsrädrigen, mobilen Roboter, der mit einem rotierenden Ultraschallsensor und einem Sehsystem ausgestattet ist. Das Sehsystem besteht aus einer CCD-Kamera auf einer aktiven Zweiachsenplattform. Die vorgestellten Experimente zeigen die Verhaltensfähigkeit *Follow\_Person*, welche durch die Anwesenheit eines Menschen aktiviert wird. *Follow\_Person* koordiniert die Fähigkeiten bezüglich Fortbewegung *Move\_To* und Wahrnehmung *Search&Track*, welche wiederum auf die in der Hierarchie darunter liegenden Fertigkeiten zugreifen. In der Fähigkeit *Search&Track* löst ein Verlust der fixierten Zielperson automatisch eine Suche aus, um die Person wiederzufinden.

---

<sup>17</sup> Bei der Darstellung des Inhalts dieser Arbeit wurden die von den Autoren verwendeten Ausdrücke mit Begriffen übersetzt, wie sie in Abschnitt 1.1 definiert sind.

## 2.3 Modellierung von Wissen und Verhalten

In der Literatur finden sich eine Vielzahl von Methoden zur Repräsentation, Modellierung und Verarbeitung von Wissen, Abläufen und Verhalten in technischen Systemen. Abschnitt 2.3.2 gibt einen groben Überblick über diese Methoden. Dem an dieser Thematik interessierten Leser sei REIMER [1991] empfohlen. Zu Beginn dieses Kapitels wird aber zunächst dargestellt, wie man glaubt, dass der Mensch Informationen verarbeitet.

### 2.3.1 Die Informationsverarbeitung durch den Menschen

RASMUSSEN [1983] entwickelte ein Modell für die Informationsverarbeitung durch den Menschen, welches heute allgemein anerkannt ist. Nach diesem Modell können die vom Menschen aufgenommenen Informationen in drei Klassen unterteilt werden: Signale, Zeichen und Symbole. Diesen Klassen ordnet der Mensch drei hierarchisch gegliederte Verhaltensebenen zu (vgl. Abb. 2.1).

Die Ebene des fertigkeitbasierten Verhaltens umfasst hochgeübte, häufig und im Wesentlichen unbewusst ablaufende, sensomotorische Leistungen. Typische Vertreter dieser Ebene sind Stabilisierungsaufgaben und manuelle Regelungsaufgaben unter stationären Prozessbedingungen. Die Verhaltensmuster werden dabei direkt von den Signalen aus der Umwelt ausgelöst. Fertigkeitbasiertes Verhalten geschieht quasi-automatisch und ohne mentale Anstrengung.

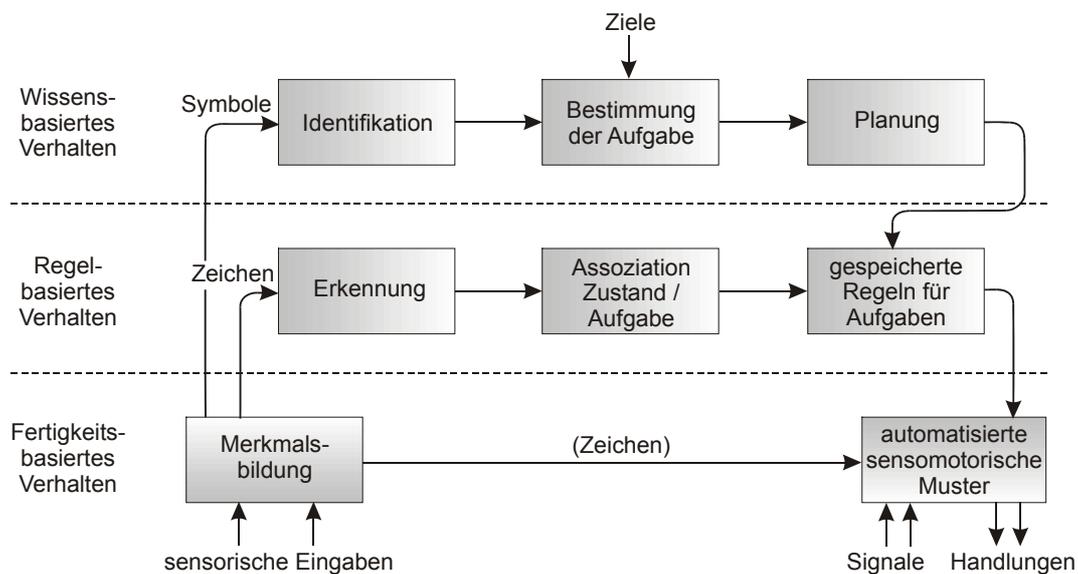


Abb. 2.1: Schematische Darstellung der menschlichen Verhaltensebenen nach [RASMUSSEN, 1983]

Auf der Ebene des regelbasierten Verhaltens werden bekannte Zeichen erkannt und mit passenden, bisweilen generalisierten Regeln weiterverarbeitet. Die Anwendung einer Regel bedingt die Klassifizierung der darin beschriebenen Situation als bekannt und vertraut. Die Regeln stellen explizites Wissen dar und haben oft prozeduralen Charakter, wie sie z. B. in der Lenkung während des Abbiegens oder bei der Bahnführung auftreten. Regelbasiertes Verhalten wird bewusst ausgeführt und koordiniert das fertigkeitbasierte Verhalten auf der darunter liegenden Ebene.

Die Ebene des wissensbasierten Verhaltens wird dann aktiv, wenn neuartige, bisher unbekannte Situationen zu bewältigen sind, für die keine passende Regel zur Erzeugung einer Handlung als Reaktion darauf bekannt ist. Bei der Lösung solcher Probleme ist zuerst tiefergehend zu abstrahieren. Unter Zuhilfenahme des Situations- und Aufgabenkontextes und von allgemeinem, explizit repräsentiertem Hintergrundwissen wird dann versucht, eine Lösung für das Problem zu finden. Bei der Problemlösung kann man auf gespeichertes Regelwissen und Fertigkeiten zurückgreifen. Beispiele für wissensbasiertes Verhalten sind z. B. Planungs-, Diagnose- und Fehlerbehandlungsaufgaben.

Tritt ein gleichartiges Problem häufiger auf, so ist eine schrittweise Verlagerung des Verhaltens in Richtung niedrigerer Ebenen beobachtbar. Dazu muss die erfolgreiche Lösung bzw. das Verhaltensmuster in den niedrigeren Ebenen als neue Regel bzw. sensomotorische Reaktion abgelegt werden. Der Aufmerksamkeitsbedarf, die kognitive Beanspruchung und der zeitliche Bedarf nehmen im Rasmussen-Schema von oben nach unten ab.

### 2.3.2 Klassische Beschreibungsmethoden

Dieser Unterabschnitt beschreibt in kurzer Form einige aus der Literatur bekannte Methoden zur Modellierung von Wissen, Abläufen und Verhalten in technischen Systemen. Nicht behandelt werden z. B. die verschiedenen Arten von Logiken (Aussagelogik, Prädikatenlogik, temporale Logik), die verschiedenen Arten von neuronalen Netzen und statistische Verfahren wie Bayes-Netze. Die Vor- und Nachteile der Methoden und weiterführende Literatur nennen z. B. RUCKDESCHEL [1996], STÜTZ [1999] und FERARIC [1996]. Die vorliegende Arbeit verwendet zur Wissensrepräsentation und Verhaltensmodellierung u. a. unscharfe Mengen und Regeln. Diese Modellierungsart wird in Kapitel 2.3.3 ausführlicher beschrieben.

#### 2.3.2.1 Rahmen und Skripte

Ein Rahmen (*frame*) stellt eine allgemeine Struktur dar, die zur Aufnahme objektspezifischen Wissens dient. Er enthält eine Menge von Fächern (*slots*), welche einzelne Aspekte des Objekts beschreiben, wie Eigenschaften, Konsequenzen, Erklärungen oder Verknüpfungen mit anderen Rahmen. Ein Fach kann wiederum einen Rahmen enthalten, was den Aufbau einer hierarchischen Wissensstruktur ermöglicht. Rahmen erlauben ferner die Einbettung von Prozeduren, die bei bestimmten Zugriffen auf die Fächer aufgerufen werden. Durch Vererbung kann man Eigenschaften von einem Objekt auf ein untergeordnetes Objekt übertragen und Redundanzen in der Wissensrepräsentation vermeiden [RUCKDESCHEL, 1996]. KOPF [1993] verwendet Rahmen zur hierarchischen Gliederung des sogenannten Situationsraums (vgl. Abschnitt 3.2.1). Skripte sind eine spezielle Form von Rahmen, welche zur Darstellung von sequentiellen Abläufen dienen können. Sie besitzen zusätzliche Fächer für charakteristische, mit dem Objekt assoziierte Handlungsabläufe [PRÉVÔT, 1995].

#### 2.3.2.2 Semantische Netze

Das Grundprinzip semantischer Netze besteht darin, Informationen über Objekte in Knoten und Informationen über Beziehungen zwischen Objekten in Kanten im Sinne der Graphentheorie darzustellen [REIMER, 1991]. Mit den Kanten sind spezielle Bedeutungen verknüpft. Häufig benutzte Kantentypen sind die Generalisierungskante („*ist ein*“) und die Bestandteilkante („*hat ein*“) [WETZEL, 1995]. Die Knoten repräsentieren Konzepte, welche einen Verarbeitungsschritt, ein physikalisches Objekt, einen Objektteil oder komplexe Vorgänge (z. B. ein Überholmanöver) modellieren können. In jedes Konzept sind eine Reihe von Methoden eingebettet, durch die das Konzept instanziiert (ausgeprägt) werden kann. Die Instanz eines Kon-

zepts enthält das Ergebnis des Abgleichs zwischen den modellierten abstrakten Begriffen oder Gegenständen und den tatsächlich vorliegenden Daten [RICHTER, 1995].

### 2.3.2.3 Endliche Automaten

Automatenmodelle sind abstrakte Beschreibungen von technischen Geräten, dynamischen Systemen oder Abläufen und erlauben eine Zwischenspeicherung von Daten. Dabei wird davon ausgegangen, dass der Prozess in diskreten Schritten abläuft oder sich in solche unterteilen lässt. Während des gesamten Prozesses befindet sich das System stets in einem wohldefinierten Zustand. In jedem Zustand können gewisse Eingaben verarbeitet und Ausgaben erzeugt werden. Der Automat kann nur endlich viele verschiedene Zustände einnehmen und die Anzahl der Ein- und Ausgangsgrößen ist begrenzt. Ein endlicher Automat lässt sich durch ein sogenanntes Zustandsübergangsdiagramm darstellen, in dem Zustände als Knoten und Zustandsübergänge als gerichtete Kanten repräsentiert werden [RUCKDESCHEL, 1996]. Partiiell gekoppelte Automaten erlauben ferner die Modellierung von Nebenläufigkeiten [KOPF, 1993].

### 2.3.2.4 Zustandskarten

HAREL [1987] erweiterte die endlichen Automaten um die Konzepte Orthogonalität und Überzustand zu den sogenannten Zustandskarten (*statecharts*). Zustandskarten unterscheiden sich von klassischen Zustandsübergangsdiagrammen durch die Möglichkeit der UND-ODER-Dekomposition der Zustände und durch die Zustandsübergänge zwischen diesen Dekompositionsebenen. Die Untergliederung der Zustände in Unterzustände erlaubt eine Beschreibung von Hierarchien von Automaten und von Nebenläufigkeiten. So kann ein Zustand einen weiteren Automaten enthalten und zu einem Überzustand werden. Ferner besteht die Möglichkeit, das Ereignis eines Zustandsübergangs an Empfänger in anderen Subnetzen zu versenden (*broadcast*-Kommunikation). Damit lassen sich z. B. Nebenläufigkeiten synchronisieren. MAURER [2000] schlägt Zustandskarten für die flexible Automatisierung von Straßenfahrzeugen vor.

### 2.3.2.5 Entscheidungsbäume und -tabellen

Beim Entscheidungsbaum findet die Entscheidungsfindung durch das Durchlaufen eines Graphen mit Baumstruktur statt; bei der Entscheidungstabelle geschieht dies durch Nachschlagen in einer Tabelle. Dabei werden boolesche Ausdrücke, bestehend aus einer oder mehreren Bedingungen, auf ihre Erfülltheit überprüft. Hinsichtlich der beschreibbaren Logik besteht zwischen beiden Darstellungsarten kein Unterschied: Ein Entscheidungsbaum lässt sich in eine äquivalente Entscheidungstabelle umwandeln und umgekehrt [MARTIN & MCCLURE, 1985]. Entscheidungsbäume erlauben die Festlegung und graphische Darstellung des Entscheidungsweges, d. h. die wohldefinierte Reihenfolge mit der die Bedingungen ausgewertet werden. Der Endknoten (Blatt) eines Entscheidungszweiges repräsentiert die jeweils auszuführende Aktion bzw. das gesuchte Resultat [PRÉVÔT, 1995]. Wenn Wissen in multiplen Hierarchien vorliegt, d. h. ein Knoten mehrere Vorgänger besitzen kann, so müssen Entscheidungstabellen verwendet und wiederholt angewandt werden. KOPF [1993] stellt eine Situationsanalyse mit Entscheidungsbäumen für die Assistenz von Kraftfahrzeugführern auf Autobahnen vor. HERRMANN [1997] zeigt eine Methode, mit der aus einer Menge von klassifizierten Trainingsbeispielen ein Entscheidungsbaum ‚gelernt‘ werden kann, welcher die Trainingsbeispiele korrekt klassifiziert.

### 2.3.2.6 Produktionssysteme

Produktionssysteme bestehen prinzipiell aus 3 Komponenten: *Langzeitspeicher*, *Kurzzeitspeicher* und *Regelinterpreter*. Der *Langzeitspeicher* beinhaltet Produktionsregeln der Form

WENN  $\langle \text{Bedingung } 1, \dots, \text{Bedingung } n \rangle$  DANN  $\langle \text{Aktion } 1, \dots, \text{Aktion } 2 \rangle$ .

und Fakten, die immer erfüllt sind. Der Inhalt des Langzeitspeichers ändert sich nicht oder nur langsam. Der *Kurzzeitspeicher* nimmt die von den Produktionen modifizierte Daten auf und stellt den aktuellen Zustand des Produktionssystems dar. Der *Regelinterpretierer* führt den Inferenzprozess durch. Er entscheidet aufgrund des aktuellen Inhalts des Kurzzeitspeichers, welche Produktion gefeuert wird und aktualisiert daraufhin den Kurzzeitspeicher [RUCKDESCHEL, 1996]. Die Bedingungsteile eines Produktionssystems können teilweise gleichzeitig erfüllt sein. In einem solchen Fall muss eine Konfliktauflösung (*matching*) nachgeschaltet werden, welche die zu feuernde Produktion ermittelt [STÜTZ, 1999]. Zur Konfliktauflösung bieten sich verschiedene Strategien an: Eine Strategie ist es, die Regeln in eine vordefinierte Reihenfolge zu bringen (z. B. sortiert nach der Priorität ihrer Ausführung), und immer die erste, erfüllte Regel in dieser Reihenfolge anzuwenden. Weitere Strategien sind, immer diejenige Regel auszuwählen, deren Vorbedingungen am meisten spezifisch ist<sup>18</sup> oder die am längsten nicht angewandt wurde. Ferner kann man durch sogenannte Meta-Regeln festlegen, unter welchen Bedingungen eine Regel einer anderen Regel vorzuziehen ist. Eine weitere Strategie ist es, alle anwendbaren Regeln in parallelen Ableitungsschritten weiterzuverfolgen [REIMER, 1991] [PRÉVÔT, 1995]. ZISMAN [1978] schlägt Methoden zur Strukturierung großer Regelbasen vor. Insbesondere untersucht er die Eignung von Produktionssystemen zur Modellierung ereignisgesteuerter, asynchroner und nebenläufiger Prozesse.

### 2.3.2.7 Petrinetze

Petrinetze eignen sich besonders zur Modellierung ereignisdiskreter und nebenläufiger Systeme [RUCKDESCHEL, 1996]. Durch sie wird versucht, Prozess-Systeme als Vernetzung elementarer Prozessatome zu betrachten. Diese Prozessatome sind Stellen, Transitionen, Kanten und Marken. Stellen dienen der Modellierung von diskreten Zuständen und werden graphisch als Kreise dargestellt. Transitionen formulieren Übergangsbedingungen und erscheinen in der graphischen Darstellung als Rechtecke oder Balken. Die Kanten verbinden die Stellen und Transitionen miteinander; über sie fließen die Marken. Die Marken in den Stellen geben den aktuellen Zustand des Petrinetzes an. Je nach Kapazität können Stellen mehrere (evtl. unbeschränkt viele Marken) aufnehmen. Der durch die Marken repräsentierte Sachverhalt hängt von der Semantik des Petrinetzes ab. Marken können z. B. die Erfülltheit eines Zustands oder die Verfügbarkeit einer Ressource anzeigen. Beim Feuern von Transitionen werden die Marken im Netz zwischen den Stellen verschoben. Das Gewicht einer Kante gibt die Anzahl der Marken an, die bei einem Schaltvorgang über die Kante fließen. Ein großer Vorteil von Petrinetzen ist die Verfügbarkeit von analytischen Verifikationsverfahren [STÜTZ, 1999].

### 2.3.2.8 Fallbasiertes Schließen

Unter fallbasierten Schließen (*case based reasoning* (CBR)) versteht man die Anwendung von gespeichertem Beispielwissen (Erfahrungen) auf aktuell zu lösende Problemstellungen. Abb. 2.2 zeigt den prinzipiellen Ablauf des CBR. Häufig separiert man beim fallbasierten Schließen die Fälle in einen Anfrage- und Lösungsteil. In der Anfrageanalyse erfolgt die charakteristische Untersuchung der Anfrage, so dass sie unter Benutzung eines zu spezifizierenden Ähnlichkeitsmaßes mit den Fällen in der Fallbasis verglichen werden kann. Das fallbasierte Schließen sucht nach jenen Fällen, deren Anfrageteile der Anfrage am ähnlichsten sind. Ziel ist es dabei, die Suchdauer möglichst kurz zu halten. Das Ergebnis der Suche ist die Fallauswahl, bestehend aus einer mehr oder minder großen Anzahl von Fällen [SCHAAF, 1998].

<sup>18</sup> Eine Vorbedingung  $v_1$  ist spezifischer als eine Vorbedingung  $v_2$ , wenn sie in weniger Fällen erfüllt ist (also  $v_1 \Rightarrow v_2$  gilt, aber nicht  $v_2 \Rightarrow v_1$ ).

Die Fallanpassung führt schließlich eine Adaption der ausgewählten Fälle an die Anfrage durch. Eine Adaptionmöglichkeit ist z. B. die Interpolation zwischen den Lösungen der ähnlichsten Fälle. Optional kann die Nützlichkeit der gefundenen Lösung getestet und gegebenenfalls zusammen mit der Anfrage explizit in der Fallbasis gespeichert werden. Dadurch wird die adaptierte Lösung wiederverwendbar [STÜTZ, 1999].

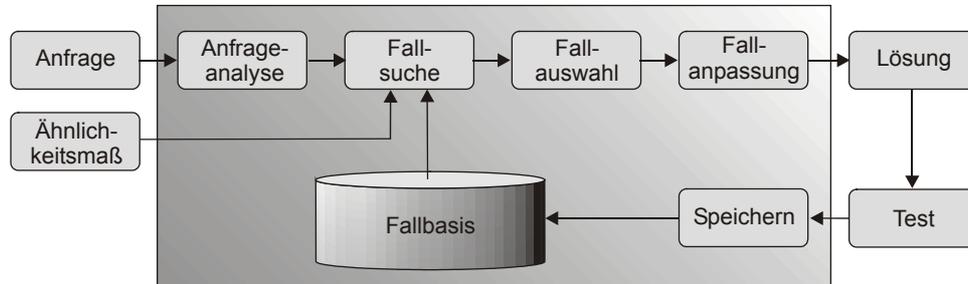


Abb. 2.2: Ablauf beim fallbasierten Schließen

### 2.3.3 Beschreibung durch unscharfe Mengen und Regeln

Scharfe Logik eignet sich gut zur Beschreibung von scharfen Phänomenen, z. B. ob ein Ereignis eingetreten ist oder nicht. Daneben gibt es jedoch unscharfe Phänomene, die sich nicht durch scharfe Ausdrücke beschreiben lassen. LIU [1993] nennt als Beispiel den Begriff der Verkehrssicherheit. So ist im Hinblick auf die Verkehrssicherheit die Grenze zwischen „sicher“ und „unsicher“ fließend und die exakte bzw. klassische Mathematik ist für deren Beurteilung weniger geeignet. Überlegungen dieser Art inspirierten den amerikanischen Systemtheoretiker ZADEH [1965] dazu, eine qualitativ neuartige Modellbildungsstrategie zu propagieren. In dieser Strategie wird ein Übergang von einer mit traditionellen mathematischen Mitteln arbeitenden Modellierung zu unscharfen Mengen durchgeführt. Seine Lehre – ‚fuzzy set‘ – begründete er mit folgendem Inkompatibilitätsprinzip:

*„In general, complexity and precision bear an inverse relation to one another in the sense that, as the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics. Thus fuzzy thinking may not be deplorable, after all, if it makes possible the solution of problems which are much too complex for precise analysis.“*

Laut obigem Zitat stehen die Komplexität eines Systems und die Genauigkeit seiner Modellierung in Diskrepanz zueinander. ZADEH [1965] geht davon aus, dass eine exakte Beschreibung eines komplexen Systems schnell an seine Grenzen stößt, und dass das (fehlerhafte) Ergebnis im zunehmenden Maße unbedeutend bzw. unbrauchbar wird. Wenn Genauigkeit und Bedeutsamkeit einander ausschließen, so müsse man sich mit einer unscharfen Darstellung des Systems abfinden. Ferner können Menschen bei der Formulierung von Verhaltensregeln mit natürlichsprachlichen Begriffen besser umgehen als mit numerischen Größen. Aufgrund der Vagheit von natürlichsprachlichen Begriffen und der Unsicherheit automatisch ermittelter Schätzgrößen ist die begriffliche Abstraktion geometrischer Größen mit Hilfe unscharfer Mengen sehr vorteilhaft.

REKERSBRINK [1994] unterscheidet zwischen zwei Arten von Unschärfe: Die sogenannte *intrinsische Unschärfe* ist der Ausdruck der Ungenauigkeit menschlicher Empfindung sowie der Kontextabhängigkeit sprachlicher Begriffe. Dagegen resultiert die *informationale Un-*

*schärfe* aus der Unmöglichkeit, Informationen exakt zu erfassen und zu einer klaren Beurteilung zusammenzufügen. So kann der Begriff ‚sicherer Abstand‘ zwar kinematisch exakt formuliert sein, im Fahrbetrieb stehen aber weder die Informationen mit der notwendigen Genauigkeit zur Verfügung, noch ist der menschliche Fahrer in der Lage, diese zu evaluieren. Zur informationalen Unschärfe zählt der Autor auch die Ungewissheit über das Verhalten anderer Verkehrsteilnehmer, welches sich nur aufgrund von Erwartungen ungefähr prognostizieren lässt und stets unsicher bleibt.

PAULI [1993] stellt zwei Möglichkeiten vor, die Güte des Wissens, d. h. die Unsicherheit der Schätzwerte, bei der Zuordnung natürlichsprachlicher (Objekt-)Eigenschaften einfließen zu lassen. So schlägt er z. B. vor, die Varianz der Zustandsgrößen  $V_{X_i}$  (= Diagonalelemente der Kovarianzmatrix eines Kalman-Filters) auf einen Zusicherungsgrad  $f(V_{X_i}) \in [0,1]$  abzubilden, der mit kleiner werdender Varianz  $V_{X_i}$  größer wird. Der Gesamtzuordnungsgrad einer Objekteigenschaft ergibt sich dann aus der Kombination des aus der begrifflichen Unschärfe kommenden Zugehörigkeitsgrades und der Unsicherheit der Schätzung. Die zweite, vorgeschlagene Möglichkeit besteht darin, die Zugehörigkeitsfunktion<sup>19</sup> direkt mit der Schätzunsicherheit  $V_{X_i}$  zu parametrisieren. Durch eine solche Parametrisierung kann z. B. die Steilheit der Flanken der Zugehörigkeitsfunktion derart beeinflusst werden, dass eine symbolische Aussage bei kleinen Varianzen schon bei kleineren Abweichungen vom Mittelwert der Zugehörigkeitsfunktion ihre Gültigkeit verliert.

Nach REKERSBRINK [1994] besteht das grundlegende Konzept der Theorie unscharfer Mengen darin, die Kluft zwischen verbaler Kategorisierung von System- und Situationsbeschreibungen und der Trennschärfe der klassischen Logik zu überbrücken und mittels geeigneter Verrechnungsmethoden der numerischen Bearbeitung zugänglich zu machen. Dadurch wird es möglich, Ungenauigkeiten von nicht-stochastischem Charakter mathematisch auszudrücken.

Aufbauend auf die *fuzzy theory* können sogenannte linguistische Aussagen in Form von linguistischen Variablen<sup>20</sup> definiert werden, deren Werte mit unscharfen Mengen verknüpft sind. Die linguistischen Aussagen ermöglichen es, eine Beschreibung oder ein Beurteilungsergebnis in umgangssprachlich üblichen Worten auszudrücken unter Ausnutzung von vorhandenem System- und Prozesswissen.

Diese linguistischen Aussagen kombiniert man dann mittels unscharfer UND- und ODER-Operatoren zu unscharfen WENN-DANN-Regeln<sup>21</sup>. Die Prämissen der Regeln machen oft zustands- und situationsbeschreibende Aussagen und die Konklusionen beinhalten unscharfe Aussagen darüber, welche Aktionen in den jeweiligen Fällen auszuführen sind. Die Verwendung von Fuzzy-Regeln zur Problemlösung wird in der Literatur auch als *fuzzy reasoning* bezeichnet. BANDER & GOTTWALD [1993] nennen dafür auch den Begriff des *approximativen Schließens*, bei dem an eine Menge von linguistischen Werten von Situationsaspekten „elastische Bedingungen“ gestellt werden. Die Aufgabe des approximativen Schließens ist es, die in den unscharfen Bedingungen und Beziehungen niedergelegte Information sinnvoll und sachgemäß zu verarbeiten, und zwar in einer Art und Weise, die dem „Alltagsdenken“ nahe kommt. Dass dieses „Alltagsdenken“ z. B. zur Verarbeitung von Verkehrsinformation in der Lage ist, zeigt das tägliche Leben.

Die Attraktivität einer Verhaltensmodellierung durch Fuzzy-Regeln liegt auch in der Tatsache begründet, dass sie eine inhaltserhaltende Repräsentation und Verarbeitung von Exper-

<sup>19</sup> Der Begriff der Zugehörigkeitsfunktion wird in Kapitel 3.2.2.7 erläutert.

<sup>20</sup> Die Eigenschaften einer linguistischen Variablen beschreibt Kapitel 3.2.2.

<sup>21</sup> Der Aufbau unscharfer Regeln wird in Abschnitt 3.4.1 ausführlich dargestellt.

tenwissen ist, dessen linguistische Form auch vom Mensch benutzt wird. Ferner kann ausgehend von wenigen Regeln schnell der erste Prototyp getestet werden (*rapid prototyping*) und es lassen sich schnell und einfach neue Regeln hinzufügen. Nach REKERSBRINK [1994] darf eine unscharfe Regelbasis auch widersprüchliche Aussagen enthalten, da der Fuzzy-Regler in der Lage ist, solche Widersprüche mit einem Kompromiss durch „gewichtete Mittelwertbildung“ aufzulösen.

Die Verkehrsteilnehmer generieren ihr Verhalten im Allgemeinen nach dem gleichen Satz von Verkehrsregeln. Daneben besitzen Menschen jedoch unterschiedliche Denkweisen, die sich auch auf die Interpretation der Regeln auswirken. Dies führt dazu, dass die gleichen Regeln von verschiedenen Menschen unterschiedlich interpretiert werden. LIU [1993] schlägt vor, in Fuzzy-Beurteilungsmodellen diese verschiedenen Interpretationen dadurch zu berücksichtigen, indem man den Aufbau und die Terme der Regeln zwar unverändert lässt, jedoch die Fuzzy-Operatoren je nach Typ des Interpreters variiert. Eine solche Variation von Fuzzy-Operationen untersucht z. B. IRMSCHER [2001].

REKERSBRINK [1994] realisierte eine mikroskopische Verkehrssimulation, in welcher die einzelnen Fahrer-Fahrzeug-Elemente<sup>22</sup> als Fuzzy-Regler formuliert sind. Der Autor weist nach, dass die Fuzzy-Regler verschiedene Fahraufgaben auch bei ungenauen und zum Teil unsicheren Informationen bewältigen, und dass sich das in der Realität beobachtete Verkehrsgeschehen durch einfache, linguistische Regeln nachbilden lässt. Der intuitiv verständliche Regelkatalog lässt sich dabei komfortabel und entkoppelt von der eigentlichen Reglerimplementierung anpassen. Die zentrale Variable für die Modellierung des Fahrverhaltens beim Fahrzeugfolgen und beim Fahrstreifenwechsel ist die potentielle Kollisionszeit TTC (*time to contact*, vgl. [GODTHELP, 1984]). Sie ist die zu einem bestimmten Zeitpunkt verbleibende Restzeit bis zur Kollision bei unveränderten kinematischen Verhältnissen. Die potentielle Kollisionszeit leitet sich aus den physiologischen Rahmenbedingungen der visuellen Wahrnehmung ab<sup>23</sup>. Dadurch wird berücksichtigt, dass dynamische Größen wie Abstände oder Geschwindigkeitsdifferenzen nur dann eine Wirkung nach sich ziehen können, wenn diese Größen für einen Fahrer überhaupt visuell wahrnehmbar sind.

Ein generelles Problem der Fuzzy-Systeme ist nach STROHAL [1999] die Festlegung der konkreten Zugehörigkeitsgrade im Intervall  $[0,1]$ , welche sich streng genommen nur durch aufwendige statistische Erhebungen in befriedigender Weise bestimmen lassen. An Lernverfahren, mit denen diese Werte anhand von Beispieldaten automatisch ermittelt werden können, besteht daher großes Interesse.

### 2.3.4 Architekturen für autonome Systeme

In der Vergangenheit wurden zwei verschiedene Wege beschritten, um komplexe Systeme zu kontrollieren: Die ältere, funktionsbasierte und klassische KI-Methode benutzt eine explizite Repräsentation der Welt, um zu planen und Verhaltensentscheidungen daraus abzuleiten. Die sogenannten verhaltensbasierten Systeme brachen Mitte der 80er Jahre mit dieser Tradition. Synonyme für funktionsbasierte Architekturen sind *deliberative*<sup>24</sup>, *top-down* oder *wissensgesteuerte (knowledge-driven)* Architekturen. Verhaltensbasierte Systeme werden auch *reaktive, bottom-up* oder *datengesteuerte (data-driven)* Architekturen genannt. Die folgenden

<sup>22</sup> Ein Fahrer-Fahrzeug-Element ist ein Modell von Fahrer- und Fahrzeugeigenschaften, das zur Interaktion mit der Straßenumgebung und mit anderen Verkehrsteilnehmern herangezogen wird.

<sup>23</sup> Nach REKERSBRINK [1994] ist beim Bremsverhalten von Kraftfahrern nicht der absolute Abstand die entscheidende und geschätzte Größe, sondern der Zeitpunkt bis zum Zusammenstoß, welchen der Fahrer aus dem Fluss des visuellen Feldes abschätzt.

<sup>24</sup> unter Ausnutzung eines expliziten Umweltmodells.

Unterabschnitte 2.3.4.1 und 2.3.4.2 betrachten die Eigenschaften dieser beiden Architekturen. Die im Abschnitt 2.3.4.3 beschriebenen hybriden Architekturen versuchen, einen Kompromiss zwischen rein reaktiven und rein planbasierten Ansätzen zu finden. Der Kompromiss besteht i. d. R. darin, in den unteren Systemebenen reaktive Module vorzusehen, jedoch auf höheren Ebenen nicht auf Planung und Koordination zu verzichten. ARKIN [1998] und HAMDI [1999] geben für alle Architekturtypen eine große Anzahl von Beispielen in der Literatur an.

#### 2.3.4.1 Funktionsbasierte, deliberative Architekturen

Die funktionsbasierte Architektur ist die klassische Art von Steuerungssystemen, bei der das Steuerungsproblem in eine Reihe von eigenständigen Funktionseinheiten zerlegt wird. Jede dieser Einheiten bearbeitet eine gewisse Aufgabe wie Wahrnehmung, Planung oder Aktion. Das Gesamtverhalten des Systems entsteht dadurch, dass jedes dieser Module parallel aktiv ist und eine komplexe Interaktion zwischen Perzeption, Kognition und Aktion abläuft [HAMDI, 1999]. Die Leistungsfähigkeit des Gesamtsystems wird durch den Entwicklungsstand der einzelnen Komponenten bestimmt. Nach ARKIN [1998] haben deliberative Architekturen folgende wesentliche Eigenschaften:

- Sie besitzen hierarchische Strukturen mit klar identifizierbaren, funktionalen Untereinheiten.
- Sie sind stark abhängig von einer symbolischen Repräsentation der Welt. Diese muss konsistent, fundiert, vollständig und sicher sein.
- Die Kommunikation und der Kontrollfluss zwischen den Hierarchieebenen und zwischen den Einheiten (*bottom-up* und *top-down*) ist klar definiert.
- Höhere Hierarchieebenen geben den tieferen Ebenen Teilziele vor.
- Die Hierarchieebenen besitzen unterschiedliche zeitliche und räumliche Planungshorizonte. Mit absteigender Hierarchieebene werden der zeitliche Horizont kürzer und die räumlichen Betrachtungen lokaler.

Der wichtigste Vorteil von deliberativen Ansätzen ist das Vorhandensein einer globalen Sicht, aufgrund derer eine komplexe Planung unter Ausnutzung von KI-Methoden durchgeführt werden kann. Diese Planung ermittelt eine Sequenz von Aktionen, die sicher zum Ziel führt; oder man kann schon von Beginn an zeigen, dass die Aufgabe unlösbar ist. Ferner ist eine Optimierung der Pläne möglich, schon bevor die erste Aktion erfolgt. Dies führt zu der charakteristischen Zielorientiertheit deliberativer Systeme.

Ein großer Nachteil von deliberativen Architekturen ist ihre mangelnde Reaktivität. Wegen den detaillierten, komplexen Berechnungen dauert der Planungszyklus relativ lange. Wenn sich nun in diesem Zeitraum die dynamische Umwelt ändert, so kann dies zu einem ungültigen Plan führen und dem System liegt bis zum Eintreffen eines neuen, gültigen Plans kein Sollverhalten vor. Ein autonomes System läuft umso mehr in diese Gefahr, je dynamischer die Umgebung ist und je unberechenbarer sich die darin vorkommenden Objekte und Subjekte bewegen.

Wegen ihrer starken Abhängigkeit von einer symbolischen Repräsentation der Welt tritt bei deliberativen Systemen das bereits beim maschinellen Sehen beobachtete „*symbol grounding problem*“ auf (vgl. Abschnitt 2.1.1), d. h. die große Schwierigkeit, eine exakte, gültige Repräsentation der (natürlichen) Welt aufzubauen und der Realität nachzuführen. Viele Systeme versuchen deshalb, die Umweltbeschreibung und deren Rekonstruktion möglichst einfach zu gestalten, was aber im ungünstigsten Fall zu einer reduzierten Autonomie führt. Als weiteren Nachteil nennt HAMDI [1999] die mangelnde Robustheit deliberativer Systeme auf-

grund der gegenseitigen Abhängigkeiten der Module. Fällt nur eine Systemkomponente aus, so ist meist das ganze System funktionsunfähig und eventuell in einem gefährlichen Zustand.

#### 2.3.4.2 Verhaltensbasierte, reaktive Architekturen

Aufgrund der dargestellten Nachteile deliberativer Architekturen wurde Mitte der 80er Jahre damit begonnen, verhaltensbasierte, reaktive Systeme zu bauen [BROOKS, 1986]. Diese Ansätze besitzen dort Vorteile, wo die Umgebung zum Teil unbekannt und Sensordaten veräuscht sind oder das Operieren in einer dynamischen Umwelt und das Reagieren auf unerwartete Ereignisse ein Arbeiten in Echtzeit erzwingen. Reaktive Systeme verfügen über eine enge Kopplung von Wahrnehmung und Aktion, wobei die Kopplungen von speziellen Teilzielen bestimmt werden. Die Schlüsselaspekte von reaktiven Systemen sind nach ARKIN [1998]:

- Wahrnehmung und Aktion sind in sogenannten Verhaltensmustern, Verhaltensmodulen oder *behaviors* direkt miteinander verbunden. Die Sensoraktivität eines Verhaltensmusters generiert die notwendige (minimale) Information über die Umwelt, so dass der Aktionsteil des Verhaltensmusters ausgeführt werden kann. Die Verhaltensmuster dienen als fundamentale Bauelemente für die Aktionen des Systems und stellen die Antwort des Systems auf eine bestimmte Umweltsituation dar. Durch die direkte Kopplung ist die Zeitverzögerung zwischen Sensorik und Aktuatorik minimal. Es besteht keine Hierarchie zwischen den Verhaltensmustern.
- Man verzichtet bewusst auf eine explizite Repräsentation von räumlichen und zeitlichen Aspekten. Der Aufbau einer solchen abstrakten Weltbeschreibung wird als zu zeitaufwendig und zu fehleranfällig erachtet, um in natürlichen Umgebungen zu robusten Verhalten führen zu können. Stattdessen interpretiert man jede Aktion als direkte Reaktion auf einen wahrgenommenen Stimulus.
- Verhaltensbasierte Systeme sind bezüglich des Programmdesigns inhärent modular. Dies ermöglicht es, durch Hinzufügen weiterer Verhaltensmuster den Anwendungsbereich des Systems zu erweitern, ohne bestehende Teile anzutasten. Dieses Ansammeln von Fähigkeiten und die Wiederverwendung von Funktionalität sind für die Entwicklung komplexer Systeme unverzichtbar.
- Das verhaltensbasierte System verfügt über einen Satz von Verhaltensmuster mit geringer Komplexität, die auch gleichzeitig aktiv sein können, d. h. Stellgrößen für die Aktuatoren vorgeben. Zur Verhaltensgenerierung muss deshalb aus der Vielzahl von Stellgrößen diejenigen ermittelt bzw. fusioniert werden, die wirklich auf die Aktuatoren zugreifen.

Bei der Verhaltensgenerierung kann man zwischen den konkurrierenden (*competitive*) und kooperativen (*cooperative*) Ansätzen unterscheiden [ARKIN, 1998]. Bei den konkurrierenden Ansätzen ist immer nur ein Verhaltensmuster aktiv. Die einfachste konkurrierende Methode ist die *winner-take-all*-Arbitrierung, bei der nur ein Verhaltensmuster von einem Schiedsrichter (*arbiter*) ausgewählt wird. Daneben gibt es noch die Möglichkeit der Priorisierung der Verhaltensmuster, wobei immer das aktive Verhaltensmuster mit der höchsten Priorität den Vorzug erhält. Die Priorität spiegelt die Wichtigkeit des Verhaltensmusters für den Umgang mit der aktuellen Umweltsituation im Zielkontext wieder. Eine dritte Möglichkeit ist es, den sogenannten Aktivierungsgrad eines Verhaltensmusters einzuführen, der von den Zielen des autonomen Systems und vom Sensorinput abhängt. Dasjenige Verhaltensmuster mit dem aktuell größten Aktivierungsgrad bekommt den Zugriff auf die Aktuatoren. Bei einer vierten Methode kommen demokratische Aspekte zum Einsatz: Jedes aktive Verhaltensmuster votiert dabei für eine gewisse Aktion und die Aktion mit der höchsten Stimmenzahl wird ausgeführt.

Bei den kooperativen Methoden gehen die Aktionen aller aktiven Verhaltensmuster mit mehr oder weniger Gewicht in die Aktion des autonomen Systems ein. Die am meisten verbreitete Methode ist die Vektoraddition, bei der die Steuervektoren aller aktiven Verhaltensmuster (evtl. gewichtet) vektoriell addiert werden. Eine weitere Methode ist die sogenannte Potentialfeldmethode. Es existieren auch Arbitrierungsverfahren, die sowohl konkurrierende als auch kooperative Komponenten besitzen.

Ein großer Vorteil verhaltensbasierter Systeme ist ihre Robustheit. Da alle Verhaltensmuster die Sensoren und Aktuatoren des Systems bedienen, führt ein Ausfall eines Verhaltensmusters lediglich zu einer Leistungsminderung. Als wichtigsten Nachteil nennt HAMDI [1999] die mangelnde Zielorientierung, da globale Information und eine globale Planung fehlen. Pure Reaktionen auf den aktuellen Sensorinput garantieren nicht, dass das autonome System jemals das Ziel erreicht. Ein weiterer Nachteil ist die mangelnde Flexibilität. Die abgeschlossene Menge von Verhaltensmustern und ihre Arbitrierung führen zu einem festverdrahteten Steuerungssystem, welches nicht für verschiedene Aufgabenstellungen programmiert werden kann [SCHNEIDER-FONTAN, 1999]. Ferner setzen verhaltensbasierte Ansätze voraus, dass der Designer des Systems in der Lage ist, die notwendigen Verhaltenselemente zu entwerfen und für die richtige Interaktion zwischen diesen zu sorgen. Dies wird mit zunehmender Komplexität des Systems immer schwieriger.

### 2.3.4.3 Hybride Architekturen

In der Zwischenzeit sind die meisten Entwickler von Robotersystemen davon überzeugt, dass für die Ausschöpfung des ganzen Potentials verhaltensbasierter Ansätze sowohl deliberative als auch reaktive Komponenten notwendig sind. Solche hybriden Systeme erlauben eine Rekonfiguration der reaktiven Systemkomponenten aufgrund von Hintergrundwissen und gemachten Erfahrungen. So nennt ARKIN [1998] eine Reihe von Methoden in der Literatur, um eine Wissensrepräsentation in reaktive Architekturen zu integrieren, und dabei die vorteilhaften reaktiven Eigenschaften zu erhalten.

Meist ist die deliberative Komponente dafür verantwortlich, längerfristige Ziele zu verfolgen und für die notwendige Sequentialisierung der Aktionen zu sorgen. Im Gegensatz dazu ist es die Aufgabe der verhaltensbasierten Komponenten, die situativ richtige und zeitadäquate Umsetzung von Sensordaten in Aktuatorbefehle durchzuführen [HAMDI, 1999]. Nach ARKIN [1988] gibt es vier prinzipielle Strategien, um deliberative mit verhaltensbasierten Systemkomponenten zu koppeln:

- **Selektion:** Der deliberative Anteil bestimmt die Auswahl der aktiven Verhaltensmuster und deren Laufzeitparameter während der gesamten Ausführung.
- **Ratgeber:** Der deliberative Anteil schlägt dem verhaltensbasierten System Veränderungen in der Verhaltensauswahl vor. Das verhaltensbasierte System berücksichtigt diese aufgrund aktueller Sensordaten sofort, erst später oder überhaupt nicht.
- **Adaption:** Der deliberative Anteil verändert kontinuierlich die reaktiven Komponenten aufgrund von Veränderungen in der Situation und der Intention des Subjekts.
- **Aufschub (*postponing*):** Der deliberative Anteil greift auf die verhaltensbasierte Ebene so spät wie möglich zu und nur wenn unbedingt notwendig. Der Ansatz geht davon aus, dass die für korrekte Entscheidungen notwendigen Informationen erst spät im Prozess verfügbar werden und die möglichst aktuellen Sensordaten eine Rückwärts-Prädiktion vermeiden helfen.

Hybride Systeme können auch mehrere dieser Kopplungsstrategien parallel verfolgen. Zu bedenken ist einerseits, dass hybride Architekturen Kompromisse an beiden Enden des Spekt-

rums erfordern und andererseits die Grenze zwischen deliberativem und reaktivem Verhalten in der Literatur noch nicht ausreichend verstanden ist. Nach HAMDİ [1999] leiden hybride Systeme oft unter dem *Horizonteffekt*: So zeigt ein hybrides System zwar ein besseres Reaktionsverhalten, die Einschränkungen des Planungssystems verlagern sich aber nur etwas weiter in die Zukunft. So ist es eine Frage der Dynamik der Umwelt, ob der planende, deliberative Teil überhaupt das Verhalten beeinflussen kann. Nichtsdestotrotz finden sich in der Biologie zahlreiche, motivierende Beispiele für hybride Systeme.

Das in den Kapiteln 1.2.2 und 5 vorgestellte EMS-VISION System besitzt eine hybride Architektur, wie sie oben beschrieben ist. Beim EMS-VISION System gibt es keine feste, direkte Verbindung zwischen den Sensoren und Aktuatoren. Vielmehr verknüpft das Softwaresystem in Abhängigkeit von seinem internen Zustand den Sensorinput dynamisch mit den Aktuatoren. Diese Verknüpfung geschieht innerhalb sogenannter Fähigkeiten. Einmal aktiviert koppeln die darin gekapselten Vorsteuergesetze und Regelungen bestimmte, wahrgenommene Zustandsgrößen direkt mit den Aktuatoren des Fahrzeugs (vgl. Abschnitt 3.5.1).

### 2.3.5 Weiterführende Literatur

Zum Schluss dieses Kapitels werden einige Arbeiten genannt, die für den Themenkomplex ‚autonome Fahrzeuge‘ von Interesse sind und eine Vielzahl weiterer Literatur angeben. Zunächst ist die Arbeit von MAURER [2000] zu nennen, welche den Stand der Technik bei der Automatisierung von Fahrfunktionen mit Rechnersehen beschreibt. LÜTZELER [2002] fasst internationale Forschungsaktivitäten auf dem Gebiet sehender autonomer Kraftfahrzeuge zusammen. GREGOR [2002] gibt einen Überblick über den Stand der Technik im Bereich der autonomen Landfahrzeuge, welche für den Einsatz auf Wegenetzen und im Gelände bzw. in unstrukturierter Umgebung konzipiert wurden. DICKMANN [2002] gibt einen guten Überblick über die Entwicklung des Rechnersehens zum Zweck der Steuerung von Landfahrzeugen oder einzelner Fahrzeugfunktionen im letzten Jahrzehnt. STROHAL [1999] beschreibt den Stand der Technik bezüglich der Fehler- und Absichtserkennung und SCHREINER [1999] bezüglich Fahrerhaltensforschung und Fahrerunterstützungssysteme. JÜRGENSOHN [1997] behandelt die Problematik der Modellierung von menschlichen Fahrerhaltens aus methodologischer Sicht und stellt hybride Fahrermodelle in der Literatur vor. FERARIC [1996] und KOPF [1993] beschreiben bereits ältere Arbeiten zur Fahrerunterstützung. KRÜGER [1991] gibt einen Überblick über die Situationsmodellierung in der Bildfolgenauswertung. HEIMES [2000] nennt Arbeiten zur sichtgestützten Fahrerassistenz für den Innenstadtbereich und für Landstraßen. CHERFAOUI et al. [2000] versuchen, Unzulänglichkeiten der Signalgewinnung und der Signalverarbeitung und deren Auswirkungen auf die Situationserkennung und Verhaltensentscheidung quantitativ zu erfassen.

## **TEIL B**

# **KONZEPTE UND REALISIERUNGEN**



## Kapitel 3

### Verhaltensentscheidung

Zu Beginn dieses Kapitels sollen zunächst nochmals die wesentlichen Aspekte aus Abschnitt 1.2 in Gedächtnis gerufen werden, welche die Konzeption der vom Autor entwickelten Verhaltensentscheidung beeinflussten. Im EMS-VISION System wird das Verhalten des Fahrzeugs durch den selektiven Einsatz von Vorsteuergesetzen und Regelkreisen bestimmt. Diese Vorsteuer- und Regelgesetze verknüpfen funktionell die Sensorsignale mit den Aktuatoren des Fahrzeugs und sind in sogenannten Fähigkeiten gekapselt. Bei der Ausführung einer komplexen Mission werden diese Fähigkeiten nacheinander angestoßen und ausgeführt. Nach DICKMANN & WÜNSCHE [1999] sind folgende drei Punkte von einem System zu erfüllen, um komplexe Missionen autonom ausführen zu können:

- Das System benötigt für die Ausführung einer Mission einen vollständigen Satz von Fähigkeiten.
- Das System muss Wissen darüber besitzen, in welchen Situationen es welche Fähigkeiten mit welcher Parametrisierung zu nutzen hat, um ein formuliertes Missionsziel zu erreichen. Dieses Wissen kann in quasi-statische Regeln gefasst werden.
- Das System muss durch Sensoren und Erkennungsprozesse in der Lage dazu sein, diese Situationen auch zu erkennen.

Für die Planung der Aktionen sind die Auswirkungen der Fähigkeiten auf den höheren Systemebenen symbolisch zu repräsentieren. Bei dieser Repräsentation ist es ausreichend, die Auswirkungen der in den Fähigkeiten gekapselten Vorsteuergesetzen und Regelungen auf die systemdynamischen Zustandsgrößen durch wenige, quantitative Größen zu beschreiben (Anfangs- und Endzustände, Ausführungsdauer, usw.). Die Verhaltensentscheidung muss systemdynamische Aspekte und Echtzeitbedingungen nicht berücksichtigen, da dieses innerhalb der Fähigkeiten geschieht.

Die eigentliche Ausführung einer Fähigkeit läuft auf der systemdynamischen Ebene ab. Nur diese Ebene hat Zugang zu allen Messdaten, Modellen und Stellgliedern. Dabei wird entweder mit einem Vorsteuergesetz ein gewünschter Zustandsübergang durchgeführt, oder mittels eines Reglers ein Sollzustand stabilisiert. Das Vorsteuersignal kann unter Beachtung der bekannten Systemdynamik als Referenz für den Rückkopplungszweig eines unterlagerten Reglers aufgeschaltet werden. Die unterlagerten Regler kompensieren die während der Ausführung der Fähigkeiten auftretenden Störungen [DICKMANN, 2001].

Kritische Verkehrssituationen werden nur dann korrekt eingeschätzt, wenn das wahrscheinliche Verhalten anderer Verkehrsteilnehmer berücksichtigt wird. MOCK-HECKER [1994] nennt als Beispiel für diesen Sachverhalt ein Überholzenario mit weit entferntem Gegenver-

kehr. Die alleinige Betrachtung der systemdynamischen Größen (Positionen und Geschwindigkeiten der Fahrzeuge) lässt die Überholsituation gefährlich erscheinen. Nur unter der Annahme, dass der Überholende nach Abschluss seines Überholvorgangs wieder auf die rechte Spur zurückkehren wird, erscheint die Situation für das entgegenkommende Auto (noch) unkritisch. Kritisch ist die Situation erst, wenn das Wiedereinscheren behindert wird.

Der eleganteste Weg das Verhalten anderer Verkehrsteilnehmer einzuschätzen, ist sich in deren Situation hineinzusetzen und die eigenen Entscheidungskriterien darauf anzuwenden [DICKMANN, 1989]. Dazu müssen die eigenen Verhaltensregeln auf andere Subjekte in der Szene anwendbar sein. Mit den vermuteten Aktionen der anderen Fahrzeuge kann das Eigenfahrzeug die Situation präzisieren und sein Verhalten danach richten. Diese Vorgehensweise setzt voraus, dass sich Fremdfahrzeuge nach den gleichen Regeln verhalten wie das Eigenfahrzeug. Bei nicht erwartungskonformen Verhalten anderer Fahrzeuge kommt es zu einer fehlerhaften Prädiktion der Situation. Diesem Problem steht aber auch der Mensch gegenüber.

Ein weiterer wichtiger Punkt bei der Beurteilung von Verkehrssituationen sind Konflikte zwischen den (geplanten) Aktionen verschiedener Fahrzeuge. Auch wenn eine Aktion eines Fahrzeugs für sich alleine konfliktfrei und ausführbar ist, so können doch die Aktionen verschiedener Fahrzeuge in Konflikt zueinander stehen. Ein Konfliktfall ist z. B. das oben dargestellte Überholmanöver mit (nahe) Gegenverkehr. Die Verhaltensentscheidung eines Fahrzeugs muss solche Konflikte mit berücksichtigen.

In den nächsten Unterkapiteln werden nacheinander die einzelnen Komponenten des im Rahmen dieser Arbeit entwickelten Konzepts zur Verhaltensentscheidung beschrieben. Zuvor gibt Abschnitt 3.1 einen Überblick über das Konzept als Ganzes und beschreibt seine Einbettung in das EMS-VISION System.

### 3.1 Überblick

Wissen ist die Grundlage jeder Entscheidung. Dabei beinhaltet das Wissen nicht nur räumliche 3D-Information von den vorhandenen Objekten in der Umgebung, sondern auch zeitliche Aspekte. Nur mit räumlich-zeitlichem Hintergrundwissen können Bewegungsabläufe von Objekten und Aktionsschemata von Subjekten verstanden werden.

Aus diesem Grund verfügt das EMS-VISION System<sup>1</sup> über eine Wissensrepräsentation, die entsprechend den unterschiedlichen Arten des darin niedergelegten Wissens in mehrere Teile gegliedert ist. Ein wichtiger Teil der Wissensrepräsentation ist die Dynamische Objektdatenbasis (DOB) (vgl. Abb. 3.1), welche wegen ihrer baumartigen Strukturierung auch Szenenbaum genannt wird. Die DOB beinhaltet das beste Wissen über die systemdynamischen Zustände der in der realen Szene vorkommenden physikalischen Objekte<sup>2</sup>. Ein physikalisches (Teil-)Objekt wird dabei durch einen sogenannten Szenenknoten repräsentiert, der über ein Lagemodell, ein Formmodell und die dazugehörigen systemdynamischen Zustandsgrößen verfügt. Jeder Szenenknoten stellt eine Schnittstelle zu den objektspezifischen Daten dar [RIEDER, 2000]. Daneben ist in der Wissensrepräsentation der Missionsplan niedergelegt, welcher vom autonomen Fahrzeug zu verfolgen ist [GREGOR, 2002].

Bei der Untersuchung der Verkehrssituation müssen situationsbeschreibende Aussagen generiert werden. Diese ergeben sich in der Regel durch Beobachtung der systemdynamischen Größen über mehrere Sekunden hinweg. Situationsbeschreibende Aussagen fassen das impli-

<sup>1</sup> Das EMS-VISION System wird in den Kapiteln 1.2.2 und 5 beschrieben.

<sup>2</sup> Die vom Autor benutzte Implementierung des Szenenbaums geht auf RIEDER [2000] zurück.

zit in der DOB vorhandene Wissen in geeigneter Weise zusammen und machen es so einer Weiterverarbeitung durch die Verhaltensentscheidung zugänglich. Im Modul zur Situationsanalyse geschehen situationsbeschreibende Aussagen innerhalb sogenannter Situationsaspekte. Ein Situationsaspekt beschreibt einen Teilaspekt der Verkehrssituation; alle Situationsaspekte zusammen umschreiben die gesamte Situation. Bei den Situationsaspekten handelt es sich um linguistische Variablen im Sinne der Fuzzy-Theorie mit unscharfen (linguistischen) Werten im Intervall [0,1]. Die linguistischen Werte erlauben nicht nur die boolesche Aussage „der Situationsaspekt trifft zu/nicht zu“, sondern es kann differenziert werden, wie gut und in welcher Weise der Situationsaspekt zutrifft.

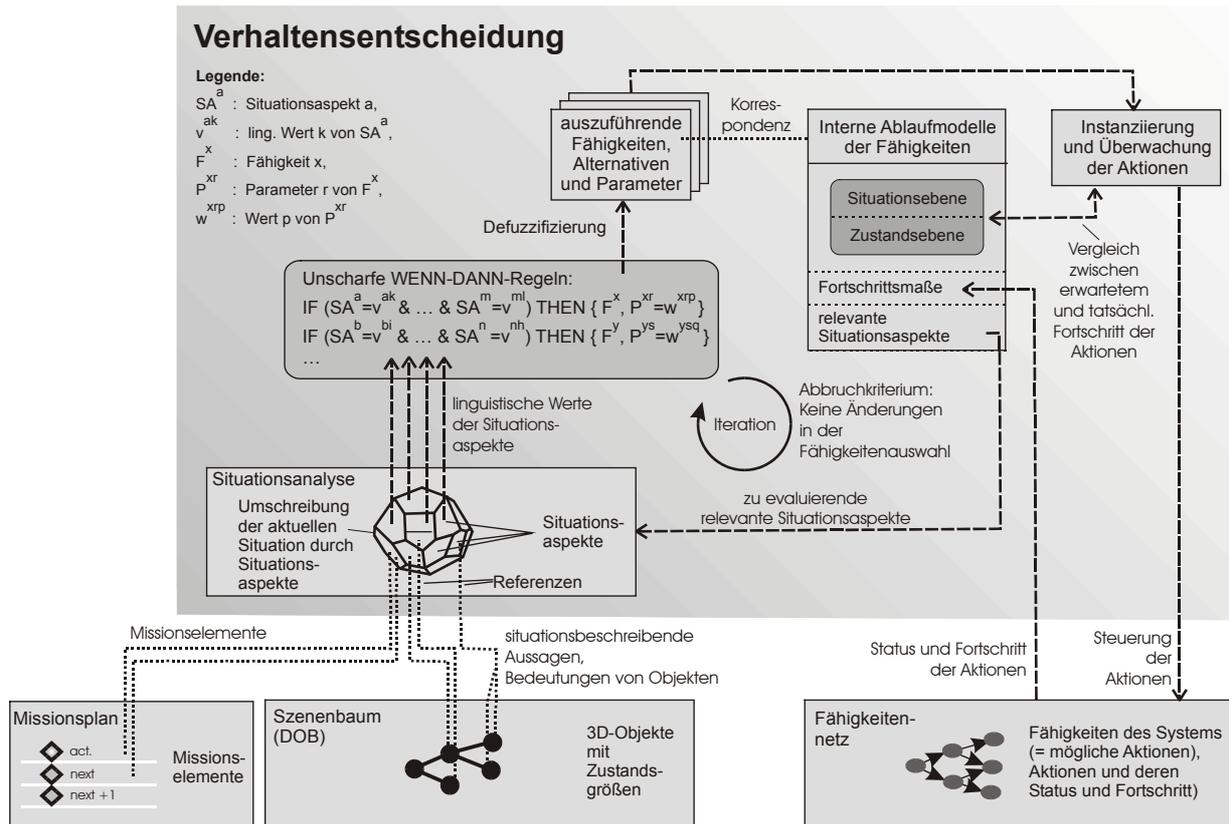


Abb. 3.1: Die Komponenten der Verhaltensentscheidung und deren Einbettung ins Gesamtsystem

Da es eine Vielzahl von Situationsaspekten gibt und diese auf eine noch größere Anzahl von Objektkombinationen angewandt werden können, ist eine grundsätzliche Berechnung aller Situationsaspekte ineffizient und bei detaillierter werdender Situationsbeschreibung in Echtzeit nicht durchführbar. Deshalb kann und soll die Berechnung nur solcher Situationsaspekte erfolgen, die von der Verhaltensentscheidung benötigt und weiterverarbeitet werden. Ob eine Situationsbeschreibung ausreichend detailliert ist, hängt davon ab, welche Fähigkeiten dem System zur Verfügung stehen und welche davon überhaupt in Frage kommen. Für eine Vorauswahl von situationsgerechten Fähigkeiten benötigt die Verhaltensentscheidung aber eine Situationsbeschreibung als Entscheidungsgrundlage. Dies führt zu einer iterativen Durchführung der Situationsanalyse und Verhaltensentscheidung.

Bei der Ausführung einer Fähigkeit sind gewisse Situationsaspekte relevant, welche – wie eben dargestellt – von der Situationsanalyse nicht automatisch berechnet werden. Nach der Auswahl einer Fähigkeit durch die Anwendung der Regelbasis erfolgt nicht unmittelbar die Instanziierung der Fähigkeit. Vielmehr spezifiziert das interne Ablaufmodell der Fähigkeit

(vgl. unten), welche Situationsaspekte auf welche Objekte im Szenenbaum anzuwenden sind, um die Fähigkeit sicher ausführen zu können (vgl. Kapitel 3.3.6). Die Situationsanalyse berechnet anschließend die neu hinzugekommenen Situationsaspekte. Die Situationsaspekte führen z. B. eine quantitative Prädiktion der Situation durch oder decken Konflikte auf.

Mit der so erweiterten Situationsrepräsentation wird anschließend erneut die Regelbasis ausgewertet. Die dadurch neu ausgewählten Fähigkeiten spezifizieren zusätzliche, relevante Situationsaspekte und ein weiterer Iterationsschritt in der Situationsanalyse folgt. Mit zunehmender Anzahl von Iterationsschritten wird die Situationsrepräsentation immer detaillierter. Die Iteration bricht ab, wenn sich beim letzten Iterationsschritt für kein Subjekt in der Szene die Fähigkeitauswahl geändert hat. Im Anschluss daran instanziiert man die für das Eigenfahrzeug ausgewählten Fähigkeiten über das Fähigkeitenetz.

Die Verhaltensentscheidung erfolgt durch eine Regelbasis mit unscharfen WENN-DANN-Regeln. Der WENN-Teil einer Regel besteht aus einer Verknüpfung von unscharfen Termen durch unscharfe UND- und ODER-Operatoren. In den Termen sind Bedingungen formuliert, welche die aktuellen, linguistischen Werte der Situationsaspekte auswerten. Im DANN-Teil einer Regel befinden sich situationsgerechte Fähigkeiten mit ihren Parametern und Alternativen. Die Fähigkeiten sind priorisiert: Die Fähigkeiten mit niedriger Priorität kommen zum Einsatz, wenn die Fähigkeiten mit höherer Priorität momentan nicht verfügbar oder gescheitert sind.

Die Verhaltensentscheidung kann natürlich nur solche Fähigkeiten auswählen, welche das autonome Fahrzeug (aktuell) auch ausführen kann. Aus diesem Grund sind die Fähigkeiten des Systems und ihre Abhängigkeiten statisch zu repräsentieren. Dies geschieht durch das sogenannte Fähigkeitenetz (vgl. Abschnitt 3.3.5). Von der prinzipiellen (statischen) Verfügbarkeit zu unterscheiden ist die aktuelle, dynamische Verfügbarkeit. Letztere richtet sich nach eventuell vorhandenen Fehlern und Ausfällen bei Systemkomponenten, nach der Güte des momentan vorhandenen Wissens, und nach den bereits genutzten Fähigkeiten im System. Die Fähigkeitenrepräsentation zeigt dazu auch die dynamische Verfügbarkeit an.

Die Ausführung komplexer Fähigkeiten erfordert die Zusammenarbeit mehrerer Experten und führt zu der Problematik, ein verteiltes System zu steuern und zu koordinieren. Die dazu notwendigen Mechanismen sind in das Fähigkeitenetz integriert. Weil es während der Ausführung von Fähigkeiten zu Fehlern und Verzögerungen kommen kann, muss der Fortschritt der Aktionen über die Verhaltensentscheidung rückgekoppelt werden. Die Fähigkeitenrepräsentation zeigt zu diesem Zweck den Fortschritt und Status der in den Experten ablaufenden Aktionen an.

Innerhalb eines hierarchischen, verteilten Systems nimmt mit steigender Hierarchieebene die Entscheidungskompetenz und der Abstraktionsgrad des bei den Entscheidungen verwendeten Wissens zu. Deshalb muss jeder Experte sein detailliertes Expertenwissen dazu nutzen, um sinnvolle Maße für den Fortschritt der bei ihm ablaufenden Aktionen zu generieren. Die Fähigkeitenrepräsentation reicht diese Maße an höhere Hierarchieebenen weiter. Das Maß für den Fortschritt muss so aussagekräftig sein, dass sich die übergeordneten Module ein symbolisches Bild von der sich entwickelnden Situation und den damit verbundenen Veränderungen bei den systemdynamischen Größen machen können (vgl. Abschnitt 3.3.6).

Wie oben bereits angeklungen ist, muss eine Verhaltensentscheidung nicht nur die zur Verfügung stehenden Fähigkeiten kennen, sondern auch eine Vorstellung von deren Ablauf besitzen. Aus diesem Grund verfügt die Verhaltensentscheidung über interne Modelle der Fähigkeiten, die deren Abläufe auf unterschiedlichen Ebenen beschreiben. Auf der Situationsebene

wird der Ablauf einer Fähigkeit in Anfangs- und Endzustand und beliebig viele Zwischenzustände unterteilt. Jeder Zustand repräsentiert einen Abschnitt im Ablauf, in dem sich Werte gewisser Situationsaspekte ändern. Mit diesen Änderungen gehen systemdynamische Zustandsänderungen einher. Auch diese können im internen Ablaufmodell einer Fähigkeit beschrieben werden. Auf diese Weise sind die Zustandsänderungen auf der Situationsebene mit Erwartungen an Verläufe gewisser systemdynamischer Größen verknüpft.

Durch die Instanziierung einer Fähigkeit und aller dafür notwendigen Teilfähigkeiten entsteht ein Netz von zusammengehörigen und voneinander abhängigen Aktionen, welche innerhalb des verteilten Systems in den verschiedenen Experten der Fähigkeiten ablaufen. Die Verhaltensentscheidung besitzt die Möglichkeit, durch gezielte Steuerbefehle über das Fähigkeitsnetz Aktionen mit hoher Priorität den Zugriff auf beschränkte Ressourcen zu gestatten und andere nur als vorgehaltene Alternativen „mental“ mitlaufen zu lassen.

Über das Fähigkeitsnetz werden automatisch der Status und Fortschritt der Aktionen kommuniziert. Die internen Ablaufmodelle innerhalb der Verhaltensentscheidung bringen diese Größen in direkten Zusammenhang mit den sich ändernden linguistischen Werten der Situationsaspekte und den sich ändernden Zustandsgrößen der Objekte. Scheitert eine Aktion oder kommt es zu einer Verzögerung bei der Ausführung, so besitzt die Verhaltensentscheidung eine Vorstellung vom momentanen Zustand auf der Situations- und Zustandsebene und kann daraus geeignete Reaktionen ableiten. Die internen Ablaufmodelle der Fähigkeiten erlauben eine sinnvolle Rückkopplung der von der Verhaltensentscheidung generierten Vorgaben.

Nach diesem Überblick über das Konzept zur Verhaltensentscheidung stellen die nun folgenden Kapitel 3.2-3.4 die einzelnen Komponenten detaillierter vor.

## 3.2 Situationsanalyse

### 3.2.1 Grundlegendes zur Analyse von Situationen

In der Literatur werden Situationsmodelle oft als Bindeglied zwischen Dateninterpretation und Handlungssteuerung angesehen, deren Aufgabe es ist, aus der Fülle von Information denjenigen Ausschnitt zu erfassen, der gerade handlungsrelevant ist. KRÜGER [1991] versteht unter dem Begriff „Situation“ die Sichtweise, wie sich ein im Straßenverkehr autonom handlungsfähiges System zu einem bestimmten Zeitpunkt in seine Umgebung eingebettet sieht. In der vorliegenden Arbeit steht der Begriff der Situation für die Summe der für die Verhaltensentscheidungen relevanten Sachverhalte (vgl. Abschnitt 1.1.3).

Auch im Bereich der kognitiven Psychologie existiert eine entsprechende Auffassung bezüglich des Begriffs der Situation. So sieht PRINZ [1983] nach einer Untersuchung der Beziehung zwischen Wahrnehmung und Tätigkeitensteuerung eine zweistufige Reizverarbeitung. Die erste Stufe beinhaltet die sogenannten „obligatorischen Grundprozesse“, die vom Reiz her zwangsläufig angestoßen werden und zunächst eine primäre Repräsentation der Reizinformation erstellen. In der zweiten Stufe wird ein Teil dieses Informationsangebots ausgewählt und gezielt weiterverarbeitet. Welcher Teil dies ist, hängt entscheidend vom „situativen Gesamtzusammenhang“ ab, in dem sich der Beobachter gerade befindet.

PRINZ [1983] unterteilt den situativen Gesamtzusammenhang in zwei Komponenten: Die erste Komponente ist der Tätigkeitszusammenhang, d. h. das hierarchische Gefüge der gerade ablaufenden Handlungen. Die selektive Wirkung des Tätigkeitszusammenhangs besteht in seiner Aussage, welche reizabhängigen Informationen zur Steuerung und Kontrolle der gerade ausgeführten Handlungen notwendig sind. Die zweite Komponente ist der Umgebungszusammenhang und beschreibt die Ereigniserwartung des Wahrnehmenden in einer bestimmten Situation. Die selektive Wirkung des Umgebungszusammenhangs besteht darin, dass Ergebnisse, die nicht mit dem aktuellen Umgebungsmodell des Beobachters zu vereinbaren sind, als überraschend gewertet werden und deshalb eine zeitweise Aussetzung oder einen Abbruch der gerade ausgeführten Handlungen nach sich ziehen.

Nach KRÜGER [1991] muss eine Situationsbeschreibung über eine Zustands-, eine Prädiktions- und eine Handlungskomponente verfügen. Die Handlungskomponente sorgt dafür, dass die in der Zustandskomponente außer Acht gelassenen Handlungen und Handlungsmöglichkeiten der Subjekte berücksichtigt werden. Die Prädiktionskomponente ist erforderlich, um die zeitliche Entwicklung einer Situation abzuschätzen. Das der Prädiktion zugrundeliegende Wissen kann in Wissen über physikalische Gesetzmäßigkeiten und Handlungswissen unterteilt werden. Das physikalische Wissen dient dazu, aus der Vielzahl von vorstellbaren Zustandsveränderungen diejenigen auszuschließen, die im Widerspruch zur Physik stehen. Das Handlungswissen schränkt die noch verbleibenden Möglichkeiten weiter ein und beinhaltet Konventionen oder Verhaltensregeln, wie sie beispielsweise in der Straßenverkehrsordnung festgehalten sind. Ferner gehört das Wissen über die Absichten der Subjekte und über die Konflikte zwischen den Absichten verschiedener Subjekte ebenfalls zum Handlungswissen.

MOCK-HECKER [1994] schlägt eine zweistufige Strategie zur Konflikterkennung zwischen den Intentionen verschiedener Verkehrsteilnehmer vor. In der ersten Stufe wird eine reine qualitative Konflikterkennung auf der Basis qualitativer, grob vereinfachender Situationsaspekte ohne Berücksichtigung der zeitlichen Zusammenhänge durchgeführt, welche die Mög-

lichkeit eines Konflikts abschätzt. Nach der ersten, qualitativen Situationsanalyse überprüft man die den Subjekten unterstellten Aktionen quantitativ nach Konflikten. Dazu sind unter Zuhilfenahme von mehr oder weniger komplexen Bewegungsmodellen die absoluten, quantitativen Orts-, Geschwindigkeits- und Zeitintervalle zu untersuchen. Die Bewegungsmodelle stellen Wissen über die Dynamik der Bewegungsabläufe bereit und ermöglichen es, aus der qualitativ extrapolierten Situation quantitative zeitliche und räumliche Fakten abzuleiten und 3D-Trajektorien hinreichend genau zu berechnen. Durch die vorgeschaltete qualitative Situationsanalyse ist die Anzahl der zu berechnenden, quantitativen Situationsaspekte jedoch beschränkt. Diese zweistufige Strategie kommt auch in dieser Arbeit zur Anwendung (vgl. Abschnitt 3.2.3.3).

Die Situationsanalyse arbeitet das implizit in den systemdynamischen Zuständen der Szenenobjekte vorhandene Wissen heraus, repräsentiert es explizit und macht es so einer Weiterverarbeitung durch die Verhaltensentscheidung zugänglich. Dazu beobachtet sie die zeitliche Entwicklung der systemdynamischen Größen über mehrere Sekunden hinweg. Können den Subjekten in der Szene hypothetische, alternative Aktionen unterstellt werden, so ist eine Abschätzung der zukünftigen Entwicklung der Situation und der zukünftigen, absehbaren Gefahren und Behinderungen möglich.

In dieser Arbeit sind die Resultate der Situationsanalyse situationsbeschreibende Aussagen, welche die aktuelle Situation umschreiben und durch sogenannte Situationsaspekte generiert werden. Prinzipiell ist es möglich, die Situationsanalyse beliebig detailliert zu gestalten, was die Anzahl und den Umfang der dabei durchzuführenden Berechnungen erhöht. Um nur notwendige Berechnungen durchzuführen, ist eine intelligente Steuerung der Situationsanalyse notwendig. Hier stößt man auf ein „Henne-Ei-Problem“: Für eine ausreichende aber handhabbare Situationsrepräsentation sollten die der Verhaltensentscheidung zur Verfügung stehenden, alternativen Fähigkeiten berücksichtigt werden. Damit eine Verhaltensentscheidung jedoch eine Vorauswahl von situationsgerechten Fähigkeiten treffen kann, benötigt sie eine Situationsbeschreibung als Entscheidungsgrundlage. Dieser Sachverhalt führt auf das in Abschnitt 3.2.4 beschriebene iterative Vorgehen bei der Situationsanalyse.

Durch eine enge Verzahnung von Situationsanalyse und Verhaltensevaluierung und durch die Möglichkeit zur Prädiktion der Situationsbeschreibung ist das System immer auf relevante Situationsveränderungen vorbereitet. So können die Sensoren und die stets beschränkte Verarbeitungskapazität zielgerichtet dort eingesetzt werden, wo sie für die Gesamtaufgabe von größtem Nutzen sind. Ferner kann die Verhaltensentscheidung schon vor den entsprechenden Situationsveränderungen Aktionen vorbereiten.

Der Rest dieses Kapitels 3.2 beschäftigt sich mit folgenden Themen: Abschnitt 3.2.2 beschreibt den Begriff des Situationsaspekts als linguistische Variable. Abschnitt 3.2.3 diskutiert die Objektbezogenheit eines Situationsaspekts und Abschnitt 3.2.4 illustriert die iterative Ausführung der Situationsanalyse anhand eines Beispiels. Auf die Realisierung des Konzepts zur Situationsanalyse wird in Abschnitt 3.2.5 und Anhang A eingegangen.

### **3.2.2 Der Situationsaspekt als linguistische Variable**

Um Situationsmodelle aufbauen zu können, müssen den Gegenständen und Sachverhalten des zu modellierenden Weltausschnitts direkt Datenobjekte als rechnerinterne Stellvertreter gegenübergestellt werden [KRÜGER, 1991]. Als Vorbild dient dabei der Mensch, der Begriffe verwendet, um ähnliche Gegenstände und Sachverhalte zusammenzufassen und von anderen zu unterscheiden. Bei der Modellierung der Situation stellt sich die Frage, welche Begriffe auf

welche Weise zu verwenden sind. So können Begriffe herangezogen werden, die physikalischen Objekten Bedeutungen zuordnen oder Beziehungen zwischen diesen ausdrücken.

Nach SCHAAF [1998] können dieselben Daten und Sachverhalte in unterschiedlichen Situationen verschiedene Bedeutungen besitzen und unterschiedliche Handlungen nach sich ziehen. Dieses Phänomen wird vom Autor als „Betrachtung verschiedener Aspekte von Dingen“ gedeutet. So teilt SCHAAF [1998] das Wissen über die Situation und die Funktionalität zum Erwerb dieses Wissens auf sogenannte Situationsaspekte auf. Mit Hilfe dieser Situationsaspekte führt er eine kontextabhängige Betrachtung von Teilsituationen durch. Die Gesamtheit der berechneten Situationsaspekte umschreiben die reale Situation. Dieses Konzept zur Situationsanalyse durch eine Menge von Situationsaspekten greift auch die vorliegende Arbeit auf. Die Situationsaspekte sind dabei linguistische Variablen, die einzelnen Objekten und Subjekten oder Gruppen von Objekten Bedeutungen, symbolische Aussagen oder Beziehungen zuordnen.

In der Literatur werden auf dem Gebiet der Situationsbeurteilung verschiedene Begriffe eingeführt und verwendet, z. B. (symbolisches) Merkmal, Eigenschaft, Attribut oder Aspekt. Dabei variiert die Granularität und Komplexität des damit assoziierten Wissens von Autor zu Autor. Der hier verwendete Begriff des Situationsaspekts ist mit diesen Begriffen verwandt.

Ein Situationsaspekt ist ein kontextabhängig mal mehr, mal weniger wichtiger Teil der Gesamtsituation und beinhaltet die Funktionalität, um aus den im Szenenbaum vorliegenden Daten eine situationsbeschreibende Aussage zu generieren. Das Ergebnis dieser Aussage wird in Form von Zugehörigkeitsgraden zu linguistischen Werten gespeichert. Bei der Generierung einer situationsbeschreibenden Aussage durch einen Situationsaspekt können verschiedene Arten von Information eine Rolle spielen, z. B.

- die Existenz von Objekten bestimmter Klassen,
- die systemdynamischen Zustandsgrößen und Eigenschaften der Objekte,
- die Existenz und die Zustände von Objekten, über die bereits ein anderer Situationsaspekt eine situationsbeschreibende Aussage formuliert hat oder
- die vermuteten oder wahrgenommenen Aktionen der Subjekte in der Szene.

Nach LUTH & WENDT [1998] besteht eine linguistische Variable aus folgenden Komponenten:

1. *den Variablennamen  $V_L$ ,*
2. *einer Menge von Regeln  $G_L$ , mit deren Hilfe die linguistischen Werte festgelegt werden,*
3. *die Menge  $\sigma_i$  der linguistischen Werte,*
4. *die Grundmenge  $X$  mit den zugehörigen Basisvariablen  $x$ ,*
5. *die Zugehörigkeitsfunktionen  $\mu_i$ , die den linguistischen Werten  $\sigma_i$  unscharfe Mengen aus der Basismenge  $X$  zuweisen.*

Abb. 3.2 zeigt die Komponenten eines Situationsaspekts. Wie im Folgenden klar wird, besitzt der in dieser Arbeit verwendete Begriff des Situationsaspekts die Eigenschaften einer linguistischen Variablen.

### 3.2.2.1 Name

Jeder Situationsaspekt besitzt einen eindeutigen Namen. Dieser Name entspricht dem Variablennamen  $V_L$  in der obigen Definition einer linguistischen Variablen. Der Name kann so gewählt werden, dass der menschliche Betrachter damit eine situationsbeschreibende Aussage assoziieren kann.

### 3.2.2.2 Liste von Objektklassen für Zuordnung

Ein Situationsaspekt macht eine Aussage bezüglich eines Objekts oder ordnet dem Objekt eine bestimmte Bedeutung im Situationskontext zu. Über welche Klassen von Objekten ein Situationsaspekt eine Aussage machen kann, wird in einer Liste innerhalb des Situationsaspekts spezifiziert. Das Objekt, dem die Aussage zugeordnet wird, heißt *Zuordnungsobjekt*.

### 3.2.2.3 Liste von Objektklassen für Referenzen

Setzt ein Situationsaspekt zwei Objekte in Beziehung zueinander, so muss man neben dem Objekt, dem die Aussage zugeordnet wird (*Zuordnungsobjekt*), auch das Objekt referenzieren, bezüglich dem diese Aussage gilt (*Referenzobjekt*). Welche Klassen von Objekten als Referenzen zulässig sind, spezifiziert eine Liste im Situationsaspekt.

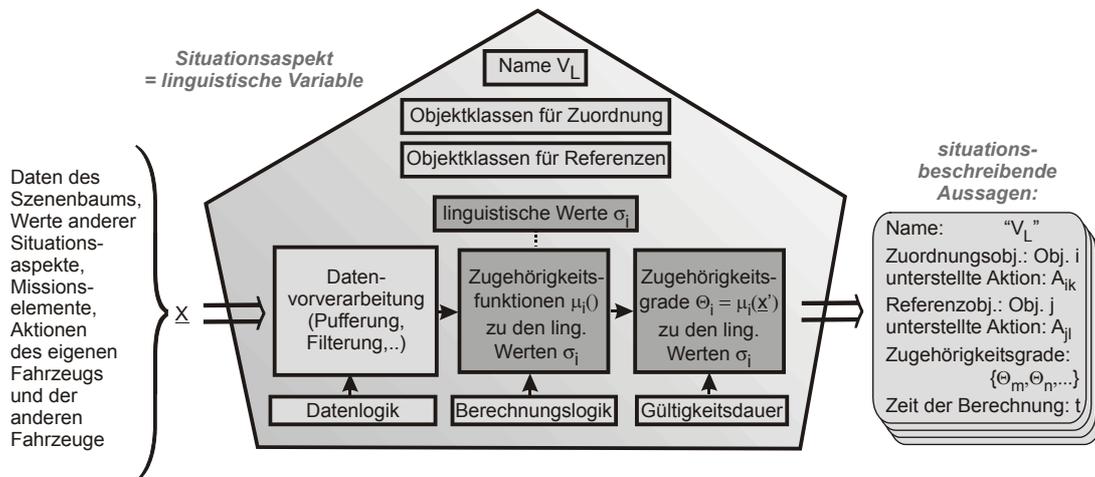


Abb. 3.2: Die Komponenten eines Situationsaspekts

### 3.2.2.4 Datenvorverarbeitung

Obwohl die Auswertung eines Situationsaspekts nicht im Systemtakt ablaufen muss, kann es notwendig sein, zyklisch ermittelte Objektdaten zu puffern oder zu filtern. Eine Untermenge dieser Daten stellen die Basisvariablen  $\underline{x}$  (mit ihrer Grundmenge  $\underline{X}$ ) in der obigen Definition einer linguistischen Variablen dar. Die Funktionalität zur Datenvorverarbeitung ist spezifisch für den Situationsaspekt. Die Datenvorverarbeitung kann auch die scharfen oder unscharfen Ergebnisse anderer Situationsaspekte verarbeiten.

### 3.2.2.5 Datenlogik

Auch die Datenvorverarbeitung muss nicht grundsätzlich zyklisch ablaufen. Die Datenlogik gibt Auskunft darüber, ob die Datenvorverarbeitung im aktuellen Zyklus aktiv werden muss oder nicht.

### 3.2.2.6 Linguistische Werte

Die linguistischen Werte  $\sigma_i$  eines Situationsaspekts benennen die alternativen Ergebnisse der situationsbeschreibenden Aussage. Die Beurteilung eines Situationsaspekts durch linguistische Werte bietet folgende Vorteile:

- Es ist nicht nur die boolesche Aussage darüber möglich, ob der Situationsaspekt zutrifft oder nicht, sondern auch eine Aussage darüber, in welchem Grad und in welcher Weise er zutrifft.

- Durch das Wissen, wie gut und in welcher Weise Situationsaspekte zutreffen, sind kontinuierliche Situationsveränderungen besser zu verfolgen und zu differenzieren. Für den menschlichen Betrachter werden kontinuierliche Situationsveränderungen damit besser nachvollziehbar.
- Neben dem Namen des Situationsaspekts besitzen auch dessen Aussagen anschauliche, symbolische Benennungen.

Welche linguistischen Werte ein Situationsaspekt besitzt, ist innerhalb des Situationsaspekts fest definiert und die Anzahl der Werte ist konstant. Da nicht für alle situationsbeschreibenden Aussagen eine linguistische Beschreibung notwendig bzw. sinnvoll ist, können Situationsaspekte auch über kontinuierliche Werte verfügen. Dies ist z. B. bei Nummerierungen, Zählungen oder beim Vergleich kontinuierlicher Größen der Fall.

### 3.2.2.7 Zugehörigkeitsfunktionen

Die Zugehörigkeitsfunktionen  $\mu_i()$  eines Situationsaspekts ordnen den linguistischen Werten  $\sigma_i$  unscharfe Mengen aus der Grundmenge  $\underline{X}$  zu. Bei der Ermittlung der Zugehörigkeitsgrade zu den linguistischen Werten gehen die Daten  $\underline{x}$  der Datenvorverarbeitung ein. Da die Datenvorverarbeitung sowohl die scharfen Werte der Zustandsgrößen als auch auf die unscharfen Werte von Situationsaspekten verarbeiten kann, stellen die Zugehörigkeitsfunktionen unscharfe Relationen mit scharfen und/oder unscharfen Mengen im Sinne der Fuzzy-Theorie dar. Bei unscharfen Relationen mit scharfen Mengen spricht man auch von Fuzzifizierung [LUTH & WENDT, 1998].

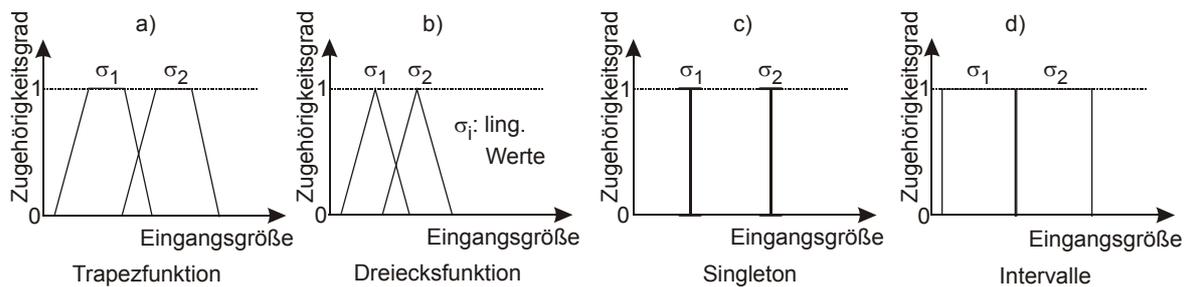


Abb. 3.3: Mögliche Zugehörigkeitsfunktionen  $\mu_i$  von linguistischen Werten  $\sigma_i$

In den Anwendungen der Fuzzy-Theorie werden am häufigsten Trapezfunktionen und Dreiecksfunktionen als Zugehörigkeitsfunktionen benutzt (Abb. 3.3 a+b). Neben diesen existieren noch ‚entartete‘ Zugehörigkeitsfunktionen wie Singletons (Abb. 3.3 c) und Intervalle (Abb. 3.3 d). Je nach Art des Situationsaspekts kann eine der oben abgebildeten Zugehörigkeitsfunktionen zweckmäßig sein. Auch Kombinationen verschiedener Typen von Zugehörigkeitsfunktionen sind möglich. Einführungen in die Algebra mit scharfen und unscharfen Mengen finden sich z. B. in [BANDER & GOTTWALD, 1993] und [KRUSE et al., 1993].

### 3.2.2.8 Gültigkeitsdauer

Da sich die systemdynamischen Zustandsgrößen der Objekte im Szenenbaum zeitlich ändern, sind die in einem Situationsaspekt ermittelten Ergebnisse nur für eine bestimmte Zeitperiode gültig. Diese Zeitperiode ist spezifisch für den Situationsaspekt und wird Gültigkeitsdauer genannt. Nach Ablauf der Gültigkeitsdauer sind die linguistischen Werte ungültig und müssen bei Bedarf neu berechnet werden. Ferner kann man die linguistischen Werte aufgrund singulärer Ereignisse von außen ungültig setzen.

### 3.2.2.9 Berechnungslogik

Die Berechnungslogik beantwortet die Frage, ob die linguistischen Werte im aktuellen Systemzyklus neu zu ermitteln sind oder nicht. Dabei geht die Gültigkeitsdauer des Situationsaspekts ein. Ferner kann die Berechnungslogik von der Verhaltensentscheidung beeinflusst werden. Dies ist der Fall, wenn die Verhaltensentscheidung den Situationsaspekt für die Verifikation der im letzten Iterationsschritt ausgewählten Fähigkeiten benötigt (vgl. Abschnitt 3.1).

#### 3.2.2.10 Zugehörigkeitsgrade

Durch die Anwendung der Zugehörigkeitsfunktionen  $\mu_i()$  auf die von der Datenvorverarbeitung gelieferten Daten  $\underline{x}$  liegt für jeden linguistischen Wert  $\sigma_i$  ein Zugehörigkeitsgrad  $\Theta_i = \mu_i(\underline{x})$  vor. Der Wertebereich von  $\Theta_i$  ist  $[0,1]$  und die Summe der Zugehörigkeitsgrade aller linguistischen Werte eines Situationsaspekts ist 1. Die Zugehörigkeitsgrade beschreiben, in welchem Maße die einzelnen linguistischen Aussagen wahr sind, und stellen die eigentlichen situationsbeschreibenden Aussagen dar.

Nach der Berechnung eines Situationsaspekts wird das Ergebnis für die Dauer der Gültigkeit in der Situationsrepräsentation niedergelegt und so der Verhaltensentscheidung zur Verfügung gestellt (vgl. Abb. 3.2, rechts). Neben dem Namen, der Gültigkeitsdauer und den Zugehörigkeitsgraden zu den linguistischen Werten werden auch Referenzen auf die Zuordnungs- und Referenzobjekte zusammen mit den zugrundegelegten Aktionen abgespeichert (vgl. Abschnitt 3.2.4)

### 3.2.3 Die Objektbezogenheit von Situationsaspekten

Die situationsbeschreibenden Aussagen eines Situationsaspekts können von unterschiedlich vielen Objekten in der Szene abhängen. Die Objektbezogenheit eines Situationsaspekts gibt an, auf wie viele Objekte bei der Berechnung des Situationsaspekts referenziert werden muss. Zur Illustration der Objektbezogenheit beschreiben die folgenden Unterabschnitte eine Autobahnsituation mit Situationsaspekten der Objektbezogenheit 0., 1. und 2. Grades.

#### 3.2.3.1 Objektbezogenheit 0. Grades

Bei Situationsaspekten mit einer Objektbezogenheit 0. Grades kann die damit verbundene situationsbeschreibende Aussage keinem Objekt in der Szene zugeordnet werden. Diese Aussagen beziehen sich nicht auf physikalische Objekte, sondern beschreiben abstrakte Sachverhalte. Solche abstrakten Sachverhalte sind z. B. aktuelle Wetterbedingungen (Nebel, Regen, Schnee), globale und lokale Verkehrsregeln (Geschwindigkeitsbegrenzung, Überholverbot) oder von außen (z. B. über Verkehrsfunk) empfangene Informationen (Warnungen, Staumeldungen).

#### 3.2.3.2 Objektbezogenheit 1. Grades

Ein Situationsaspekt mit der Objektbezogenheit 1. Grades macht eine situationsbeschreibende Aussage für ein Objekt, ohne dass diese Aussage von einem weiteren Objekt abhängt. Diese Aussagen beschreiben also keine Beziehungen zwischen Objekten. Als Beispiel soll die in Abb. 3.4 a gezeigte Autobahnszene durch zwei Situationsaspekte mit einer Objektbezogenheit 1. Grades untersucht werden.

Der erste Situationsaspekt besitzt den Namen *Spurnummer* und nummeriert alle Objekte der Klasse *Spur* nach ihrer Position auf dem gehärteten Bereich von rechts nach links. Der zweite Situationsaspekt hat den Namen „*Fährt auf der Spur*“ und kann für jedes Objekt der

Klasse Fahrzeug die Spurzugehörigkeit ermitteln. Dabei geht die Nummerierung der Fahrspuren mit ein. Der Situationsaspekt „Fährt auf der Spur“ ist auf alle Objekte des Typs Fahrzeug anwendbar, also sowohl auf Fremdfahrzeuge als auch auf das Eigenfahrzeug.

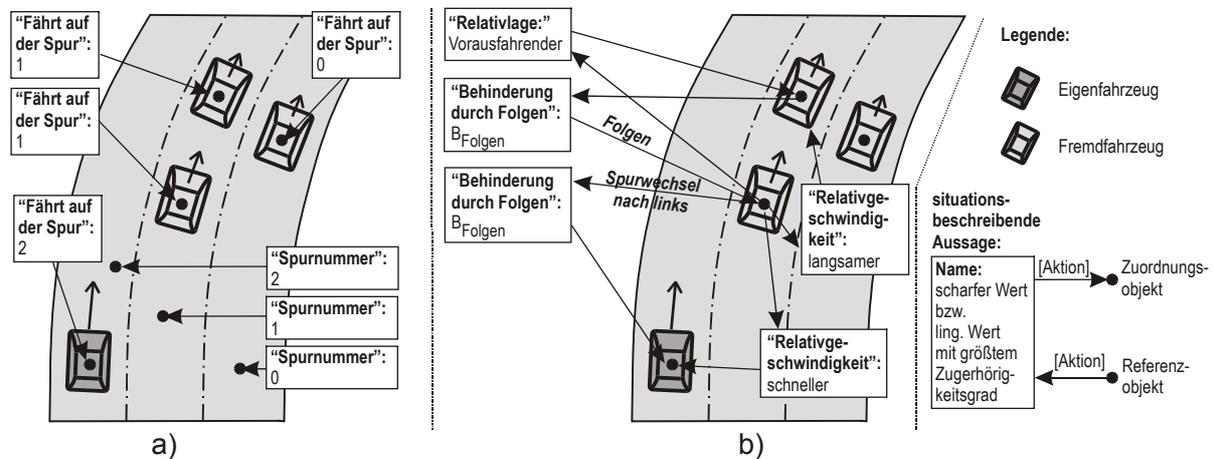


Abb. 3.4: Situationsaspekte mit einer Objektbezogenheit 1. Grades (a) und 2. Grades (b)

### 3.2.3.3 Objektbezogenheit 2. Grades

Bei der Berechnung eines Situationsaspekts mit der Objektbezogenheit 2. Grades muss neben dem Zuordnungsobjekt ein Referenzobjekt angegeben werden (vgl. Abschnitte 3.2.2.2 und 3.2.2.3). Situationsaspekte mit einer Objektbezogenheit 2. Grades beschreiben Beziehungen zwischen zwei Objekten. Die Autobahnszene von Abb. 3.4 b wird nun mit Situationsaspekten diesen Typs untersucht.

Die Situationsaspekte *Relativlage* und *Relativegeschwindigkeit* beschreiben in linguistischer Form die Relativlage bzw. Relativegeschwindigkeit zwischen zwei Fahrzeugen. Bei beiden Situationsaspekten wird die Aussage einem Objekt der Klasse Fahrzeug zugeordnet. Zusätzlich ist ein Referenzobjekt der Klasse Fahrzeug anzugeben.

In der Syntax von Abschnitt 3.2.2 stellen sich die linguistischen Werte des Situationsaspekts *Relativegeschwindigkeit* wie folgt dar (vgl. Abb. 3.2):

$$\mathcal{O}_{\text{Relativegeschwindigkeit}} = \{ \text{viel\_schneller}, \text{schneller}, \text{gleich\_schnell}, \text{langsamer}, \dots \}$$

Die Menge der Zuordnungsfunktionen dieses Situationsaspekts sind:

$$\underline{\mu}_{\text{Relativegeschwindigkeit}} = \{ \mu_{\text{viel\_schneller}}(), \mu_{\text{schneller}}(), \mu_{\text{gleich\_schnell}}(), \dots \}$$

Werden diese Zuordnungsfunktionen auf *Fahrzeug*<sub>1</sub> angewandt, wobei *Fahrzeug*<sub>2</sub> das Referenzobjekt ist, so ergeben sich folgende situationsbeschreibende Aussagen:

$$\mathcal{O}_{\text{Relativegeschwindigkeit}}(\text{Fahrzeug}_1, \text{Fahrzeug}_2) = \{ (\text{Fahrzeug}_1, \text{Fahrzeug}_2, \mu_{\text{viel\_schneller}}(\underline{x})), (\text{Fahrzeug}_1, \text{Fahrzeug}_2, \mu_{\text{schneller}}(\underline{x})), (\text{Fahrzeug}_1, \text{Fahrzeug}_2, \mu_{\text{gleich\_schnell}}(\underline{x})), \dots \}$$

Der Zustandsvektor  $\underline{x}$  beinhaltet diejenigen Größen, die von den Zuordnungsfunktionen ausgewertet werden. Der Ausdruck

$$\mathcal{O}_{\text{Relativegeschwindigkeit}}(\text{Fahrzeug}_1, \text{Fahrzeug}_2, \mu_{\text{schneller}}(\underline{x}))$$

besagt, dass die situationsbeschreibende Aussage

„Das Objekt Fahrzeug<sub>1</sub> hat bezüglich des Situationsaspekts Relativgeschwindigkeit relativ zu Objekt Fahrzeug<sub>2</sub> die Eigenschaft schneller“

zum Grade  $\mu_{\text{schneller}}(\underline{x})$  wahr ist, wobei  $0 \leq \mu_{\text{schneller}}(\underline{x}) \leq 1$  gilt.

Die Situationsaspekte *Relativgeschwindigkeit* und *Relativlage* führen zu einer groben Einteilung der Fahrzeuge, so dass z. B. eine daran anschließende Kollisions- oder Behinderungsuntersuchung einen direkten Zugriff auf die in Frage kommenden Fahrzeuge hat.

Der Situationsaspekt „*Behinderung durch Folgen*“ berechnet ein Maß für die Behinderung, die einem Fahrzeug durch ein in der selben Spur vorausfahrendes oder einscherendes, langsames Fahrzeug aufgezwungen wird. Für das damit erzwungene Abbremsen und anschließende Folgen wurde von KOPF [1993] das folgende Behinderungsmaß vorgeschlagen:

$$B_{\text{Folgen}} = \frac{(v_L - v_R)^2}{2(\Delta x - t_S v_R)} \quad (\text{Gl. 3.1})$$

dabei gilt:

- $v_R$  = Geschwindigkeit des behindernden Fahrzeugs (als konstant angenommen) [m/s]
- $v_L$  = Geschwindigkeit des nachfolgenden Fahrzeugs vor der Bremsung ( $v_L > v_R$ ) [m/s]
- $t_S$  = Sicherheitszeitabstand zwischen den Fahrzeugen nach der Bremsung des hinteren Fahrzeugs [s]
- $\Delta x$  = Abstand zwischen den Fahrzeugen vor der Bremsung ( $\Delta x > t_S \cdot v_R$ ) [m]

Bei diesem Modell wird von der Vorstellung ausgegangen, dass die als konstant angenommene Verzögerung  $B_{\text{Folgen}}$ , die das nachfolgende Fahrzeug eine gewisse Zeit aufbringen muss, um danach im sicheren Zeitabstand  $t_S$  dem Hindernisfahrzeug mit gleicher Geschwindigkeit folgen zu können, ein Maß für die Behinderung ist.

Beim Situationsaspekt „*Behinderung durch Folgen*“ ist neben dem behinderten Fahrzeug auch das behindernde Fahrzeug zu referenzieren. So ist es z. B. möglich, dass ein Fahrzeug durch zwei Fahrzeuge behindert wird und deshalb zwei Werte  $B_{\text{Folgen}}$  zugewiesen bekommt.

Neben dem Behinderungsmodell wurde von KOPF [1993] auch ein Gefährdungsmodell aufgestellt. Dieses Gefährdungsmodell liefert ein Maß für die Gefährlichkeit einer Situation welches von der zeitlich-räumlichen Konstellation der Gefahresträger und der antizipierten gefährdungsvermindernden Reaktion abhängt. Die zugrundeliegende Gefährdungsfunktion ist definiert als

$$G_i = \frac{1}{T_{\text{res}}} \quad (\text{Gl. 3.2})$$

wobei die Zeitreserve  $T_{\text{res}}$  die Zeit ist, die noch verstreichen darf, bis die antizipierte gefährdungsvermindernde Aktion  $i$  einsetzen muss, damit das zeitlich-räumliche Zusammentreffen der Gefahresträger gerade noch verhindert werden kann.  $G_i$  ist ein Maß für den Zeitdruck, unter dem das Subjekt eine gefährdungsvermindernde Aktion auswählen und ausführen muss.

In einer Gefahrensituation sind mehrere gefährdungsvermindernde Aktionen denkbar. Für jede von ihnen lässt sich eine Gefährdungsfunktion  $G_i$  aufstellen. Ist  $G_i$  singular, so heißt dies, dass die Situation mit der Aktionsalternative  $i$  nicht mehr beherrscht werden kann. KOPF [1993] definiert Gefährdungsfunktionen für die drei gefährdungsvermindernde Aktionen *Bremsen*, *Ausweichen* und *Spurfolgen*. Bei den ersten beiden Aktionen ist die Zeitreserve  $T_{\text{res}}$  gleich dem Zeitintervall bis zu dem Punkt, an dem (durch Ausnutzung der maximal mögli-

chen Verzögerung bzw. der maximal möglichen Lenkrate) ein Zusammenstoß gerade noch verhindert werden kann. Bei der dritten Aktion ist es die Zeit bis zu dem Punkt, an dem eine Berührung der Fahrbahnbegrenzung (durch Ausnutzung der maximal möglichen Lenkrate) gerade noch zu vermeiden ist.

Durch die Modelle für die Behinderung und Gefährdung wird eine implizite Prädiktion in die Zukunft durchgeführt, d. h. die gegenwärtige Situation wird unter Annahme einer Aktion prädiert, so dass nicht nur die gegenwärtige Situation sondern auch mögliche, zukünftige Situationen bei der Verhaltensentscheidung Berücksichtigung finden.

In analoger Weise sind auch Situationsaspekte mit einer Objektbezogenheit höheren Grades denkbar. Sie beschreiben Beziehungen zwischen mehr als zwei Objekten. Je größer der Grad der Objektbezogenheit eines Situationsaspekts ist, desto mehr Permutationen gibt es zwischen den aktuell vorhandenen Szenenobjekten, desto öfter kann der Situationsaspekt auf ein Objekt angewandt werden und umso wichtiger ist seine selektive Anwendung. Es verbleibt zu vermerken, dass die hier dargestellte Objektbezogenheit von Situationsaspekten und die z. B. in [KRÜGER, 1991] beschriebene Darstellung situativer Sachverhalte durch Begriffsgraphen Analogien aufweisen.

### 3.2.4 Beispiel für die rekursive Ausführung der Situationsanalyse

In der Literatur wird oft betont (z. B. in [KOPF, 1993]), dass es sich bei der Intentionbildung um einen iterativen Prozess handelt, bei dem mehrere alternative Intentionen in Erwägung gezogen, geprüft und ggf. wieder verworfen werden, ehe dann eine Intention zur Ausführung gelangt. Auch KRÜGER [1991] stellt klar heraus, dass ein retrospektives Verarbeitungsprinzip<sup>3</sup> für eine Situationsmodellierung und Situationsanalyse unbrauchbar ist. Vielmehr muss die Situationsmodellierung schritthaltend mit dem Aufkommen der Eingangsdaten ausgeprägt werden.

Die situationsspezifische, selektive Datenverarbeitung und Datenbeschaffung ist von zentraler Bedeutung. So muss die Verhaltenssteuerung auf die Situationsanalyse zurückwirken können, um gezielt benötigte Information zu beschaffen [KRÜGER, 1991]. Für ein handelndes System bedeutet diese Art der Selektion nichts anderes, als dass zu bestimmten Verarbeitungszeitpunkten die in naher Zukunft möglichen Situationen vorhergesagt werden können.

Die hier beschriebene Situationsanalyse verfügt über eine Menge von Situationsaspekten, mit denen sie schrittweise eine Situationsrepräsentation aufbauen kann. Dabei steigt mit dem Grad der Objektbezogenheit der Situationsaspekte die Anzahl der möglichen Anwendungen auf die physikalischen Szenenobjekte. Es ist für die Verhaltensentscheidung nicht notwendig und in Echtzeit auch nicht durchführbar, Beziehungen zwischen allen möglichen Permutationen von Objekten zu ermitteln. Es stellt sich also die Frage, welche Situationsaspekte zu welchen Zeitpunkten berechnet werden. Hier ist zu berücksichtigen, dass sich die Relevanz der Situationsaspekte für die Verhaltensentscheidung dynamisch ändert. Die Abarbeitung eines zur Laufzeit unveränderbaren Skripts ist aus diesem Grund nicht sinnvoll. Durch die in diesem Kapitel vorgeschlagene iterative Durchführung von Situationsanalyse und Verhaltensentscheidung werden die Situationsaspekte nur auf diejenigen Objekte angewandt, welche wirklich relevant für die Verhaltensentscheidung sind. Das iterative Vorgehen wird nun anhand der Autobahnscene von Abb. 3.4 und Abb. 3.5 illustriert.

---

<sup>3</sup> Bei einem retrospektiven Verarbeitungsprinzip wird davon ausgegangen, dass vor Beginn der Situationsanalyse eine vollständige geometrische Szenenbeschreibung vorliegt.

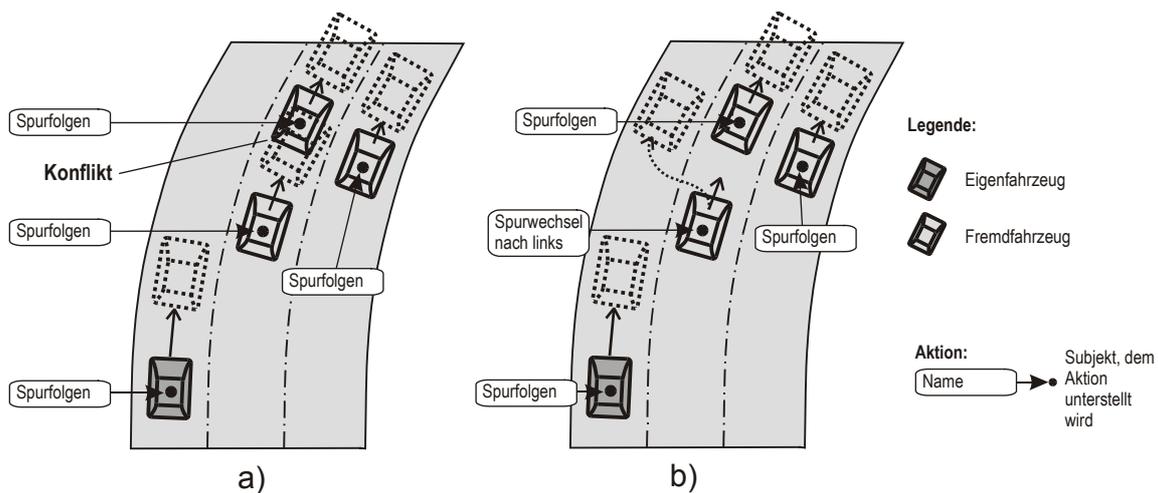


Abb. 3.5: Aktionen der Fahrzeuge nach der 1. Iteration (a) und nach der 2. Iteration (b)

Situationsaspekte mit einer Objektbezogenheit 0. und 1. Grades können für jedes Szenenobjekt nur einmal berechnet werden. Ihre Berechnung ist deshalb seitens des Rechenaufwandes eher unkritisch. So wendet man im ersten Schritt alle Situationsaspekte mit einer Objektbezogenheit 0. und 1. Grades auf die entsprechenden Szenenobjekte an. Nach diesem ersten Schritt sind die Spuren auf dem gehärteten Bereich nummeriert und von jedem Fahrzeug ist die Eigenspur bekannt (vgl. Abb. 3.4 a). Diese rudimentäre Situationsanalyse wird an die Verhaltensentscheidung gesandt.

Die Vorgänge in der Verhaltensentscheidung sind in Abschnitt 3.1 skizziert und in Kapitel 3.4 ausführlich beschrieben. Wichtig an dieser Stelle ist nur, dass die Verhaltensentscheidung auf eine mehr oder minder ausführliche Situationsbeschreibung mit einer situationsgerechten Auswahl von Fähigkeiten antwortet. Da in der Situationsbeschreibung von Abb. 3.4 nur sehr wenige Situationsaspekte ausgewertet sind, werden mögliche Konflikte noch nicht erkannt. Die Verhaltensentscheidung wählt deshalb die Wunschaktion eines jeden Fahrzeugs aus<sup>4</sup>: *Spurfolgen* (vgl. Abb. 3.5 a).

Innerhalb der Verhaltensentscheidung existiert für jede Fähigkeit ein internes Ablaufmodell (vgl. Abschnitt 3.1). Teil jedes Ablaufmodells sind die für die Fähigkeit relevanten Situationsaspekte und Angaben darüber, auf welche Objekte sie anzuwenden sind. Aufgabe dieser relevanten Situationsaspekte ist es, die Auswirkungen und Konflikte zu erkennen, welche sich aus der Ausführung der Fähigkeit ergeben würden. In Abb. 3.5 a ist ein Konfliktfall auf der 1. Spur ersichtlich. Bei der Aktion *Spurfolgen* sind u. a. die Situationsaspekte *Relativlage* und *Relativgeschwindigkeit* relevant. Der Situationsaspekt *Relativlage* nimmt die durch den Situationsaspekt „*Fährt auf der Spur*“ ermittelte Spurzugehörigkeit der Fahrzeuge und betrachtet die symbolische, relative Lage der Fahrzeuge zueinander. Nach der 2. Iteration in der Situationsanalyse ist Folgendes bekannt (vgl. Abb. 3.4 b): Es befinden sich zwei Fahrzeuge auf der 1. Spur, wobei das hintere Fahrzeug eine größere Geschwindigkeit besitzt, als das vorausfahrende. Die Fahrzeuge in den anderen Spuren haben kein vorausfahrendes Fahrzeug.

Auf diese erweiterte Situationsrepräsentation wird nun ein zweites mal die Regelbasis der Verhaltensentscheidung angewandt. Da nun zusätzliche Information über die Situation vorliegt, ist es möglich, dass sich eine andere Fähigkeitenauswahl ergibt. In der Beispielsituation von Abb. 3.5 b änderte sich die Aktion nur eines Fahrzeugs: Weil das in der 1. Spur vorausfahrende Fahrzeug langsamer fährt als das nachfahrende Fahrzeug, wird das letztere behindert. Das nachfahrende Fahrzeug besitzt zwei Handlungsalternativen: Entweder es bremst ab

<sup>4</sup> Aus Gründen der Übersichtlichkeit sei in diesem Beispiel das Rechtsfahrgebot nicht modelliert.

und folgt dem vorausfahrenden Fahrzeug oder es führt einen Spurwechsel nach links aus. Da die zweite Alternative der Intention des Fahrzeugs eher entgegenkommt, wird die Verhaltensentscheidung einen „*Spurwechsel nach links*“ vorschlagen.

Auch dieser Aktion „*Spurwechsel nach links*“ sind relevante Situationsaspekte zugeordnet und auch diese werden in einem weiteren, dritten Iterationsschritt von der Situationsanalyse evaluiert. So betrachtet man die Behinderung durch ein Folgen, die Fahrzeugen zugemutet wird, welche auf der Zielspur des Spurwechsels fahren (hier ist es die 2. Spur). Der Situationsaspekt „*Behinderung durch Folgen*“ ermittelt dieses Behinderungsmaß (vgl. Abschnitt 3.2.3), wobei dem Referenzfahrzeug die Aktion „*Spurwechsel nach links*“ unterstellt ist (vgl. Abb. 3.4 b). Der gleiche Situationsaspekt wird nochmals auf das nachfahrende Fahrzeug unter der Annahme der Aktion *Folgen* angewandt, wobei das Referenzfahrzeug das vorausfahrende Fahrzeug ist. Damit ist ein Maß für die Behinderung bekannt, die dem nachfahrenden Fahrzeug durch die Handlungsalternative *Folgen* zugemutet würde. Die Verhaltensentscheidung vergleicht nun diese beiden Behinderungsmaße und wählt eine der beiden Handlungsalternativen aus. So ist es z. B. denkbar, dass sie einen Spurwechsel vorschlägt, wenn die Behinderung des auf der 1. Spur nachfahrenden Fahrzeugs durch ein *Folgen* größer wäre als die Behinderung, die dem Fahrzeug auf der 2. Spur durch einen Spurwechsel zugemutet wird.

Schlägt die Verhaltensentscheidung weiterhin einen Spurwechsel vor, so hat sich im letzten Iterationsschritt keine Aktion eines Subjekts geändert und die Iteration bricht ab. Das Ergebnis der Iteration ist eine im notwendigen Maß ausgearbeitete Situationsrepräsentation, und jedem Subjekt in der Szene – hier waren es ausschließlich PKWs – ist eine Aktion zugeordnet. Dass es sich bei einem der Fahrzeuge um das autonome Eigenfahrzeug handelt, ist für die Situationsanalyse und Verhaltensentscheidung zunächst transparent.

### 3.2.5 Realisierung der Situationsanalyse

Wie Abschnitt 3.2.4 illustriert, erfolgt die Situationsanalyse iterativ zusammen mit der Verhaltensentscheidung. Aus diesem Grund kann die prozedurale Ausführung der Situationsanalyse erst nach der Beschreibung der internen Ablaufmodelle der Fähigkeiten (Abschnitt 3.3.6) und der Regelbasis (Abschnitt 3.4) in Abschnitt 3.5.2 erfolgen.

Anhang A stellt die Basisklassen der Situationsanalyse vor: *CSituationAspect*, *CMembershipMatrixEntry*, *CValency*, *CMembershipFunction*, *CDataProcessing*, *CSubject* und *CEgo*. Insbesondere beschreibt Anhang A.7 den Prozessfluss, welcher durchlaufen wird, wenn ein (neues) Objekt oder Subjekt im Szenenbaum erscheint oder verschwindet

Aufbauend auf den eben genannten Basisklassen wurden vom Autor verschiedene Datenvorverarbeitungsmodule und Situationsaspekte implementiert. Zwei Datenvorverarbeitungsmodule filtern z. B. die Positions- und Bewegungsdaten der Fahrzeuge und Hindernisse. Realisierte Situationsaspekte beleuchten die Relativlagen von Hindernissen zum Fahrzeug und Fahrschlauch oder beschreiben die Geschwindigkeit des Fahrzeugs mit unscharfen Werten. Beim Stop&Go-Fahren weist ein Situationsaspekt einem Fremdfahrzeug die Bedeutung „*relevantes Objekt für die Längsführung*“ zu (vgl. [SIEDERSBERGER, 2003]). Bei diesem Situationsaspekt wird u. a. die zeitliche Entwicklung der Fahrtrajektorie von in den Nebenspuren fahrenden Fahrzeugen beobachtet, um auf einscherende Fahrzeuge frühestmöglich zu reagieren. Näheres über die realisierten Situationsaspekte findet sich im experimentellen Teil C dieser Arbeit.

Prinzipiell können Situationsaspekte im verteilten System von mehreren Prozessen berechnet werden. Dies bedingt jedoch eine Kommunikation der Ergebnisse dieser Analysen zur

zentralen Verhaltensentscheidung. Zum anderen können Prozesse im System an den Ergebnissen der (zentralen) Situationsanalyse interessiert sein. Ein Beispiel dafür ist die Fahrzeugführung, welche an allen Hindernissen mit der symbolischen Eigenschaft „*liegt im Fahr-schlauch*“ interessiert ist (vgl. Kapitel 6.1). Aus diesem Grund wurde vom Autor die dynamische Wissensrepräsentation DKR um einen weiteren Zweig – den Situationszweig – erweitert (vgl. Kapitel 5.3). Dieser Situationszweig übernimmt die Kommunikation situationsbeschreibender Aussagen zwischen Prozessen. Der Situationszweig bietet eine Schnittstelle an, um alle Objekte mit einer bestimmten symbolischen Bedeutung oder Eigenschaft im Szenenbaum zu finden. Zusätzlich können alle symbolischen Aussagen zu einem Szenenobjekt abgerufen werden.

## 3.3 Fähigkeiten im verteilten System

### 3.3.1 Grundlegendes zur Repräsentation von Fähigkeiten

Situationsgerechtes Handeln ist davon abhängig, welche Handlungsalternativen überhaupt zur Verfügung stehen. Deshalb sind bei der Verhaltensentscheidung neben der Situation auch die prinzipiell und aktuell vorhandenen Fähigkeiten des Systems zu berücksichtigen. Der 4D-Ansatz erlaubt auf sehr einfache Weise Fähigkeiten zu definieren und zu implementieren, da alle physikalischen Zustände und ihre zeitlichen Entwicklungen explizit repräsentiert sind [DICKMANN, 1992, 1994]. Es ist die Aufgabe einer zentralen Verhaltensentscheidung, diese Fähigkeiten in einer geeigneten Reihenfolge zu nutzen, um ein Missionsziel zu erreichen.

Werden von den einzelnen Prozessen im System Fähigkeiten mit darin gekapselten Vorsteuerungen und unterlagerten Reglern angeboten, so genügt es der zentralen Verhaltensentscheidung, in deutlich längeren Zeittakten die Verwendung der Fähigkeiten anzustoßen, den Fortschritt der Aktionen zu überwachen und diese gegebenenfalls bei signifikanten Situationsveränderungen anzupassen und zu beenden. Die Reaktionszeiten können dabei einige hundert Millisekunden betragen.

Andererseits ist es für den Experten einer Fähigkeit ausreichend zu wissen, ob und wie seine Fähigkeit benutzt werden soll. Dieses (quasi-statische) Wissen wird dem Experten der Fähigkeit durch die Verhaltensentscheidung mitgeteilt. Einmal gestartet, laufen die damit verbundenen Vorsteuerungen und Regelungen entkoppelt von den höheren Systemebenen ab, bis der Endzustand erreicht ist oder von der Verhaltensentscheidung neue Steuerbefehle eintreffen. Die Verhaltensentscheidung geht zunächst davon aus, dass die unterlagerten Regler es schaffen, trotz äußerer Störungen die Differenzen zwischen den Istverläufen und den Sollverläufen in den Zustandsgrößen klein zu halten.

Durch diese duale Repräsentation – systemdynamisches Wissen auf der Expertenebene und quasi-statisches Wissen auf Ebene der Verhaltensentscheidung – hat eine Fähigkeit sowohl eine quasi-statische Bedeutung, als auch eine exakte, systemdynamische Entsprechung. Damit lässt sich das „*symbol grounding problem*“ aus dem KI-Bereich lösen.

Fähigkeiten repräsentieren potentiell ausführbare Aktionen des Systems mit unterschiedlichen Abstraktionsgraden. Entsprechend ihres Abstraktionsgrades lassen sich die Fähigkeiten hierarchisch gliedern. In der Literatur zur Fahrzeugführung werden häufig folgende Hierarchieebenen unterschieden:

- Auf der sogenannten **Navigationsebene** oder **strategischen Ebene** werden Zielfindungs- und Planungsaufgaben verfolgt. Der menschliche Fahrer bestimmt auf dieser Ebene sein Reiseziel und plant die Fahrtroute, d. h. die sukzessive zu folgenden Straßen. Beim autonomen Fahrzeugen ist der Missionsplan – bestehend aus einer Folge von Missionselementen – mit der Intention des Eigenfahrzeugs auf dieser obersten Hierarchiestufe gleichzusetzen.
- Innerhalb der **Führungsebene** oder **taktischen Ebene** werden kurzfristige Intentionen und Aktionen verfolgt, die für die Realisierung einer strategischen Intention in der aktuellen Situation notwendig sind. Als Beispiele sind das Ausweichen vor einem Hindernis, das Verlassen einer Straße und die Ausführung von Spurwechseln und Überholmanövern zu nennen. Bei autonomen Fahrzeugen werden die taktischen Aktionen zwar von der Missionsplanung nicht explizit verlangt, sie sind aber für das Erreichen des Missionsziels und

für die Sicherheit des Eigenfahrzeugs zwingend erforderlich. Dabei kommen oft Vorsteuergesetze zum Einsatz, die den unterlagerten Reglern der operationalen Ebene Sollgrößenverläufe vorgeben.

- Die Aktivitäten auf der **operationalen Ebene** sind sehr kurzfristig geplant und dienen direkt der Fahrzeugführung. Sie benutzen einfache Regelgesetze um stabilisierend auf den Fahrzeugzustand einzuwirken. Beispiele sind Beschleunigen, Bremsen, Spurhalten oder Abstandhalten.

Aufbauend auf den Feststellungen in diesem Abschnitt wurde in der Arbeitsgruppe VERHALTEN [2000] das sogenannte Fähigkeitenkonzept entwickelt. Das nun folgende Unterkapitel 3.3.2 stellt dieses Fähigkeitenkonzept vor. Dabei werden für die Fähigkeiten und Aktionen der verschiedenen Hierarchiestufen spezielle Begriffe eingeführt. Abschnitt 3.3.3 beschreibt die Kapselung der Funktionalitäten des Systems in den Fähigkeiten. Abschnitt 3.3.4 stellt eine Reihe von Maßen vor, die den Fortschritt einer Aktion abstrakt beschreiben können. Anschließend legt Abschnitt 3.3.5 dar, wie die Fähigkeiten des Systems durch das sogenannte Fähigkeitsnetz repräsentiert werden. Abschnitt 3.3.6 beschreibt, wie sich die Verhaltensentscheidung mittels interner Ablaufmodelle ein symbolisches Bild vom Fortschritt der Aktionen macht, um so auf eventuelle Fehlentwicklungen angemessen reagieren zu können. Abschnitt 3.3.7 zeigt den zeitlichen Ablauf bei der Steuerung der kooperierenden Aktionen und Abschnitt 3.3.8 listet die bisher im EMS-VISION System realisierten Fähigkeiten auf. Das Klassendesign des Fähigkeitenkonzepts ist im Anhang B dargestellt.

### 3.3.2 Das Fähigkeitenkonzept<sup>5</sup>

Ein autonomes Fahrzeug besteht aus einer Vielzahl von Hardware-Modulen und Software-Prozessen. Jede dieser Komponenten bietet Fähigkeiten an, welche für die sichere Fortbewegung des autonomen Fahrzeugs notwendig sind und nach Abstraktion, Koordinationskomplexität und Wechselhäufigkeit geordnet werden können. Nach MAURER [2000] kann man die Fähigkeiten unterschiedlichen Kategorien zuordnen:

- Fähigkeiten zur Wahrnehmung und Szenenbeschreibung
- Fähigkeiten zur Verhaltensentscheidung
- Fähigkeiten zur Fortbewegung und Blickrichtung
- Fähigkeiten zur Planung

Abb. 3.6 zeigt einige Teilfunktionalitäten eines autonomen Fahrzeugs. Das Fahrzeug verfügt über Aktuatoren wie Gas, Bremse und Lenkung. Diese elektromechanischen Komponenten erledigen klar definierte Aufgaben bezüglich der Fortbewegung. Ist das autonome Fahrzeug mit einer aktiven Blickrichtungssteuerung und einer Zweiachsenplattform ausgerüstet, so besitzt es als weitere Aktuatoren die Motoren der Kameraplattform. Mit ihnen kann das autonome System die Ausrichtung der Sensoren aktiv beeinflussen.

Die Aktuatoren werden durch geeignete Steuerungen und Regelungen bedient. So verfügt das autonome Fahrzeug z. B. über Regelalgorithmen, welche die Beschleunigung, die Geschwindigkeit oder den Abstand zum vorausfahrenden Fahrzeug auf vorgegebene Sollwerte regeln. Die Querdynamik des Fahrzeugs wird ebenfalls von einer Regelung bestimmt, welche die Ablage zu einer Spur konstant hält [SIEDERSBERGER, 2003]. Eine Blickrichtungssteuerung, welche das Blickverhalten der Wirbeltiere imitiert, führt eine Sequenz von Sakkaden und Blickfixationen aus (vgl. Kapitel 4). Auch diese Fähigkeiten sind in Abb. 3.6 dargestellt.

---

<sup>5</sup> Der Inhalt dieses Abschnitts wurde innerhalb der Arbeitsgruppe VERHALTEN [2000] erarbeitet.

Fähigkeiten dieser Hierarchiestufe werden in der vorliegenden Arbeit **Fertigkeiten** genannt (vgl. Abb. 3.6, links). Instanzen dieser Fertigkeiten heißen **Handlungsprimitive**. Fertigkeiten und Handlungsprimitive besitzen im EMS-VISION System folgende charakteristische Eigenschaften:

- Sie realisieren relativ verfestigte und automatisierte Abläufe,
- sie sind schwerpunktmäßig der operationalen Ebene zuzuordnen (vgl. Abschnitt 3.3.1),
- in ihnen kommen steuerungs- und regelungstechnische Algorithmen zum Einsatz<sup>6</sup>,
- sie wirken stabilisierend auf den Fahrzeugzustand<sup>6</sup>,
- sie greifen direkt auf die Aktuatoren und/oder Sensoren des Fahrzeugs zu,
- sie sind eindeutig einer Kategorie zuzuordnen (vgl. oben) und
- sie werden i. d. R. nicht direkt von der zentralen Verhaltensentscheidung angesprochen.

Die Abhängigkeiten der Fertigkeiten von den Aktuatoren sind in Abb. 3.6 (unten) durch Pfeile dargestellt.

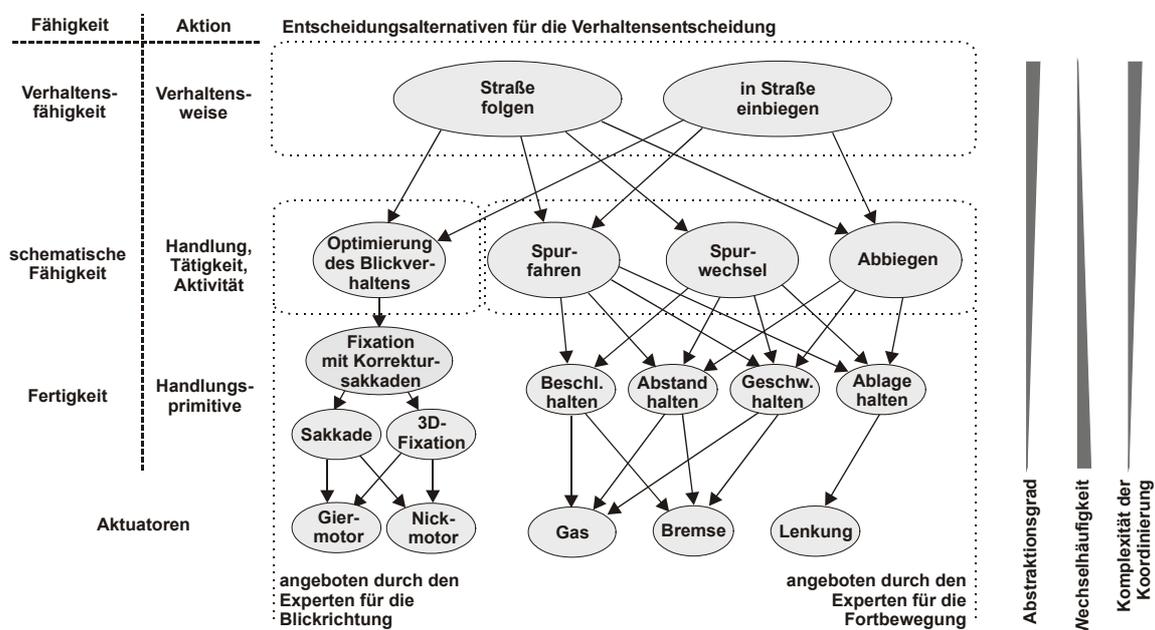


Abb. 3.6: Die Fähigkeiten von Systemkomponenten auf unterschiedlichen Ebenen

Die niederfrequent ablaufende Verhaltensentscheidung soll sich nicht um die Auswahl eines Reglers oder um das Starten und Beenden einer Sakkade kümmern müssen. Aus diesem Grund bieten die Experten für die Fortbewegung und Blickrichtung Fähigkeiten einer mittleren Hierarchiestufe an. Der Experte für die Fortbewegung bietet z. B. die Fähigkeiten Spur-fahren, Spurwechsel oder Abbiegen an und der Experte für die Blickrichtungssteuerung die Fähigkeit, ein optimales Blickverhalten zu planen, welches den Bedürfnissen der Wahrnehmungsprozesse und dem Aufmerksamkeitsbedarf der Objekte genügt. Fähigkeiten dieser mittleren Hierarchiestufe heißen **schematische Fähigkeiten** und eine Instanz einer schematischen Fähigkeit heißt **Handlung, Tätigkeit** oder **Aktivität**<sup>7</sup>. Schematische Fähigkeiten und Handlungen zeichnen sich im EMS-VISION System durch folgende Eigenschaften aus:

- Sie sind von den Fertigkeiten, die sie benutzen, abhängig,

<sup>6</sup> Nur bei Fähigkeiten bzgl. Fortbewegung und Blickrichtung.

<sup>7</sup> Diese drei Begriffe werden in der vorliegenden Arbeit als Synonyme benutzt.

- sie beinhalten Algorithmen, Automaten und Logiken zur Auswahl, Ansteuerung und Parametrisierung der Fertigkeiten,
- sie sind schwerpunktmäßig der taktischen Ebene zuzuordnen (vgl. Abschnitt 3.3.1),
- sie greifen nicht direkt auf die Aktuatoren und Sensoren des Fahrzeugs zu und
- sie werden von der zentralen Verhaltensentscheidung angestoßen, parametrisiert, gesteuert und überwacht.

Die Abhängigkeiten der schematischen Fähigkeiten von den Fertigkeiten sind in Abb. 3.6 wieder durch Pfeile symbolisiert.

Die schematischen Fähigkeiten können von der Verhaltensentscheidung zu komplexeren Fähigkeiten – **Verhaltensfähigkeiten** genannt – kombiniert werden. Die Instanzen dieser Verhaltensfähigkeiten heißen **Verhaltensweisen**. Die Verhaltensfähigkeiten und Verhaltensweisen besitzen im EMS-VISION System folgende charakteristische Eigenschaften:

- Sie entstehen aus einer Kombination und Koordination von schematischen Fähigkeiten und sind von diesen abhängig,
- sie sind schwerpunktmäßig der strategischen Ebene zuzuordnen (vgl. Abschnitt 3.3.1),
- sie können i. d. R. nicht einer einzelnen Kategorie zugeordnet werden,
- sie besitzen den gleichen Abstraktionsgrad wie die Missionselemente des Missionsplans und werden im Missionskontext ausgeführt.

Der Missionsplan stellt eine abstrakte Aufgabenbeschreibung dar. Die Verhaltensentscheidung versucht, dieser durch eine passende Auswahl von Verhaltensfähigkeiten nachzukommen. Beispiele für Verhaltensfähigkeiten sind „*einer Straße folgen*“, „*in eine Straße einbiegen*“ oder „*Erkundung im Stand*“. Die Abhängigkeiten der Verhaltensfähigkeiten von den schematischen Fähigkeiten sind in Abb. 3.6 wieder durch Pfeile visualisiert.

Betrachtet man das in Abb. 3.6 entstandene Bild, so sieht man einen netzartigen, gerichteten und zyklenfreien Graphen. Die Knoten dieses Graphen repräsentieren die Fähigkeiten der Systemkomponenten. Von unten nach oben nimmt der Abstraktionsgrad und die Komplexität der notwendigen Koordination zu. Die gerichteten Verknüpfungen stellen die Abhängigkeiten zwischen den Fähigkeiten dar. Jede Fähigkeit ist nur dann verfügbar, wenn alle im Netz darunter liegenden Fähigkeiten ebenfalls verfügbar sind.

Die Wechselhäufigkeit zwischen den Fähigkeiten einer Ebene nimmt in Abb. 3.6 von oben nach unten zu. So ändert sich die von der Verhaltensentscheidung ausgewählte Verhaltensfähigkeit nur relativ selten. Die dabei benutzten schematischen Fähigkeiten wechseln schon häufiger. So kann z. B. die Verhaltensweise „Straße folgen“ häufige Spurwechsel erfordern. Eine schematische Fähigkeit benutzt im allgemeinen mehrere Fertigkeiten und wechselt zwischen diesen relativ oft. Die Sollgrößen für die Aktuatoren ändern sich quasi-kontinuierlich.

### 3.3.3 Die Fähigkeiten der Systemkomponenten<sup>5</sup>

Das Fähigkeitenkonzept geht davon aus, dass jede Funktionalität im System in einer Fähigkeit gekapselt ist. Das Modul, welches die Fähigkeit anbietet, heißt **Experte der Fähigkeit**. Soll die Fähigkeit genutzt werden, so legt der Experte eine Instanz der Fähigkeit (Aktion) an. Für die Ausführung einer (komplexen) Fähigkeit sind in der Regel mehrere Teilfähigkeiten notwendig. Ist eine Fähigkeit von anderen Fähigkeiten abhängig und wird diese aktiviert, so führen die Experten der abhängigen Fähigkeiten ebenfalls die dafür notwendigen Aktionen aus (vgl. Abb. 3.7 a). Diese Teilaktionen sind in geeigneter Weise zu synchronisieren. Der Experte der übergeordneten Fähigkeit muss dazu neben dem Fortschritt seiner eige-

nen Aktion auch den Fortschritt und Status der Aktionen überwachen, von denen seine Aktion abhängt [VERHALTEN, 2000].

Die Experten sind prinzipiell in der Lage, die von ihnen angebotenen Fähigkeiten parallel auszuführen (vgl. Abb. 3.7 a). Dieser Parallelität sind natürlich expertenspezifische Grenzen gesetzt. Parallele Aktionen können die Aktionen unterschiedlicher Fähigkeiten sein, oder es können mehrere Aktionen der gleichen Fähigkeit parallel mit unterschiedlichen Parametern ablaufen.

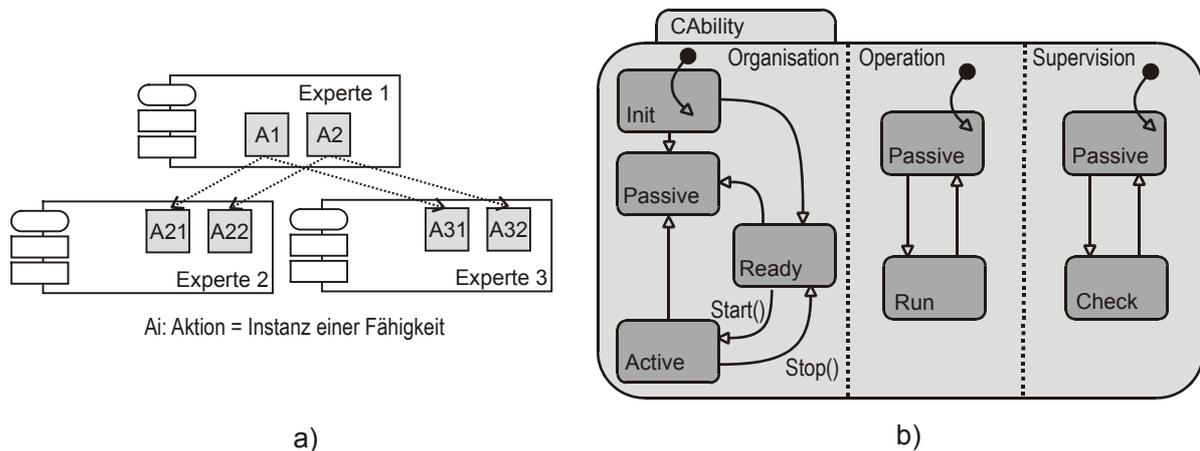


Abb. 3.7: Aktionen in den Experten (Komponentendiagramm nach UML<sup>8</sup>)(a) und die einheitliche Zustandskarte für alle Fähigkeiten (b)

Abb. 3.7 b zeigt die einheitliche Zustandskarte<sup>9</sup> aller Fähigkeiten. Eine Fähigkeit besitzt einen organisatorischen, operativen und überwachenden Teil. In der Zustandskarte erscheinen diese als parallele Zustände. Jeder dieser Zustände ist in Unterzustände aufgeteilt. Der Zustand Organisation verfügt über vier Unterzustände. Im Anfangszustand Init wird die Funktionalität initialisiert und vorbereitet. Nach erfolgreicher Initialisierung geht der organisatorische Teil in den Zustand Organisation.Passive<sup>10</sup> über; bei nicht erfolgreicher Initialisierung wird der Zustand Organisation.Passive eingenommen. Im Zustand Ready werden bereits alle („mentalen“) Funktionalitäten bearbeitet, die keinen Zugriff auf Aktuatoren oder begrenzte Ressourcen verlangen. In diesem Zustand konkurriert die Aktion nicht mit anderen Aktionen. Erst wenn von außen der Aufruf Start() getätigt wird, geht der organisatorische Teil in Active über. Ist der Zustand Active eingenommen, so ist es der Aktion erlaubt, eventuell begrenzte Ressourcen zu belegen und damit in Konkurrenz zu anderen Aktionen zu gehen. Der eigentliche Zugriff auf die Aktuatoren geschieht im Run-Zustand des operativen Teils. Der Anfangszustand Operation.Passive darf in den Zustand Operation.Run übergehen, wenn sich der organisatorische Teil im Zustand Organisation.Active befindet. Wird der Zustand Organisation.Active verlassen, z. B. durch einen externen Stop()-Aufruf, so erzwingt dies einen Übergang vom Zustand Operation.Run in Operation.Passive. Die belegten Ressourcen werden dabei freigegeben.

Durch die Aufteilung in Organisations- und Operationsteil lässt sich von außen vorgeben, welche der Aktionen den Zugriff auf beschränkte Ressourcen erhält. Bei konkurrierenden Aktionen darf sich immer nur eine der Aktionen im Zustand Operation.Run befinden. Auf

<sup>8</sup> Das Komponentendiagramm ist Teil der *Unified Modeling Language* (UML). Eine Einführung in UML findet sich z. B. in [OESTEREICH, 1997].

<sup>9</sup> vgl. Abschnitt 2.3.2.4.

<sup>10</sup> Diese verkürzende Notation wurde in Anlehnung an die Syntax der objektorientierten Programmiersprache C++ gewählt: Der Ausdruck X.Y steht dabei für den Unterzustand Y innerhalb des übergeordneten Zustands X.

diese Weise ist es möglich, konkurrierende und alternative Aktionen in organisatorischer Hinsicht parallel ablaufen zu lassen. Alternative Aktionen laufen lediglich „mental“ ohne Zugriff auf die Aktuatorik ab. Man hält sie vor, um – falls die im Zustand *Operation.Run* befindliche Aktion scheitert – schnell auf die Alternative wechseln zu können.

Im Teilzustand *Supervision* einer Fähigkeit befindet sich die Funktionalität des Experten, um die Ausführung der Aktion zu überwachen (*monitoring*). Wechselt der Status *Supervision.Check* in *Supervision.Passive*, so ist vom Experten ein Fehler festgestellt worden, der zum Scheitern der Aktion führt oder geführt hat. Ein Experte überwacht sowohl den Fortschritt und Zustand seiner eigenen Aktionen, als auch den Zustand der Aktionen, von denen seine Aktionen abhängen. Zu diesem Zweck wird der vollständige Zustand und Status aller Aktionen über die Fähigkeitenrepräsentation kommuniziert (vgl. Abschnitt 3.3.5).

### 3.3.4 Fortschrittsmaße für Aktionen<sup>5</sup>

Grundsätzlich tritt bei einem verteilten System von Experten die Problematik auf, dass mit höher werdender Hierarchieebene die Entscheidungskompetenz zunimmt, aber das bei den Entscheidungen benutzte Expertenwissen abstrakter wird. Zum Beispiel soll sich die Verhaltensentscheidung nicht um die Parametrisierung und Überwachung der Längs- oder Querregelkreise des Fahrzeugs kümmern müssen, da für dieses sehr detaillierte Expertenwissen notwendig ist. Andererseits ist es nicht die Aufgabe der Fahrzeugsteuerung, sich um das Vorhandensein von Wahrnehmungsprozessen und um die Berücksichtigung des vollen Missions- und Situationskontextes zu kümmern.

Da es bei Aktionen zu Fehlern und Verzögerungen kommen kann, muss für eine robuste Ausführung der Status und Fortschritt der Aktionen in geeigneter Weise rückgekoppelt werden. Dazu ist es notwendig, dass die Experten den Fortschritt der bei ihnen ablaufenden Aktionen bemessen und anzeigen. Dabei benutzt jeder Experte sein detailliertes Expertenwissen dazu, sinnvolle Maße für den Fortschritt seiner Aktionen zu generieren. Die Fähigkeitenrepräsentation reicht diese Maße an höhere Hierarchieebenen weiter. Das Maß für den Fortschritt muss so aussagekräftig sein, dass sich die Verhaltensentscheidung ein symbolisches Bild von der sich entwickelnden Situation und den damit verbundenen Veränderungen bei den systemdynamischen Größen machen kann.

Im Vergleich zu den Experten besitzt die Verhaltensentscheidung nur eine ungefähre Vorstellung vom Sollverlauf der Zustandsgrößen. Andererseits hat aber nur sie die Übersicht über die Zusammenhänge zwischen Zustandsgrößen, welche sich aufgrund von Aktionen verschiedener Experten verändern. Deshalb ist es sinnvoll, die Überwachung der im System ablaufenden Aktionen sowohl dezentral in den Experten als auch zentral in der Verhaltensentscheidung durchzuführen.

Die zentrale Überwachung in der Verhaltensentscheidung betrachtet die sich entwickelnde Verkehrssituation und berücksichtigt nur niederfrequente Änderungen der Zustandsgrößen. Dabei sind die Auswirkungen der Aktionen durch wenige Größen repräsentiert (z. B. Anfangs- und Endwerte der sich verändernden Zustandsgrößen, Dauer der Ausführung, usw.). Die Verhaltensentscheidung geht davon aus, dass die Ausführung und Überwachung der schematischen Fähigkeiten durch die Experten auf detaillierteren Form- und Dynamikmodellen beruht, und dass Abweichungen von den Solltrajektorien aufgrund von außen einwirkender Störungen durch stabilisierende Regelungen kompensiert werden.

Dieser Abschnitt stellt fünf allgemeine Maße für den Fortschritt einer Aktion vor (vgl. Abb. 3.8), welche sowohl den räumlichen als auch den zeitliche Aspekt der Ausführung be-

rücksichtigen [VERHALTEN, 2000]. Zusätzlich zu diesen Maßen können die Experten noch weitere, spezifische Fortschrittsmaße definieren und kommunizieren.

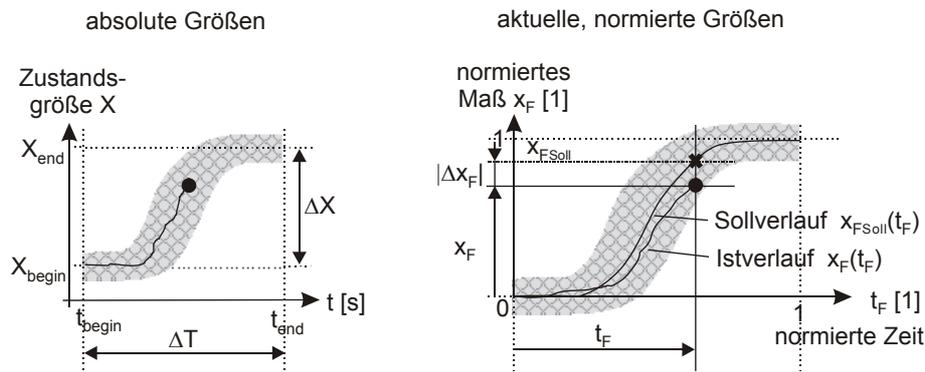


Abb. 3.8: Die den Fortschritt einer Aktion beschreibenden Größen<sup>11</sup>

### 1. Absolute Zeit $\Delta T$ :

Die Größe  $\Delta T$  beschreibt die für die Ausführung der Aktion vom Experten veranschlagte Zeit:

$$\Delta T = t_{end} - t_{begin} \quad (\text{Gl. 3.3})$$

Bei einer Fähigkeit ohne zeitliche Begrenzung (z. B. *Spurhalten*) ist  $\Delta T$  unendlich. Die Einheit der Größe  $\Delta T$  ist Sekunden.

### 2. Absolute Zustandsänderung $\Delta X$ :

Die Größe  $\Delta X$  beschreibt die absolute Zustandsänderung, welche bei der Ausführung der Fähigkeit durchlaufen wird:

$$\Delta X = X_{end} - X_{begin} \quad (\text{Gl. 3.4})$$

Welche systemdynamische Zustandsgröße verwendet wird und welche Einheit damit  $\Delta X$  besitzt, hängt von der Fähigkeit ab. Bei der Fähigkeit *Anhalten* ist dies die metrische Entfernung zwischen dem Fahrzeug und dem geometrischen Haltepunkt. Die Einheit von  $\Delta X$  ist in diesem Fall Meter.

### 3. Aktuelle, normierte Zeit $t_F$ :

Die aktuelle, normierte Zeit  $t_F$  ist die seit dem Start der Aktion abgelaufene Zeit dividiert durch die absolute Zeit  $\Delta T$ :

$$t_F(t) = \frac{t - t_{begin}}{\Delta T} \quad t(t_F) = \Delta T * t_F + t_{begin} \quad (\text{Gl. 3.5})$$

Sie besitzt am Anfang der Aktion den Wert 0, wächst während der Aktion stetig und streng monoton und erreicht beim planmäßigen Ende der Aktion den Wert 1. Kommt es während der Ausführung der Aktion zu einer Verzögerung, so kann  $t_F$  auch größer als 1 werden.

<sup>11</sup> Die Bedeutung der grau unterlegten Bereiche wird in Abschnitt 3.3.6 erklärt.

4. **Aktuelles, normiertes Maß  $x_F$ :**

Analog zur aktuellen, normierten Zeit ist das aktuelle normierte Maß die bereits erreichte Zustandsänderung dividiert durch die absolute Zustandsänderung:

$$x_F(t) = \left| \frac{x(t) - x_{begin}}{\Delta X} \right| \qquad x_F(t_F) = \left| \frac{x(\Delta T * t_F + t_{begin}) - x_{begin}}{\Delta X} \right| \qquad (Gl. 3.6)$$

Die Größe  $x_F$  ist ein Maß dafür, welcher Teil der Aktion bereits abgearbeitet wurde. Mit ihr kann direkt auf den aktuellen, absoluten systemdynamischen Zustand  $X$  und auf die Zustände auf der Situationsebene geschlossen werden.

5. **Maß für die Verzögerung/Vorzeitigkeit  $\Delta x_F$ :**

Es ist nicht notwendig, dass der Wert  $x_F$  streng monoton mit der Zeit  $t_F$  steigt, sondern es können auch Zeitperioden ohne Anstieg oder mit einer Verminderung auftreten. Welchen Wert die Größe  $x_F$  zu einer gegebenen Zeit  $t_F$  erreicht haben muss, ist i. d. R. nur dem Experten der Fähigkeit bekannt. Deshalb kommuniziert der Experte die Größe  $\Delta x_F$ , welche folgendermaßen definiert ist:

$$\Delta x_F(t_F) = x_F(t_F) - x_{F_{Soll}}(t_F) \qquad (Gl. 3.7)$$

Die Größe  $\Delta x_F$  dient dazu, nicht nur den durch  $x_F$  gemessenen Ausführungsgrad der Aktion im Auge zu behalten, sondern auch den zeitlichen Aspekt. Dabei ist die Größe  $x_{F_{Soll}}(t_F)$  das aktuelle, normierte Sollmaß ohne Verzögerung bei der Ausführung der Fähigkeit (= Sollverlauf zur Zeit  $t_F$ ). Dieser Sollverlauf ist nur dem Experten der Fähigkeit bekannt. Wird z. B. mit fortschreitender Zeit  $t_F$  kein Fortschritt mehr erzielt, so verändert sich  $x_{F_{Soll}}(t_F)$  weiterhin und der Betrag von  $\Delta x_F$  wird größer (vgl. Abb. 3.9). Der Betrag von  $\Delta x_F$  ist ein Maß für die Verzögerung bzw. das Vorseilen der Aktion. Bei Verzögerungen ist  $\Delta x_F$  negativ, beim Vorseilen positiv. Ist die Aktion exakt im Zeitplan, so ist  $\Delta x_F$  null.

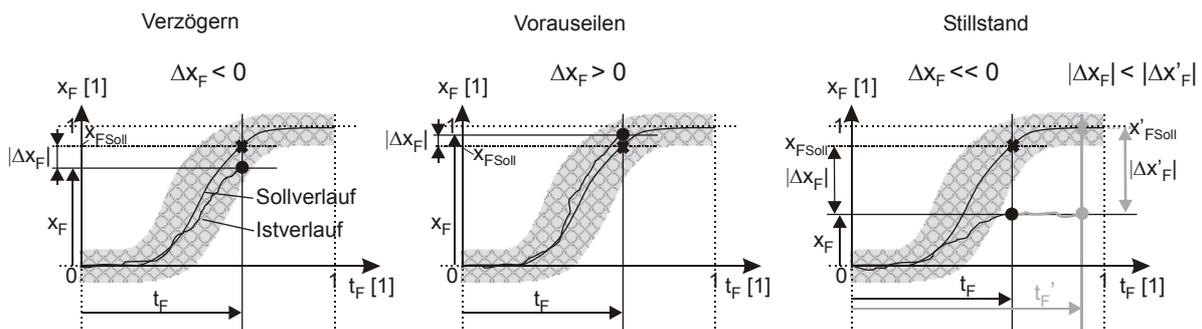


Abb. 3.9: Unterschiedliche Fortschritte bei der Ausführung einer Fähigkeit<sup>11</sup>

3.3.5 **Repräsentation der Fähigkeiten durch das Fähigkeitsnetz**

Die prinzipiellen Fähigkeiten eines Systems sind durch dessen Hard- und Softwaremodule fest vorgegeben und deshalb statisch repräsentierbar. Zwischen den einzelnen Fähigkeiten bestehen i. A. Abhängigkeiten, welche für eine Verhaltensentscheidung, die unter komplexen Fähigkeiten auswählt, nur bedingt von Interesse sind und vor dieser verborgen werden können. Die Fähigkeitenrepräsentation muss dazu die Abhängigkeiten zwischen den Fähigkeiten beinhalten. In dem von der Arbeitsgruppe VERHALTEN [2000] entwickelten Fähigkeitenkon-

zept geschieht die Repräsentation der Fähigkeiten durch das sogenannte Fähigkeitsnetz (vgl. Abschnitt 3.3.2).

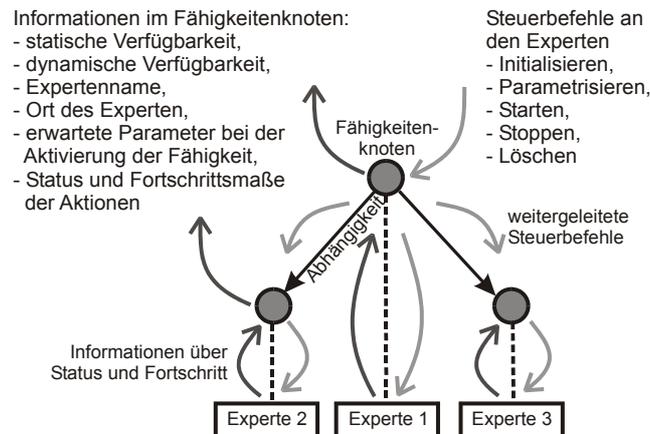


Abb. 3.10: Funktionalität des Fähigkeitsnetzes

Das **Fähigkeitsnetz** ist ein unidirektionaler, zyklensfreier, gerichteter Graph, der auf statische Weise die im System implementierten und integrierten Fähigkeiten repräsentiert. Die Knoten des Netzes – **Fähigkeitenknoten** genannt – repräsentieren die einzelnen Fähigkeiten. Die Fähigkeitenknoten sind durch gerichtete Kanten, welche Abhängigkeiten repräsentieren, miteinander verbunden. Neben der eigentlichen Repräsentationsaufgabe bietet das Fähigkeitsnetz den Experten und Nutzern von Fähigkeiten komfortable Funktionalität an. Folgende Informationen werden von einem Fähigkeitenknoten bereitgestellt (vgl. Abb. 3.10):

- Es wird neben der prinzipiellen, statischen Verfügbarkeit einer Fähigkeit auch deren dynamische, zum aktuellen Zeitpunkt geltende Verfügbarkeit angezeigt. Letztere richtet sich nach eventuell vorhandenen Fehlern und Ausfällen bei Systemkomponenten, nach der Güte des momentan vorhandenen Wissens und nach den bereits genutzten Fähigkeiten im System.
- Ein Fähigkeitenknoten enthält eine Referenz zum Experten der Fähigkeit (Name und Ort im verteilten System) und die Parameter, welche bei der Aktivierung der Fähigkeit anzugeben sind.
- Ein Fähigkeitenknoten gibt Auskunft darüber, wie viele Instanzen einer Fähigkeit (Aktionen) bereits beim Experten ablaufen und wie viele weitere Instanzen zusätzlich gestartet werden können.
- Ein Fähigkeitenknoten beinhaltet den Status und die Fortschrittsmaße der beim Experten ablaufenden Aktionen der Fähigkeit (vgl. Abschnitt 3.3.4).

Der Experte einer Fähigkeit schreibt die entsprechenden Daten in seinen Fähigkeitenknoten<sup>12</sup>, wo sie automatisch zu potentiellen Nutzern der Fähigkeit kommuniziert werden.

Die Ausführung (komplexer) Fähigkeiten erfordert die Zusammenarbeit mehrerer Experten und führt zu der Problematik, ein verteiltes System zu steuern und zu koordinieren. Die dazu notwendigen Mechanismen sind in das Fähigkeitsnetz integriert. Wenn ein Prozess eine fremde Fähigkeit nutzen möchte, kann er seine Steuerbefehle über das Fähigkeitsnetz senden. Dabei dienen die Fähigkeitenknoten als einheitliche Schnittstellen zu den Experten der Fähigkeiten. Die Steuerbefehle werden automatisch zum Experten der Fähigkeit und zu den

<sup>12</sup> Dieser Abgleich ist durch den Prozessrahmen CClientScene automatisiert (vgl. Anhang B).

Experten der Teilfähigkeiten gesandt. Mit den entsprechenden Steuerbefehlen lässt sich eine Instanz einer Fähigkeit initialisieren, parametrisieren, starten, stoppen und wieder löschen (vgl. Abschnitt 3.3.7 und Abb. 3.10). Durch das Fähigkeitsnetz ist es für den Nutzer einer (komplexen) Fähigkeit transparent, ob und aus welchen Teilfähigkeiten sich eine Fähigkeit zusammensetzt. Das Wissen über die Abhängigkeiten der Fähigkeiten ist im Fähigkeitsnetz und in den Experten angelagert.

Wie dieser Abschnitt zeigt, dient das Fähigkeitsnetz nicht nur der Repräsentation der im System vorhandenen Funktionalität, sondern unterstützt auch deren einheitliche Steuerung und Koordination. Der gesamte Kontrollfluss im System kann so auf einheitliche Weise gestaltet und über das Fähigkeitsnetz abgewickelt werden

### 3.3.6 Die internen Ablaufmodelle der Fähigkeiten

Nachdem sich die Verhaltensentscheidung für gewisse Fähigkeiten entschieden hat, werden diese im verteilten System instanziiert und gestartet. Im System sind daraufhin zusammengehörige und kooperierende Aktionen dieser Fähigkeiten aktiv (vgl. Abb. 3.7 a). Um feststellen zu können, ob die Aktionen erwartungskonform ablaufen oder nicht, muss die Verhaltensentscheidung eine Vorstellung (mentales Modell) vom Ablauf der Aktionen besitzen. Dazu verfügt die Verhaltensentscheidung über sogenannte *interne Ablaufmodelle*. Die Ablaufmodelle bilden die Entscheidungsgrundlage, um auf den Erfolg oder Misserfolg der Aktionen angemessen zu reagieren.

Wie KRAISS [1985] betont, „lernt“ auch der menschliche Fahrer die Dynamik seines Fahrzeugs. Dabei baut er im Verlauf des Lernens ein inneres, mentales Modell des zu regelnden Systems auf, welches die Erwartung hinsichtlich des dynamischen Systemverhaltens beschreibt. Zu diesem mentalen Modell gehören z. B. genaue Vorstellungen über Integrationsstufen, Zeitkonstanten, Totzeiten und etwaige Kopplungen zwischen den Zustandsgrößen. Die Korrekturingriffe erfolgen aus einem laufenden Sollwert-Istwert-Vergleich zwischen tatsächlichem Ablauf und mentaler Erwartung. Auch die Bediener hochautomatisierter Systeme müssen ein mentales Modell besitzen, um bei Fehlfunktionen im System angemessen darauf reagieren zu können und u. U. katastrophale Fehlreaktionen zu vermeiden. Aus diesem Grund werden oft voll funktionsfähige automatische Systeme regelmäßig manuell betrieben, um die Operateure mit ihren „mental Modellen“ auf einem akzeptablen Stand zu halten.

Die Verwendung von mentalen Modellen von Fähigkeiten zum Zweck der Überwachung ist nicht auf die zentrale<sup>13</sup> Verhaltensentscheidung beschränkt. So können auch unterlagerte Entscheidungsinstanzen oder Experten von Fähigkeiten interne Modelle für die Überwachung des erwartungskonformen Ablaufs einsetzen.

Die Beschreibung des Ablaufs einer Fähigkeit kann auf der Situationsebene und/oder auf der (systemdynamischen) Zustandsebene erfolgen (vgl. Abb. 3.11 a). Auf der Situationsebene wird der Ablauf einer Fähigkeit in eine Sequenz von Teilabläufen zerlegt und durch einen Zustandsautomaten repräsentiert. Die Sequenz beginnt mit einem Anfangszustand, führt über beliebig viele Zwischenzustände und schließt mit einem Endzustand. Neben einer einfachen Kette von aufeinanderfolgenden Zuständen sind beliebig komplexe Zustandsautomaten mit hierarchischen Strukturen und parallelen Zweigen möglich. MAURER [2000] schlägt für die

---

<sup>13</sup> Wie in Kapitel 3.5.1 dargestellt wird, befindet sich das hier beschriebene Konzept zur Situationsanalyse und Verhaltensentscheidung innerhalb des EMS-VISION Systems in der sogenannten „zentralen Entscheidungsinstanz“ (Central Decision). Dieser zentralen Verhaltensentscheidung sind weitere, spezialisierte Entscheidungsmodulare unterlagert, welche taktische Entscheidungen bzgl. Blickrichtung und Fortbewegung treffen.

Modellierung komplexer, paralleler Abläufe Zustandskarten (*statecharts*) vor (vgl. Abschnitt 2.3.2.4).

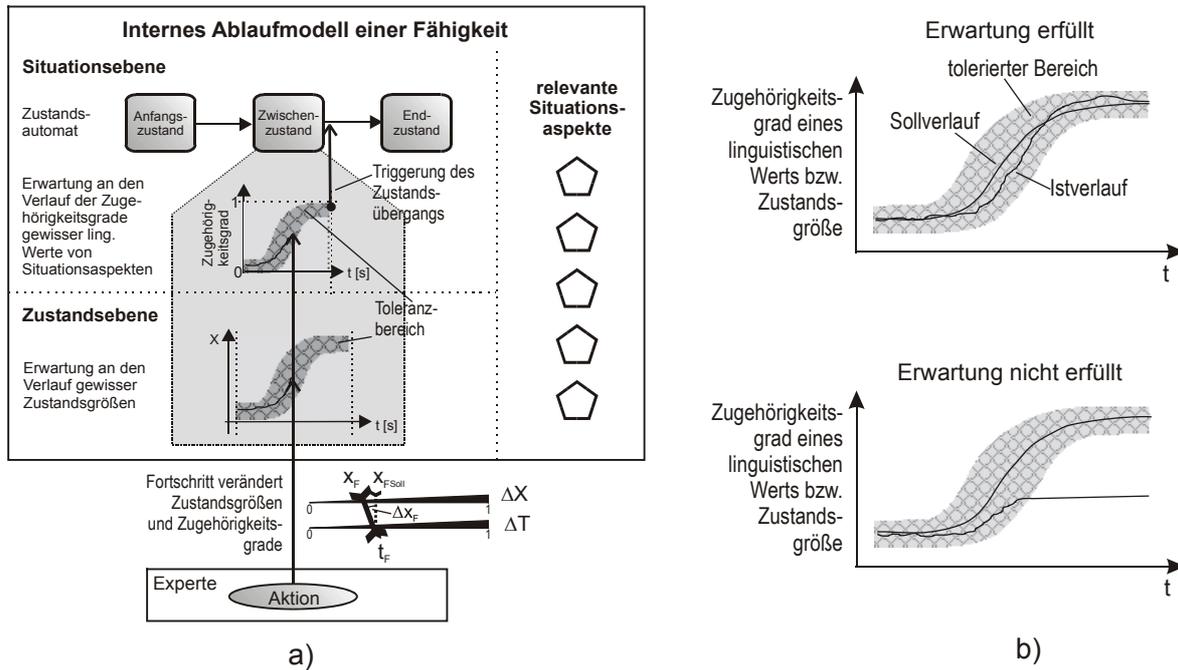


Abb. 3.11: Überwachung einer Aktion auf der Zustands- und Situationsebene (a) und Vorgabe von Sollverläufen mit Toleranzbereichen (b)

Ein (Zwischen-)Zustand ist dadurch gekennzeichnet, dass sich in diesem die linguistischen Werte gewisser Situationsaspekte in vorgegebener Weise entwickeln. Das Erreichen des erwarteten Werts triggert i. A. den Übergang in den nächsten Zustand. Innerhalb der zentralen Verhaltensentscheidung treibt auf diese Weise der in den Experten gemessene und kommunizierte Fortschritt der Aktionen die internen Ablaufmodelle. Ferner ist es möglich, die Zustandswechsel im Zustandsautomaten an die Zustandswechsel in den ablaufenden Aktionen zu koppeln (vgl. Abschnitte 3.3.3 und 3.3.6.2). Scheitert eine Fähigkeit, so wird der zuletzt eingenommene Zustand im Zustandsautomaten nicht mehr verlassen, und es ist bekannt, in welchem Zwischenzustand die Aktion gescheitert ist. Die zentrale Verhaltensentscheidung kann in diesem Fall in Abhängigkeit vom Zwischenzustand alternativen Aktionen den Zugriff auf die Aktuatoren gewähren oder eine Neuplanung anstoßen.

Erwartungen an den Verlauf von Fähigkeiten können neben der Situationsebene auch auf der (systemdynamischen) Zustandsebene formuliert werden (vgl. Abb. 3.11 a). Dieses ist insbesondere deshalb wichtig, weil es für die Verhaltensentscheidung eine Rolle spielt, welche systemdynamischen Zustände mit der teilweise ausgeführten, aber letztendlich gescheiterten Fähigkeit erreicht wurden.

Die Erwartungen der Verhaltensentscheidung an die Verläufe gewisser linguistischer Werte oder gewisser systemdynamischer Zustandsgrößen werden in Form von zeitfunktionalen Sollverläufen mit Toleranzbereichen im internen Ablaufmodell niedergelegt. Diese Toleranzbereiche sind so gestaltet, dass sie bei einem erwartungskonformen Ablauf der Aktion nicht verlassen werden (vgl. Abb. 3.11 b). Je schmaler ein Toleranzbereich ist, desto strenger spezifiziert er die Erwartung an den Istverlauf. Ein modellierter Sollverlauf erstreckt sich i. d. R. über den Zeitraum, in dem der zugehörige Zustand auf der Situationsebene gültig ist (vgl. Abb. 3.11 a). Für die Ermittlung des Istverlaufs empfängt das Ablaufmodell die Maße für den

Fortschritt der zugehörigen Aktion vom Experten der Fähigkeit ( $\Delta T$ ,  $\Delta X$ ,  $t_F$ ,  $x_F$ ,  $\Delta x_F$ , vgl. Abschnitt 3.3.4). Die Fortschrittsmaße der Experten lösen damit die Zustandsübergänge auf der Situationsebene aus und treiben das Ablaufmodell.

In den internen Ablaufmodellen der Verhaltensentscheidung ist ferner niedergelegt, welche Situationsaspekte von der Situationsanalyse auf welche Objekte in der Szene anzuwenden sind, um eine sichere und erfolgreiche Ausführung der Fähigkeiten zu gewährleisten (vgl. Abb. 3.11 a). Sie ermöglichen auf diese Weise das in Abschnitt 3.2.4 beschriebene iterative Vorgehen bei der Situationsanalyse und Verhaltensentscheidung: Nach der Auswahl einer Fähigkeit werden die im zugehörigen internen Ablaufmodell befindlichen, relevanten Situationsaspekte und die Anweisungen bzgl. deren Anwendung an die Situationsanalyse gesandt und dort im nächsten Iterationsschritt berechnet.

Für die Verhaltensentscheidung ist es nicht notwendig, alle Daten und Ereignisse, welche bei den unterlagerten Fähigkeiten eine Rolle spielen, zu kennen. Sie kann sich bis zu einem gewissen Grad darauf verlassen, dass Störungen von den Experten der unterlagerten Fähigkeiten ausgeglichen werden. Die Verhaltensentscheidung muss jedoch wissen, welche Übergänge mit einer Fähigkeit zu erreichen sind und wie sie die Parameter der Fähigkeit an die aktuelle Situation anzupassen hat. Dieses Wissen korrespondiert mit dem quasi-statischen Wissen, das durch die Methoden der Künstlichen Intelligenz (KI) gehandhabt werden kann [DICKMANN, 2002A]. Aus diesem Grund nehmen die Ablaufmodelle eine zentrale Rolle bei der Einbindung von KI-Methoden in das EMS-VISION System ein. Die Ablaufmodelle beschreiben die funktionellen Abläufe von Aktionen und ihre Auswirkungen auf der Situations- und Zustandsebene und können deshalb neben der oben diskutierten Überwachungsaufgabe auch für Planungsaufgaben und zur Prädiktion der Situation herangezogen werden.

Die folgenden Unterkapitel beschreiben zwei Beispiele von Ablaufmodellen. Das erste Beispiel überwacht die Ausführung der Fähigkeit *Überholen* auf der Situationsebene. Dieses Ablaufmodell ist Teil der zentralen<sup>13</sup> Verhaltensentscheidung. Wie oben bereits angedeutet wurde, ist die Verwendungsmöglichkeit von mentalen Ablaufmodellen nicht auf die zentrale Verhaltensentscheidung beschränkt. So beschreibt das zweite Unterkapitel ein Ablaufmodell der Fähigkeit *Anhalten*, welches entsprechend der in Kapitel 3.5.1 dargestellten organisatorischen Hierarchie von Entscheidungsinstanzen des EMS-VISION Systems nicht der zentralen Verhaltensentscheidung zuzuordnen ist, sondern dem Experten für die Fortbewegung (vgl. [SIEDERSBERGER, 2003]). Anhand dieses Beispiels wird dargestellt, wie die Ablaufmodellierungen auf der Situationsebene und auf der Zustandsebene zusammenwirken, um eine einfache Anpassungsfunktionalität zu verwirklichen. Kapitel 4.3 schildert als weiteres Beispiel eine Überwachung und Anpassung einer Fähigkeit bezüglich Blickrichtung.

### 3.3.6.1 Das Ablaufmodell der Fähigkeit *Überholen*

Dieses Kapitel betrachtet das Ablaufmodell der Fähigkeit *Überholen* auf der Situationsebene (vgl. Abb. 3.12). Auf der Situationsebene kann ein Überholmanöver durch folgende Kette von Zuständen beschrieben werden:

Beschleunigen → Ausscheren → Vorbeiziehen → Einscheren

Im Zustand *Beschleunigen* verlangen die Toleranzbereiche für den Situationsaspekt *Relativgeschwindigkeit*<sup>14</sup>, dass die Geschwindigkeit des überholten Fahrzeugs kleiner ist oder kleiner wird, als die Geschwindigkeit des überholenden Fahrzeugs. Trifft dies nicht zu, so zeigt diese

<sup>14</sup> Der Situationsaspekt wird dabei auf das überholte Fahrzeug angewandt (Zuordnungsobjekt); das Referenzobjekt ist das überholende Fahrzeug (vgl. Abb. 3.12).

Diskrepanz ein Scheitern des Beschleunigungsvorgangs an. In analoger Weise wird im Zustand Vorbeiziehen verlangt, dass das überholte Fahrzeug langsamer fährt als das überholende Fahrzeug.

Die linguistischen Werte des Situationsaspekts *Relativlage*<sup>14</sup> ändern sich in den Zuständen Ausscheren, Vorbeiziehen und Einscheren. Die Verhaltensentscheidung setzt voraus, dass vor dem Ausscheren das überholte Fahrzeug der Vorausfahrende<sup>15</sup> des überholenden Fahrzeugs ist, und nach dem Ausscheren der Rechts\_Vorausfahrende<sup>15</sup>. Beim Vorbeiziehen ändert sich die Relativlage von Rechts\_Vorausfahrender<sup>15</sup>, über Rechts\_Nebenfahrender<sup>15</sup> zu Rechts\_Nachfahrender<sup>15</sup> (nicht in Abb. 3.12 dargestellt). Beim Einscheren wird aus dem Rechts\_Nachfahrender<sup>15</sup> der Nachfahrende<sup>15</sup> und der Überholvorgang ist abgeschlossen.

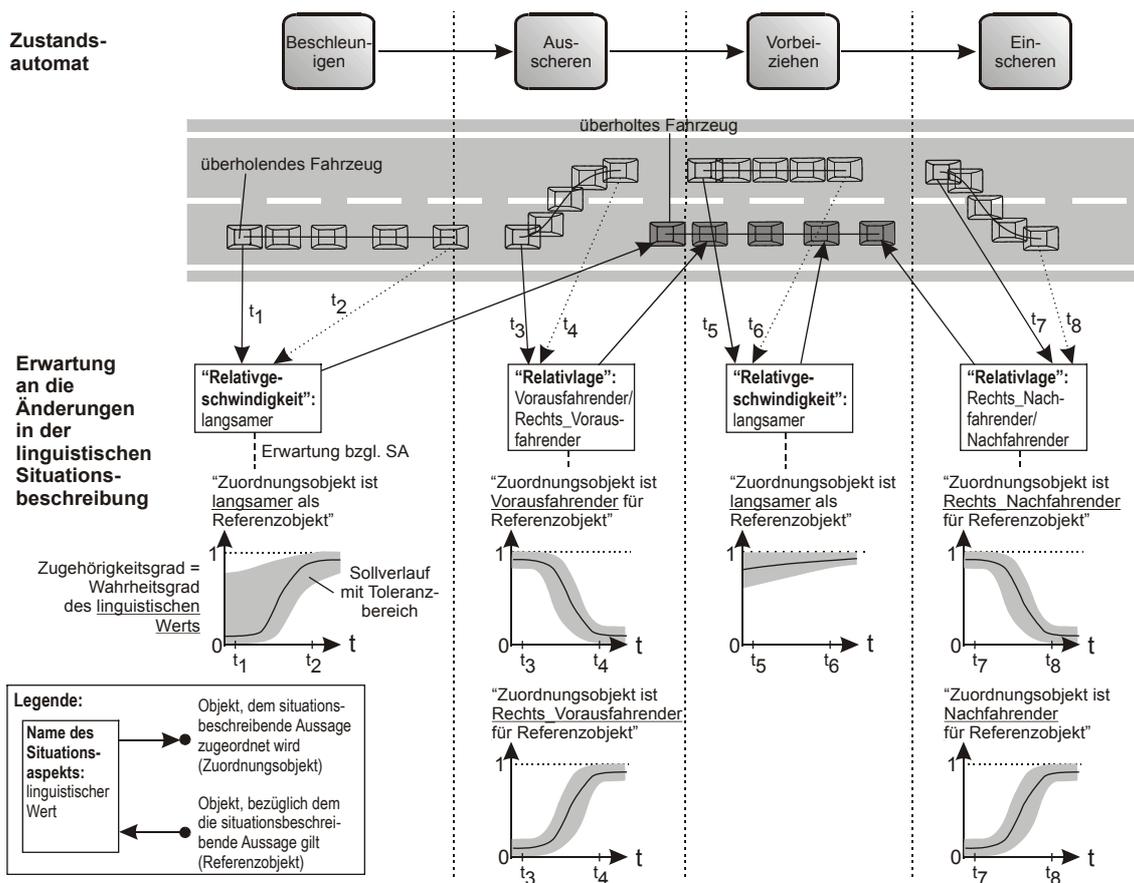


Abb. 3.12: Das interne Ablaufmodell der Fähigkeit „Überholen“ auf der Situationsebene

Wie dieses Beispiel zeigt, werden im internen Ablaufmodell einer Fähigkeit mal mehr mal weniger strenge Bedingungen an die Verläufe der linguistischen Werte gewisser Situationsaspekte gestellt. Die Strenge wird dabei durch zeitfunktionale Toleranzbereiche um die erwarteten Sollwertverläufe beschrieben (vgl. Abb. 3.12, unten). Je schmaler der Toleranzbereich ist, desto strenger ist die damit ausgedrückte Bedingung. In den Teilzuständen *Beschleunigen* und *Vorbeiziehen* wird direkt nach dem Eintritt in den Zustand eine relativ große Abweichung vom Sollwert akzeptiert. Die tolerierte Abweichung reduziert sich jedoch mit zunehmender Dauer. Verlässt der Istverlauf den Toleranzbereich, so werden seitens der Verhaltensentscheidung Reaktionen notwendig. Im dargestellten Beispiel kann dies ein Abbruch des Überholvorgangs sein.

<sup>15</sup> linguistischer Wert des Situationsaspekts *Relativlage*.

### 3.3.6.2 Das Ablaufmodell der Fähigkeit *Anhalten*

Als weiteres Beispiel soll nun das Ablaufmodell der Fähigkeit *Anhalten* gezeigt werden. Wie oben bereits erwähnt, ist dieses Beispiel entsprechend der in Kapitel 3.5.1 beschriebenen Hierarchie von Entscheidungsinstanzen des EMS-VISION Systems dem Experten für die Fortbewegung zuzuordnen.

Die Darstellung in Abb. 3.13 folgt der Darstellung von Zustandskarten nach HAREL [1987], bei der u. a. Transitionsbedingungen und Ereignisbehandlungen in Abhängigkeit von Zustandswechsel definiert werden können: Ein Zustandswechsel erfolgt, wenn die Transitionsbedingung T erfüllt ist. Die Bezeichnungen entry /Ai und exit /Ai sagen aus, dass die Ereignisbehandlung Ai beim Eintritt bzw. beim Verlassen des Zustands ausgeführt wird.

Ziel des Ablaufmodells einer Fähigkeit ist die Beurteilung der Güte und Leistungsfähigkeit einer Aktion, so dass der Anwender der Fähigkeit in geeigneter Weise darauf reagieren kann. Eine dieser Reaktionen kann sein, dass systematische Abweichungen, welche bei der wiederholten Ausführung der Fähigkeit immer wieder auftreten, kompensiert werden. So ist es möglich, dass die Anhaltstrecke und/oder die während des Bremsens eingehaltene Verzögerung systematisch um einen gewissen Prozentsatz von den kommandierten Sollwerten abweichen. Beim Anhalten können solche systematischen Abweichungen z. B. von einem Verschleiß an der Bremsanlage verursacht werden.

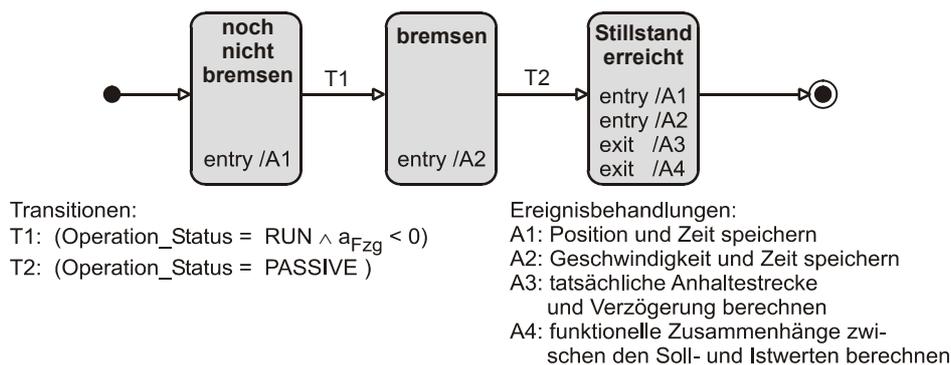


Abb. 3.13: Der Zustandsautomat des Ablaufmodells der Fähigkeit „Anhalten“ (Zustandskarte nach HAREL [1987])

Die Handlung *Anhalten* lässt sich auf der Situationsebene in die Zustände „noch nicht bremsen“, „bremsen“ und „Stillstand erreicht“ aufteilen. Zu Beginn des Anhaltens wird der erste Zustand eingenommen. Wechselt der Operationsstatus der Handlung *Anhalten* in RUN, d. h. der eigentliche Bremsvorgang mit der Ansteuerung der Fahrzeugbremse beginnt (vgl. [SIEDERSBERGER, 2003]), so schaltet die Transition T1 und es erfolgt im Zustandsautomaten ein Wechsel in den Zustand „bremsen“. Die Transition T2 schaltet, wenn der Operationsstatus der Handlung *Anhalten* in den Zustand PASSIVE zurückkehrt und dadurch der Fortbewegungsexperte das Anhalten für beendet erklärt.

Die Fähigkeit *Anhalten* besitzt als Parameter die Anhaltstrecke und die Anhalteverzögerung (Tab. 3.1 auf Seite 80). Das Ablaufmodell dieser Fähigkeit soll nun überwachen, ob diese Parameter tatsächlich eingehalten werden oder nicht. Zu diesem Zweck speichert das Ablaufmodell beim Wechsel in den Zustand „noch nicht bremsen“ die aktuelle Position und die aktuelle Zeit (Ereignisbehandlung A1 in Abb. 3.13). Beim Eintritt in den Zustand „bremsen“ wird die Geschwindigkeit zusammen mit dem Zeitstempel gespeichert (Ereignisbehandlung A2). Beim Eintritt in den Zustand „Stillstand erreicht“ fragt das Ablaufmodell wieder die Position, die Geschwindigkeit und die aktuelle Zeit ab und überprüft, ob die Geschwindigkeit

Null ist. Aus den ermittelten Daten lassen sich dann die tatsächliche Anhaltstrecke und die durchschnittliche Verzögerung berechnen (Ereignisbehandlung A3 in Abb. 3.13).

Ein Ablaufmodell hat die Möglichkeit, die Ausführung der zugehörigen Fähigkeit zu abstrahieren und als *black box* mit einem messbaren Übertragungsverhalten zu betrachten, ohne Wissen darüber, wie und warum es zu dazu kommt. Zum Beispiel kann es die Ausführung als P-Übertragungsglied betrachten, bei dem der kommandierte Sollwert  $x_i^{soll}$  in einem linearen Zusammenhang mit dem tatsächlich eingehaltenen Istwert  $x_i^{ist}$  steht:

$$x_i^{ist} = K_i \cdot x_i^{soll} \quad (\text{Gl. 3.8})$$

$K_i$  ist die Konstante des P-Übertragungsglieds und beschreibt den prozentualen Fehler, der bei der Ausführung  $i$  auftritt. Wird der Sollwert um 10 % unterschritten, so gilt  $K_i = 0,9$ ; tritt kein Fehler auf, so ist  $K_i = 1$ . Neben dem linearen sind auch beliebig komplexe funktionale Zusammenhänge zwischen den Ist- und Sollwerten modellierbar.

Im Ablaufmodell der Fähigkeit *Anhalten* werden nach jeder Ausführung die  $K_i$ 's für die Sollgrößen Anhaltstrecke und Anhalteverzögerung berechnet und in einer Datei abgelegt. In dieser „Erfahrungsdatei“ befinden sich die  $K_i$ 's der letzten  $n$  Anhaltenmanöver. Diese Erfahrung bleibt über die Laufzeit des Systems hinaus erhalten und wird bei einem Neustart des Systems geladen.

Bei einer erneuten Ausführung und Parametrisierung der Fähigkeit geht der Anwender der Fähigkeit davon aus, dass es zu den gleichen prozentualen Abweichungen kommen wird. Um diese zu kompensieren, kommandiert er nicht den Nominalwert  $x_{i+1}^{nom}$ , sondern einen um die systematische Abweichung korrigierten Wert. Dazu werden die  $K_i$ 's der letzten  $n$  Ausführungen gemittelt, was zu  $\bar{K}_i$  führt. Der kommandierte Sollwert  $x_{i+1}^{soll}$  berechnet sich dann durch

$$x_{i+1}^{soll} = \frac{x_{i+1}^{nom}}{\bar{K}_i} \quad (\text{Gl. 3.9})$$

Kommt es bei der Ausführung zu dem erwarteten Übertragungsverhalten  $K_{i+1} = \bar{K}_i$ , so gilt:

$$x_{i+1}^{ist} = x_{i+1}^{nom} \quad (\text{Gl. 3.10})$$

Bei der Fähigkeit *Anhalten* wird diese Korrektur sowohl für die Anhaltstrecke als auch für die Anhalteverzögerung durchgeführt. Auf diese Weise lernt das Ablaufmodell die systematischen Abweichungen bei der Ausführung und kann sie kompensieren. Zudem ist es denkbar, dass das System bei einer zu großen prozentualen Abweichung eine Warnung ausspricht und/oder die Fähigkeit durch alternative Fähigkeiten ersetzt (z. B. Nothalt). Abschnitt 6.3 beschreibt experimentelle Ergebnisse dieser Kompensation.

### 3.3.7 Die Steuerung kooperierender Aktionen im System über das Fähigkeitsnetz

In Anhang B sind die Basisklassen des Fähigkeitskonzepts beschrieben: CNetNode, CAbilityNode, CClientScene und CAbility. Insbesondere zeigt Abb. B.1 in einem detaillierten Klassendiagramm die Beziehungen zwischen den Klassen. Aufbauend auf diesem Klassendesign, soll nun anhand eines Beispiels der über das Fähigkeitsnetz abgewickelte Kontrollfluss illustriert werden. Abb. 3.14 zeigt ein Fähigkeitsnetz mit den drei Fähigkeitsknoten 1, 2 und 3. Die Fähigkeitsknoten sind Objekte von Klassen, die von der Basisklasse CAbil-

ityNode abgeleitet sind, und dienen als Schnittstellen zu den Experten. Ein weiterer Prozess möchte die Fähigkeit 1 nutzen.

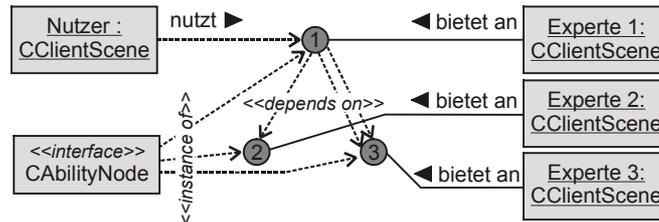


Abb. 3.14: Beispiel eines Fähigkeitsnetzes mit 3 Experten und einem Nutzer (Klassendiagramm nach UML)

Das Fähigkeitsnetz von Abb. 3.14 spezifiziert, dass die Fähigkeit 1 von den Fähigkeiten 2 und 3 abhängt, wobei zur Ausführung von Fähigkeit 1 zwei Instanzen der Fähigkeit 3 notwendig sind (Fähigkeitenknoten 1 besitzt zwei Verknüpfungen zum Fähigkeitenknoten 3). Durch das Fähigkeitsnetz wird gewährleistet, dass der Nutzer der Fähigkeit 1 diese Abhängigkeiten nicht zu kennen braucht. Alle Prozesse im System – sowohl der Nutzer als auch die Experten – verfügen über den gleichen, durch die Klasse CClientScene realisierten Prozessrahmen zur Kommunikation, Wissensrepräsentation und Ereignisbehandlung (vgl. Anhang B).

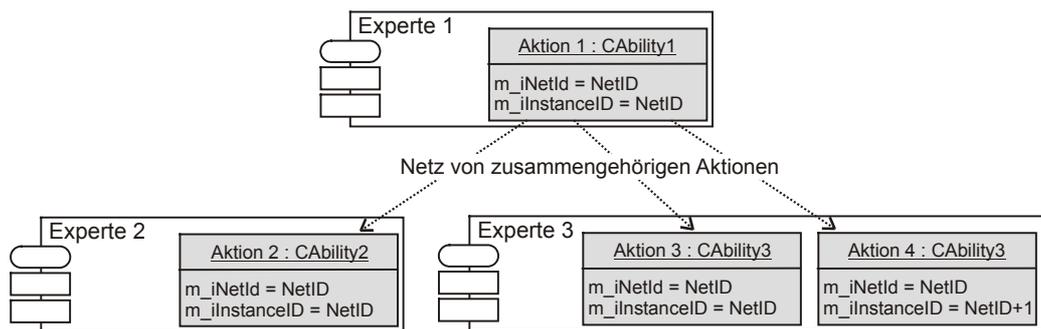


Abb. 3.15 Aktionsnetz nach Instanziierung der komplexen Fähigkeit 1 von Abb. 3.14 (Komponentendiagramm nach UML)

Im ersten Schritt ruft der Nutzer die Funktion Initialize() des Fähigkeitenknotens 1 auf (vgl. Abb. 3.16). Dieser Aufruf führt dazu, dass an den Experten 1 der Fähigkeit 1 eine Prozessnachricht gesandt wird. Diese Prozessnachricht hat wiederum den Aufruf der Ereignisbehandlungsfunktion OnInitializeAbility() beim Experten 1 zur Folge<sup>16</sup>, welche eine Instanz der Fähigkeit 1 anlegt. Die Fähigkeit 1 sei in diesem Beispiel in der Klasse CAbility1 gekapselt. Die Klasse CAbility1 besitzt die Basisklasse CAbility und füllt alle (virtuellen) Klassenfunktionen der Basisklasse mit fähigkeitspezifischer Funktionalität. Ist eine Instanz der Klasse CAbility1 mit dem Namen „Aktion 1“ angelegt, so erfolgt anschließend der Aufruf ihrer Klassenfunktion OnInitialize() zur Initialisierung.

<sup>16</sup> Die Übermittlung von Ereignissen zwischen Prozessen ist in Abb. 3.16 aus Gründen der Übersichtlichkeit nur vereinfacht dargestellt.

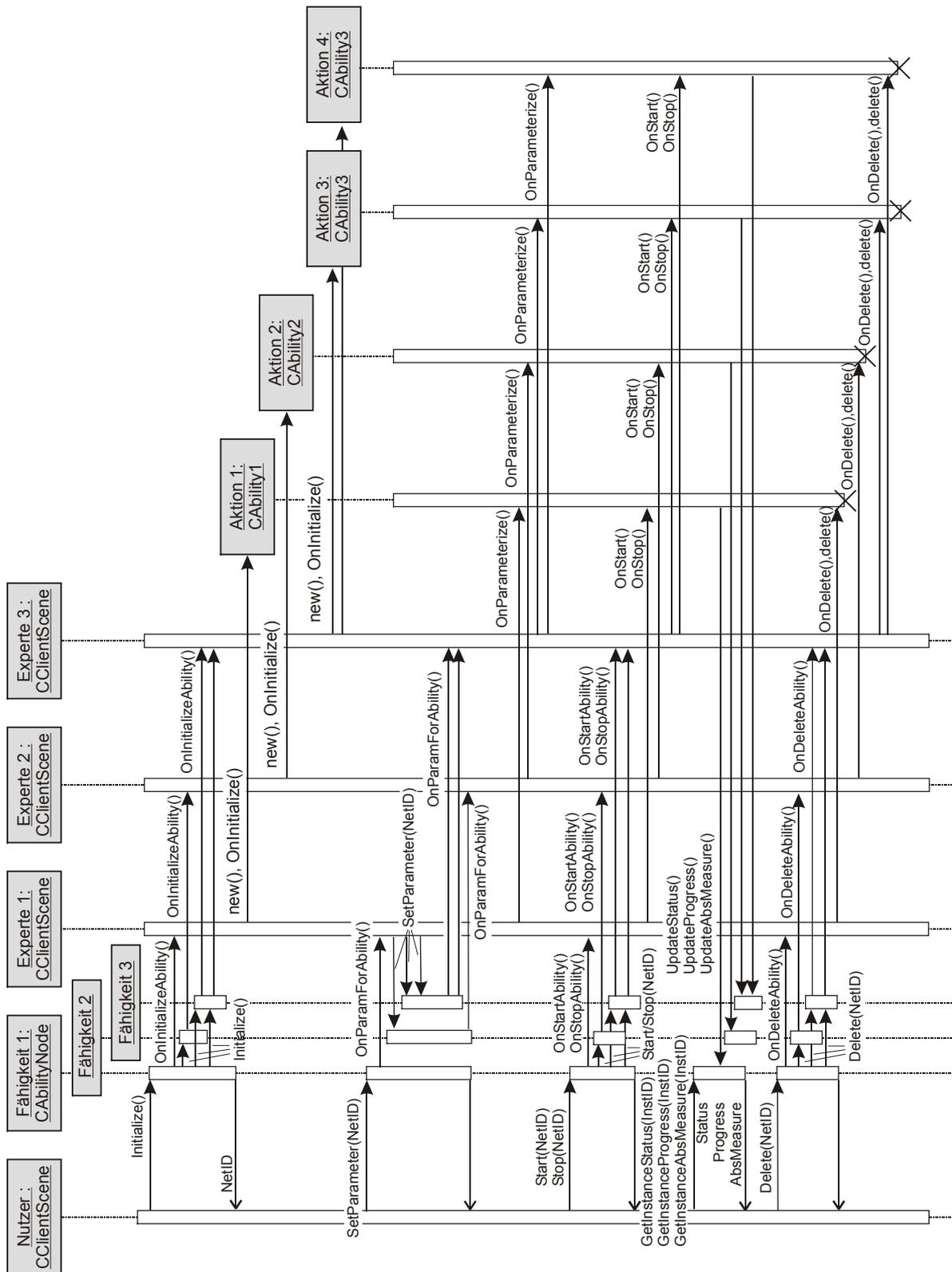


Abb. 3.16: Sequenzdiagramm für das Fähigkeitenkonzept (Darstellung nach UML)

Das Fähigkeitsnetz leitet automatisch alle Aufrufe der Steuerfunktionen eines Fähigkeitsknotens rekursiv an die Knoten der Teilfähigkeiten weiter. So zieht im obigen Beispiel der Aufruf der Funktion `Initialize()` beim Fähigkeitsknoten 1 den Aufruf der gleichlautenden Funktionen bei den Fähigkeitsknoten 2 und 3 nach sich. Auch diese Aufrufe bewirken die zuvor beschriebene Benachrichtigung der zugehörigen Experten und das Anlegen der entsprechenden Instanzen der Teilfähigkeiten (Aktionen 2 bis 4). Dabei ist zu erwähnen, dass die

doppelte Verknüpfung zwischen den Fähigkeitenknoten 1 und 3 bewirkt, dass zwei Aktionen der gleichen Fähigkeit `CAbility3` angelegt werden. Alle Instanzen, die aus einem Aufruf `Initialize()` eines Fähigkeitenknotens resultieren, gehören zusammen, kooperieren und besitzen die gleiche Netz-ID (vgl. Abb. 3.15). Sie bilden zusammen ein Aktionsnetz. Die Netz-ID, welche die Zusammengehörigkeit der Aktionen repräsentiert, bekommt der Nutzer beim Aufruf der Funktion `Initialize()` zurückgegeben.

Eine Instanz einer Fähigkeit besitzt neben der Netz-ID auch eine Instanz-ID. Diese Instanz-ID ist fähigkeitenintern eindeutig, d. h. zwei Instanzen einer Fähigkeit haben immer unterschiedliche Instanz-IDs. In der derzeitigen Implementierung wird in der jeweils ersten Instanz einer Fähigkeit innerhalb eines Aktionsnetzes die Instanz-ID gleich der Netz-ID gesetzt. Bei jeder weiteren Instanz wird die Instanz-ID um 1 inkrementiert (vgl. Aktion 3 und 4 in Abb. 3.15).

Der Nutzer eines Aktionsnetzes kann dieses durch weitere Steuerbefehle beeinflussen. Dabei muss er die zugehörige Netz-ID angeben. Im Beispiel ruft der Nutzer die Funktion `SetParameter()` auf und sendet so dem Experten 1 Parameter für die bei ihm ablaufende Aktion 1 (vgl. Abb. 3.16). Es ist nun die Aufgabe des Experten 1, evtl. unter Zuhilfenahme und Auswertung der empfangenen Parameter, die Parameter für die Teilaktionen 2 - 4 zu bestimmen und diese über die Fähigkeitenknoten an die jeweiligen Experten der Teilaktionen zu senden.

Bei den Steueraufrufen `Start()`, `Stop()` und `Delete()` ist der gleiche Automatismus wirksam wie bei `Initialize()`: Ruft der Nutzer beim Fähigkeitenknoten 1 eine dieser Funktionen für das Aktionsnetz unter Angabe der Netz-ID auf, so wird durch einen rekursiven Aufruf der gleichnamigen Funktion bei den Fähigkeitenknoten 2 und 3 dieses Kommando parallel an alle am Aktionsnetz beteiligten Experten gesandt. Auf diese Weise besitzt der Nutzer die Möglichkeit, durch ein Start- oder Stop-Kommando dem ganzen Aktionsnetz die Erlaubnis zu erteilen, auf begrenzte Ressourcen zuzugreifen bzw. diesem den Zugriff wieder zu entziehen (vgl. Abschnitt 3.3.2). Welche Teilaktionen wo im verteilten System ablaufen, ist für den Nutzer transparent und belanglos. Zum Schluss kann der Nutzer das Aktionsnetz mit dem Aufruf von `Delete()` beim obersten Fähigkeitenknoten löschen (vgl. Abb. 3.16).

Jeder Prozess im System – nicht nur der Nutzer – besitzt die Möglichkeit, den Status und die relativen und absoluten Fortschrittsmaße einer Aktion abzufragen. Diese Abfrage geschieht beim zugehörigen Fähigkeitenknoten durch die Aufrufe `GetInstanceStatus()`, `GetInstanceProgress()` bzw. `GetInstanceAbsMeasure()`, wobei die Instanz-ID der Aktion anzugeben ist. Diese Informationen werden in den Experten durch den Prozessrahmen (`CClientScene`) automatisch von den Aktionen in die Fähigkeitenknoten übertragen.

### 3.3.8 Implementierte Fähigkeiten

Tab. 3.1 zeigt eine tabellarische Zusammenstellung der schematischen Fähigkeiten mit ihren Parametern, so wie sie derzeit im EMS-VISION System implementiert und im Fähigkeitsnetz repräsentiert sind<sup>17</sup>. Die Fähigkeiten zur Fortbewegung wurden von SIEDERSBERGER [2003] und die Fähigkeit zur Wahrnehmung von HOFMANN [2003] implementiert und beschrieben. Die Fertigkeiten und schematischen Fähigkeiten zur Blickrichtungssteuerung sind Thema der vorliegenden Arbeit (vgl. Kapitel 4).

<sup>17</sup>Neben den hier genannten Fähigkeiten gibt es im System noch weitere Funktionalitäten, die bisher nicht an das Fähigkeitenkonzept angepasst und im Fähigkeitsnetz repräsentiert wurden (z. B. Straßenerkennung).

Name der Fähigkeit	Beschreibung	Parameter
<i>Fähigkeiten zur Fortbewegung (nach [SIEDERSBERGER, 2003]):</i>		
Stand	<i>Stehen</i>	<i>keine Parameter</i>
Stop	<i>Anhalten in x Meter</i>	<i>Distanz x, Sollverzögerung, relevantes Objekt für die Querführung, Sollablage</i>
StopInFrontOf	<i>Anhalten vor Objekt</i>	<i>relevantes Objekt für die Längsführung, Sollabstand zum Objekt nach dem Anhalten, Sollverzögerung</i>
EmergencyStop	<i>Nothalt</i>	<i>keine Parameter</i>
FollowLane	<i>Spurfolgen</i>	<i>Sollgeschwindigkeit, Spur, Sollablage</i>
LeaveLane	<i>Spur verlassen</i>	<i>Abstandsintervall in dem Verlassen erlaubt ist, Spur</i>
EnterLane	<i>in Spur einbiegen.</i>	<i>relevantes Objekt für die Querführung, (Ziel-)Spur, Sollablage</i>
TurnOff	<i>Abbiegen</i>	<i>relevantes Objekt für die Querführung, (Ziel-)Spur, Sollablage</i>
CrossOver	<i>gerade über die Kreuzung fahren</i>	<i>relevantes Objekt für die Querführung, (Ziel-)Spur, Sollablage</i>
AvoidObstacle	<i>einem Hindernis ausweichen</i>	<i>relevantes Objekt für die Querführung (Hindernis), Vorschlag für die Richtung des Ausweichens (unter Berücksichtigung des Situationskontextes), minimaler Abstand zum Hindernis</i>
Stop&Go	<i>Staufahren</i>	<i>relevantes Objekt für die Querführung (Spur), relevantes Objekt für die Längsführung (Fahrzeug), Zeitabstand</i>
<i>Fähigkeiten zur Wahrnehmung (nach [HOFMANN, 2003]):</i>		
PNO	<i>Erkennung negativer Hindernisse</i>	<i>Modus der Wahrnehmung: ganzer Graben, linker Rand, rechter Rand, linker und rechter Rand, Grabensegment vor Fahrzeug</i>
<i>Fähigkeiten zur Blickrichtungssteuerung (vgl. Kapitel 4.7):</i>		
OVB	<i>Optimierung des Blickverhaltens (OVB) durch Planung von Sakkaden und Blickverfolgungen</i>	<i>maximale Anzahl von Sakkaden, Planungshorizont, evtl. konst. Nickwinkel oder Gierwinkel, Möglichkeit des Anstoßens einer Neuplanung der Blickrichtung, Aufmerksamkeitsbedarf<sup>18</sup> und Kombinationen von ROA<sup>18</sup> der Szenenobjekte</i>
3D-Search	<i>Suche in einem 3D-Bereich</i>	<i>Spezifikation des 3D-Bereichs</i>

Tab. 3.1: Die schematischen Fähigkeiten des EMS-VISION Systems<sup>17</sup>

<sup>18</sup> Diese Größen werden über die zugehörigen Objektknoten im Szenenbaum kommuniziert.

## 3.4 Unscharfe Regelbasis

### 3.4.1 Grundlegendes zu unscharfen Regeln

Die in diesem Kapitel beschriebene Regelbasis bildet den Kern der Verhaltensentscheidung und formuliert das vom autonomen Fahrzeug zu zeigende Verhalten. Die Regelbasis besteht aus einer Menge von unscharfen WENN-DANN-Regeln (vgl. Abschnitt 2.3.3). In den Prämissen (WENN-Teilen) der Regeln werden linguistische (unscharfe) Aussagen formuliert und durch unscharfe UND- und ODER-Operatoren miteinander verknüpft. Diese Verknüpfungen umschreiben Situationen mehr oder weniger detailliert. In den Konklusionen (DANN-Teilen) der Regeln sind situationsgerechte Fähigkeiten und Angaben zu ihrer Parametrisierung niedergelegt. Eine Regel ordnet einer Situationsumschreibung ein situationsgerechtes Verhalten zu (Situations-Aktions-Paar) (vgl. Abb. 3.17, links).

Die in der Konklusion einer Regel niedergelegten Fähigkeiten sind priorisiert. Die alternativen Fähigkeiten mit niedriger Priorität kommen zum Einsatz, wenn die höher priorisierten Fähigkeiten zur Zeit im System nicht verfügbar sind. Dabei ist zu betonen, dass die alternativen Fähigkeiten nicht zur Auflösung der sich aus der Situation ergebenden Konflikte dienen, da die Konflikte durch die iterative Ausführung der Situationsanalyse und die wiederholte Anwendung der Regelbasis aufgedeckt und berücksichtigt werden (vgl. Abschnitt 3.2.4). Vielmehr dienen die alternativen Fähigkeiten als Rückfallposition bei nicht vollständiger oder fehlerhafter Funktionalität im System. Die Gründe dafür können innerhalb des autonomen Systems (z. B. Funktionsausfall bei Systemkomponenten) oder außerhalb des Systems (z. B. schlechte Sichtverhältnisse) liegen. Durch die alternativen Fähigkeiten ist bei einem plötzlichen Eintreten solcher Ereignisse auch ohne erneutes Durchlaufen der Verhaltensentscheidung ein „Sollverhalten“ spezifiziert.

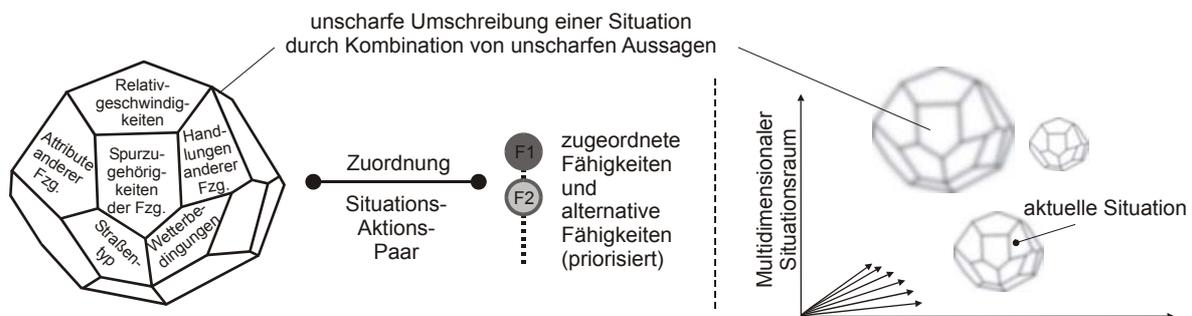


Abb. 3.17: Aufgabe der Regelbasis

Durch die verschiedenen Situationsaspekte stehen eine Vielzahl von situationsbeschreibenden Aussagen zur Verfügung. Die situationsbeschreibenden Aussagen spannen den sogenannten Situationsraum auf, dessen Dimension gleich der Anzahl der berücksichtigten Aussagen ist (vgl. Abb. 3.17, rechts). Die aktuelle Situation lässt sich als Punkt in diesem Situationsraum darstellen. Bei der Bildung von Situationsklassen wird der Situationsraum in Unterräume aufgeteilt [GRASHEY, 1999]. Eine Verhaltensmodellierung durch Klassifikation ordnet jeder Situationsklasse (d. h. jedem Unterraum im Situationsraum) ein gewisses Verhalten zu.

Der hier benutzte Ansatz mit unscharfen Regeln versucht nicht, den Situationsraum in Bereiche mit scharfen Grenzen zu unterteilen. Vielmehr wird ein Vergleich mit unscharfen Umschreibungen von Situationen durchgeführt, um passende Aussagen bezüglich der zu wählen-

den Fähigkeiten und ihrer Parametrisierung zu gewinnen. Dies geschieht durch die Anwendung der Regelbasis. Durch die Anwendung vergleicht man die (unvollständigen) Situationsumschreibungen der Regeln mit den tatsächlichen, in den Situationsaspekten berechneten linguistischen Werten (vgl. Abb. 3.18). Eine Umschreibung einer Situation trifft umso mehr zu, je erfüllter die zugehörige Regel ist.

SCHAFF [1998] benutzt folgende Metapher: Die Situationsaspekte umschreiben die aktuelle Situation wie die Seitenflächen eines Polyeders den Polyeder umschreiben. Diese Metapher wurde auch in Abb. 3.17 zur Veranschaulichung herangezogen, indem jede Regel durch einen (unscharfen) Polyeder symbolisiert ist. Anschaulich gesprochen ist eine Regel um so mehr wahr, je näher der zugehörige Polyeder bei der aktuellen Situation (= Punkt im Situationsraum) liegt. Es können auch mehrere Regeln gleichzeitig wahr sein. Im Situationsraum bedeutet dies, dass die Polyeder der Regeln ineinander verschachtelt sind.

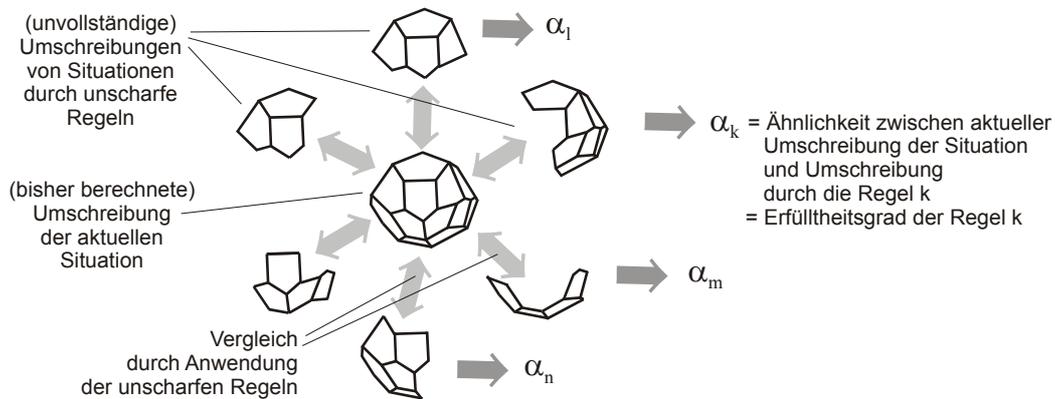


Abb. 3.18: Anwendung der Regelbasis

In Abb. 3.19 ist der Datenfluss vom Szenenbaum über die Situationsanalyse bis hin zu den unscharfen Regeln dargestellt. Der Szenenbaum liefert zyklisch Objektdaten. Eine Menge von Datenvorverarbeitungsmodulen verarbeitet die ankommenden Objektdaten weiter. Diese Verarbeitung kann eine Pufferung, Filterung, Mittelung oder sonstige signalverarbeitende Maßnahme beinhalten. Die Situationsaspekte greifen auf die Ergebnisse der Datenvorverarbeitungsmodulen zu, wobei die Ergebnisse eines Datenvorverarbeitungsmoduls allen Situationsaspekten zur Verfügung stehen. Die Situationsaspekte versuchen nun, wesentliche Zusammenhänge zwischen den Daten zu erkennen, um die Situation signifikant zu beschreiben.

Innerhalb eines Situationsaspekts  $SA_k$  werden die Zugehörigkeitsfunktionen  $\mu_{ki}$  dazu benutzt, um unter Berücksichtigung des (vorverarbeiteten) Zustandsvektors  $\underline{x}$  die Zugehörigkeitsgrade  $\Theta_{ki}$  zu den linguistischen Werten  $\sigma_i$  zu ermitteln (vgl. Abschnitt 3.2.2):

$$\Theta_{ki} = (SA_k, \mu_{ki}(\underline{x}))$$

Die Prämisse einer Regel verknüpft unscharfe Terme durch unscharfe UND- und ODER-Operatoren. Die Terme formulieren Bedingungen für die linguistischen Aussagen  $\Theta_{ki}$  der Situationsaspekte. Der Wahrheitswert der Terme liegt im Intervall  $[0,1]$ . Die Auswertung der Verknüpfung der Terme ergibt den Erfülltheitsgrad  $\alpha$  der Regel. Der Erfülltheitsgrad  $\alpha$  ist ein Maß für die Ähnlichkeit zwischen der durch die Prämisse der Regel formulierten Situationsumschreibung und der aktuellen Situation.

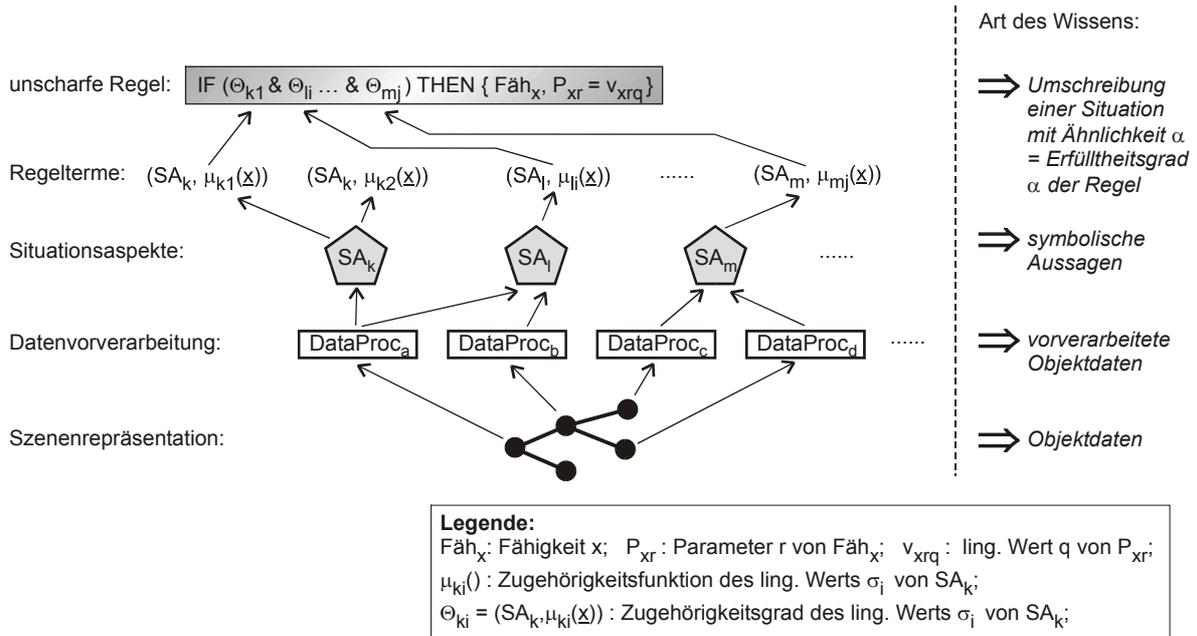


Abb. 3.19: Die Datenverarbeitung bis zur Regelbasis

Die Verknüpfung von Situation und Aktion durch unscharfe Regeln besitzt den Vorteil, dass die Ergebnisse für den menschlichen Betrachter transparent und nachvollziehbar sind. So kann der Entwickler im Falle einer Diskrepanz zwischen erwartetem und gezeigtem Verhalten durch Hinzufügen weiterer linguistischer Regeln das gewünschte Verhalten feiner modellieren, ohne dazu eine Codierung in einer Beschreibungssprache benutzen zu müssen. Dies ist ein Vorteil, der z. B. bei einer Verhaltensmodellierung durch neuronale Netze nicht gegeben ist.

Es ist nicht notwendig, dass die linguistischen Aussagen der berechneten Situationsaspekte exakt mit einer Regel übereinstimmen. Im allgemeinen tragen mehrere Regeln der Regelbasis zum endgültigen Ergebnis bei. Dies entspricht der menschlichen Logik, bei der ebenfalls nicht für jede Situation eine Regel angegeben werden kann. Die formulierten Regeln umschreiben nur typische Situationen und durch das unscharfe Schließen lassen sich approximierende Lösungen finden [STROHAL, 1999].

Nach diesen grundsätzlichen Bemerkungen bezüglich unscharfer Regeln beschreibt der Abschnitt 3.4.2 zwei Methoden, mit der die Verwaltung einer Regelbasis effizient gestaltet werden kann. Abschnitt 3.4.3 zeigt anhand eines Beispiels die sinnvolle Gestaltung der Regelbasis. Die realisierten und getesteten Regelbasen werden in Abschnitt 3.4.4 vorgestellt. Das programmtechnische Klassendesign der Regelbasis ist Thema des Anhangs C.

### 3.4.2 Die effiziente Verwaltung der Regelbasis

Mit zunehmender Anzahl von Regeln wird das Verhalten des Systems immer genauer beschrieben. Andererseits steigt damit die Anforderung an die Schnelligkeit der Auswertung der Regeln, wenn eine Verhaltensmodellierung in Echtzeit aufrechterhalten werden soll. Obwohl der Autor den Standpunkt vertritt, dass durch eine geeignete Gestaltung der Situationsaspekte die Zahl der notwendigen Regeln klein gehalten werden kann, seien hier zwei Möglichkeiten genannt, die den Berechnungsaufwand bei der Auswertung der Regelbasis reduzieren. Die erste Möglichkeit ist das dynamische Laden nur aktuell in Frage kommender Regeln und die

zweite ist die Nutzung *offline* berechneter und explizit repräsentierter Ähnlichkeiten zwischen den Regeln der Regelbasis.

### 3.4.2.1 Dynamisches Laden der Fallbasis

Nicht alle situationsbeschreibenden Aussagen sind unscharf und hängen vom Situationskontext ab. So ist zum Beispiel die Frage nach der Domäne (z. B. Autobahn oder Landstraße) nur mit ja oder nein zu beantworten. In Abhängigkeit von der Domäne kommen aber ganze Gruppen von Regeln für die Verhaltensentscheidung nicht in Betracht. So sind Regeln für Querfeldeinfahren auf Autobahnen nicht relevant und sollten sich nicht in der Regelbasis befinden. Ändert sich andererseits die Domäne (z. B. beim Ausfahren aus der Autobahn), so muss man ganze Gruppen von Regeln aus der Regelbasis löschen und andere laden. Durch das dynamische Laden ist die auszuwertende Regelbasis möglichst klein, was Speicherplatz und Rechenzeit einspart.

### 3.4.2.2 Explizite Repräsentation von Ähnlichkeit

Man kann die Reihenfolge, in der die Regeln angewandt werden, situationsabhängig verändern, indem man Ähnlichkeiten und Unähnlichkeiten zwischen den Regeln in Form von gewichteten Kanten in der Regelbasis mit abspeichert. Durch diese Kanten entsteht eine vernetzte Struktur (siehe Abb. 3.20, links). Diese Vorgehensweise, Such- und Rechenzeit auf Kosten von Speicherplatz zu sparen, wurde u. a. von SCHAAF [1998] für das fallbasierte Schließen vorgeschlagen. Dieses Vorgehen erscheint auch bei einer unscharfen Regelbasis vorteilhaft.

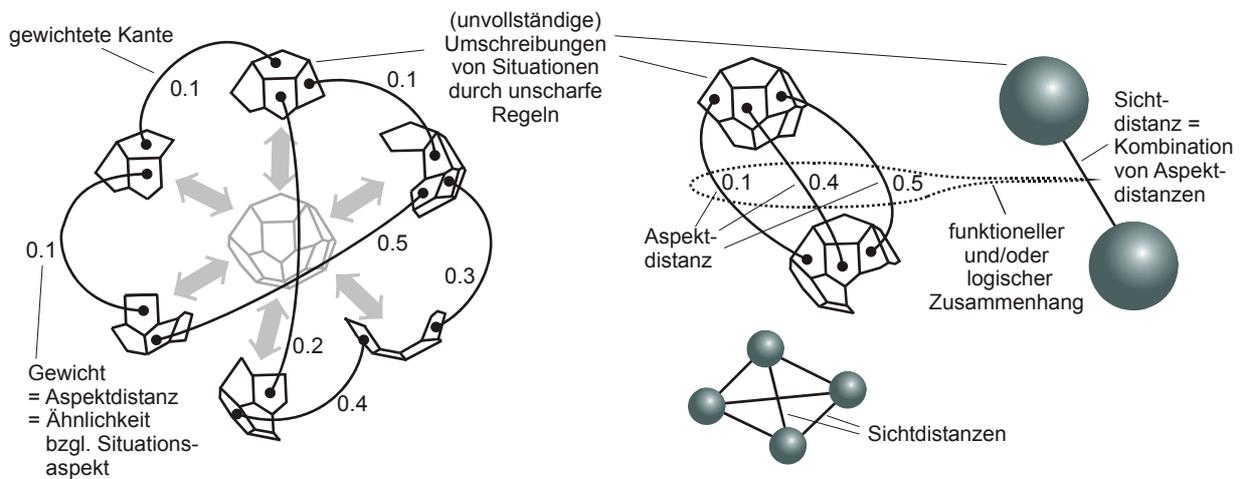


Abb. 3.20: Explizite Speicherung von Ähnlichkeiten zwischen den Regeln

Die Kanten verbinden die Terme verschiedener Regeln und verfügen über ein Gewicht, welches ein Maß für die Gleichheit bzw. Ungleichheit der darin berücksichtigten Werte der Situationsaspekte ist. Je ähnlicher die Werte sind, desto kleiner ist das Gewicht. Diese Ähnlichkeit betrifft die Regeln nicht als Ganzes, sondern nur hinsichtlich eines gewissen Situationsaspekts und ist nicht kontextabhängig. SCHAAF [1998] bezeichnet das Gewicht einer Kante als *Aspekt-distanz* und zwei Fälle als *benachbart*, wenn diese eine gemeinsame Kante besitzen. Beim Einfügen einer neuen Regel muss man die Aspekt-distanzen zwischen der neuen und den bereits existierenden Regeln nur einmalig berechnen und in der Fallbasis ablegen. Auf diese Weise werden die berechneten Ähnlichkeiten wiederverwendbar.

SCHAAF [1998] geht noch einen Schritt weiter und kombiniert mehrere, mit einer Gewichtung versehene Aspekt-distanzen zu einer *Sichtdistanz* (siehe Abb. 3.20, rechts). Die Sichtdis-

tanz beschreibt die Ähnlichkeit zweier Regeln nicht nur hinsichtlich eines Situationsaspekts, sondern hinsichtlich einer Menge von Situationsaspekten. Die Sichtdistanz ist kontextabhängig. Der Name „Sichtdistanz“ beruht auf der Tatsache, dass zwei Situationsbeschreibungen einmal ähnlich und ein andermal unähnlich sein können, je nachdem aus welcher Sicht man sie betrachtet. Zur Erläuterung kann man die zu Beginn dieses Kapitels benutzte Metapher, bei der die Situationsbeschreibungen durch Polyeder und die Situationsaspekte durch die Seiten der Polyeder symbolisiert werden, nochmals heranziehen:

*Betrachtet man mehrere Polyeder im Raum, so sind immer nur gewisse Flächen dieser Polyeder sichtbar. Bekommt ein Betrachter die Aufgabe gestellt, die Polyeder nach ihrer Ähnlichkeit zu gruppieren, so kann dieser beim Vergleich nur die sichtbaren Flächen heranziehen. Es ergeben sich daraus gewisse Gruppen und die Polyeder einer Gruppe werden hinsichtlich der sichtbaren Flächen ähnlich sein. Wechselt der Betrachter nun seinen Standort und betrachtet die gleichen Polyeder aus einer anderen Sicht, so sind zuvor verdeckte Flächen sichtbar und bisher sichtbare Flächen nicht mehr sichtbar. Bekommt der Betrachter nun die gleiche Gruppierungsaufgabe wie zuvor gestellt, so setzen sich die Gruppen anders zusammen.*

Die gespeicherten Kanten beschreiben sowohl Ähnlichkeiten als auch Unähnlichkeiten hinsichtlich gewisser Situationsaspekte. Fehlt eine Kante zwischen Situationsaspekten, so bedeutet dies lediglich, dass über die Ähnlichkeit hinsichtlich dieses Aspekts kein explizites Wissen vorhanden ist. Nicht repräsentierte Ähnlichkeit ist dabei nicht mit Unähnlichkeit gleichzusetzen, da man ansonst von einer vollständigen Wissensrepräsentation ausginge (*closed world assumption*). Kriterien für das Speichern von Nachbarschaften werden von SCHAAF [1998] diskutiert.

Ergab die Anwendung einer unscharfen Regel einen sehr kleinen Erfülltheitsgrad, so sollten Regeln mit einer großen Ähnlichkeit zu dieser bei der weiteren Auswertung zunächst zurückgestellt werden. Im entgegengesetzten Fall, d. h. bei einem großen Erfülltheitsgrad, sollte man die ähnlichen Regeln bevorzugt auswerten. Aus dieser Vorgehensweise entwickelte SCHAAF [1998] einen Suchalgorithmus mit dem Namen „*Fischen und Versenken*“. Ein Vorteil dieses Algorithmus ist es, dass die Auswertung der Regelbasis jederzeit unterbrochen werden kann und dabei immer sichergestellt ist, dass die zur Anfrage ähnlichsten Regeln bereits ausgewertet wurden. Auf diese Weise kann man die bereits ausgewerteten Regeln berücksichtigen, während man die nächstähnlichsten Regeln weiter prüft.

Das in dieser Arbeit vorgestellte Klassendesign für die Situationsanalyse und Verhaltensentscheidung ist derart gestaltet, dass eine situationsbeschreibende Aussage oder eine Aktion immer relativ zu einem Subjekt in der Szene berechnet und gespeichert werden kann. Dies ist erforderlich, um die wahrscheinlichen Aktionen der anderen Subjekte in der Szene bei der Prädiktion der Situation und bei der Verhaltensentscheidung berücksichtigen zu können (vgl. Kapitel 3.1). Das Eigenfahrzeug ist zwar das herausragendste und am exaktesten beschriebene Subjekt in der Szene, aber nur eines unter mehreren. Durch die Anwendung der Regelbasis auf alle Subjekte in der Szene wird jedem Subjekt eine (vermutete) Aktion zugeordnet. Die nun folgenden Unterabschnitte beschäftigen sich mit Realisierungsaspekten der Regelbasis. Die dabei verwendeten Basisklassen sind in Anhang C dargestellt.

### 3.4.3 Die sinnvolle Gestaltung der Regelbasis

Es muss das Ziel sein, das gewünschte Verhalten eines autonomen Fahrzeugs mit möglichst wenigen, prägnanten Regeln zu modellieren. Eine geringe Anzahl von Regeln mit großer Aussagekraft dient der Übersichtlichkeit und Handhabbarkeit der Regelbasis und spart Speicherplatz und Rechenzeit. Bei der Formulierung der Verhaltensregeln zeigen sich die Vorteile des Fuzzy-Ansatzes: Es lassen sich leichtverständliche Vorschriften in Form von verbalen Regeln aufstellen.

Nach KOPF [1993] ist es sinnvoll, die Situationsbewertung im Hinblick auf Begriffe wie Behinderung und Gefährdung (der eigenen und von anderen) und hinsichtlich der Beachtung von Verkehrsregeln durchzuführen. Für die Beachtung von Verkehrsregeln bieten sich logische Funktionen an, wogegen für die Gefährdung und Behinderung von Kopf [1993] kontinuierliche, mathematische Modelle auf der Basis von Zeitreserven entwickelt wurden. Diese Maße wurden bereits in Abschnitt 3.2.3 vorgestellt.

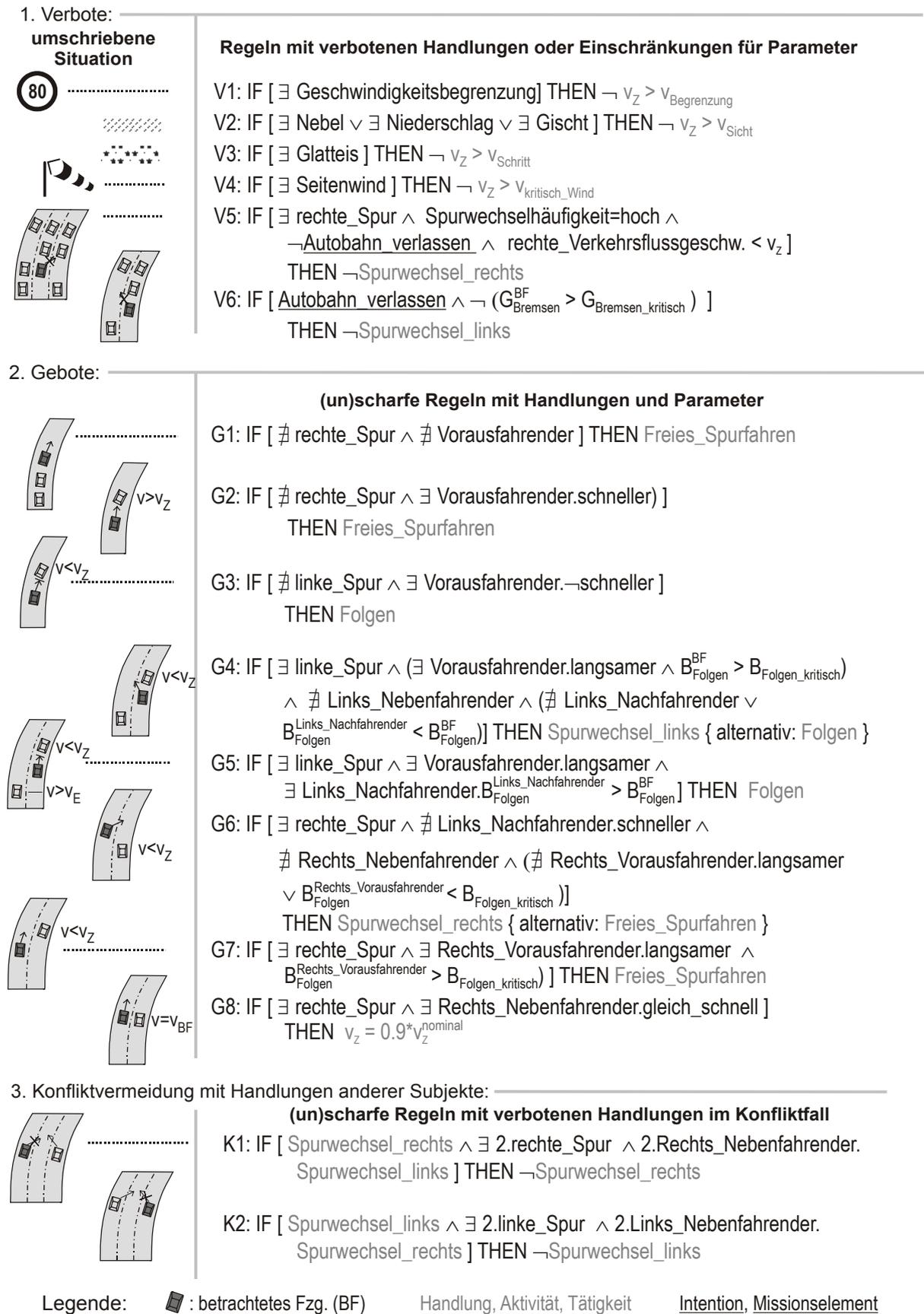
Anhand der Regelbasis für das Autobahnfahren in Abb. 3.21 wird nun die sinnvolle Gestaltung einer Regelbasis diskutiert. Zunächst ist es vorteilhaft, die Regeln nach der Art ihrer Konklusionen in Verbote, Gebote und Konfliktvermeidungen zu ordnen und sie in dieser Reihenfolge auch auszuwerten. Verbote beschreiben keine komplette Fähigkeitsauswahl, sondern spezifizieren lediglich Zwangsbedingungen an die Parameter von Fähigkeiten oder verbieten bestimmte Fähigkeiten (negative Auswahl). So kann z. B. eine (wahrgenommene) Geschwindigkeitsbegrenzung die Zielgeschwindigkeit  $v_z$  beschränken (vgl. Verbot V1 in Abb. 3.21). In analoger Weise können Sichtbehinderungen, Glatteis oder Seitenwind dazu führen, dass  $v_z$  einen bestimmten Schwellwert nicht überschreiten darf (V2 bis V4). Verbote müssen vor den Geboten ausgeführt werden, da sie die durch die Gebote getroffene Fähigkeitsauswahl und deren Parametrisierung einschränken.

Um den gesamten Situationsraum durch Regeln abzudecken und die Anzahl der dazu notwendigen Regeln zu beschränken, sind zunächst möglichst allgemeingültige Regeln aufzustellen. Ausnahmen von diesen allgemeingültigen Regeln können dann in effizienter Weise durch Verbote oder weitere, konkreter beschreibende Regeln formuliert werden. Wie Abb. 3.21 zeigt, unterdrückt das Verbot V5 einen Spurwechsel auf die rechten Spur bei hoher Spurwechselhäufigkeit und geringer Verkehrsflussgeschwindigkeit<sup>19</sup>. Das Verbot V6 unterdrückt einen Spurwechsel nach links, wenn die Autobahn verlassen werden soll und die Gefährdung<sup>20</sup> durch das damit verbundene Bremsen und Folgen einen Schwellwert nicht überschreitet.

---

<sup>19</sup> Die Syntax des letzten Terms in V5 (Abb. 3.21) beschreibt die unscharfe Aussage „Die Verkehrsflussgeschwindigkeit rechts ist kleiner als die Zielgeschwindigkeit“. Die Zugehörigkeitsfunktion dieser unscharfen Menge ist abhängig von der momentanen Zielgeschwindigkeit  $v_z$ .

<sup>20</sup> Das Gefährdungsmaß  $G_{\text{Bremsen}}$  hinsichtlich der gefährdungsvermindernden Handlung *Bremsen* wird in Abschnitt 3.2.3 vorgestellt.



Legende:  : betrachtetes Fzg. (BF)      Handlung, Aktivität, Tätigkeit      Intention, Missionselement

Abb. 3.21: Beispiel einer Regelbasis für Autobahnfahren (Syntaxbeschreibung siehe Text)<sup>21</sup>

<sup>21</sup> Diese Regelbasis wurde im Projektrahmen nicht implementiert. Sie erhebt daher keinen Anspruch auf Vollständigkeit sondern soll lediglich die prinzipielle Gestaltung einer komplexen Regelbasis illustrieren.

Wie oben dargestellt wurde, ist es das Ziel, durch möglichst allgemeingültige Regeln einen großen Bereich im Situationsraum abzudecken. So ist die durch das Gebot G1 in Abb. 3.21 umschriebene Situation sehr allgemein gehalten: Wenn keine rechte Spur und kein vorausfahrendes Fahrzeug existiert<sup>22</sup> (beides relativ zum betrachteten Subjekt), soll ein freies Spurfahren erfolgen. Freies Spurfahren soll auch dann erfolgen, wenn ein Vorausfahrender<sup>23</sup> vorhanden ist, dieser jedoch schneller als das betrachtete Fahrzeug fährt (G2)<sup>24</sup>.

Das Gebot G3 sagt aus, dass bei dem Fehlen einer linken Spur einem nicht schneller fahrenden Vorausfahrenden<sup>23</sup> gefolgt werden soll. Die Gebote G1 bis G3 sind sowohl bei ein- als auch bei mehrspurigen Autobahnen relevant. Die jetzt folgenden Gebote G4 bis G8 kommen dagegen nur dann zur Anwendung, wenn die Anzahl der Fahrspuren mindestens 2 ist. Sie umschreiben also schon etwas detailliertere Situationen.

Das Gebot G4 umschreibt eine Situation, in der ein Spurwechsel erfolgen soll. In dieser Situation muss – relativ zum betrachteten Fahrzeug – eine linke Spur und ein Vorausfahrender<sup>23</sup> existieren. Zusätzlich muss der Vorausfahrende das betrachtete Fahrzeug zu einem gewissen Grad – spezifiziert durch den Schwellwert  $B_{\text{Folgen\_kritisch}}$  – behindern<sup>25</sup>. Ein Spurwechsel darf nur erfolgen, falls es keinen Links\_Nebenfahrenden<sup>23</sup> gibt und entweder kein Links\_Nachfahrender<sup>23</sup> existiert, oder dieser durch einen Spurwechsel weniger behindert wird, als das betrachtete Fahrzeug (BF) durch ein Verbleiben auf der Spur und Folgen. Man beachte, dass ein Spurwechsel in dieser Situation unterdrückt wird, wenn die Prämisse des Verbots V6 zutrifft. Steht in der durch G4 beschriebenen Situation die Fähigkeit Spurwechsel\_links nicht zur Verfügung, so ist die alternative Handlung Folgen auszuführen (vgl. Konklusion von G4).

Das Gebot G5 beschreibt den entgegengesetzten Fall zu G4: Es existiert ein Links\_Nachfahrender<sup>23</sup> und dieser würde durch ein Ausscheren des betrachteten Fahrzeugs stärker behindert als das betrachtete Fahrzeug durch ein Folgen. Deshalb soll das betrachtete Fahrzeug in der Spur verbleiben und dem Vorausfahrenden<sup>23</sup> folgen. Die Gebote G6 und G7 regeln das Wiedereinscheren in die rechte Spur. Das Gebot G8 behandelt die besondere Situation, in welcher das betrachtete Subjekt einen Überholvorgang nicht abschließen kann, weil der Rechts\_Nebenfahrende<sup>23</sup> mit der gleichen Geschwindigkeit fährt wie er. In dieser Situation soll das betrachtete Subjekt seine Zielgeschwindigkeit um 10 % reduzieren, um eine Rückkehr auf die rechte Spur zu ermöglichen.

Die bisher beschriebenen Regeln können auf alle Fahrzeuge angewandt werden, um damit deren Verhalten abzuschätzen. Daneben ist es möglich, dass zwischen den Aktionen verschiedener Fahrzeuge Konflikte auftreten. Zwei solcher Konfliktsituationen beschreiben die beiden Regeln K1 und K2 in Abb. 3.21: Will das betrachtete Fahrzeug einen Spurwechsel nach rechts/links durchführen und existiert eine übernächste rechte/linke Spur auf der ein Fahrzeug auf gleicher Höhe fährt und ebenfalls einen Spurwechsel nach links/rechts durchführt oder durchführen will, so ist der Spurwechsel zu unterdrücken. Es ist sinnvoll, die Re-

<sup>22</sup> Der Wahrheitsgehalt dieses Terms wird dadurch ermittelt, dass zunächst das Fahrzeug gesucht wird, für welches die Bedeutung Vorausfahrender<sup>23</sup> den größten Zugehörigkeitsgrad  $\delta$  besitzt. Der Wahrheitsgehalt des Terms „es existiert kein Vorausfahrender“ ist dann  $1-\delta$ .

<sup>23</sup> Diese Bedeutung eines Fahrzeugs wird durch den Situationsaspekt Relativlage ermittelt (vgl. Abb. 3.4).

<sup>24</sup> Die Syntax  $\exists$  Vorausfahrender.schneller beschreibt die Verknüpfung der linguistischen Aussagen „es existiert ein Vorausfahrender“ und „Fahrzeug fährt schneller als das referenzierte Fahrzeug“ durch einen unscharfen UND-Operator. Die zweite Aussage wird durch den Situationsaspekt Relativgeschwindigkeit ermittelt. Die Zugehörigkeitsfunktion dieses Situationsaspekts ist u. a. von der aktuellen Geschwindigkeit des referenzierten Fahrzeugs abhängig.

<sup>25</sup>  $B_{\text{Folgen}}$  als Maß für die Behinderung bei einem erzwungenen Folgen wird in Abschnitt 3.2.3 definiert.

geln zur Konfliktvermeidung nur auf das autonome Eigenfahrzeug und nicht auf die Fremdfahrzeuge anzuwenden. Würde man in den Situationen von K1 und K2 die Konfliktvermeidung auf beide Fahrzeuge anwenden, so wäre der Spurwechsel für beide Fahrzeuge unterdrückt und keines der Fahrzeuge würde auf die mittlere Spur wechseln, was in letzter Konsequenz einen Spurwechsel für das Eigenfahrzeug erlaubt.

In den Regeln der Regelbasis von Abb. 3.21 werden verschiedene Schwellwerte benutzt. So stellt  $G_{\text{Bremsen\_kritisch}}$  die Schwelle dar, ab der zugunsten einer gefährdungsvermindernden Reaktion (= Spurwechsel) das nächste Missionselement Autobahn verlassen nicht mehr berücksichtigt wird.  $B_{\text{Folgen\_kritisch}}$  beschreibt die gerade noch akzeptierte Behinderung, bei der auf einen Spurwechsel verzichtet und dem Vorausfahrenden<sup>23</sup> gefolgt wird. Es ist denkbar, auch diese Schwellen durch entsprechende Regeln anzupassen. Solche (Meta-)Regeln zur Schwellwertanpassung sind noch vor den Verboten und Geboten auszuwerten, so dass sie von diesen berücksichtigt werden können. Hier seien zwei Beispiele genannt, in denen eine solche Anpassung zweckmäßig erscheint: Ein Fahrzeug, welches bei einer Autobahneinfahrt auf dem Beschleunigungsstreifen mit beschränkter Länge fährt und auf die rechte Fahrspur wechseln möchte, wird einem auf der rechten Fahrspur Nachfahrenden<sup>23</sup> eine größere Behinderung  $B_{\text{Folgen}}$  zumuten, als bei einem „normalen“ Spurwechsel zwischen zwei Fahrspuren. Ferner ist denkbar,  $B_{\text{Folgen\_kritisch}}$  zugunsten eines schnelleren Vorankommens zu reduzieren, wenn dies von der Missionsplanung gewünscht wird.

Die Regelbasis von Abb. 3.21 beschreibt die Verhaltensregeln von Subjekten des Typs PKW. Neben PKWs gibt es noch andere Arten von Subjekten (z. B. LKWs, Motorräder oder Fußgänger), die sich etwas anders verhalten. Um das Verhalten solcher Subjekte abzuschätzen und zu berücksichtigen, benötigt man für jeden Subjekttyp eine spezielle Regelbasis, welche in analoger Art und Weise gestaltet sein kann.

Die in Abb. 3.21 gezeigte Regelbasis für Autobahnfahren konnte im Rahmen dieser Arbeit nicht vollständig entwickelt und getestet werden. Nichtsdestotrotz ist ihre Darstellung wichtig, um die Funktionsweise des vorgestellten Konzepts zur Situationsanalyse und Verhaltensentscheidung zu verstehen und deren Vorteile zu erkennen. Den Nachweis, dass komplexes Verkehrsgeschehen mittels solcher unscharfer Regelbasen modelliert werden kann, führt REKERSBRINK [1994] anhand von Simulationen.

#### 3.4.4 Implementierte Regelbasen

Die im Rahmen dieser Arbeit implementierten Regelbasen beschreiben das Verhalten für die Domänen „*Fahren auf Straßen niederer Ordnung*“, *Querfeldeinfahren* und *Stop&Go-Fahren* auf Autobahnen. Die in der Abb. 3.22 gezeigten Regeln werden nicht gleichzeitig ausgewertet, sondern je nach Domäne ist der entsprechende Teil in der Regelbasis geladen. Beim Übergang zwischen zwei Domänen – z. B. beim Verlassen der Straße oder beim Wiederauffahren auf die Straße – müssen sich die Regeln von zwei Domänen in der Regelbasis befinden.

Die Regelbasis wertet u. a. den Missionsplan aus. Der Missionsplan gibt z. B. vor, welcher Straße gefolgt oder in welche Straßen abgelenkt werden soll und spezifiziert den Startort und den Zielort in einer digitalen Karte. Der Missionsplan besteht aus einer Liste von Missionselementen, deren erster Eintrag das aktuelle Missionselement ist. Jedes Missionselement beinhaltet eine zeitliche und/oder räumliche Entfernung zum nächsten Missionselement. An der zeitlich/räumlichen Entfernung zwischen aktuellem und nächstem Missionselement kann man den Missionsfortschritt ablesen. Der Missionsplan und die Entfernungsmaße zwischen den

Missionselementen werden von dem von GREGOR [2002] realisierten Modul zur Missionsplanung und -überwachung generiert und aktualisiert. Die zeitlich/räumlichen Entfernungsmaße werden in der Verhaltensentscheidung dazu benutzt, um die für die Abarbeitung der nächsten und übernächsten Missionselemente notwendigen Handlungen ausreichend früh zu initiieren. Bei der Aktivierung der Fähigkeiten spielt es eine Rolle, wie nah (sowohl zeitlich als auch räumlich) das nächste Missionselement ist.

	G1: IF [ ( <u>kein Missionselement</u> $\vee$ <u>Halt</u> ) $\wedge$ Fahrzeug_bewegt_sich ] THEN 1: Anhalten, 2: Stehen <sub>wenn 1 aktiv</sub>
	G2: IF [ ( <u>kein Missionselement</u> $\vee$ <u>Halt</u> ) $\wedge$ Fahrzeug_steht ] THEN 1: Stehen
	G3: IF [ <u>Straße folgen</u> $\wedge$ $\nexists$ Hindernis ] THEN 1: Spurfahren, { alternativ: 2: Anhalten, 3: Stehen <sub>wenn 2 aktiv</sub> }
	G4: IF [ <u>Straße folgen</u> $\wedge$ $\exists$ Hindernis ] THEN 1: Spurfahren, 2: Ausweichen, { alternativ: 3: Anhalten_vor_Objekt, 4: Stehen <sub>wenn 3 aktiv</sub> }
	G5: IF [ <u>Straße folgen</u> $\wedge$ (dann) <u>Stehen</u> $\wedge$ $\nexists$ Hindernis ] THEN 1: Spurfahren, 2: Anhalten, 3: Stehen <sub>wenn 2 aktiv</sub> }
	G6: IF [ <u>Straße folgen</u> $\wedge$ (dann) <u>Stehen</u> $\wedge$ $\exists$ Hindernis ] THEN 1: Spurfahren, 2: Anhalten_vor_Objekt, 3: Stehen <sub>wenn 2 aktiv</sub> }
	G7: IF [ <u>Straße folgen</u> $\wedge$ (dann) <u>in Straße einbiegen</u> ] THEN 1: Spurfahren, 2: Abbiegen
	G8: IF [ <u>in Straße einbiegen</u> $\wedge$ (dann) <u>Straße folgen</u> ] THEN 1: Abbiegen, 2: Spurfahren
	G9: IF [ <u>Wegpunktnavigation</u> $\wedge$ $\nexists$ Hindernis ] THEN 1: Wegpunkte_anfahren, { alternativ: 2: Anhalten, 3: Stehen <sub>wenn 2 aktiv</sub> }, 4: Hindernisse wahrnehmen
	G10: IF [ <u>Wegpunktnavigation</u> $\wedge$ $\exists$ Hindernis ] THEN 1: Wegpunkte_anfahren, 2: Ausweichen, { alternativ: 3: Anhalten_vor_Objekt, 4: Stehen <sub>wenn 3 aktiv</sub> }, 5: Hindernisse_wahrnehmen
	G11: IF [ <u>Wegpunktnavigation</u> $\wedge$ (dann) <u>Stehen</u> $\wedge$ $\nexists$ Hindernis ] THEN 1: Wegpunkte_anfahren, 2: Anhalten_vor_Objekt (Wegpkt.), 3: Stehen <sub>wenn 2 aktiv</sub> , 4: Hindernisse_wahrnehmen
	G12: IF [ <u>Wegpunktnavigation</u> $\wedge$ (dann) <u>Stehen</u> $\wedge$ $\exists$ Hindernis ] THEN 1: Wegpunkte_anfahren, 2: Anhalten_vor_Objekt (Hind.), 3: Stehen <sub>wenn 2 aktiv</sub> , 4: Hindernisse_wahrnehmen
	G13: IF [ <u>Wegpunktnavigation</u> $\wedge$ (dann) <u>auf Straße auffahren</u> ] THEN 1: Wegpunkte_anfahren, 2: in_Spur_einbiegen, 3: Hindernisse_wahrnehmen
 Domänenwechsel	G14: IF [ <u>auf Straße auffahren</u> $\wedge$ (dann) <u>Straße folgen</u> ] THEN 1: in_Spur_einbiegen, 2: Spurfahren
	G15: IF [ <u>Straße folgen</u> $\wedge$ (dann) <u>Straße verlassen</u> ] THEN 1: Spurfahren, 2: Spur_verlassen, { alternativ: 2: Anhalten, 3: Stehen <sub>wenn 2 aktiv</sub> }
	G16: IF [ <u>Straße verlassen</u> $\wedge$ (dann) <u>Wegpunktnavigation</u> ] THEN 1: Spurfahren, 2: Wegpunkte_anfahren, 3: Hindernisse_wahrnehmen
	G17: IF [ <u>Straße folgen</u> $\wedge$ Verkehrsdichte = gross $\wedge$ Geschwindigkeit < 30 km/h $\wedge$ Geschwindigkeit < v <sub>z</sub> ] THEN 1: Spurfahren, 2: Stop&Go
	Legende: Handlung, Aktivität, Tätigkeit <u>Intention</u> , <u>Missionselement</u>

Abb. 3.22: Implementierte Regelbasis für „Fahren auf Straßen niederer Ordnung“, Querfeldeinfahren und Stop&Go-Fahren

Die Terme der Regeln sind scharfe und unscharfe Bedingungen. Diese Bedingungen können an Missionselemente, Aktionen oder an linguistische Werte von Situationsaspekten geknüpft sein. So treffen die Regeln G1 und G2 in Abb. 3.22 nur zu, wenn kein Missionselement vorhanden ist, oder das aktuelle Missionselement Halt ist. Wenn sich in diesem Fall das

Fahrzeug noch bewegt, wird durch G1 eine Instanz der Fähigkeit Anhalten gestartet. An die Benutzung der zweiten Fähigkeit Stehen von G1 ist eine Bedingung geknüpft: Erst wenn der Operationsteil der ersten Handlung Anhalten in den Zustand RUN wechselt, d. h. der Bremsvorgang begonnen hat, wird das Stehen gestartet (vgl. auch [SIEDERSBERGER, 2003]). Steht das Fahrzeug, so startet Regel G2 sofort und ausschließlich die Handlung Stehen.

Ist das aktuelle Missionselement Straße folgen und kein Hindernis vorhanden, so soll das Fahrzeug die Fähigkeit Spurfahren ausführen (Regel G3). Versagt das Spurfahren oder steht diese Fähigkeit aktuell nicht zur Verfügung, so wird als Alternative zuerst das Anhalten und dann das Stehen gestartet. Taucht beim Spurfahren ein Hindernis auf, so initiiert Regel G4 eine Instanz der Fähigkeit Ausweichen. Auch hier ist eine Alternative spezifiziert: Steht die Fähigkeit Ausweichen aktuell nicht zur Verfügung oder scheitert das Ausweichen, so erfolgt ein Anhalten vor dem Objekt. Regel G5 behandelt das Anhalten nach dem Fahren einer vorgegebenen Strecke. Regel G6 legt das Verhalten fest, wenn während des Anhaltens ein Hindernis auftaucht: In diesem Fall erfolgt kein Ausweichen, sondern ein Anhalten vor dem Objekt.

Die Regel G7 verlangt als aktuelles Missionselement Straße folgen und als nächstes Missionselement in Straße einbiegen. Ist die Entfernung vom aktuellen zum nächsten Missionselement noch ausreichend groß, so hat die Regel G3 den größten Erfülltheitsgrad und es wird hinsichtlich der Fortbewegung ausschließlich die Fähigkeit Spurfahren ausgeführt (vgl. G3 in Abb. 3.22). Ist das autonome Fahrzeug dem Abzweig (und damit dem nächsten Missionselement) ausreichend nahe, so startet die dann gültige Regel G7 zusätzlich die Fortbewegungsfähigkeit Abbiegen. Beim Erreichen des Abzweigs ist kurzzeitig in Straße einbiegen das aktuelle Missionselement und Regel G8 gültig (vgl. Kapitel 6.2). Die Regel G8 führt die Handlung Abbiegen weiter aus und bereitet das Spurfahren (auf der neuen Straße) vor. Wann genau die Übergänge in den operativen Teilen der Handlungen vollzogen werden, entscheidet der Experte für die Fortbewegung (vgl. [SIEDERSBERGER, 2003] und Kapitel 3.5.1).

Die Regeln für das Querfeldeinfahren G9 bis G12 sind analog zu den Regeln G3 bis G6 aufgebaut: An die Stelle der Handlung Spurfahren tritt die Handlung Wegpunkte\_anfahren. Auch hier weicht das Fahrzeug Hindernissen aus, falls das Fortbewegungsziel noch nicht erreicht ist. Zusätzlich ist die Wahrnehmungsfähigkeit Hindernisse\_wahrnehmen aktiv<sup>26</sup>. Die Regeln G13 + G14 und G15 + G16 behandeln das Auffahren auf eine Straße bzw. das Verlassen einer Straße. Sie sind die Bindeglieder zwischen den Domänen „*Fahren auf Straßen niederer Ordnung*“ und *Querfeldeinfahren*. Die Aktivierung der Fähigkeiten läuft hier analog zum Abbiegen ab (vgl. mit Regeln G7 + G8). Schließlich beschreibt Regel G17 die Situation, in der ein Staufahren ausgeführt werden soll. Diese Situation zeichnet sich durch eine große Verkehrsdichte und eine Geschwindigkeit aus, die unter der Zielgeschwindigkeit des Fahrzeugs und unter einem absoluten Schwellwert (30 km/h) liegt. Die Aktivierung der (taktischen) Stop&Go-Fähigkeit geschieht in der gleichen Art und Weise wie bei der Fähigkeit Ausweichen (vgl. [SIEDERSBERGER, 2003]).

---

<sup>26</sup> Beim Fahren auf Straßen niederer Ordnung sind ebenfalls Wahrnehmungsfähigkeiten aktiv (z. B. Eigen- und Querstraßenerkennung), welche jedoch z. Zt. nicht an das Fähigkeitenkonzept angepasst sind.

### 3.5 Realisierung der Verhaltensentscheidung

Dieses Kapitel befasst sich mit einigen wichtigen Realisierungsaspekten bezüglich der in dieser Arbeit vorgestellten Konzepte zur Situationsanalyse und Verhaltensentscheidung. So beschreibt Abschnitt 3.5.1 die Einbettung der Verhaltensentscheidung ins EMS-VISION System. Die prozedurale Ausführung der Situationsanalyse und Verhaltensentscheidung ist Thema des Abschnitts 3.5.2. Abschnitt 3.5.3 beleuchtet die Auswertung der Regelbasis näher. Ferner stellt Anhang D den Prozessrahmen mit den Basisklassen CDecisionBaseScene und CCentralDecisionScene vor und Anhang E zeigt die Klassen zur Verwaltung der Aktionen im verteilten System: CAbilityModel, CAbilityInstance, CAbilityCtrl und CInstanceState.

#### 3.5.1 Die Einbettung der Verhaltensentscheidung ins Gesamtsystem

##### 3.5.1.1 Die organisatorische Hierarchie von Entscheidungsinstanzen

MAURER [2000] schlägt für die flexible Automatisierung von Fahrzeugfunktionen eine organisatorische Hierarchie von Entscheidungsinstanzen vor. Dieses Konzept geht davon aus, dass eine in der Hierarchie höher angesiedelte Entscheidungsinstanz die unteren Entscheidungsinstanzen koordiniert, ihnen aber maximalen Handlungsspielraum lässt. Je höher sich eine Entscheidungsinstanz in der Hierarchie befindet, umso größer ist der Ausschnitt des von ihr überblickten Gesamtsystems, desto länger ist ihre Entscheidungsperiode und desto langsamer sind die Aspekte des berücksichtigten Systemverhaltens. Die auf höheren Ebenen zu handhabenden Beschreibungen und Probleme sind weniger gut strukturiert, schwieriger quantitativ zu formalisieren und mit mehr Unsicherheit behaftet.

Es ist die Aufgabe der übergeordneten Entscheidungsinstanz, Konflikte zwischen den Entscheidungen der unterlagerten Entscheidungsinstanzen aufzulösen, sei es durch Priorisierung oder durch die Vorgabe alternativer Lösungen. Sie muss auch in Aktion treten, wenn eine Situation die Leistungsfähigkeit einer unterlagerten Entscheidungsinstanz übersteigt. Ferner kann man es durch klar spezifizierte Schnittstellen erlauben, dass die unteren Entscheidungsinstanzen Randbedingungen austauschen. Durch die kooperative Einhaltung der ausgetauschten Randbedingungen lassen sich viele Konflikte vermeiden, so dass die überlagerte Entscheidungsinstanz nicht aktiv zu werden braucht.

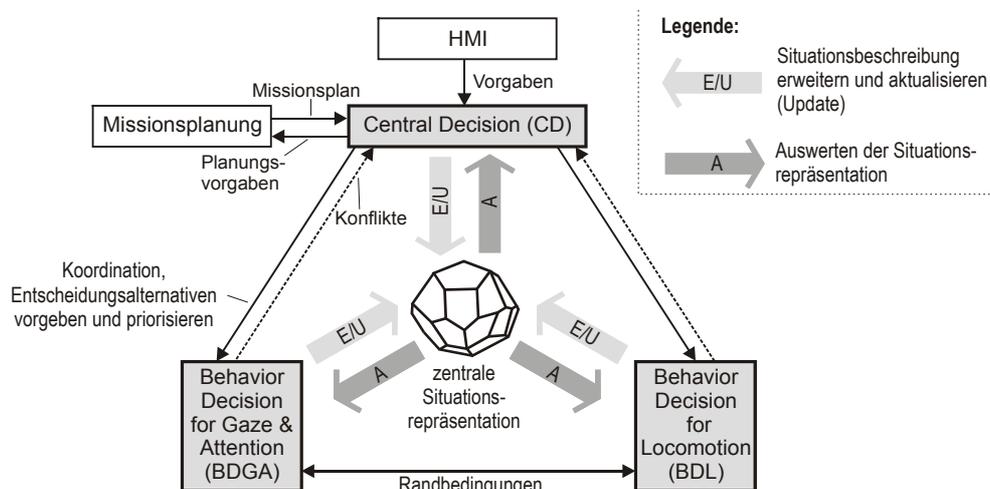


Abb. 3.23: Die Hierarchie von Entscheidungsinstanzen des EMS-VISION Systems

Diese von MAURER [2000] vorgeschlagene organisatorische Hierarchie von Entscheidungsinstanzen wurde in der vorliegenden Arbeit übernommen und ist Teil des EMS-VISION Systems<sup>27</sup> (vgl. Abb. 3.23). Die in Kapitel 3 vorgestellte Situationsanalyse und Verhaltensentscheidung ist in der sogenannten *zentralen Entscheidungsinstanz* Central Decision (CD) angelagert. Neben CD gibt es zwei untergeordnete Entscheidungsinstanzen: Detaillierte Entscheidungen hinsichtlich der Blickrichtung werden vom Prozess Behavior Decision for Gaze & Attention (BDGA) getroffen und hinsichtlich der Fortbewegung vom Prozess Behavior Decision for Locomotion (BDL).

Die Verhaltensentscheidungen der drei Prozesse CD, BDGA und BDL unterscheiden sich hinsichtlich ihres Zeithorizontes: Die Entscheidungen von BDL und BDGA wechseln im Allgemeinen häufiger als die Entscheidungen von CD. Die Entscheidungen von CD sind mehr strategischer Natur und stehen im direkten Zusammenhang mit dem Erreichen des Missionsziels. Die Entscheidungen von BDL und BDGA sind dagegen taktisch geprägt: So legt z. B. BDL den zeitlichen Beginn, die Übergänge zwischen den einzelnen Phasen und das Ende der Handlungen bzgl. Fortbewegung fest. BDGA entscheidet, wann und wie lange und auf welche Objektkombinationen die (hochauflösenden) Kameras gerichtet werden. BDGA und BDL können sich durch den Austausch von Randbedingungen in einem gewissen Rahmen gegenseitig beeinflussen. Treten trotz dieser Abstimmung Konflikte auf, so wird CD mit der Konfliktlösung beauftragt. Die drei Entscheidungsmodule CD, BDGA und BDL werden in den folgenden Unterkapiteln näher betrachtet.

### 3.5.1.2 Die zentrale Entscheidungsinstanz Central Decision (CD)

Der Prozess CD befindet sich an der Spitze der Hierarchie und besitzt die höchste Entscheidungskompetenz aller drei Entscheidungsmodule. CD ist der Experte der Verhaltensfähigkeiten und setzt dementsprechend deren Verfügbarkeit (vgl. Abb. 3.24, oben). Möchte CD eine Verhaltensfähigkeit ausführen, so instanziiert, parametrisiert und startet sie die dazu notwendigen Handlungen. Zusätzlich werden alternative Handlungen vorbereitet, um auf diese bei einem Scheitern der höher priorisierten Handlungen schnell wechseln zu können. Nicht mehr notwendige oder beendete Handlungen werden von CD gelöscht.

CD berücksichtigt bei ihren Entscheidungen den gesamten Situationskontext inklusive der Wahrnehmungsaspekte und den Missionsplan. Durch das Instanziiieren alternativer Handlungen gibt sie den unterlagerten Entscheidungsinstanzen BDL und BDGA vor, welche taktischen Entscheidungsalternativen im Situations- und Missionskontext erlaubt sind und wie diese von ihr priorisiert werden. Die unterlagerten Entscheidungsinstanzen können sich zwischen diesen Alternativen entscheiden und die entsprechenden Handlungen ausführen. Auf der anderen Seite wird von CD nicht vorgegeben, wann die Handlungen genau ausgeführt werden, da sie das dazu notwendige Expertenwissen nicht besitzt und ihr auch die dazu notwendigen Signale und Parameter i. A. nicht zugänglich sind. CD besitzt jedoch immer die Möglichkeit des direkten „Durchgriffs“: Gibt CD den unterlagerten Entscheidungsinstanzen nur eine Entscheidungsalternative vor, so muss diese von BDL bzw. BDGA ausgeführt werden.

Die von der Missionsplanung generierte Liste von Missionselementen dient CD als abstrakte Aufgabenbeschreibung. CD versucht, durch eine geeignete Auswahl, Aktivierung und Überwachung von Verhaltensfähigkeiten dieser Aufgabenbeschreibung gerecht zu werden. Parallel zur Missionsplanung kann der menschliche Benutzer über das Human Machine Interface (HMI) Vorgaben an CD senden. So ist es z. B. möglich, die Zielgeschwindigkeit oder den Grad der Autonomie vorzugeben. Diese Randbedingungen werden an die Missionsplanung weitergeleitet. Die Missionsplanung kann dann überprüfen, ob die neuen Randbedin-

<sup>27</sup> Eine ausführliche Beschreibung des EMS-VISION Systems findet sich in Kapitel 5.

gungen dem aktuellen Missionsplan widersprechen oder nicht. Gegebenenfalls passt sie den bestehenden Missionsplan an oder erstellt einen neuen. Die Missionsplanung kann mittels des Fähigkeitsnetzes die Verfügbarkeit oder Nichtverfügbarkeit von Verhaltensfähigkeiten bei der Planung mit berücksichtigen und so verhindern, dass ein Missionsplan bis zum Ende ausgearbeitet wird, wenn das System die dazu notwendigen Verhaltensfähigkeiten aktuell nicht unterstützt. Das Wissen im Fähigkeitsnetz dient auf diese Weise der Optimierung und Lenkung der Missionsplanung [GREGOR, 2002].

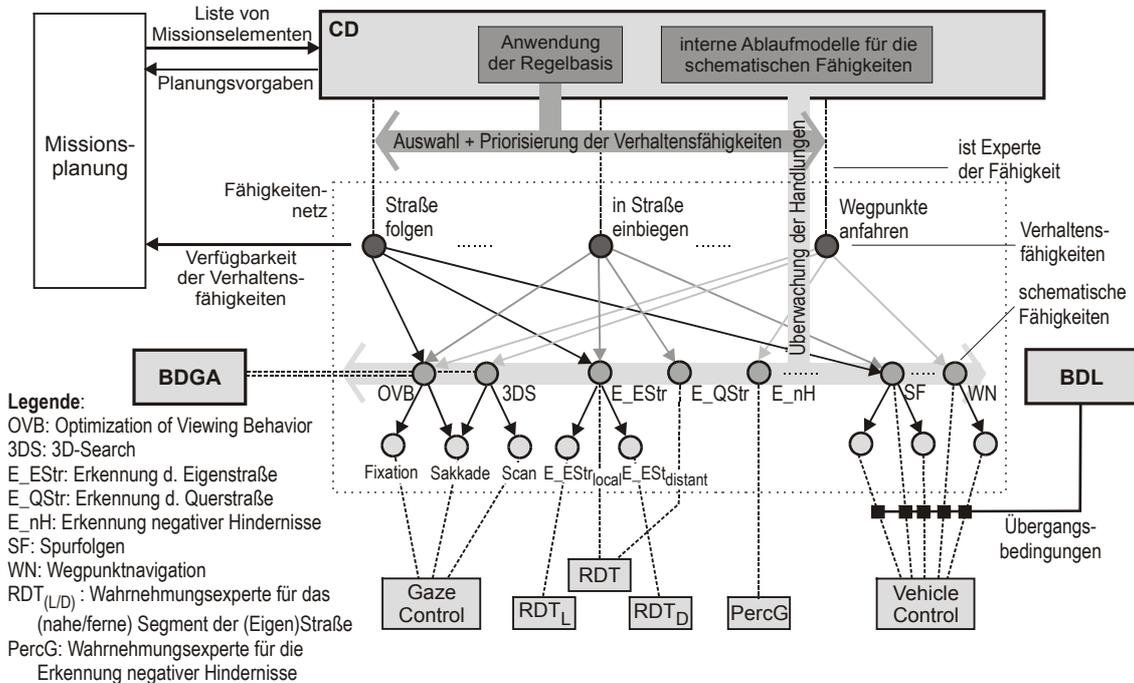


Abb. 3.24: Aktivierung, Priorisierung und Überwachung der schematischen Fähigkeiten durch CD

### 3.5.1.3 Die unterlagerten Entscheidungsinstanzen BDGA und BDL

In den unterlagerten Entscheidungsinstanzen BDL und BDGA kommen detaillierte Modelle, Signale und Parameter zur Anwendung, welche nicht im gesamten System verfügbar sind. So besitzt BDL ein größeres Expertenwissen hinsichtlich der Fortbewegung als CD, arbeitet mit detaillierteren Form- und Dynamikmodellen für das Eigenfahrzeug und hat Zugang zu allen Daten des Prozesses Vehicle Control (VC) [SIEDERSBERGER, 2003]. In BDGA ist dagegen das Wissen angelagert, wie und wann die relevanten Objekte in der Umgebung abgebildet werden müssen, um den Informationsbedarf des autonomen Systems im Situationskontext zu befriedigen. BDGA berücksichtigt dabei die vom Prozess Gaze Control (GC) angebotenen Blickrichtungsfertigkeiten, die Leistungsfähigkeit des eingebetteten Systems zur Blickrichtungssteuerung, die optischen Eigenschaften der Kameras und die Eigenschaften der Wahrnehmungsprozesse (vgl. Kapitel 4).

BDL und BDGA werden über das Fähigkeitsnetz davon informiert, welche schematischen Fähigkeiten bezüglich Fortbewegung bzw. Blickrichtung von CD favorisiert werden und welche Alternativen zur Verfügung stehen. Die Entscheidung, welche der von CD zugelassenen sukzessiven und alternativen Handlungen wirklich auf die Aktuatoren des Fahrzeugs zugreifen und wann genau die operativen Wechsel zwischen den Handlungen vollzogen werden, liegt in der Entscheidung von BDL und BDGA. CD wird mittels der im Fähigkeitsnetz repräsentierten Handlungen über die Entscheidungen von BDL und BDGA informiert.

### 3.5.1.4 Die zentrale Situationsrepräsentation

Eine wichtige Komponente bei der Verhaltensgenerierung innerhalb des EMS-VISION Systems bildet die zentrale Repräsentation des Wissens über die Situation. So können alle drei Entscheidungsmodule CD, BDGA und BDL die während ihrer Entscheidungstätigkeiten generierten situationsbeschreibenden Aussagen der Situationsrepräsentation hinzufügen. Dabei kann es sich um linguistische oder kontinuierliche Größen handeln. Die Gesamtheit der im System generierten situationsbeschreibenden Aussagen stehen damit allen Entscheidungsmodulen zur weiteren Auswertung zur Verfügung. Realisiert wird die zentrale Situationsrepräsentation durch einen separaten Zweig innerhalb der dynamischen Wissensrepräsentation DKR (vgl. Kapitel 5.3).

### 3.5.1.5 Das Fähigkeitsnetz als Schnittstelle zwischen den Entscheidungsinstanzen

Neben der zentralen Situationsrepräsentation bildet das in Kapitel 3.3 beschriebene Fähigkeitsnetz eine weitere Schnittstelle zwischen den Entscheidungsinstanzen und den Experten der Fähigkeiten. In der obersten Ebene des Fähigkeitsnetzes stehen die Verhaltensfähigkeiten, deren Experte CD ist (vgl. Abb. 3.24). Im Fähigkeitsnetz ist repräsentiert, welche schematischen Fähigkeiten für welche Verhaltensfähigkeiten notwendig sind. Entsprechend den Aussagen in Kapitel 3.3.3 überwacht CD als Experte der Verhaltensfähigkeiten die Ausführung und den Fortschritt der schematischen Fähigkeiten. Dies geschieht durch die internen Ablaufmodelle in der Verhaltensentscheidung (vgl. Abschnitt 3.3.6). Man kann 3 Arten von schematischen Fähigkeiten unterscheiden:

- Fähigkeiten bzgl. Fortbewegung werden durch den Prozess Vehicle Control (VC) angeboten. CD kann zur Abarbeitung des Missionsplans nacheinander mehrere Fortbewegungsfähigkeiten instanziiieren. Welche dieser angestoßenen Handlungen sich im Operationsstatus RUN befindet und damit auf die Aktuatoren des Fahrzeugs zugreift, entscheidet BDL (vgl. [SIEDERSBERGER, 2003]).
- Die Fähigkeiten zur Wahrnehmung werden von den verschiedenen Wahrnehmungsexperten im System zur Verfügung gestellt. In Abb. 3.24 ist die vom Prozess Perceive Ground (PercG) angebotene Fähigkeit „Erkennung negativer Hindernisse“ eingezeichnet. Ferner sind die Wahrnehmungsfähigkeiten „Erkennung der Eigenstraße“ und „Erkennung der Querstraße“ dargestellt, welche beide vom Experten RDT angeboten werden. Die erstere der beiden Fähigkeiten gliedert sich in die Teilfähigkeiten „Erkennung des lokalen Segments der Eigenstraße“ (unter Verwendung der Weitwinkelkameras) und „Erkennung des fernen Segments der Eigenstraße“ (im Bild der Telekamera). Die Experten dieser Teilfähigkeiten befinden sich aufgrund der unterschiedlichen Bildquellen auf verschiedenen Rechnern im verteilten System (vgl. Kapitel 5.3). Der Prozess CD braucht dieses Detail nicht zu kennen.
- BDGA bietet die schematischen Fähigkeiten bzgl. Blickrichtung an, welche in Kapitel 4.7 detailliert beschrieben sind.

Die Fähigkeiten bezüglich Fortbewegung, Wahrnehmung und Blickrichtung werden durch die Verhaltensfähigkeiten miteinander verknüpft. So beinhaltet z. B. die schematische Fähigkeit *Spurfolgen* zwar die Funktionalität für die Steuerung und Regelung des Fahrzeugzustands; sie kümmert sich jedoch nicht um das Vorhandensein einer Spurerkennung. Diese Zusammenschau und Koordination wird durch die Verhaltensfähigkeit *„Straße folgen“* geleistet. So wie BDL die Fähigkeiten bezüglich Fortbewegung und BDGA die Fähigkeiten bezüglich Blickrichtung koordiniert, so koordiniert CD die Verhaltensfähigkeiten und triggert die dazu notwendigen schematischen Fähigkeiten.

### 3.5.1.6 Die hybride Architektur des EMS-VISION Systems

Anhand den Ausführungen in den vorangegangenen Unterabschnitten sieht man deutlich den hybriden Charakter der Architektur des EMS-VISION Systems mit deliberativen und verhaltensbasierten Elementen (vgl. Abschnitt 2.3.4). So besitzt das System als deliberative Teile die zentrale Wissensrepräsentation, die zentrale Entscheidungsinstanz und ein Planungsmodul mit Kartenwissen. Die Experten mit ihren Fähigkeiten bilden den verhaltensbasierten Systemanteil, welcher durch den deliberativen Teil kurzfristige Aktionen vorgeschlagen bekommt. Die Experten der Fähigkeiten setzen die Aktionen selbständig, situativ richtig und zeitadäquat um, wobei gewisse Sensordaten mittels unterlagerten Regelungen direkt auf die Aktuatoren des Fahrzeugs rückgekoppelt werden. Diese für verhaltensbasierte Systeme so charakteristische enge Verknüpfung zwischen Wahrnehmung und Aktion ist im EMS-VISION System nicht statisch; vielmehr entscheiden deliberative Systemkomponenten darüber, welche Verknüpfungen (Fähigkeiten) aktuell anzuwenden sind. Die Kopplung zwischen den deliberativen und verhaltensbasierten Teilen beruht beim EMS-VISION System auf Selektion, Adaption und ratgebender Funktion (vgl. Abschnitt 2.3.4.3).

### 3.5.2 Die prozedurale Ausführung der Situationsanalyse und Verhaltensentscheidung

Dieses Kapitel befasst sich mit der prozeduralen Ausführung der Situationsanalyse und Verhaltensentscheidung, welche innerhalb der zyklisch aufgerufenen Funktion `OnEveryCycle()` des Prozessrahmens `CCentralDecisionScene` abläuft.

Zu Beginn der zyklischen Behandlung werden die Datenvorverarbeitungsmodul durch den Funktionsaufruf `DoDataProcessing()` (vgl. Abb. 3.25 und Anhang A) und die Situationsaspekte durch den Aufruf von `DoSituationAnalysis()` darüber informiert, dass ein neuer Systemzyklus begonnen hat. Jede Datenvorverarbeitung und jeder Situationsaspekt entscheidet anhand der Datenlogik bzw. Berechnungslogik für sich alleine darüber, ob sie bzw. er im aktuellen Zyklus angewandt werden muss (vgl. Abschnitt 3.2.2). Die Datenvorverarbeitungen und Situationsaspekte werden nur auf zu berechnend markierte Objektkombinationen angewandt.

Innerhalb der Funktion `DoBehaviorDecision()` erfolgt durch den Aufruf der Funktion `Compare()` für jedes Subjekt die Anwendung der Regelbasis auf die berechneten Situationsaspekte. Die dabei ausgewählten Regeln werden dann mit absteigendem Erfülltheitsgrad durch den Aufruf von `SpecifySituationAspects()` dazu aufgefordert, ihre relevanten Situationsaspekte einzutragen. Die Regeln leiten diese Aufforderungen durch den Aufruf der Funktion `SetSituationAspects()` an die in den Konklusionen spezifizierten Ablaufmodelle weiter. In den internen Ablaufmodellen sind relevante Situationsaspekte und Anweisungen über deren Anwendung gespeichert (vgl. Abschnitt 3.3.6). Diese Information überträgt die Funktion `InsertMembershipMatrixEntry()` in die Matrix `m_MembershipMatrix` des Situationsaspekts (vgl. Anhang A.1).

Nach diesen Neueinträgen beginnt die nächste Iteration: Es werden in den Funktionen `DoSituationAnalysis()` und `DoBehaviorDecision()` die neu hinzugekommenen Einträge in den Situationsaspekten berechnet und das erweiterte, linguistische Wissen von den Regeln der Regelbasis ausgewertet. Gegebenenfalls spezifizieren neu ausgewählte Fähigkeiten weitere Anforderungen an die Situationsanalyse und lösen damit eine erneute Iteration aus. Werden keine neuen Fähigkeiten ausgewählt, so bricht die Iteration ab.

Die von den gültigen Regeln vorgeschlagenen Fähigkeiten gelangen nun zur Ausführung. Bei neu gültigen Regeln ruft der Prozessrahmen die Funktion `OnBegin()` auf. War die Regel

im letzten Zyklus bereits ausgewählt, so führt dies zum Funktionsaufruf von OnCycle(). Ist eine zuvor gültige Regel jetzt nicht mehr gültig, so erfolgt der Aufruf der Funktion OnEnd(). Diese drei Ereignisbehandlungsfunktionen senden entsprechende Steuerbefehle an die internen Ablaufmodelle. Welche Steuerbefehle möglich sind und was sie bewirken, beschreibt Abschnitt 3.5.3. Zum Abschluss der Zyklusbehandlung erfolgt für alle Ablaufmodelle der Aufruf der Funktionen DoCommands() und Cycle(), deren Inhalte ebenfalls Thema von Abschnitt 3.5.3 ist.

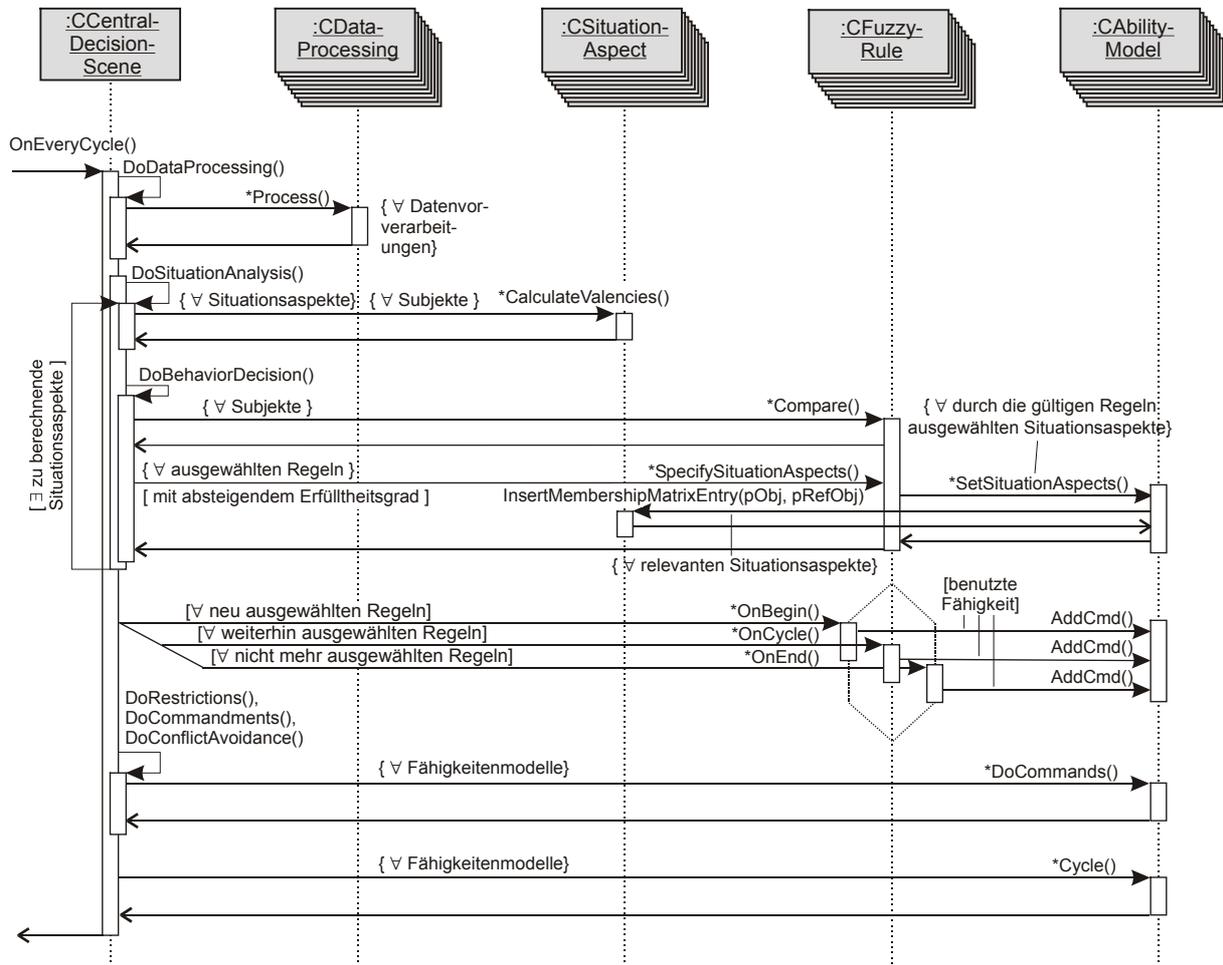


Abb. 3.25: Der Ablauf von Situationsanalyse und Verhaltensentscheidung (Sequenzdiagramm nach UML)

### 3.5.3 Die Auswertung beim Regelwechsel

Wie im vorangegangenen Abschnitt dargestellt wurde, senden die ausgewählten Regeln der Regelbasis Steuerbefehle an die internen Ablaufmodelle der Fähigkeiten, um die Ausführung der Aktionen in ihrem Sinne zu beeinflussen. Wann und wie dies geschieht, illustriert Abb. 3.26.

In Abb. 3.26 sind drei Regeln dargestellt, welche die Verwendung von drei Fähigkeiten vorschlagen: Regel 1 benützt die Fähigkeit 1, wobei deren Parameter 1 mit dem linguistischen Wert *mittel* angegeben ist. Die Regel 2 benützt die Fähigkeiten 1 und 2 mit der Parameterbelegung Parameter 1 = *groß*. Die dritte Regel schlägt ebenfalls zwei Fähigkeiten vor: Fähigkeit 1 mit der Parametrisierung Parameter 1 = *klein* und Fähigkeit 3. Die Erfülltheitsgrade der Regeln liegen im Intervall [0,1] und ändern sich mit der Zeit. In Abb. 3.26 sind die Erfülltheitsgrade durch die Grauschattierungen angedeutet: Regel 1 besitzt einen durchschnittlichen Er-

fülltheitsgrad, der jedoch mit fortschreitender Zeit nur wenig kleiner wird. Die Erfüllung der Regel 2 reduziert sich mit fortschreitender Zeit und die Erfüllung der Regel 3 wird im Gegensatz dazu größer. Ferner ist in Abb. 3.26 angenommen, dass sich die Benutzung der Fähigkeiten 2 und 3 widersprechen, d. h. sie können nicht gleichzeitig ausgeführt werden<sup>28</sup>.

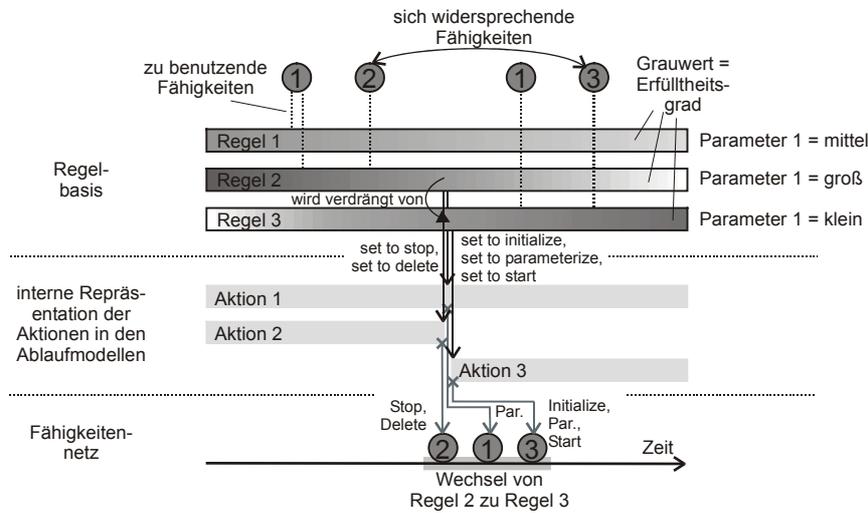


Abb. 3.26: Wechsel in der Regelbasis

Anfänglich besitzt Regel 2 die größere Erfüllung, was zur Ausführung von Fähigkeit 1 und 2 führt (Aktion 1 und 2). Nach einer gewissen Zeit sinkt der Erfülltheitsgrad der Regel 2 unter den der Regel 3, so dass wegen ihrer sich widersprechenden Konklusionen Regel 3 die Regel 2 verdrängt. Dies führt bei Regel 2 zum Aufruf der Ereignisbehandlungsfunktion OnEnd() und bei Regel 3 zum Aufruf von OnBegin() (vgl. Abschnitt 3.5.2). Diese Funktionen setzen Steuerbefehle an die entsprechenden internen Ablaufmodelle der Fähigkeiten ab. Im Beispiel von Abb. 3.26 markiert OnEnd() von Regel 2 die Aktionen 1 und 2 als zu beenden und zu löschen und OnBegin() von Regel 3 die Aktion 1 und die (noch nicht instanziierte) Aktion 3 der Fähigkeit 3 als zu initialisieren, parametrisieren und zu starten.

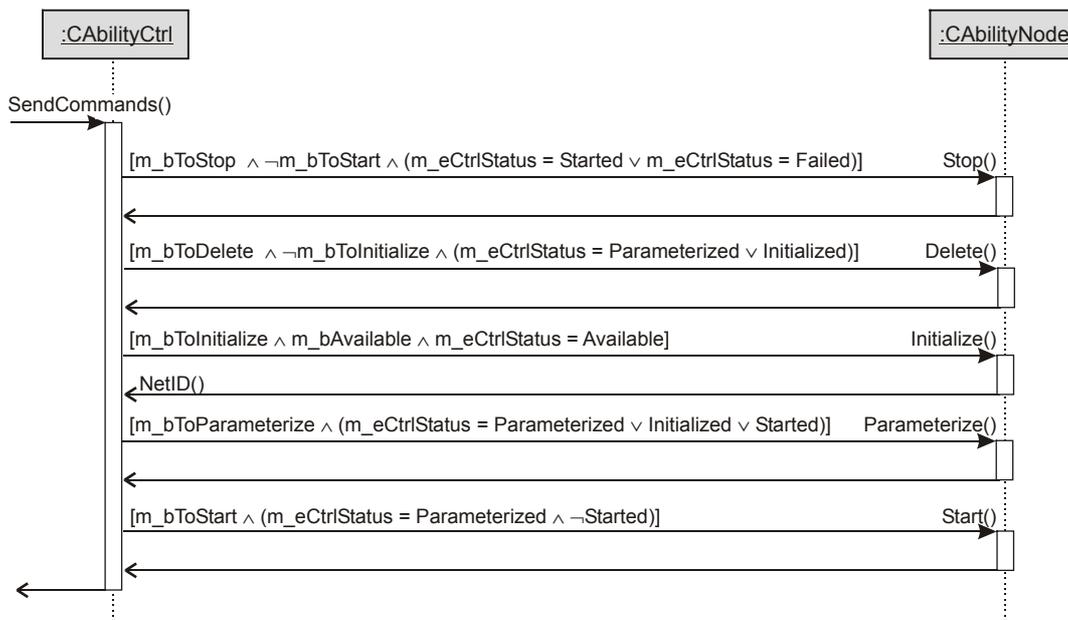


Abb. 3.27: Die Logik bei der Steuerung von Aktionen (Sequenzdiagramm nach UML)

<sup>28</sup> Zwei sich widersprechende Fähigkeiten hinsichtlich der Querführung eines Fahrzeugs sind z. B. Spurfolgen und Spurwechsel.

In den internen Ablaufmodellen befinden sich Verwaltungsobjekte der Klasse *CAbilityCtrl*, welche die Markierungen aufnehmen (vgl. Anhang E). Anhand des Kontrollstatus der aktuell ausgeführten Aktionen und der Markierungen wird innerhalb der Funktion *CAbility-Model::DoCommands()* entschieden, welche Kommandos über das Fähigkeitsnetz abgesetzt werden müssen, um die Ausführung der Fähigkeiten im Sinne der neu gültigen Regeln anzupassen. Im Beispiel von Abb. 3.26 markierte zwar Regel 2 die Aktion 1 als zu löschen, da aber Regel 3 diese gleichzeitig als zu starten markiert, läuft Aktion 1 weiter. Es erfolgt lediglich eine neue Parametrisierung. Abb. 3.27 zeigt, in welcher Reihenfolge und mit welcher Logik die Markierungen ausgewertet werden.

Bei der Parametrisierung einer Aktion können alle Regeln durch ein Defuzzifizierungsverfahren<sup>29</sup> mitwirken, sofern sie die Aktion unterstützen. So wird gemäß Abb. 3.26 der Parameter 1 von allen drei Regeln beeinflusst, wobei natürlich die aktuellen Erfülltheitsgrade der Regeln in das Defuzzifizierungsverfahren mit eingehen. Im Beispiel liegt vor dem Regelwechsel der defuzzifizierte Parameter 1 im Schwerpunkt zwischen den linguistischen Werten *mittel* und *groß*, und nach dem Wechsel zwischen *mittel* und *klein*.

### 3.6 Informationsfluss im Gesamtsystem

Zum Abschluss der Beschreibung der Verhaltensentscheidung sei nochmals der Informationsfluss im Gesamtsystem betrachtet. Abb. 3.28 zeigt die entsprechende Übersicht. Die grau unterlegten Ellipsen symbolisieren Daten und Informationen, welche von den Systemkomponenten – dargestellt durch Rechtecke – gemessen bzw. generiert werden. Zunächst gewinnt die Zustands- und Formschatzung aus den Bildmerkmalen die Zustandsgrößen der Objekte (4D-Ansatz) und trägt diese in den Szenenbaum der dynamischen Objektdatenbasis (DOB) ein. Die Situationsanalyse nimmt die im Szenenbaum vorliegende Information und berechnet damit die Zugehörigkeitsgrade der linguistischen Werte der als relevant markierten Situationsaspekte. Die Zugehörigkeitsgrade werden in die Situationsrepräsentation eingetragen. Die Verhaltensentscheidung nimmt die aktuelle Situationsrepräsentation und wendet auf sie die in der Regelbasis befindlichen unscharfen Regeln an. In diesen Regeln gehen als weitere Terme die von der Missionsplanung spezifizierten Missionselemente ein. Aus der Anwendung der Regelbasis ergibt sich eine situationsgerechte Auswahl von priorisierten Fähigkeiten. Die zu den ausgewählten Fähigkeiten gehörenden internen Ablaufmodelle spezifizieren eventuell neu zu berechnende, relevante Situationsaspekte. Diese werden an die Situationsanalyse gesandt und dort evaluiert. Durch die Berechnung der linguistischen Werte der neu hinzugekommenen Situationsaspekte liegt eine erweiterte Situationsrepräsentation vor, auf welche die Regelbasis erneut anzuwenden ist. In Abb. 3.28 (links oben) ist der bei der Iteration mehrfach durchlaufene Kreis

*Situationsanalyse*  $\rightleftharpoons$  *Verhaltensentscheidung*

ingezeichnet.

Ergab sich bei der letzten Iteration für kein Subjekt in der Szene eine Änderung in der Fähigkeitsauswahl, so bricht die Iteration ab. Die für das Eigenfahrzeug ausgewählten Fähigkeiten sind damit verifiziert und gelangen zur Ausführung. Die Instanziierung, Steuerung und Überwachung der Aktionen geschieht über das Fähigkeitsnetz. Dabei dienen die Knoten des Fähigkeitsnetzes als einheitliche Schnittstelle zu den Experten der Fähigkeiten. Das Fähigkeitsnetz leitet die Steuerbefehle automatisch an die beteiligten Experten weiter.

<sup>29</sup> Als Defuzzifizierungsverfahren bietet sich z. B. eine Schwerpunktmethod an.

In den Bildverarbeitungsprozessen laufen Aktivitäten bzgl. Wahrnehmung ab, welche die Zustands- und Formschätzung beeinflussen. Die Blickrichtungssteuerung legt die Blickrichtung hinsichtlich des Aufmerksamkeitsbedarfs der Szenenobjekte und der Wünsche der Bildverarbeitungsprozesse fest und bestimmt damit, welche Merkmale in den Bildern sichtbar sind. Parallel dazu beeinflusst auch die Fortbewegung des Fahrzeugs die Sichtbarkeit der Merkmale.

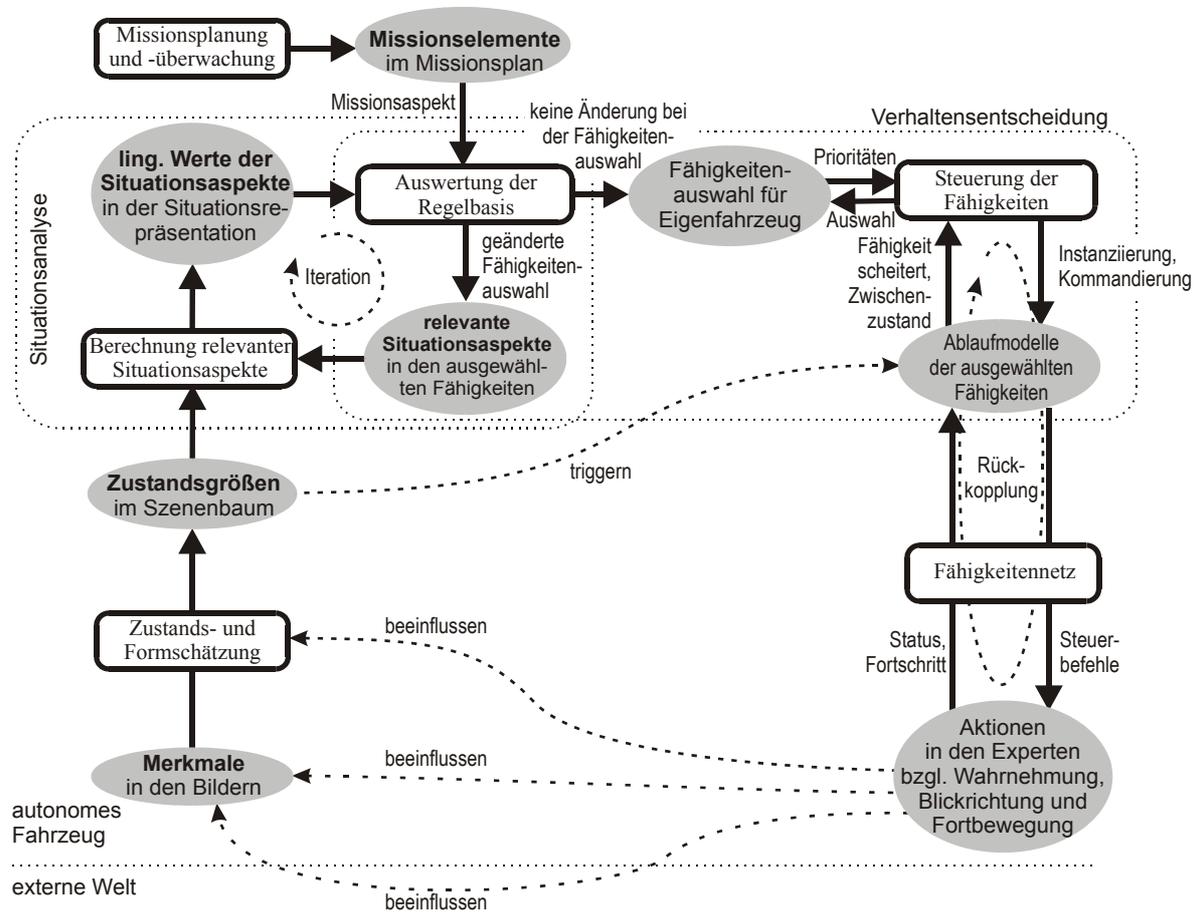


Abb. 3.28: Informationsfluss im Gesamtsystem

Die Experten berechnen den Status und Fortschritt ihrer Aktionen. Die Verhaltensentscheidung greift über das Fähigkeitsnetz auf diese Größen zu und reagiert in geeigneter Weise darauf. Die dafür notwendige Vorstellung vom Sollablauf einer Fähigkeit ist im internen Ablaufmodell der Fähigkeit niedergelegt. Dabei treibt der Verlauf der Zustandsgrößen im Szenenbaum die Zustandsübergänge im Ablaufmodell. Beim Scheitern einer Fähigkeit aktiviert die Verhaltensentscheidung diejenigen Fähigkeiten, welche die (nach Abbruch der oben beschriebenen Iteration) ausgewählten Regeln als alternative Fähigkeiten spezifizieren. Parallel dazu kann die Verhaltensentscheidung von der Missionsplanung einen neuen Missionsplan anfordern. In Abb. 3.28 (rechts) ist der Rückkopplungskreis

*Verhaltensentscheidung*  $\rightleftharpoons$  *Ablaufmodelle der Fähigkeiten*  $\rightleftharpoons$  *Fähigkeitsnetz*  $\rightleftharpoons$  *Experten*  
klar ersichtlich.

## Kapitel 4

### Blickrichtungssteuerung

Das Fahren in unterschiedlichen Domänen erfordert die parallele Wahrnehmung vieler Objekte mit unterschiedlicher Größe, Relativlage und Orientierung. Besonders bei hohen Geschwindigkeiten ist eine große Vorausschauweite unabdingbar für eine frühzeitige Erkennung von Fremdfahrzeugen. Im Stadtverkehr ist ein weites Gesichtsfeld notwendig, um Kollisionen z. B. mit Fußgängern oder Radfahrern zu vermeiden. Für die Gewinnung von räumlicher Information (Position und Geschwindigkeit) ist eine Stereobildverarbeitung vorteilhaft. Schließlich bietet sich im unebenen Gelände eine inertielle Stabilisierung der Blickrichtung zur Reduzierung der Bewegungsunschärfe an.

Eine große Vorausschauweite und gleichzeitig ein großes Gesichtsfeld lässt sich z. B. dadurch erreichen, indem man viele Kameras mit unterschiedlichen optischen Eigenschaften und Orientierungen auf das Fahrzeug montiert [ULMER, 1994]. Diese Vorgehensweise ist jedoch teuer, unflexibel, fehleranfällig und verlangt die Kommunikation und Handhabung großer Video-Datenströme. Der Ansatz der hier verfolgt wird und auf DICKMANN'S [1994] zurückgeht, ist die Imitation des Blickverhaltens des menschlichen Fahrers, wie es in Kapitel 2.1.2 beschrieben wurde. Man bildet dazu das menschliche Auge durch eine aktive Kameraplattform mit multifokalem Kameraaufbau nach. Dieses Fahrzeugauge gibt dem autonomen Fahrzeug die Möglichkeit, die Kameraausrichtung und damit den Sensorinput dem aktuellen Wissensstand anzupassen und auf eventuelle Wissensdefizite zu reagieren. Neben den Aktuatoren für die Fortbewegung verfügt das aktiv sehende Fahrzeug damit über weitere Aktuatoren, mit denen es mit der Umwelt interagieren kann.

Für die Blickrichtungssteuerung eines autonomen Fahrzeugs ist die Gesamtsituation relevant, welche sich aus dem externen Verkehrsgeschehen und dem Eigenfahrzeug als Subjekt zusammensetzt. Die Art und Weise wie das autonome Fahrzeug die Freiheitsgrade in der Blickrichtung nutzt, ist Teil des Verhaltens des Systems und die in diesem Kapitel beschriebene Optimierung des Blickverhaltens ist Teil der Verhaltenssteuerung (vgl. dazu auch Kapitel 3.5.1). Auch BALLARD & BROWN [1982] betonen, dass ein Sehsystem komplexe, zusammengesetzte Fähigkeiten benötigt und in der Lage sein muss, seine eigenen Ziele und seinen Erfolg zu modellieren und zu bewerten.

#### 4.1 Das Fahrzeugauge MARVEYE

Das Schlüsselement des EMS-VISION Systems ist das multifokale, aktive/reaktive Fahrzeugauge namens MARVEYE. Der Name MARVEYE steht für *Multi-focal active/reactive Vehicle Eye*. Dieser Kameraaufbau resultiert aus den oben geschilderten Anforderungen, welche sich aus dem autonomen Fahren in unterschiedlichen Domänen wie Autobahnen, Straßen nie-

derer Ordnung und Querfeldeinfahren ergeben. Die Gründe, die zur Entwicklung des MARVEYE-Fahrzeugauges in dieser Form geführt haben, sind in [LÜTZELER, 2002] dargestellt. Die MARVEYE-Kamerakonfiguration kombiniert ein weites Gesichtsfeld ( $\sim 105^\circ$ ) mit zentralen Bereichen hoher Auflösung. MARVEYE besteht aus bis zu vier Kameras mit drei verschiedenen Brennweiten (vgl. Abb. 4.1 a). Zwei Kameras sind mit Weitwinkelobjektiven ausgestattet (Brennweite 6 mm, Öffnungswinkel ca.  $58^\circ$ )<sup>30</sup> und bilden ein horizontales Stereopaar mit parallelen oder divergenten optischen Achsen. Der divergente Aufbau wird für das Navigieren auf engem Raum auf Wegenetzen und für das frühzeitige Erkennen von Einscherern auf Autobahnen benutzt. Der parallele Aufbau ist geeignet für die Schätzung der Vertikalkrümmung der befahrenen Oberfläche und zur Detektion von Hindernissen und Fremdfahrzeugen durch Stereobildverarbeitung (vgl. Abb. 4.1 b).

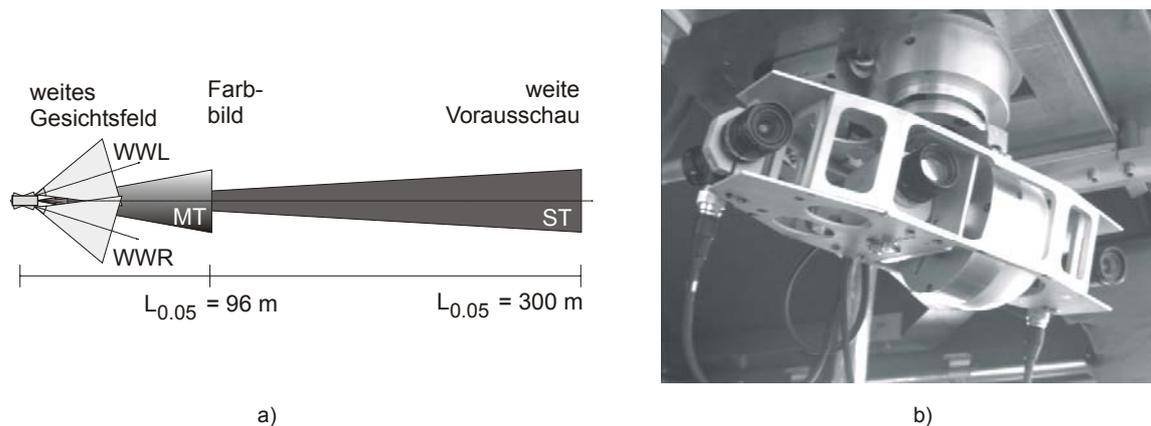


Abb. 4.1: Die MARVEYE-Kamerakonfiguration: Skizze (a) und montiert auf dem Kameraträger des Versuchsfahrzeugs VAMORS (mit parallelem Weitwinkelpaar und ohne starke Telekamera<sup>31</sup>) (b)

Auf der dritten und vierten Kamera sind ein mildes bzw. eine starke Teleobjektiv mit Brennweiten von 24 bzw. 75 mm befestigt (Öffnungswinkel  $11^\circ$  bzw.  $5^\circ$ ). Die Auflösung der von ihnen gelieferten Bilder ist um einen Faktor 3 - 4 bzw. 10 größer als die Auflösung der Weitwinkelbilder. Die hochaufgelösten Kamerabilder und die korrespondierenden großen Vorausschauweiten<sup>32</sup> unterschiedlicher Länge sind nützlich zur (frühzeitigen) Identifizierung neuer Objekte, für das Schätzen von Zustandsgrößen bekannter Objekte mit großer Genauigkeit und zur Landmarkennavigation [GREGOR, 2002]. Die milde Telekamera ist eine 3-Chip Farbkamera und ermöglicht Farbbildverarbeitung. Die restlichen Kameras sind Monochromkameras.

## 4.2 Das eingebettete System zur Blickrichtungssteuerung

Die optimale Nutzung des MARVEYE-Fahrzeugauges bedingt, dass die innerhalb des weiten Gesichtsfeldes liegenden Objekte durch schnelle Blickzuwendungen (Sakkaden) in ein höherauflösendes Kamerabild gebracht werden können. Dazu ist eine aktive Steuerung der Blickrichtung der Kameras notwendig. Das MARVEYE-Fahrzeugauge ist aus diesem Grund auf einem Kameraträger montiert, der hochdynamisch um die Nick- und Gierachse gedreht

<sup>30</sup> Es kamen auch andere Brennweiten (z. B. 4,8 mm) zum Einsatz.

<sup>31</sup> Die starke Telekamera kann unterhalb der milden Telekamera am Kameraträger befestigt werden (siehe [LÜTZELER, 2002]).

<sup>32</sup> Die Vorausschauweite  $L_{0,05}$  spezifiziert die Entfernung, bei der ein Objekt von 0,05 m Breite bei der Projektion ins Bild auf genau 1 Pixel abgebildet wird.

werden kann (vgl. Abb. 4.1 b). Die Regelung des Kameraträgers übernimmt das eingebettete System zur Blickrichtungssteuerung.

Dieses eingebettete System zur Blickrichtungssteuerung – kurz auch (Kamera-)Plattform genannt – besteht aus zwei Teilen: Plattformelektronik und Plattformmechanik (vgl. Abb. 4.2). Die Plattformmechanik trägt die Kameras und ist mit Winkelsensoren und Winkelratensensoren für die Regelung und Stabilisierung der Plattform ausgestattet. Ferner verfügt die Mechanik über bürstenlose Gleichstrommotoren.

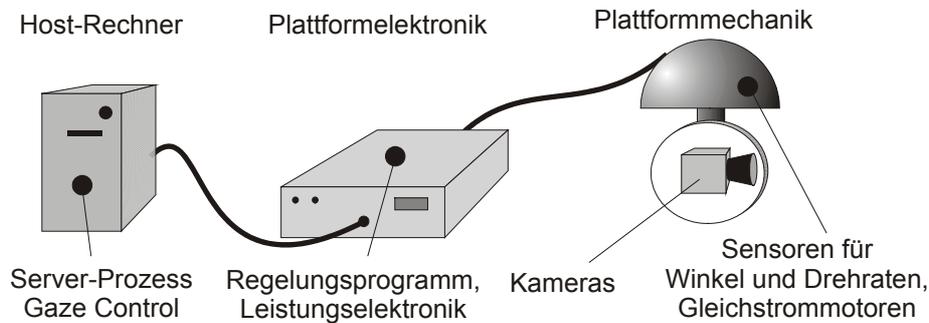


Abb. 4.2: Das eingebettete System zur Blickrichtungssteuerung mit Host-Rechner

Die Plattformelektronik besteht aus dem eingebetteten Rechner und der Leistungselektronik zur Motoransteuerung. Der eingebettete Rechner ist im Versuchsfahrzeug VAMORS ein Transputer und im VAMP ein Industrie-PC. Der Rechner ist über einen Transputer-Link (VAMORS) bzw. CAN-Link (VAMP) mit dem Fahrzeugrechner (Host) verbunden (vgl. Abb. 4.2). Auf dem Fahrzeugrechner befindet sich der Server-Prozess Gaze Control, der das eingebettete System zur Blickrichtungssteuerung anspricht (vgl. Abschnitte 4.3).

Das Regelungsprogramm läuft auf dem eingebetteten Rechner und basiert auf den von SCHIEHLEN [1995] entwickelten Algorithmen. So wird z. B. durch eine sogenannte Führungsgrößenbeschränkung die Führungsgröße (Sollwinkel  $\varphi$  und/oder Sollwinkelgeschwindigkeit  $\dot{\varphi}$ ) reduziert, wenn die Stellgröße ihren maximalen Wert zu überschreiten droht. Ein Führungsgrößenbeobachter verbessert das Folgeverhalten bei nicht sprunghaften Führungsgrößenverläufen. Der eigentliche Regler ermittelt aus den (eventuell beschränkten/beobachteten) Führungsgrößen die Stellgröße. Dabei kann man zwischen zwei unterschiedlichen Reglern auswählen: Der erste Regler benutzt das Winkelsignal, um eine vorgegebene Winkelstellung anzufahren bzw. eine Winkelgeschwindigkeit einzuhalten und ermöglicht z. B. schnelle Blickzuwendungen. Der zweite Regler verarbeitet die Signale der Winkel- und Winkelratensensoren, um den Kameraträger in einem Frequenzbereich von 0,7 - 2 Hz inertial zu stabilisieren. In Fahrzeugen können damit die beim Bremsen und Beschleunigen auftretenden Nickbewegungen des Fahrzeugaufbaus ausgeglichen werden<sup>33</sup>. Einzelheiten bezüglich der Plattformregelung finden sich in der Dokumentation von RIEDER [1998]. Die Leistungsdaten der Regelung der Kameraplattform mit dem MARVEYE-Fahrzeugaug als Last werden in [PELLKOFER & DICKMANN, 2000] beschrieben.

<sup>33</sup> Die Messung und Kompensation hochfrequenter Eigenbewegungen wurde z. B. von VIÉVILLE [1997] als eine von drei Hauptfunktionalitäten eines aktiven Sehsystems für bewegliche Roboter und Fahrzeuge herausgestellt.

### 4.3 Der Aufbau der Blickrichtungssteuerung

Außerhalb des eingebetteten Systems realisieren drei Prozesse die Blickrichtungssteuerung: Die zentrale Entscheidungsinstanz Central Decision (CD), der Prozess Behavior Decision for Gaze & Attention (BDGA) und der Server-Prozess Gaze Control (GC). BDGA ist der planende Teil der Blickrichtungssteuerung und besteht aus den Teilmodulen Visibility Analysis for Gaze & Attention (VAGA) und Optimization of Viewing Behavior (OVB) (vgl. Abb. 4.3). Parallel dazu findet im Prozess CD eine Situationsanalyse bezüglich Blickrichtung statt (Situation Analysis for Gaze & Attention (SAGA)).

SAGA hat die Aufgabe, im Situationskontext die notwendige Genauigkeit der Zustandsgrößen und Güte des Objektwissens zu ermitteln. Dabei handelt es sich um absolute Gütemaße (Unsicherheitsintervalle, Varianzen) mit einer konkreten geometrischen Bedeutung (z. B. die Genauigkeit, mit der die Position oder Breite eines Fremdfahrzeugs bekannt sein muss, um das Fahrzeug seitlich zu passieren). Im nächsten Schritt bildet SAGA die Differenz zwischen den notwendigen und den bisher erreichten Genauigkeiten und leitet daraus die Parameterbelegung der Blickrichtungsfähigkeiten ab. Die wesentlichen Eingangsparameter der in Abschnitt 4.5 dargestellten Blickrichtungsfähigkeit OVB sind der Aufmerksamkeitsbedarf der um die Ressource Blickrichtung konkurrierenden Objekte, die maximale Anzahl der Sakkaden pro Planungszeitraum und der Planungshorizont (Planungszeitraum in Sekunden).

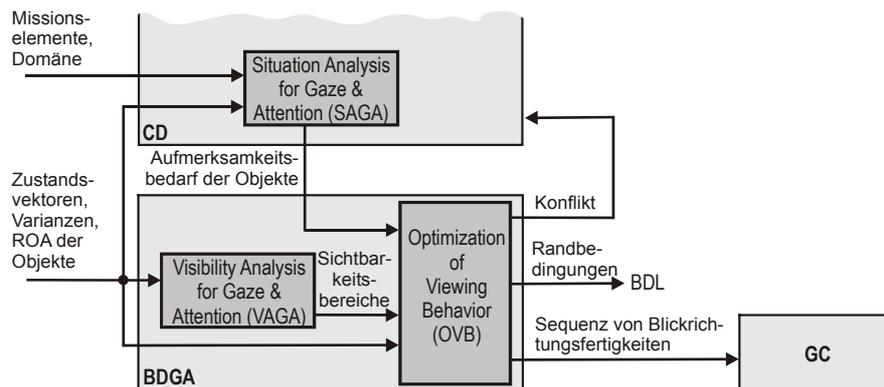


Abb. 4.3: Die an der Blickrichtungssteuerung beteiligten Prozesse CD, BDGA und GC

Der Aufmerksamkeitsbedarf ist eine kontinuierliche Größe im Intervall  $[0,1]$  und legt den prozentualen Anteil der Aufmerksamkeit fest, welcher dem Objekt im Situationskontext zugestanden wird. Der Aufmerksamkeitsbedarf bemisst die Relevanz des Objekts für die Blickrichtung, oder anders ausgedrückt, beschreibt die Notwendigkeit, über das Objekt Informationen zu sammeln. Eine Skalierung gewährleistet, dass die Summe des Aufmerksamkeitsbedarfs aller um die Blickrichtung konkurrierenden Objekte gleich 1 ist. SAGA legt z. B. beim Autobahnfahren fest, dass dem vorausfahrenden Fahrzeug je nach Zeitabstand zwischen 50 % und 100 % der Aufmerksamkeit gewidmet werden soll. Die restlichen Objekte (z. B. in der Nachbarspur fahrende Fahrzeuge oder Verkehrszeichen) teilen sich den Rest der Aufmerksamkeit. Ein Aufmerksamkeitsbedarf von 1,0 (100 %) bedeutet, dass das Objekt ausschließlich abgebildet werden soll und 0,5 (50 %) zieht eine Halbierung der Ressource Blickrichtung nach sich. Hat ein Objekt einen Aufmerksamkeitsbedarf kleiner als 0,05 (5 %), so wird es von BDGA bei der Planung nicht berücksichtigt.

Die oben beschriebene funktionelle Verknüpfung zwischen notwendiger Genauigkeit/Güte des Objektwissens und der Verteilung der Aufmerksamkeit auf die für die Blickrichtung relevanten Objekte ist sehr komplex und hängt von vielen Eingangsgrößen ab. So sind z. B. die

Eigenschaften der Wahrnehmungsprozesse und der zur Verfügung stehenden Kameras, die Güte des Vorwissens, die Leistungsdaten der Blickrichtungssteuerung und die aktuell ausgeführten Handlungen zu berücksichtigen (vgl. [PELLKOEFER & DICKMANN, 2000]). Diese funktionelle Verknüpfung ist wegen ihrer Komplexität und ihren vielen Eingangsgrößen nur schwer beschreibbar und muss durch die Auswertung experimenteller Erfahrungen und durch die Verwendung von Lernalgorithmen gefunden und angepasst werden. BDGA vergleicht dazu laufend den mit der letzten Aufmerksamkeitsverteilung erzielten Informationsgewinn, welcher mit der verbesserten Güte und Genauigkeit des Objektwissen im funktionellen Zusammenhang steht, mit dem erwarteten Informationsgewinn. Tritt hier eine systematische Diskrepanz auf, so ist die zugrundegelegte funktionelle Verknüpfung nicht korrekt und muss angepasst werden. Diese Anpassung kann bei einer Verfügbarkeit von absoluten, geometrisch aussagekräftigen Genauigkeitsmaßen (Unsicherheitsintervalle, Varianzen) durch die Wahrnehmungsprozesse in ähnlicher Weise geschehen, wie in Abschnitt 3.3.6.2 geschildert wurde.

Wie Abschnitt 4.4 zeigen wird, spezifiziert jeder Wahrnehmungsexperte sogenannte Aufmerksamkeitsbereiche (*region of attention* (ROA)) für die von ihm wahrzunehmenden Objekte. VAGA benutzt diese ROA um Sichtbarkeitsbereiche<sup>34</sup> für alle um die Ressource Blickrichtung konkurrierenden Objekte zu berechnen (vgl. Abb. 4.3). OVB ermittelt daraus eine optimale Sequenz von Blickverfolgungen von Objektgruppen, welche möglichst die Wünsche aller Wahrnehmungsexperten befriedigt. Beim Übergang von einer Objektgruppe zur nächsten erfolgt eine Sakkade. Der Server-Prozess GC ist der exekutive Teil der Blickrichtungssteuerung und führt die von BDGA berechnete Sequenz aus.

Der Rest des Kapitels 4 gliedert sich folgendermaßen: Abschnitt 4.4 beschreibt die Schnittstelle zwischen den Wahrnehmungsprozessen und der Blickrichtungssteuerung. Der Algorithmus zur Berechnung der optimalen Sequenz von Blickverfolgungen und Sakkaden ist Thema des Abschnitts 4.5. Abschnitt 4.6 behandelt die Blickrichtungsfertigkeiten des Prozesses GC. Schließlich beschreibt Abschnitt 4.7, wie die Blickrichtungsfähigkeiten in das Fähigkeitenkonzept von Kapitel 3.3 eingebettet sind.

#### 4.4 Die Schnittstelle zur Blickrichtungssteuerung

Im EMS-VISION System sind die Wahrnehmungsprozesse im verteilten System an den Orten lokalisiert, wo das benutzte Kamerasignal auch verfügbar ist (vgl. Kapitel 5.3). Ferner sind die Wahrnehmungsmodule spezialisiert: So gibt es z. B. Wahrnehmungsexperten für Straßen und Fremdfahrzeuge. Die wahrgenommenen Objekte werden durch spezielle Szenenknoten mit einem Form- und Dynamikmodell repräsentiert. Für jedes Objekt benötigt die Blickrichtungssteuerung Informationen, wie es von den Kameras am besten erfasst werden kann. Dabei spielt nicht nur der Ort der Einspeisung des Kamerasignals im verteilten System eine Rolle, auch das Kamerasignal selbst und die Auflösung des Kamerabildes muss für den Wahrnehmungsexperten verwertbar sein.

Das Wissen, wie ein Objekt abgebildet werden kann, ist die Expertise des Wahrnehmungsprozesses. Aus diesem Grund benötigt man eine Schnittstelle zur Blickrichtungssteuerung, die es den Wahrnehmungsexperten erlaubt, ihre Anforderungen und Wünsche hinsichtlich der Blickrichtung zu spezifizieren und zur Blickrichtungssteuerung zu senden.

Für die Festlegung der optimalen Abbildungsverhältnisse für ein Objekt reicht die geometrische Beschreibung der Form des Objekts nicht aus, weil sie keine Information darüber ent-

---

<sup>34</sup> Der Begriff des Sichtbarkeitsbereichs wird in Kapitel 4.5 definiert.

hält, welche Objektmerkmale wahrgenommen und welche Messmethoden bei der Wahrnehmung benutzt werden. Um dieses Problem zu lösen, spezifiziert jeder Wahrnehmungsexperte sogenannte Aufmerksamkeitsbereiche (*region of attention* (ROA)) und eine zweistufige Logik, welche die ROA kombiniert. Die ROA werden bezüglich des jeweiligen 3D-Objektkoordinatensystems beschrieben und überdecken relevante Objektteile, die für die Wahrnehmung im Kamerabild sichtbar sein müssen. Da das EMS-VISION System über mehrere Kameras verfügt, sind auch die Kameras zu referenzieren, in welchen die ROA abzubilden sind.

Die ROA werden in der ersten Logikebene mittels Operatoren unterschiedlichen Typs verknüpft. Der Operator AND besagt, dass die verknüpften ROA für eine Vermessung gleichzeitig sichtbar sein müssen. Dagegen fordert der Operator OR, dass nur einer der ROA abzubilden ist. Mit dem Operator AND\_SEQU kann der Wahrnehmungsexperte bewirken, dass die ROA innerhalb des Planungszeitraums der Blickrichtungssteuerung nacheinander (sequentiell) abgebildet werden. Die Kombinationen von ROA der ersten Logikebene sind in einer zweiten Ebene durch eine OR-Verknüpfung miteinander verbunden. Jede Kombination von ROA besitzt ein relatives Maß welches besagt, wie groß der Informationsgewinn der Kombination relativ zur besten Kombination ist. Die Kombinationen von ROA sind in der zweiten Ebene gemäß ihres Informationsgewinns geordnet, beginnend mit der besten Kombination, für die der relative Informationsgewinn definitionsgemäß 1 ist.

Anhand dieser von den Wahrnehmungsexperten gelieferten Informationen berechnet BDGA ein optimales Blickverhalten, wobei sich die Anzahl der Wahrnehmungsexperten und die Anzahl der wahrzunehmenden Objekte dynamisch ändern können.

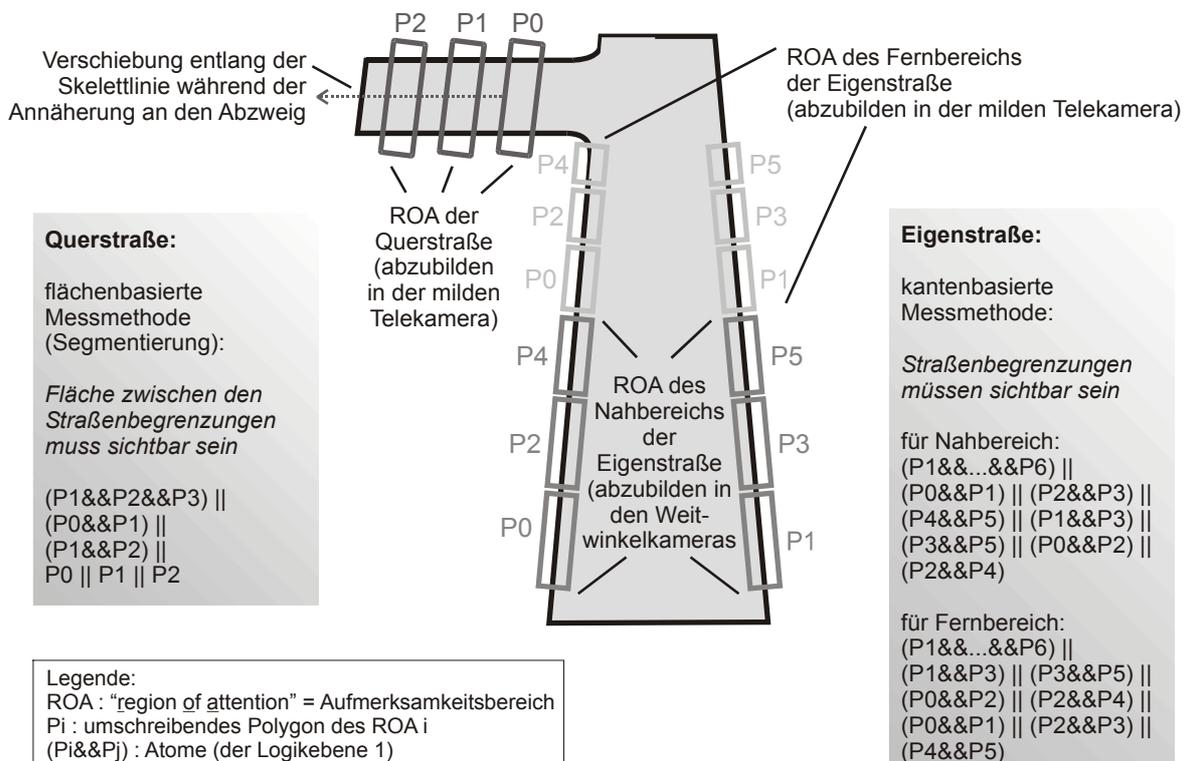


Abb. 4.4: Die Aufmerksamkeitsbereiche der Eigenstraße (im Nah- und Fernbereich) und der Querstraße beim autonomen Abbiegen

Die Schnittstelle zur Blickrichtungssteuerung soll nun anhand eines Anwendungsbeispiels erläutert werden. Abb. 4.4 zeigt schematisch, wie die ROA beim autonomen Straßefolgen und Abbiegen (von Kapitel 7.1) auf den Formmodellen positioniert sind. Das lokale und ferne

Segment der Straße, auf der das autonome Fahrzeug fährt (im folgenden Eigenstraße genannt), wird vom Straßenexperten durch eine kantenbasierte Messmethode in den Weitwinkelbildern bzw. im Telebild wahrgenommen (vgl. [LÜTZELER, 2002]). Aus diesem Grund müssen lediglich die Straßenbegrenzungen in den Kamerabildern sichtbar sein, und nicht die gesamte Straßenoberfläche zwischen diesen. Ein einzelnes Rechteck, welches die rechte und linke Straßenbegrenzung umschließt, würde (insbesondere bei einer gekrümmten Straße) die Abbildung großer Objektteile verlangen, welche der Messprozess gar nicht auswertet.

Um dies zu verhindern, ist jede Straßenbegrenzung in drei ROA unterteilt (jeweils für das nahe und ferne Segment). Die zugehörige Logik spezifiziert, dass es die beste Lösung ist, alle ROA gleichzeitig wahrzunehmen (vgl. Abb. 4.4, rechts). Ist dies nicht möglich, so gibt es weitere Lösungen: Die zweitbeste Lösung ist, die ROA 0 und 1 gleichzeitig abzubilden, die drittbeste Lösung ist die Kombination von ROA 2 und 3, usw.. Laut der Logik ist es besser, zwei auf unterschiedlichen Straßenseiten liegende ROA abzubilden als zwei auf der gleichen Seite positionierte, da dadurch die Straßenbreite geschätzt werden kann<sup>35</sup>. Ist dies jedoch nicht möglich, so kann man durch die Abbildung von zwei auf einer Straßenseite liegenden ROA zumindest die Krümmung der Straße schätzen<sup>36</sup>.

Die Querstraße, auf die das autonome Fahrzeug einbiegen soll, wird durch eine flächenbasierte Messmethode (Segmentierung) wahrgenommen [LÜTZELER, 2002]. Diese Segmentierung ist aber nur dann möglich, wenn die gesamte Oberfläche eines Abschnitts der Querstraße im Kamerabild sichtbar ist. Dementsprechend sind die ROA der Querstraße auf dem Formmodell so positioniert, dass sie die gesamte Straßenbreite überdecken. Die zweistufige Logik kombiniert wieder die ROA: Die beste Lösung ist eine gleichzeitige Abbildung aller drei ROA. Weniger gute Lösungen sind die Abbildungen von Paaren von ROA und die schlechtesten Lösungen bestehen aus der Abbildung nur eines ROA. Während des Annäherns des Fahrzeugs an den Abzweig werden die ROA der Querstraße entlang der Skelettlinie der Querstraße vom Abzweig weg verschoben, so dass die Kamera weiter in die Querstraße hineinblickt (vgl. Abb. 4.4). Einzelheiten bezüglich der Positionierung der ROA auf den Formmodellen der Eigen- und Querstraße können in [PELLKOFER et al., 2002] und [LÜTZELER, 2002] nachgelesen werden.

Abb. 4.5 zeigt das Klassendesign der Schnittstelle zur Blickrichtungssteuerung. Jeder Knoten im Szenenbaum (gemeinsame Basisklasse `CBasicSceneNode`) besitzt ein Dynamikmodell, von dem die Relativlage in den 6 Freiheitsgraden abgefragt werden kann. Ein für die Blickrichtung relevantes Szenenobjekt muss weitere Eigenschaften besitzen, welche in der Klasse `CVisObjBase` zusammengefasst sind. So beinhaltet die Klassenvariable `m_Polygons` eine Anzahl (= `m_iNoOfPolygons`) von 3D-Polygonzügen, welche die ROA des Objekts umschreiben. Eine zweistufige Logik kombiniert diese ROA. Die Variable `m_FeatureAtoms` beinhaltet die Atome der 1. Logikebene, welche Polygone verknüpfen. Diese Kombinationen von Polygonen werden in der 2. Ebene durch die Einträge in `m_FeatureEntry` nochmals verknüpft, wobei die Anzahl der Verknüpfungen durch `m_iNoOfFeatures` bekannt ist. Der Aufmerksamkeitsbedarf des Objekts (`m_fAttentionDemand`) wird von der Situationsanalyse bezüglich Blickrichtung SAGA bemessen.

<sup>35</sup> Die ROA des linken Straßenrandes können dabei mit der linken Weitwinkelkamera abgebildet werden und die ROA des rechten Straßenrandes mit der rechten Weitwinkelkamera.

<sup>36</sup> Die beiden letzten Aussagen gelten nur für das lokale Segment der Eigenstraße. Für das ferne Segment gelten die umgekehrten Aussagen: Hier hat die Schätzung der Krümmung des Straßenverlaufs eine höhere Priorität als die Schätzung der Straßenbreite und ihrer Änderung, da die Straßenkrümmung in der großen Vorausschauweite der Telekamera besser beobachtbar ist.

Die Basisklasse CPolygon3DEx besitzt neben den allgemeinen Eigenschaften eines 3D-Polygons eine ID ( $m\_iID$ ), mit der ihre Objekte in der Logik  $m\_FeatureAtoms$  referenziert werden können. Die Klassenvariable  $m\_iCameraID$  gibt an, in welcher Kamera das Polygon abzubilden ist. Ist die Kamera-ID nicht spezifiziert, so sind immer diejenigen Kameras zu verwenden, deren Bilder auf dem Rechner eingezeichnet werden, auf dem sich der Wahrnehmungsexperte befindet.

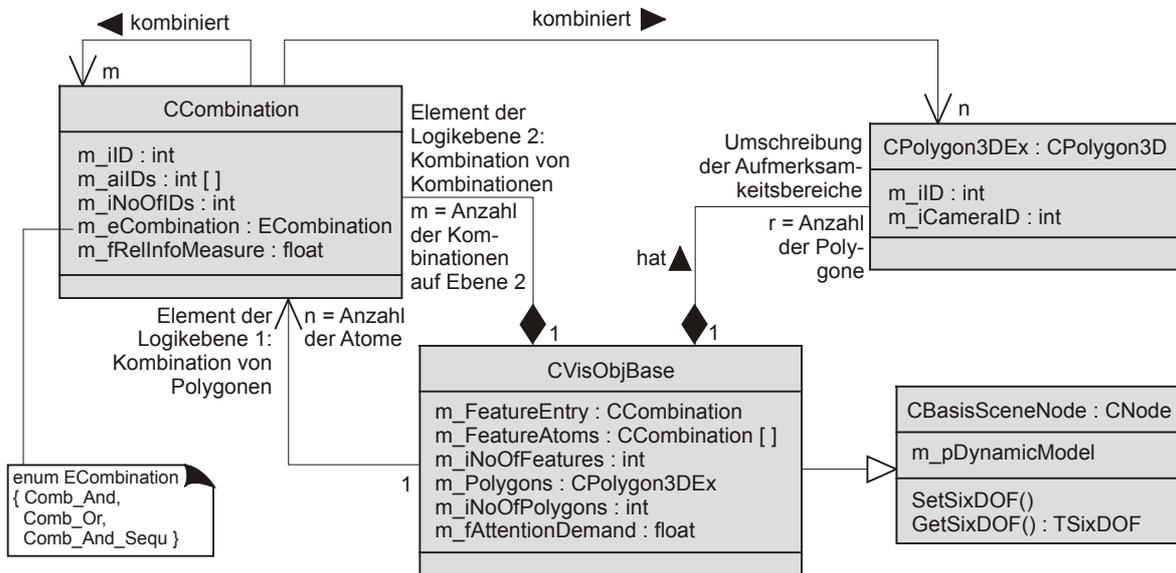


Abb. 4.5: Die Schnittstellenklasse CVisObjBase und ihre Klassenvariablen (Darstellung nach UML, vgl. auch [LÜTZELER, 2002])

Die Klasse CCombination dient der Spezifikation einer Kombination. Dabei können die kombinierten Objekte Polygone sein (Logikebene 1), oder selbst wieder Kombinationen (Logikebene 2). Die Objekte der Klasse CCombination verfügen mit  $m\_iID$  über eine eigene ID und referenzieren mit  $m\_iIDs$  die kombinierten Objekte. Die Variable  $m\_eCombination$  gibt den Typ des Verknüpfungsoperators an und  $m\_fRelInfoMeasure$  den relativen Informationsgewinn der Kombination (vgl. Seite 106). Das in Abb. 4.5 dargestellte Klassendesign wurde in Zusammenarbeit mit LÜTZELER [2002] entwickelt und von diesem implementiert.

## 4.5 Der Algorithmus zur Blickrichtungssteuerung

Dieser Absatz beschreibt den Algorithmus zur Blickrichtungssteuerung, mit welchem die Vorgaben und Wünsche der verschiedenen Wahrnehmungsexperten weiterverarbeitet und berücksichtigt werden. Das Ergebnis dieses Algorithmus ist eine optimale Sequenz von Verfolgungsphasen und Sakkaden.

Der Algorithmus zur Blickrichtungssteuerung läuft innerhalb des Prozesses BDGA ab und gliedert sich in eine Planungsphase und eine Ausführungsphase. Das Ziel der Planungsphase ist das Finden eines Kompromisses in Form einer Sequenz von Blickverfolgungen und Sakkaden, welcher die Bedürfnisse aller Wahrnehmungsexperten im System zufrieden stellt. In der Ausführungsphase wird der gefundene Kompromiss unter Verwendung der vom Prozess GC angebotenen Blickrichtungsfertigkeiten ausgeführt (vgl. Abschnitt 4.6). Wie Abb. 4.7 zeigt, legt BDGA für jedes für die Blickrichtung relevante Szenenobjekt ein Verwaltungsobjekt der Klasse CVAGAObject an. Diese Verwaltungsobjekte heißen im folgenden VAGA-Objekte.

In der Mitte von Abb. 4.8 ist die Klasse CVAGAObject dargestellt. Sie hat mit m\_pNode einen Verweis auf den zugehörigen Szenenknoten, von dem erwartet wird, dass dessen Klasse die Basisklasse CVisObjBase besitzt und damit die in Abschnitt 4.4 beschriebene Schnittstelle zur Blickrichtungssteuerung unterstützt. Die Klassenvariable m\_Cameras ist eine Liste von Referenzen auf Kameras, welche für die Abbildung des Objekts herangezogen werden können. In Abb. 4.10 ist der sequentielle Ablauf des Algorithmus dargestellt, anhand dessen nun die schrittweise Erläuterung des Algorithmus erfolgt.

#### 4.5.1 Die Planungsphase

Wenn keine Sequenz vorliegt oder die letzte geplante Sequenz vollständig ausgeführt oder verworfen wurde, beginnt im nächsten Zyklus die Planungsphase des Algorithmus. Im ersten Schritt der Planungsphase wird der Aufmerksamkeitsbedarf aller Objekte skaliert und in der Klassenvariablen m\_fAttentionDemand der Klasse CVAGAObject abgelegt, so dass die Summe gleich eins ist. Im zweiten Schritt der Planungsphase berechnet man die Sichtbarkeitsbereiche aller Aufmerksamkeitsbereiche (ROA). Der Sichtbarkeitsbereich eines ROA sind die Winkelintervalle für die Gier- und Nickachse des Kameraträgers, innerhalb deren der Aufmerksamkeitsbereich im gewünschten Kamerabild sichtbar ist (vgl. Abb. 4.6). An den Rändern des Sichtbarkeitsbereichs berührt eine Ecke oder Kante des ROA den Bildrand.

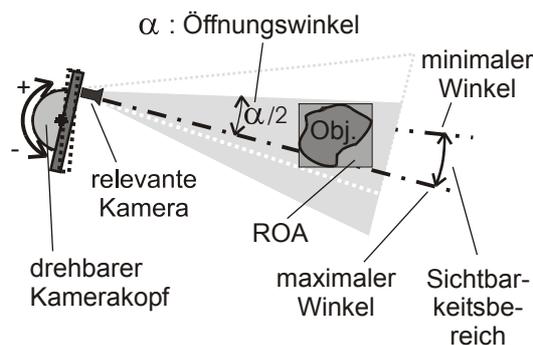


Abb. 4.6: Die Definition des Sichtbarkeitsbereichs

Im dritten Schritt werden unter Anwendung der vom Wahrnehmungsexperten spezifizierten zweistufigen Logik die Sichtbarkeitsbereiche der ROA kombiniert und ihre Schnittmengen berechnet. Für jede gefundene Schnittmenge trägt man ein Objekt der Klasse CVisibilityRange in die Liste CVAGAObject::m\_VisibilityRanges ein. Ein Objekt der Klasse CVisibilityRange bezieht nicht nur den Winkelbereich der Schnittmenge (m\_afAngles), sondern auch die Kombination, aus der diese entstanden ist (m\_aiCombination) (vgl. Abb. 4.8, links oben). Verlangt eine Kombination die sequentielle Abbildung zweier ROA (durch die Verknüpfung mit dem Operator AND\_SEQ, vgl. Abschnitt 4.4), so gibt die Variable m\_iPartNolnSequDeal an, welchen Teil der Kombination der Sichtbarkeitsbereich berücksichtigt.

Auf diese Weise wird die Information, welche sich aus der zweistufigen Logik und aus der räumlichen Lage der ROA ergibt, in eine Liste von Sichtbarkeitsbereichen transferiert. Die Liste ist nach dem Informationsgewinn der Kombinationen geordnet. Je weiter oben eine Kombination in der Liste steht, desto mehr Information kann über das Objekt durch die Abbildung gewonnen werden. So ist die Fixation des ersten Listeneintrages die beste Lösung für das Objekt alleine, ermöglicht jedoch nur ein Minimum an Variation der Blickrichtung, um evtl. andere Objekte mit zu berücksichtigen.

Im vierten Schritt der Planungsphase berechnet man Schnittmengen zwischen Sichtbarkeitsbereichen verschiedener VAGA-Objekte. Durch die oben dargestellte Sortierung der Listen nach dem Informationsgewinn und durch heuristische Abbruchkriterien ist gewährleistet, dass die Anzahl der Schnittmengenberechnungen möglichst klein ist, aber trotzdem die optimale Lösung gefunden wird. Für jede Schnittmenge legt man ein Objekt der Klasse COVObject an, welches eine Gruppe gleichzeitig wahrnehmbarer VAGA-Objekte repräsentiert (vgl. Abb. 4.8, links unten). Ein OVB-Objekt beinhaltet den Sichtbarkeitsbereich der Schnittmenge (`m_IntersectionRange`), referenziert die bei der Schnittmengenbildung beteiligten Sichtbarkeitsbereiche der VAGA-Objekte (`m_IntersectionReference`) und berechnet den Informationsgewinn, der aus der Kombination von Sichtbarkeitsbereichen resultiert (`m_fInformationInput`). Der Informationsgewinn eines OVB-Objekts ergibt sich aus folgender Gleichung:

$$I_{OVB}(k) = \sum_i^{n_{VAGA}(k)} (A_i \cdot \sigma_i^j) \quad (\text{Gl. 4.1})$$

- $I_{OVB}(k)$ : Informationsgewinn durch OVB-Objekt  $k$   
 $A_i$ : Aufmerksamkeitsbedarf des VAGA-Objekts  $i$   
 $\sigma_i^j$ : relativer Informationsgewinn<sup>37</sup> der Kombination  $j$  des VAGA-Objekts  $i$   
 $n_{VAGA}(k)$ : Anzahl der vom OVB-Objekt  $k$  berücksichtigten VAGA-Objekte

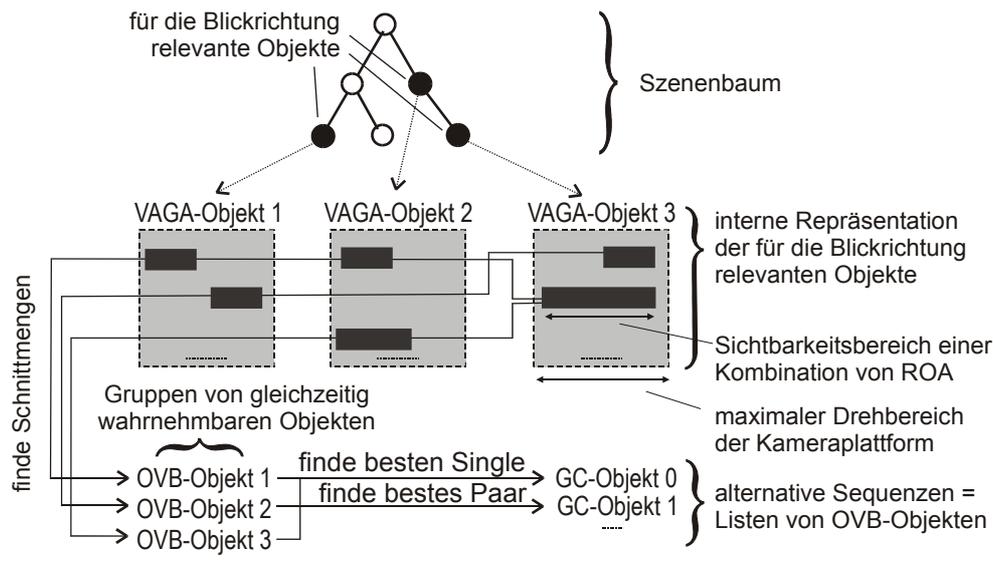


Abb. 4.7: Die Bedeutungen der dynamischen Objekte des Algorithmus

Nach dem Aufbau einer Tabelle von OVB-Objekten, wird eine Liste von Objekten der Klasse CGCObject generiert. Dies geschieht anhand folgender Anweisungen:

- Suche ein OVB-Objekt, welches alle VAGA-Objekte berücksichtigt und den höchsten Informationsgewinn einbringt. Diese Lösung wird ‚Best Single‘-Lösung genannt und stellt ein Blickverhalten ohne Sakkaden dar (abgesehen von einer evtl. notwendigen Sakkade zu Beginn der Ausführungsphase (Anfangssakkade)). Für diese Lösung erzeugt man ein Objekt der Klasse CGCObject und trägt es in eine Liste ein.

<sup>37</sup> niedergelegt in CCombination::m\_fRelInfoMeasure (vgl. Abb. 4.5, links).

- Suche das Paar von OVB-Objekten mit dem höchsten Informationsgewinn, welches alle VAGA-Objekte berücksichtigt. Auch für diese Lösung – genannt die ‚Best Pair‘-Lösung – generiert man ein GC-Objekt. Die ‚Best Pair‘-Lösung stellt eine Lösung dar, bei der nacheinander zwei Objektgruppen (OVB-Objekte) fixiert werden. Beim Wechsel zwischen diesen erfolgt eine Sakkade.
- Wenn keine ‚Best Single‘-Lösung existiert, wird auf analoge Weise die ‚Best Triple‘-Lösung – eine Gruppe von drei OVB-Objekten mit maximalem Informationsgewinn – gesucht und in die Liste von GC-Objekten eingetragen. Existiert eine ‚Best Single‘-Lösung, so verzichtet man auf die Berechnung der ‚Best Triple‘-Lösung, da man davon ausgehen kann, dass eine Lösung mit einer um zwei höheren Sakkadenanzahl einen geringeren Informationsgewinn erzielt.

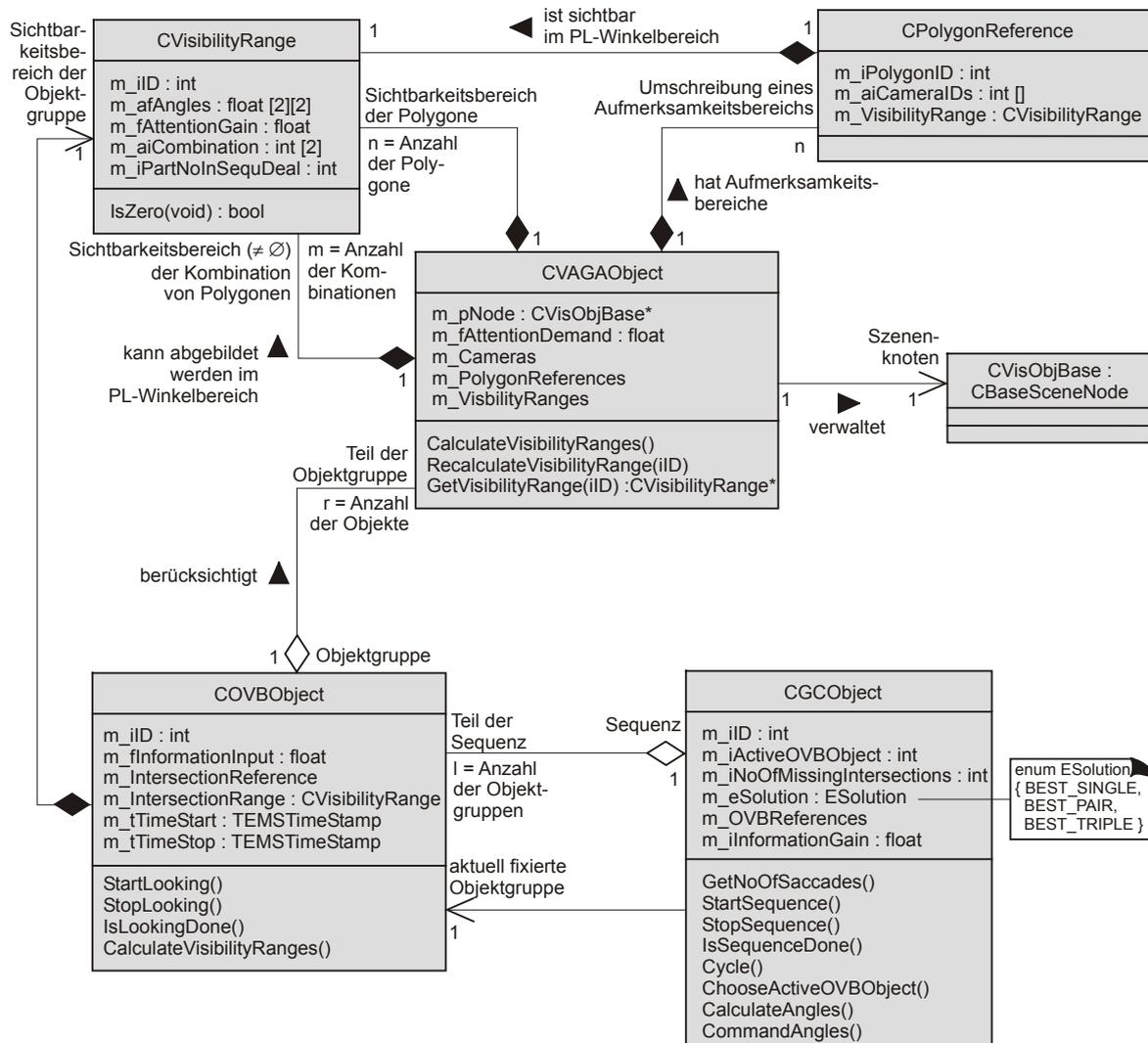


Abb. 4.8: Das Klassendiagramm des Algorithmus (nach UML)

Da während einer Sakkade mit den visuellen Sensoren keine Information über die Umwelt aufgenommen werden kann, reduziert sich der Informationsgewinn einer Sequenz mit zunehmender Anzahl von Sakkaden. Aus diesem Grund ist bei einem Planungszeitraum von einigen Sekunden nur eine beschränkte Anzahl von Sakkaden sinnvoll<sup>38</sup>. Die oben dargestellte Suche

<sup>38</sup> In den Anwendungen von Kapitel 6 und 7 war der Planungszeitraum der Blickrichtungssteuerung 4 s und die maximale Anzahl der Sakkaden pro Planungszeitraum drei (inklusive Anfangssakkade).

ermittelt deshalb Kombinationen mit maximal drei OVB-Objekten. Ein GC-Objekt repräsentiert die beste Kombination von OVB-Objekten für eine bestimmte Anzahl von OVB-Objekten. Der Informationsgewinn eines GC-Objekts berechnet sich durch folgende Gleichung:

$$I_{GC}(k) = \sum_i^{n_{OVB}(k)} (I_{OVB}^i \cdot T_{Fix}^i) \quad (\text{Gl. 4.2})$$

- $I_{GC}(k)$ : Informationsgewinn durch GC-Objekt  $k$   
 $I_{OVB}^i$ : Informationsgewinn des OVB-Objekts  $i$   
 $T_{Fix}^i$ : effektive Fixationszeit des OVB-Objekts  $i$   
 $n_{OVB}(k)$ : Anzahl der OVB-Objekte des GC-Objekts  $k$

Obige Formel geht davon aus, dass der Informationsgewinn linear mit der Fixationszeit größer wird. Diese Modellierung der Zeitabhängigkeit lässt sich nicht für alle Fixationszeiten rechtfertigen. Zu Beginn der Fixationszeit eines Objekts kommt es bei einer Bildverarbeitung, welche auf einer Zustandsschätzung durch Kalman-Filter beruht, zu Einschwingvorgängen. Geht man davon aus, dass diese Einschwingvorgänge nach einer Zeitperiode  $T_1$  abgeklungen sind, so beinhalten die Resultate der Zustandsschätzung erst nach Ablauf dieser Zeitperiode aussagekräftige Objektdaten. Durch Fixationen mit einer Dauer kleiner als  $T_1$  können deshalb keine Objektinformationen gewonnen werden. Typischerweise sind Einschwingvorgänge nach ca. 10 Messzyklen abgeklungen, so dass  $T_1$  mit  $10 \cdot 40 \text{ ms} = 0,4 \text{ s}$  abgeschätzt werden kann. Der Wert  $T_1$  hängt jedoch von den Schätzparametern des Kalman-Filters und der Güte des a-priori-Wissens ab.

Auf der anderen Seite verbessert sich das Objektwissen nach einer bestimmten Fixationsdauer  $T_2$  nicht mehr. Der Zugewinn an Objektinformation pro Zeiteinheit vermindert sich deshalb ab diesem Zeitpunkt und geht gegen Null. Dies gilt insbesondere bei statischen Objekten, bei denen man davon ausgehen kann, dass deren Zustand nach 50 Messzyklen (d. h.  $T_2 = 50 \cdot 40 \text{ ms} = 2 \text{ s}$ ) bekannt ist.  $T_2$  hängt zum einen vom Typ und der Dynamik des wahrgenommenen Objekts ab; zusätzlich spielt es eine Rolle, welche Zustandsgrößen des Objekts geschätzt werden.

Im Zeitintervall  $[T_1, T_2]$  findet die eigentliche Aufnahme der Objektinformation statt. Die in Gl. 4.2 formulierte lineare Zunahme des Objektwissens mit der Dauer der Fixation ist in dieser Zeitperiode gerechtfertigt. Bei einer ‚Best Pair‘- oder ‚Best Triple‘-Lösung (vgl. oben) beträgt die Dauer einer Fixation – bei ungefähr gleich langen Fixationsphasen – zwischen 1 und 2 Sekunden. Diese Werte liegen im Intervall  $[T_1 = 0,4 \text{ s}, T_2 = 2 \text{ s}]$ , was die Verwendung des linearen Zusammenhangs von Gl. 4.2 rechtfertigt. Ferner ist es möglich, dass die Blickrichtungssteuerung die Einschwingphase berücksichtigt, indem sie bei der Planung die Zeitperiode  $T_1$  zur Sakkadendauer  $T_{Sac}$  (vgl. Gl. 4.5) hinzuaddiert.

Die in Gl. 4.2. berücksichtigte effektive Fixationszeit  $T_{Fix}^i$  des OVB-Objekts  $i$  ergibt sich aus folgendem Zusammenhang:

$$T_{Fix}^i = \frac{I_{Komb}^i}{\sum_j^{n_{OVB}} I_{Komb}^j} \cdot (T - \sum_m^{n_{OVB}} T_{Sac}^m) \quad (Gl. 4.3)$$

- $T$  : Planungszeitraum der Blickrichtungssteuerung (4 s)
- $T_{Sac}^m$  : Dauer der Sakkade beim Wechsel von OVB-Objekt  $m$  zu  $m+1$
- $I_{Komb}^i$  : Maximaler Informationsgewinn einer Kombination von ROA innerhalb des OVB-Objekts  $i$

$I_{Komb}^i$  ergibt sich wiederum aus

$$I_{Komb}^i = \text{Max}_{k \in K_i} \{ A_n \cdot \sigma_n^k \} \quad (Gl. 4.4)$$

- $A_n$  : Aufmerksamkeitsbedarf des VAGA-Objekts  $n$
- $\sigma_n^k$  : relativer Informationsgewinn der Kombination  $k$  des VAGA-Objekts  $n$
- $K_i$  : Menge der Kombinationen von ROA innerhalb der VAGA-Objekte, die durch OVB-Objekt  $i$  berücksichtigt werden

und die Dauer einer Sakkade  $T_{Sac}$  wird durch folgende lineare Näherung ermittelt:

$$T_{Sac} = T_{Sac}^0 + \Delta T_{Sac} \cdot \Delta A \quad (Gl. 4.5)$$

- $T_{Sac}^0$  : Zeit für die Initialisierung einer Sakkade (~ 170 ms)
- $\Delta T_{Sac}$  : Inkrementelle Änderung der Sakkadendauer pro Grad Sakkadenamplitude (~ 3,7 ms/°)
- $\Delta A$  : Amplitude der Sakkade [°]

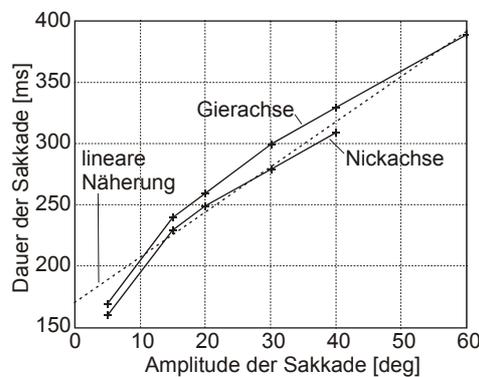


Abb. 4.9: Sakkadendauer in Abhängigkeit von der Sakkadenamplitude<sup>39</sup>

Die lineare Näherung von Gl. 4.5 ergibt sich aus einer Untersuchung des eingebetteten Systems zur Blickrichtungssteuerung des Versuchsfahrzeugs VAMORS (vgl. [PELLKOFER & DICKMANN, 2000]). Darin wurde die Abhängigkeit der Sakkadendauer von der Sakkaden-

<sup>39</sup> An dieser Stelle sei bemerkt, dass es nicht der Schwerpunkt dieser Arbeit war, die Sakkadendauer zu minimieren. Dementsprechend liegen die Werte erheblich über den Werten, wie sie von SCHIEHLEN [1995] erzielt wurden. SCHIEHLEN [1995] betrieb ähnliche Kameraplattformen mit einer oder maximal 2 Kameras, wobei diese nahe den Drehachsen angebracht waren. Ferner lief die Regelung der Kameraplattform zusammen mit der Bildverarbeitung auf der gleichen Hardware (Transputer-Netz). SCHIEHLEN [1995] konnte mit diesem System Sakkadendauern von ca. 60 ms (bei einer Amplitude von ca. 20°) erzielen. Dagegen befinden sich bei dem hier benutzten Kameraaufbau die Weitwinkelkameras in einem Abstand von ca. 12 cm von der Gierachse (vgl. Abb. 4.1 b). Da das Trägheitsmoment mit dem Quadrat der Entfernung zwischen Massepunkt und Drehachse zunimmt, wirkt sich dies bei gleich bleibendem Drehmoment sehr negativ auf die Schnelligkeit der Sakkaden aus.

amplitude experimentell ermittelt. Die Kameraplattform war dabei mit 3 Kameras inklusive den Videokabeln bestückt, so wie sie in Abb. 4.1 b dargestellt ist. In Abb. 4.9 sind die Ergebnisse dieser Messungen aufgetragen. Aus der Abbildung geht hervor, dass die Dauer der Sakkade mit zunehmender Amplitude zunimmt. Hinzu kommt eine konstante Zeitperiode für die Initialisierung der Sakkade, welche auch bei kleinen Amplituden auftritt und z. B. die Kommunikationstotzeiten zwischen dem Prozess GC und dem eingebetteten System zur Blickrichtungssteuerung beinhaltet. In Abb. 4.9 ist auch die in Gl. 4.5 formulierte lineare Näherung gepunktet eingezeichnet.

In Abb. 4.8 (rechts unten) ist die Klasse CGCObject dargestellt. In ihrer Klassenvariable  $m\_iInformationGain$  wird der Informationsgewinn  $I_{GC}$  abgelegt. In  $m\_OVReferences$  sind Referenzen auf diejenigen OVB-Objekte eingetragen, die Teil der Sequenz sind. Die Variable  $m\_iActiveOVBObject$  gibt an, welches dieser OVB-Objekte zur Zeit fixiert wird und  $m\_eSolution$  bezeichnet den Typ der Lösung.

Nach der Berechnung der Liste von GC-Objekten initialisiert der Algorithmus das GC-Objekt mit dem größten Informationsgewinn. Die Anzahl der Sakkaden ist gleich der Zahl der vom GC-Objekt berücksichtigten OVB-Objekte  $n_{OVB}$  (incl. Anfangssakkade).

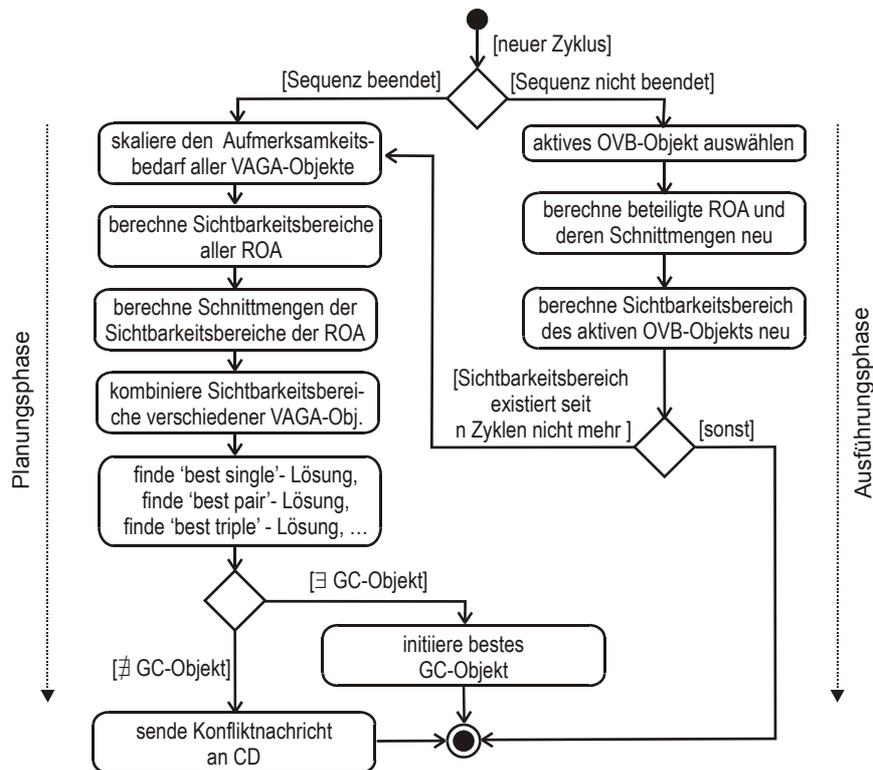


Abb. 4.10: Der sequentielle Ablauf des Algorithmus

Konnte in der Planungsphase keine Lösung, d. h. kein GC-Objekt gefunden werden, so sendet BDGA eine Konfliktnachricht an CD (vgl. Abb. 4.10). Es ist dann die Aufgabe von CD, in Abhängigkeit vom Situations- und Missionskontext die Anzahl der für die Blickrichtung relevanten Objekte zu reduzieren. Erreicht BDGA von CD keine konfliktauflösende Nachricht, so ignoriert BDGA das Objekt mit dem geringsten Aufmerksamkeitsbedarf und stößt eine Neuplanung der Blickrichtung an.

### 4.5.2 Die Ausführungsphase

Ist eine Blickrichtungssequenz zu Beginn eines neuen Zyklus noch nicht beendet, so durchläuft der Algorithmus die Ausführungsphase (vgl. Abb. 4.10). Im ersten Schritt der Ausführungsphase wird getestet, ob die Fixationszeit  $T_{Fix}$  des aktiven OVB-Objekts abgelaufen ist. Ist dies der Fall, so beendet der Algorithmus die Blickfixation dieses OVB-Objekts und startet die Fixation des in der Sequenz nachfolgenden OVB-Objekts. War das bisher aktive OVB-Objekt das letzte in der Sequenz, so ist die Sequenz beendet und im nächsten Zyklus erfolgt eine neue Planungsphase.

Im nächsten Schritt der Ausführungsphase berechnet man den Sichtbarkeitsbereich des aktiven OVB-Objekts – welcher in `COVBOject::m_IntersectionRange` niedergelegt ist – neu. Dabei sind auch die daran beteiligten Sichtbarkeitsbereiche der ROA der VAGA-Objekte und ihre Schnittmengen neu zu berechnen. Durch die Speicherung der Kombinationen der Sichtbarkeitsbereiche und ROA in den Klassenvariablen `COVBOject::m_IntersectionReference` und `CVisibilityRange::m_aiCombination` ist gewährleistet, dass nur notwendige Daten neu berechnet werden.

Existiert die Schnittmenge des aktiven OVB-Objekts weiterhin, so fixiert die unterlagerte Blickrichtungsfertigkeit den Mittelpunkt der Schnittmenge. Ist die Schnittmenge durch die Relativbewegungen der wahrgenommenen Objekte verschwunden, so erfolgt eine Inkrementierung des Zählers `CGCObject::m_iNoOfMissingIntersections` um 1 und die Plattformposition verweilt in der zuletzt eingenommenen Position. Bleibt die Schnittmenge für mehrere Zyklen verschwunden, so erreicht der Zähler einen Schwellwert<sup>40</sup> und es erfolgt ein Übergang zum nächsten OVB-Objekt in der Sequenz.

Zusammenfassend lässt sich feststellen, dass bei der Berechnung der optimalen Sequenz von Sakkaden und Blickfixationen folgende Kriterien Berücksichtigung finden:

- Der Aufmerksamkeitsbedarf der Szenenobjekte, der diesen von der Situationsanalyse zugestanden wird;
- die Anzahl der ROA, die während der Verfolgungsphasen abgebildet werden;
- der relative Informationsgewinn der alternativen Kombinationen von ROA und
- die Anzahl der während der Sequenz notwendigen Sakkaden, deren Amplituden und Dauer.

## 4.6 Die Fertigkeiten der Blickrichtungssteuerung

Der Prozess Gaze Control (GC) ist der ausführende Teil der Blickrichtungssteuerung (vgl. Abschnitt 4.3) und bietet Blickrichtungsfertigkeiten an, mit deren Hilfe die von BDGA getroffenen Entscheidungen ausgeführt werden können.

Die Basisklasse für alle Blickrichtungsfertigkeiten ist die Klasse `CGazeSkill` (vgl. Abb. 4.11). Jede Blickrichtungsfertigkeit besitzt einen Satz von Reglerparametern, die bei der Initialisierung der Fertigkeit (`Initialize()`) an das eingebettete System zur Blickrichtungssteuerung gesandt werden (vgl. Abschnitt 4.2). Die Reglerparameter stammen aus einer speziell auf die Fertigkeit abgestimmten Initialisierungsdatei (`m_plnifile`). Die Arrays `m_afSetAngles` und `m_afSetVelocities` nehmen die Sollgrößen auf, die der Fertigkeit durch BDGA vorgegeben werden. Mit `m_pCameraCarrier` besitzt die Blickrichtungsfertigkeit eine Referenz auf den Szenenknoten des Kameraträgers. Diese Referenz dient u. a. dazu, die Winkellage des Kameraträgers im Szenenbaum laufend zu aktualisieren und eine Sakkade anzuzeigen. In

<sup>40</sup> In den Anwendungen von Kapitel 7 war der Schwellwert i. d. R. mit 10 bemessen.

`m_atCarrierModus` ist der gegenwärtige Reglermodus für die zwei Achsen eingetragen. Die Klassenvariable `m_iSuccess` spezifiziert, ob die Fertigkeit erfolgreich abgeschlossen oder fehlgeschlagen ist.

Alle Blickrichtungsfertigkeiten sind von der Basisklasse `CGazeSkill` abgeleitet und gliedern sich in einfache und komplexe (zusammengesetzte) Fertigkeiten. Bei der (einfachen) Fertigkeit `CRemote` kann man von außen (z. B. über einen Joystick) Sollgrößen vorgeben. `CPause` ist immer dann aktiv, wenn keine andere Blickrichtungsfertigkeit ausgewählt wurde (*default*-Fertigkeit). In `CPause` verharret die Plattform in der zuletzt eingenommenen Position. Mit `CFunction` lassen sich funktionelle Sollwertverläufe für Winkel oder Winkelgeschwindigkeit vorgeben (z. B. Sinus-, Dreieck- oder Rechteckfunktionen).

Die mittlere Zeile in Abb. 4.11 zeigt die während des autonomen Betriebs genutzten einfachen Blickrichtungsfertigkeiten. Mit `CScan` kann eine gewisse Winkelposition unter Einhaltung einer vorgegebenen Winkelgeschwindigkeit angefahren werden. Die Variable `m_tTimeLeft` spezifiziert eine Zeitperiode, in der die Blickrichtungsfertigkeit abgeschlossen sein muss. Nach Ablauf dieser Zeitperiode zeigt `m_iSuccess` ein Scheitern an. Mit `Initialize(TScan&)` erfolgt die Initialisierung und Parametrisierung der Fertigkeit. Die Funktion `IsReached()` gibt an, ob die Zielposition bereits erreicht und die Fertigkeit damit beendet ist.

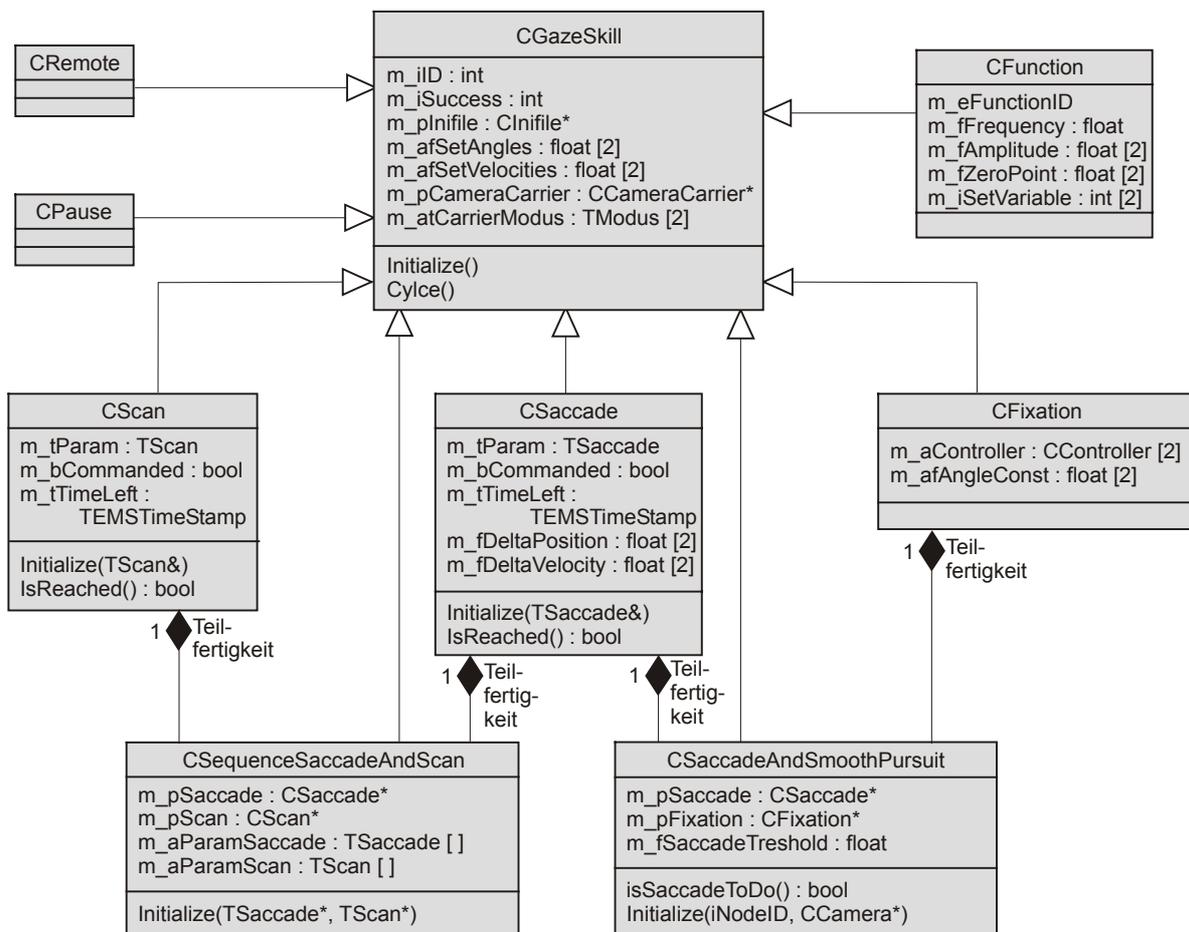


Abb. 4.11: Die vom Prozess GC angebotenen Blickrichtungsfertigkeiten (Klassendiagramm nach UML)

Die Blickrichtungsfertigkeit `CSaccade` führt eine schnelle Blickzuwendung aus. Deren Initialisierung und Verwaltung geschieht nach dem gleichen Schema wie bei `CScan`. Eine Sak-

kade gilt als beendet, wenn die Istwerte für Winkel und Winkelgeschwindigkeit sich den entsprechenden Sollgrößen mit der Genauigkeit `m_fDeltaPosition` bzw. `m_fDeltaVelocity` angenähert haben.

Die Fertigkeit `CFixation` realisiert eine Blickfixation eines Raumwinkels. Dabei baut ein Regler (`m_aController`) die Abweichung zwischen Soll- und Istwinkel ab. Als Regler stehen ein P-Regler und ein PI-Regler zur Verfügung<sup>41</sup>. Mit `m_afAngleConst` kann für eine der Drehachsen des Kameraträgers ein konstanter Sollwinkel vorgegeben werden. Der von BDGA für diese Achse berechnete Sollwinkel bleibt dann unberücksichtigt.

Die bisher beschriebenen, einfachen Blickrichtungsfertigkeiten sind Teilfertigkeiten von komplexeren Blickrichtungsfertigkeiten. Die komplexe Fertigkeit `CSequenceSaccadeAndScan` kann eine Sequenz von Sakkaden und Phasen mit konstanter Winkelgeschwindigkeit ausführen (vgl. Abschnitt 4.7). `CSequenceSaccadeAndScan` wird von der Blickrichtungsfähigkeit *3D-Search* benutzt und ist von den Teilfertigkeiten `CSaccade` und `CScan` abhängig. Für jede dieser Teilfertigkeiten sind bei der Initialisierung entsprechende Parameter vorzugeben (`m_aParamSaccade` und `m_aParamScan`). Nach erfolgreicher Beendigung der Sakkade wird automatisch die daran anschließende Bewegung mit konstanter Winkelgeschwindigkeit eingeleitet. Bei jedem Wechsel zwischen den einfachen Blickrichtungsfertigkeiten erhalten die Regler auf dem eingebetteten System neue Parameter.

Die zweite komplexe Blickrichtungsfertigkeit `CSaccadeAndSmoothPursuit` kommt innerhalb der Blickrichtungsfähigkeit *OVB* zur Anwendung (vgl. Abschnitt 4.7). Auch diese Fertigkeit benutzt zwei elementare Teilfertigkeiten: `CSaccade` und `CFixation`. Solange die Abweichung zwischen Soll- und Istwinkel während einer Fixation einen gewissen Schwellwert<sup>42</sup> (`m_fSaccadeTreshold`) nicht überschreitet, ist nur die Teilfertigkeit `CFixation` aktiv. Beim Überschreiten dieses Schwellwerts, erfolgt durch die Teilfertigkeit `CSaccade` ein schneller Abbau der Abweichung. Nach Beendigung der Sakkade wird die Fixation fortgeführt.

## 4.7 Die Fähigkeiten der Blickrichtungssteuerung

Zum Schluss dieses Kapitels sollen die Fertigkeiten und schematischen Fähigkeiten bezüglich Blickrichtung des EMS-VISION Systems dargestellt und diskutiert werden. Abb. 4.12 zeigt den entsprechenden Teil des Fähigkeitsnetzes (vgl. auch Abschnitt 3.3.2). Den untersten Teil des Fähigkeitsnetzes bilden die Aktuatoren (Motoren) des Kameraträgers. Sie sind von keinen weiteren Systemkomponenten abhängig<sup>43</sup>. Der Experte der Aktuatoren ist das Regelungsprogramm auf dem eingebetteten System zur Blickrichtungssteuerung (vgl. Abschnitt 4.2). Die Aktuatoren bekommen ihre Parameter und Sollgrößen durch die in Abschnitt 4.6 beschriebenen, einfachen Blickrichtungsfertigkeiten *Fixation*, *Saccade* und *Scan* zugesandt. Diese Fertigkeiten sind von den Aktuatoren abhängig. Entsprechend den Ausführungen in Abschnitt 3.3.2 werden die Abhängigkeiten im Fähigkeitsnetz durch gerichtete Kanten repräsentiert. Die einfachen Blickrichtungsfertigkeiten sind Teilfertigkeiten von den komplexeren Blickrichtungsfertigkeiten *Saccade & SmoothPursuit* und *Sequence Saccade & Scan*, mit denen Sequenzen von Sakkaden und Blickverfolgungen bzw. Sakkaden und Bewegungen mit konstanter Winkelgeschwindigkeit ausgeführt werden können (vgl. Abschnitt 4.6). Der aus-

<sup>41</sup> Komplexere Regleralgorithmen für eine hochdynamische Objektverfolgung mit Kameras stellt SCHIEHLEN [1995] vor.

<sup>42</sup> In den Anwendungen von Kapitel 6 und 7 war dieser Schwellwert mit 5° bemessen.

<sup>43</sup> Die Abhängigkeit aller Systemmodule von der elektrischen Stromversorgung ist nicht im Fähigkeitsnetz repräsentiert.

führende Prozess und damit der Experte aller Blickrichtungsfertigkeiten ist Gaze Control (GC).

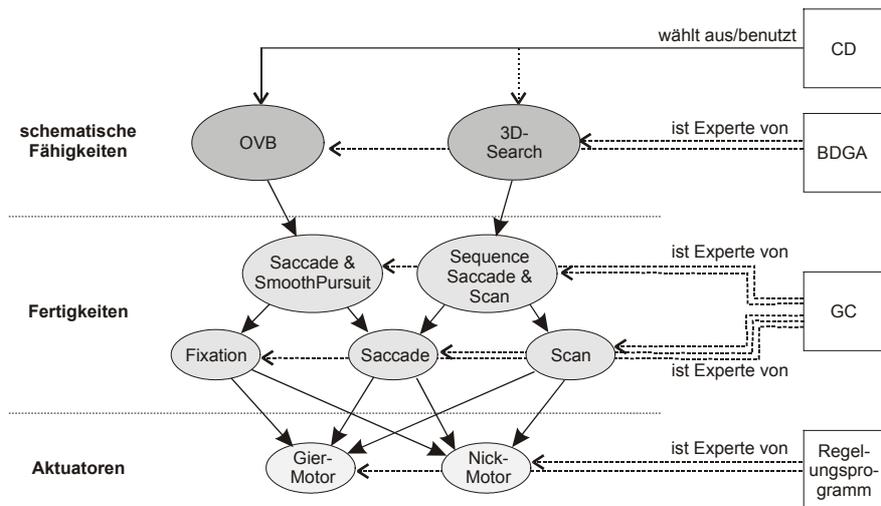


Abb. 4.12: Die Fähigkeitsnetz bezüglich Blickrichtung

Der in Abschnitt 4.5 dargestellte Algorithmus plant eine Sequenz von Sakkaden und Blickverfolgungen und läuft innerhalb des Prozesses BDGA ab. Der Algorithmus wird im Fähigkeitsnetz durch den Fähigkeitsknoten *OVB* (*Optimization of Viewing Behavior*) repräsentiert. Durch die Repräsentation im Fähigkeitsnetz ist der Algorithmus für die zentrale Entscheidungsinstanz (CD) nutzbar und parametrisierbar. Die Fähigkeit *OVB* ist von der Blickrichtungsfertigkeit *Saccade & SmoothPursuit* abhängig und besitzt als Parameter die maximale Anzahl von Sakkaden, den Planungshorizont, evtl. eine konstante Winkelvorgabe für eine der Plattformachsen und die Möglichkeit, eine Neuplanung der Blickrichtung anzustoßen. Der Aufmerksamkeitsbedarf und die Kombinationen von ROA der Objekte werden über die zugehörigen Szenenknoten kommuniziert.

Neben der Blickrichtungsfähigkeit *OVB* bietet der Prozess BDGA eine weitere Fähigkeit namens *3D-Search* an, mit der ein gewisses Gebiet in der Umgebung des Fahrzeugs mit einer Kamera durchsucht werden kann. Der Kameraträger führt dabei eine Sequenz von Sakkaden und Bewegungen mit konstanter Winkelgeschwindigkeit aus. Die Blickrichtungsfähigkeit *3D-Search* ist deshalb von der Blickrichtungsfertigkeit *Sequence Saccade & Scan* abhängig.

Die zentrale Entscheidungsinstanz CD spricht ausschließlich die beiden Blickrichtungsfähigkeiten *OVB* und *3D-Search* an, deren Experte BDGA ist (vgl. Abb. 4.12). Dass diese von Blickrichtungsfertigkeiten und damit von Aktuatoren abhängen, ist für CD irrelevant und transparent. Während der Ausführung bestimmen die verschiedenen einfachen Blickrichtungsfertigkeiten *Fixation*, *Saccade* und *Scan* abwechselnd die Sollgrößen und Parameter für die unterlagerten Regler.

**TEIL C**

**EXPERIMENTELLE**

**ERGEBNISSE**



# Kapitel 5

## Das EMS-VISION System

Das EMS-VISION System bildet die experimentelle Grundlage dieser Arbeit. In Kapitel 1.2 wurde der Entwicklungshintergrund des EMS-VISION Systems bereits ausführlich diskutiert. Der nun folgende Abschnitt 5.1 zeigt die beiden Testfahrzeuge VAMORS und VAMP der Uni-BwM, in welchen das EMS-VISION System integriert wurde. Abschnitt 5.2 geht auf die Hardwareausstattung der Fahrzeuge ein; bevor Abschnitt 5.3 die im EMS-VISION System vorhandenen Prozesse und ihre Lokalisierung im verteilten System beschreibt.

### 5.1 Testfahrzeuge

Das ISF verfügt über zwei Testfahrzeuge: der 5t-Kastenwagen VAMORS (Versuchsfahrzeug für autonome Mobilität und Rechnersehen) und der PKW VAMP (Versuchsfahrzeug für autonome Mobilität und Rechnersehen im PKW) (vgl. Abb. 5.1). Beide Fahrzeuge sind nahezu identisch ausgestattet und für unterschiedliche Domänen bestimmt. Die für den Experimentalbetrieb notwendigen Umbauten der Fahrzeuge beschreibt SIEDERSBERGER [2003]. Die Domäne für VAMP ist das Fahren mit hoher Geschwindigkeit, das Konvoi-Fahren und das Stop&Go-Fahren auf Autobahnen. Das Szenario für das Testfahrzeug VAMORS ist das Navigieren auf Wegnetzen niederer Ordnung und das Querfeldeinfahren.



a)



b)

Abb. 5.1: Die Testfahrzeuge VAMORS (a) und VAMP (b)

## 5.2 Hardware

Bei der Hardwareausstattung der Versuchsfahrzeuge wurde im Wesentlichen auf Standardkomponenten zurückgegriffen. Bei den Rechnerknoten fiel die Wahl auf ein Netz von vier Intel-basierten PCs, welche unter dem Betriebssystem Windows NT 4.0 laufen. Die Gründe, die zur Auswahl von Hardware, Betriebssystem und Programmiersprache (Visual C++) führten, beschreibt RIEDER [2000]. Drei der PCs sind Bildverarbeitungsrechner (BV-PC); der vierte PC (Fahrzeug-PC) dient zur Verhaltensgenerierung (vgl. Abb. 5.2). Die vier PCs sind über ein leistungsfähiges Kommunikationsnetzwerk miteinander verbunden. Dieses *Scalable Coherent Interface* (SCI) Netzwerk ermöglicht den schnellen und exklusiven Datenaustausch zwischen den PCs während der Echtzeitphase und erreicht eine Bandbreite von 400 MByte/s. Ein Datenaustausch ist innerhalb weniger Mikrosekunden abgeschlossen. Auf dem Gateway-PC befindet sich die gesamte EMS-VISION-Software. Dieser PC bildet die Verbindung nach außen und hat während des Echtzeitbetriebs keine Funktion. Die Systemsoftware wird zu Beginn der Echtzeitphase über ein Fast-Ethernet Netzwerk auf die anderen PCs geladen.

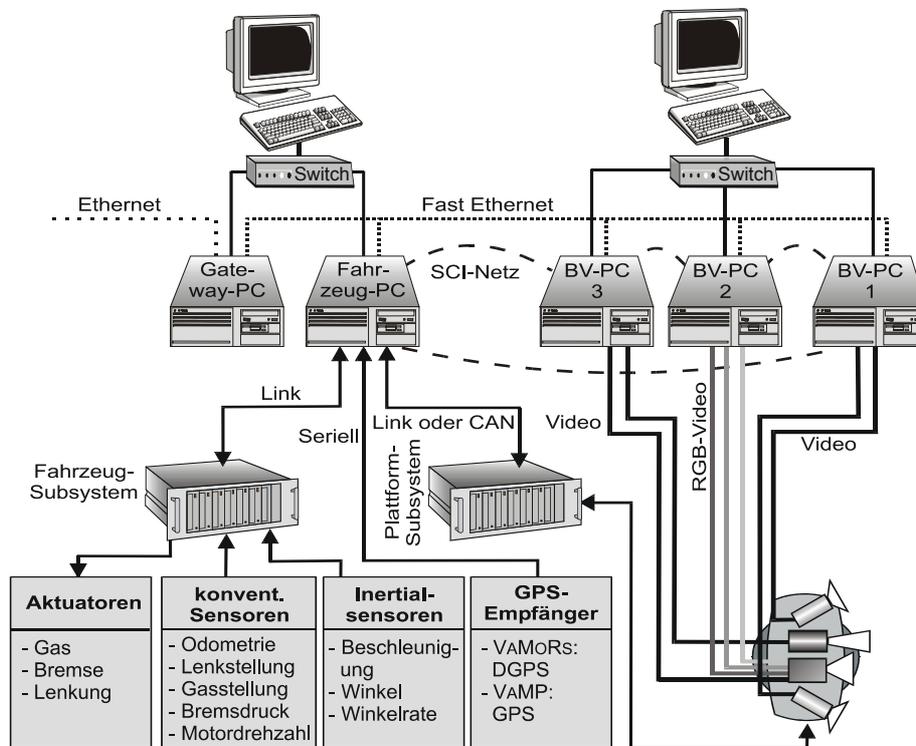


Abb. 5.2: Die Hardware-Ausstattung der Testfahrzeuge

Die BV-PCs sind mit Bilddigitalisierungskarten ausgestattet, welche analoge Videosignale digitalisieren. Die Videosignale aller Kameras sind synchronisiert, und das Synchronisationssignal wird im gesamten System inklusive den Subsystemen verteilt. Das eingebettete System zur Blickrichtungssteuerung (Plattform-Subsystem) kontrolliert den Kameraträger mit dem MARVEYE-Fahrzeugauge und ist mit dem Fahrzeugrechner verbunden. Das Fahrzeug-Subsystem besteht aus einem kleinen Transputer-Netz, welches eine Echtzeitregelung des Fahrzeugs ermöglicht. Das Fahrzeug-Subsystem kontrolliert die Aktuatoren (Gas, Bremse, Lenkung) des Fahrzeugs. Ferner ist ein GPS-Empfänger direkt mit dem Fahrzeugrechner verbunden. Das GPS-Signal ermöglicht zusammen mit digitalen Landkarten eine globale Navigation.

### 5.3 Prozesse

Die Abb. 5.3 zeigt, wie die Prozesse des EMS-VISION Systems auf die 4 Rechnerknoten verteilt sind. Auf jedem Rechnerknoten läuft im Hintergrund ein Prozess namens DKR (Dynamic Knowledge Repräsentation). Die DKR-Prozesse verteilen das dynamische Wissen im gesamten System und kommunizieren miteinander über das SCI-Netz. Ändert sich das Wissen in einem DKR-Prozess, so sendet der Prozess die Änderungen an alle anderen DKR-Prozesse, so dass die Wissensrepräsentationen in den DKR-Prozessen konsistent bleiben. Die anderen Prozesse auf den PCs – Client-Prozesse genannt – melden sich bei ihren lokalen DKR-Prozessen an, um einen skalierbaren Teil der Wissensrepräsentation gespiegelt zu bekommen. Wenn ein Client-Prozess Änderungen an seiner gespiegelten Wissensrepräsentation vornimmt und diese Änderungen für öffentlich erklärt, so werden sie automatisch an den lokalen DKR-Prozess gesandt. Dieser wiederum sendet die geänderten Daten an die DKR-Prozesse auf den anderen Rechnern. Empfängt ein DKR-Prozess neue Daten von einem anderen Prozess, so bringt er die gespiegelten Wissensrepräsentationen seiner Client-Prozesse auf den neuesten Stand. Die Client-Prozesse kommunizieren mit den lokalen DKR-Prozessen über *shared-memory*.

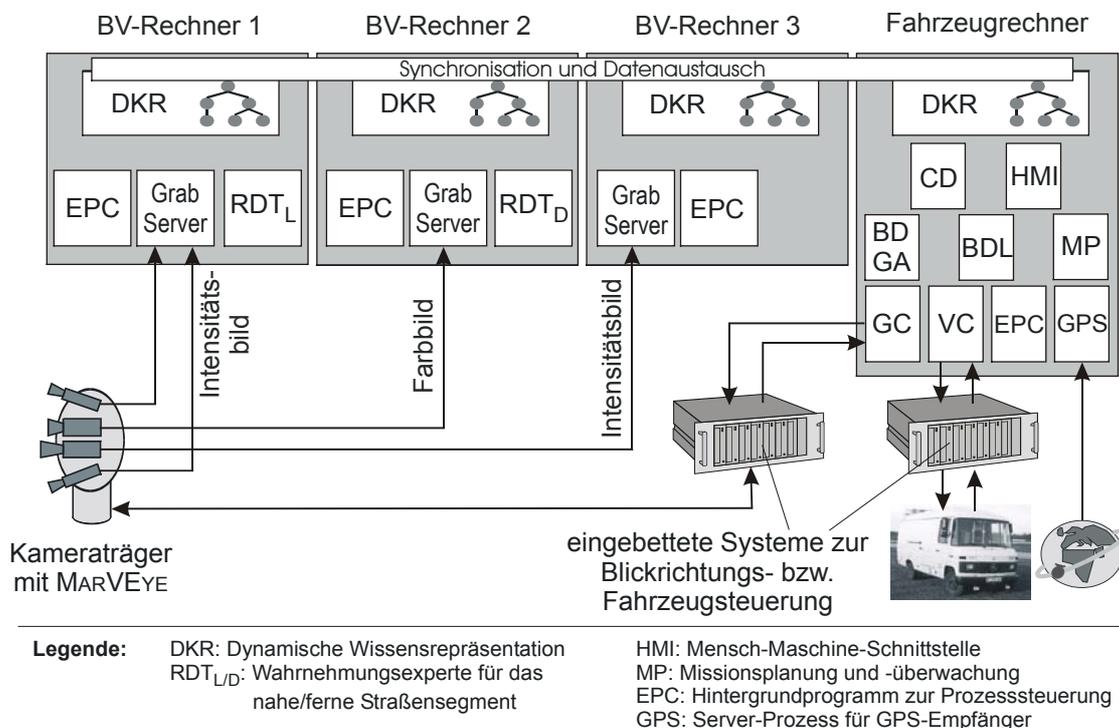


Abb. 5.3: Die Prozessaufteilung auf den Rechnerknoten

Die dynamische Wissensrepräsentation DKR ist baumartig strukturiert und ist sowohl im programmtechnischen also auch im physikalischen Sinn objektorientiert gestaltet. Sie beinhaltet die folgenden Arten von Wissen (vgl. auch Anhang B.3):

- Jeder Prozess im System wird durch ein sogenanntes Prozess-Objekt repräsentiert. Ein Prozess-Objekt beinhaltet allgemeine Informationen über den Prozess und bietet eine Schnittstelle zur Punkt-zu-Punkt-Kommunikation an.
- Jeder Rechenknoten im System ist durch ein Rechner-Objekt repräsentiert. In diesen Rechner-Objekten sind z. B. die belegten und noch verfügbaren Rechenressourcen beziffert. Die Liste von Rechner-Knoten wird dynamisch während des Hochfahrens des Systems erstellt.

- Der Szenenbaum besteht aus einer Menge von baumartig strukturierten Szenenknoten [RIEDER, 2000]. Die Szenenknoten repräsentieren physikalische Objekte, Objektteile oder virtuelle Koordinatensysteme. Die Kanten im Baum beschreiben homogene Koordinatentransformationen zwischen den durch die Szenenknoten aufgespannten Koordinatensystemen. Die Transformationsmatrizen haben eine Dimension von  $4 \times 4$  und beschreiben die Relativlage zwischen den repräsentierten Objekten mit 6 Freiheitsgraden (6DOF) oder führen eine perspektivische Projektion in ein Bildkoordinatensystem unter Nutzung eines Kameramodells (z. B. Lochkameramodell) durch. Aufgrund dieser Eigenschaften kann ein beliebiger Punkt in einem Objektkoordinatensystem in jedes andere Objektkoordinatensystem oder in ein Kamerabild transformiert werden. Neben dem Lagemodell kann ein Szenenknoten auch ein Formmodell oder andere objektspezifische Attribute aufnehmen. Szenenknoten, die Teile des Eigenfahrzeugs repräsentieren (z. B. Kameras, Kameraplattform, Fahrzeugaufbau) gehören zum permanenten Teil des Szenenbaums. Im Gegensatz dazu sind die Knoten von externen Objekten wie Straßen, Fahrzeuge oder Landmarken im Szenenbaum nur temporär vorhanden.
- Der Missionszweig der dynamischen Wissensrepräsentation repräsentiert den vom Modul zur Missionsplanung und -überwachung generierten Missionsplan. Der Missionsplan besteht aus einer Liste von Missionselementen. Das erste Eintrag in der Liste beschreibt das aktuelle Missionselement, welches das autonome Fahrzeug zum gegenwärtigen Zeitpunkt ausführt. Jedes Missionselement besitzt ein räumliches und/oder zeitliches Entfernungsmaß zum darauffolgenden Missionselement. Diese Entfernungsmaße werden ebenfalls vom Modul zur Missionsplanung und -überwachung gesetzt und aktualisiert (vgl. [GREGOR, 2002]).
- Der Situationszweig der dynamischen Wissensrepräsentation ermöglicht die Kommunikation von symbolischen Aussagen über Objekte oder Objektgruppen im Szenenbaum (vgl. Abschnitt 3.2). Dieser Zweig der Wissensrepräsentation wurde im Rahmen dieser Arbeit entwickelt und bietet eine Schnittstelle an, mit der ein Prozess Referenzen auf alle Objekte mit einer bestimmten symbolischen Bedeutung abfragen kann. Ferner kann ein Prozess sich alle symbolischen Bedeutungen eines Szenenobjekts geben lassen.
- Das Fähigkeitsnetz bildet einen weiteren Zweig der dynamischen Wissensrepräsentation und repräsentiert die Fähigkeiten und Aktionen des verteilten Systems. Auch dieser Teil der Wissensrepräsentation wurde im Rahmen dieser Arbeit entwickelt (vgl. Kapitel 3.3).

Die DKR stellt die Schnittstelle zu den Algorithmen der „*Künstlichen Intelligenz*“ dar. Sie dient als „Schwarzes Brett“, über das Informationen zwischen den Prozessen aller Systemebenen ausgetauscht werden können. Auf der systemdynamischen Ebene repräsentiert sie das relevante, optimal gefilterte Objektwissen zum aktuellen Zeitpunkt und für die nahe Vergangenheit (Pufferung).

Die Wahrnehmungsmodule für Straßen, Fahrzeuge oder Landmarken des EMS-VISION Systems basieren auf dem 4D-Ansatz von DICKMANN [1987] und hatten das von DICKMANN D. [1997] entwickelte „*Rahmensystem für die visuelle Wahrnehmung veränderlicher Szenen durch Computer*“ als Ausgangsbasis. Der 4D-Ansatz verwendet generische Modelle zur Auswertung von Bildfolgen, welche die räumliche Gestalt eines Objekts im 3D-Raum und sein Bewegungsverhalten über der Zeit beschreiben (vgl. Abschnitt 1.2).

Bei den in den Kapiteln 6.1.2, 7.1 und 7.2 dargestellten Testfahrten wurde die vom Eigenfahrzeug befahrene Straße durch drei Kameras wahrgenommen. Für die Wahrnehmung des nahen<sup>1</sup> Segments der Eigenstraße dienen die beiden monochromen Weitwinkelkameras. Wie

---

<sup>1</sup> nahe zum Eigenfahrzeug.

in Abb. 5.3 ersichtlich ist, sind diese beiden Videoströme im Hauptspeicher des BV-PC 1 verfügbar. Deshalb ist auch der Wahrnehmungsexperte für das nahe Segment der Straße  $RDT_L$  auf diesem Rechner positioniert. Das ferne Segment der Eigenstraße wird mit der milden Telekamera wahrgenommen, dessen Bild im Hauptspeicher des BV-PC 2 verfügbar ist. Dementsprechend befindet sich der Wahrnehmungsexperte für das ferne Straßensegment  $RDT_D$  auch auf diesem Rechner. Der Prozess Embedded PC Daemon (EPC) läuft auf jedem PC im Hintergrund und ermöglicht das Starten und Beenden von Prozessen von fremden Rechnern aus. Die restlichen auf dem Fahrzeugrechner befindlichen Prozesse gehen aus Abb. 5.3 hervor oder wurden bereits an anderer Stelle in dieser Arbeit beschrieben. Eine ausführlichere Darstellung des EMS-VISION Systems findet sich z. B. in [GREGOR et al., 2000].



## Kapitel 6

### Resultate der Verhaltensentscheidung

Bei der Ausführung komplexer Missionen stößt die zentrale Entscheidungsinstanz Central Decision (CD) passende, im System vorhandene (schematische) Fähigkeiten an. Die Unterabschnitte des Kapitels 6 zeigen Daten und Ergebnisse, die während autonomer Testfahrten in den Teilkomponenten des Prozesses CD aufgezeichnet wurden.

#### 6.1 Resultate in der Situationsanalyse

Die jetzt vorgestellten Versuchsfahrten wurden im Rahmen des deutsch-amerikanischen Projekts AUTONAV durchgeführt. Dieses Projekt hatte u. a. zum Ziel, den an der UniBwM entwickelten 4D-Ansatz für das maschinelle Sehen mit den hierarchischen Stereobildverarbeitungsalgorithmen der Firma SARNOFF CORPORATION (Princeton, USA) zu kombinieren. Die Echtzeit-Stereobildverarbeitung läuft dabei auf einer Spezialhardware namens Vision Front End (VFE) 200 und errechnet aus einem Stereo-Videosignal (25 Doppelbilder pro Sekunde) eine Disparitätenkarte. Zum Zweck der Objekthypothesengenerierung wird die Umgebung des Fahrzeugs in Gitterzellen der Größe  $0,25 \times 0,25 \text{ m}^2$  aufgeteilt. Mithilfe der Disparitätenkarte ermittelt man für jede Gitterzelle ein Maß für den Abstand zwischen dem durchschnittlichen Bodenniveau innerhalb der Gitterzelle zu einer virtuellen Ausgleichsebene (Horopterebene). Das VFE 200 ist innerhalb des EMS-Systems an den BV-PC 3 angeschlossen (vgl. Abb. 5.2). Auf diesem Rechner läuft der Prozess Obstacle Detection for Terrain (ODfT), welcher mittels der Disparitätenkarte Objekthypothesen generiert, in die Gitterzellen einträgt und diese anschließend durch Beobachtung über der Zeit verifiziert. Für jedes verifizierte Objekt (Hindernis) wird im Szenenbaum ein Szenenknoten eingehängt, welcher das Objekt repräsentiert und dessen Größe und Form beschreibt. Einzelheiten über die Stereobildverarbeitung und die Objekthypothesengenerierung findet man in [SIEDERSBERGER et al., 2001].

Wahrgenommene Hindernisse haben für das autonome Fahrzeug unterschiedliche Bedeutungen. Liegt ein Hindernis im Fahrschlauch, so muss eine Ausweichstrategie erarbeitet und ein Ausweichmanöver durchgeführt werden, um so das Fahrzeug vor Schaden zu bewahren. Eben diese Aussage „*das Hindernis liegt im Fahrschlauch*“ ist eine symbolische Aussage und wird durch einen entsprechenden Situationsaspekt generiert.

### 6.1.1 Die verwendeten Situationsaspekte

Bei den Testfahrten dieses Abschnitts 6.1 kommen im Wesentlichen zwei Situationsaspekte zur Anwendung. Der erste Situationsaspekt hat den Namen „*Relativlage zum Fahrschlauch und Eigenfahrzeug*“ (im folgenden *Relativlage* abgekürzt) und klassifiziert ein Objekt mit einem Formmodell nach dessen Lage relativ zu einem Fahrzeug mit seinem Fahrschlauch. Das Objekt kann ein stationäres (positives oder negatives) Hindernis oder ein Fremdfahrzeug sein. Die linguistischen Werte des Situationsaspekts lauten:

<i>im Fahrschlauch</i>	<i>links/rechst neben Fahrschlauch</i>
<i>links/rechts neben Fahrzeug</i>	
<i>links/rechts hinter dem Fahrzeug</i>	<i>hinter dem Fahrzeug</i>

Ist keine Fahrspur vorhanden, so stützt sich die Berechnung des Fahrschlauchs alleine auf den Lenkwinkel und die Lenkrate des Fahrzeugs. Ist jedoch eine Repräsentation der Fahrspur vorhanden, so wird die Krümmung und die Krümmungsänderung der Fahrspur in den Fahrschlauch übertragen. Dabei geht man implizit von der Annahme aus, dass das Fahrzeug der Fahrspur folgen wird. Dieser direkte Übertrag der Krümmungsinformation ist natürlich bei einem (geplanten) Spurwechsel nicht möglich. In diesem Fall müssen die Formparameter der alten und der neuen Spur zur Berechnung eines s-förmigen Fahrschlauchs herangezogen werden. Die Zugehörigkeitsgrade zu den linguistischen Werten des Situationsaspekts *Relativlage* werden jede Sekunde oder aufgrund externer Ereignisse neu berechnet. Ein solches Ereignis ist z. B. das Einhängen eines neuen Hindernisses in den Szenenbaum durch einen Wahrnehmungsexperten.

Der zweite Situationsaspekt lautet „*Reihenfolge im Fahrschlauch*“. Dieser Situationsaspekt wird auf alle Objekte angewandt, für die der linguistische Wert „*im Fahrschlauch*“ des Situationsaspekts *Relativlage* einen Zugehörigkeitsgrad größer als 0,5 hat. Der Situationsaspekt „*Reihenfolge im Fahrschlauch*“ sortiert die Objekte im Fahrschlauch nach ihrer Reihenfolge relativ zum Fahrzeug. In Abschnitt 6.1.3 wird der Situationsaspekt dazu benutzt, um vor dem 1. Objekt im Fahrschlauch anzuhalten.

### 6.1.2 Fahren auf Feldwegen mit Anhalten vor positiven<sup>1</sup> Hindernissen

Bei den Testfahrten dieses Abschnitts fährt das Testfahrzeug VAMORS autonom längs- und quergeführt auf einem Feldweg mit einer Geschwindigkeit von ca. 16 km/h. Die Position und Orientierung der Straße relativ zum Eigenfahrzeug und die Straßenparameter (Straßenbreite und -krümmung) werden von den Wahrnehmungsexperten für die Straße  $RDT_L$  und  $RDT_D$  geschätzt (vgl. Abschnitt 5.3). Die Blickrichtungssteuerung hält dabei das ferne Straßensegment im Zentrum der milden Telekamera (vgl. Abb. 6.1 a). Parallel dazu nehmen das VFE 200 und der Prozess ODfT mögliche Hindernisse wahr.

Als positive<sup>1</sup> Hindernisse dienen drei Pylone. Die ersten beiden Pylone befinden sich am rechten und linken Straßenrand auf gleicher Höhe. Nach weiteren ca. 10 Metern ist der dritte Pylon in der Straßenmitte positioniert (vgl. Abb. 6.1 a).

<sup>1</sup> Als positive Hindernisse werden Hindernisse bezeichnet, welche aus dem gemittelten Bodenniveau herausragen. Im Gegensatz dazu werden Löcher oder Gräben im Untergrund negative Hindernisse genannt.

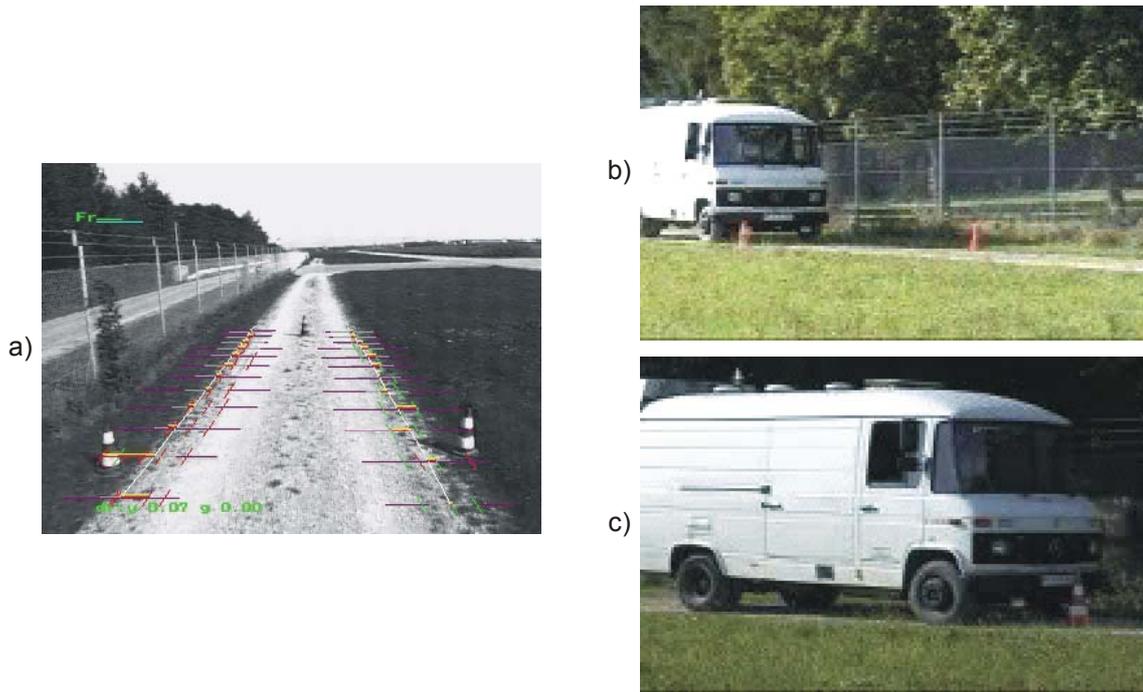


Abb. 6.1: Fixieren des Weges mit der milden Telekamera und Straßenerkennung (a), Passieren der Pylone am Wegrand (b) und Stoppen vor dem Pylon in der Wegmitte (c)

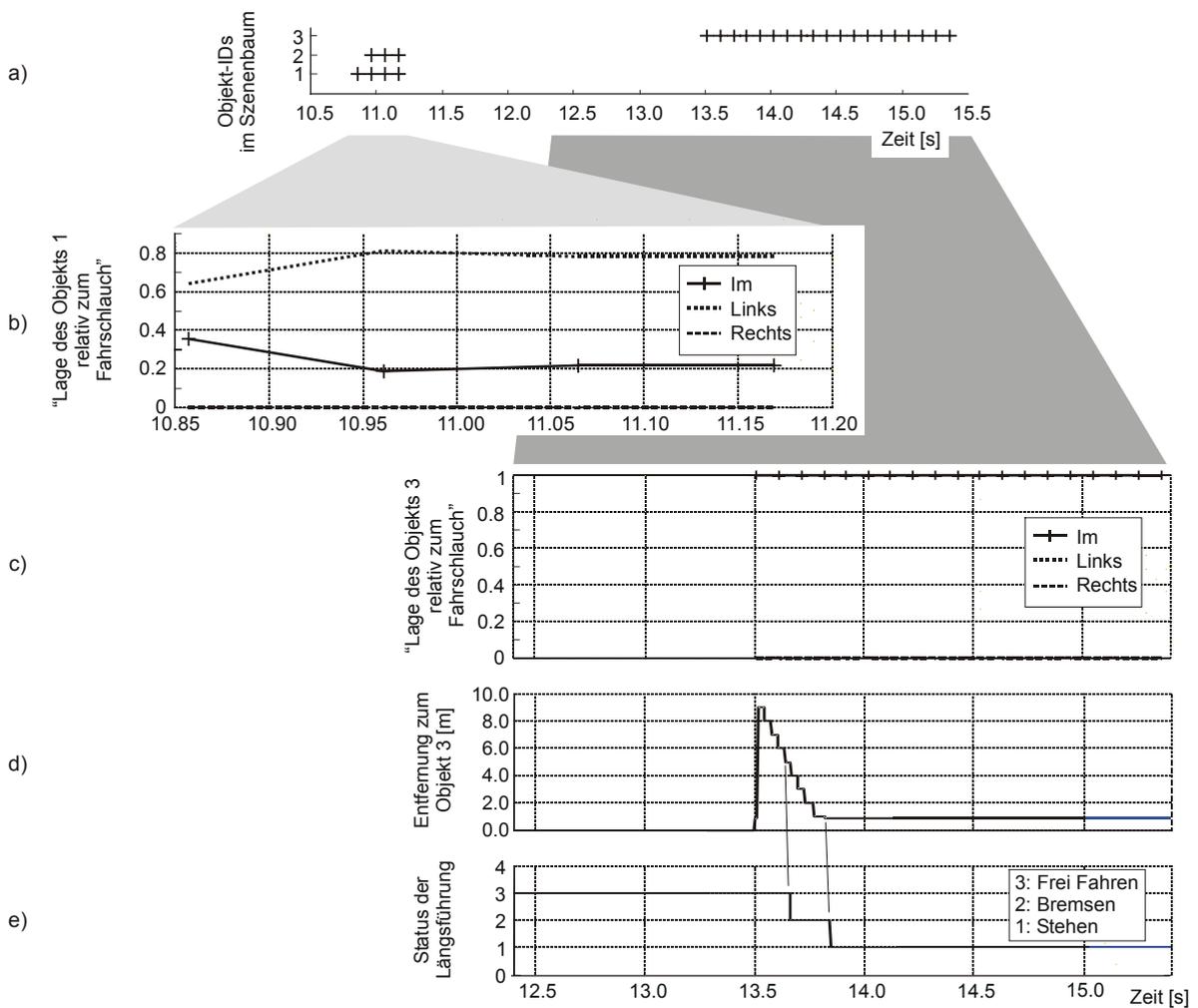


Abb. 6.2: Anhalten vor positiven Hindernissen auf einem Feldweg nach [SIEDERSBERGER et al., 2001]

Abb. 6.2 zeigt Daten einer solchen autonomen Fahrt. Die ersten beiden, an den Straßenrändern befindlichen Pylone werden nach ca. 10,8 Sekunden erstmalig detektiert und im Szenenbaum als Objekte 1 und 2 eingehängt. Wie aus Abb. 6.2 b hervorgeht, hat der linguistische Wert *Relativlage.links\_neben\_Fahrschlauch*<sup>2</sup> für das Objekt 1 den größten Zugehörigkeitsgrad. Aus diesem Grund initiiert die Verhaltensentscheidung wegen Objekt 1 kein Anhalten und das Fahrzeug passiert das Objekt (vgl. Abb. 6.1 b). Die analogen Aussagen gelten für das Objekt 2. Nach 13,5 Sekunden wird der dritte Pylon detektiert. Aus Abb. 6.2 c geht hervor, dass für dieses Objekt 3 der linguistische Wert *Relativlage.im\_Fahrschlauch* den größten Zugehörigkeitsgrad besitzt. Objekt 3 erzwingt deshalb ein autonomes Anhalten (Abb. 6.2 d-e und Abb. 6.1 c). Einzelheiten über die Ausführung des autonomen Fahrens und Anhaltens in der Fahrzeugführung finden sich in [SIEDERSBERGER et al., 2001].

### 6.1.3 Querfeldeinfahren mit Anhalten vor negativen<sup>1</sup> Hindernissen

Aufgrund ihrer Eigenverdeckung sind negative Hindernisse von Fahrzeugen aus gesehen sehr schwierig zu erkennen. In einer Entfernung von 20 m vor einem Graben mit 0,5 m Breite sind Punkte in diesem Graben, die tiefer als 5 cm liegen, nicht sichtbar (vgl. Abb. 6.3 d+e). Das Problem ist, dass ein Algorithmus, der einen solchen Graben bei 20 m erkennt, auch alle anderen Unebenheiten mit einer Tiefe von 5 cm wahrnimmt. Eine zeitliche Integration zur Reduktion der Falschalarme ist deshalb dringend notwendig [SIEDERSBERGER et al., 2001].

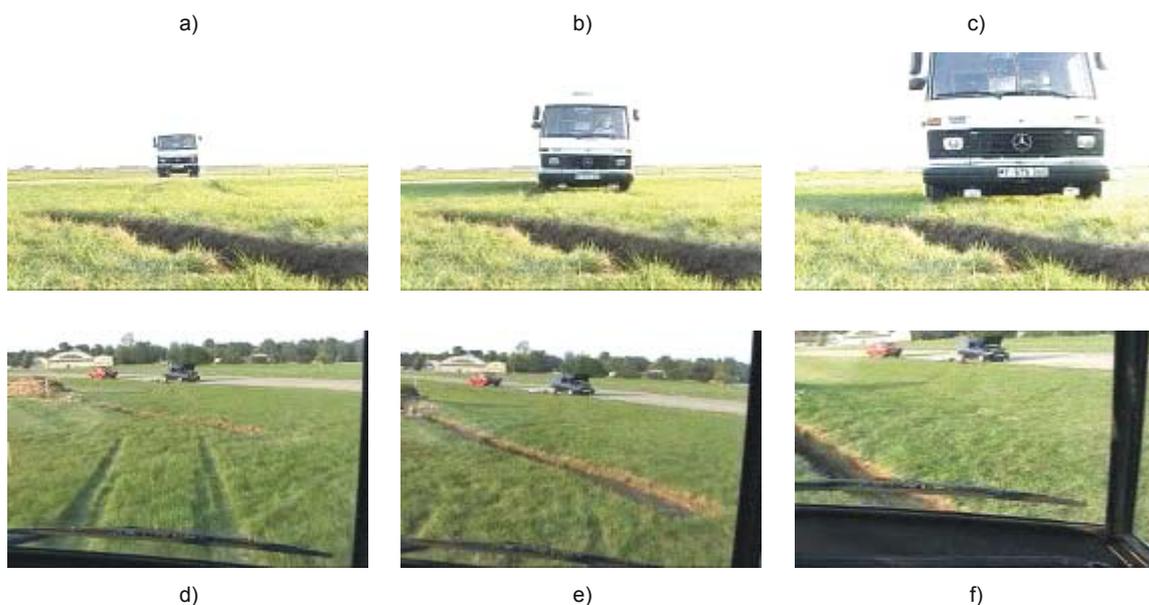


Abb. 6.3: Das autonome Anhalten vor einem Graben in der Außenansicht (a-c) und aus der Sicht des Beifahrers (d-f)

Abb. 6.3 zeigt ein autonomes Anhalten vor einem Graben. Das Versuchsfahrzeug VAMORS fährt dabei auf unbefestigtem Grund mit einer Geschwindigkeit von 16 km/h autonom längsgeführt. Die Querführung erfolgt manuell. Wie aus Abb. 6.4 a hervorgeht, wird nach ca. 32,4 Sekunden ein im Fahrschlauch befindliches Objekt mit der ID 4 detektiert. Da neben diesem Objekt noch weitere Objekte erkannt werden, ist Objekt 4 anfänglich nicht das erste und damit das am nächsten liegende Objekt im Fahrschlauch (Abb. 6.4 b, nach 32,5 s). Im Zuge der Verifikation der Objekthypothesen über der Zeit verschwinden aber näher lie-

<sup>2</sup> Der linguistische Wert *lingWert* des Situationsaspekts *Name* wird im folgenden durch *Name.lingWert* abgekürzt.

gende Objekte wieder, so dass nach 32,7 Sekunden das Objekt 4 das am nächsten liegende ist. Objekt 4 bestätigt sich auch in den folgenden Zyklen und die Verhaltensentscheidung für die Fortbewegung hält deshalb vor dem Objekt an. Weitere Informationen bezüglich der Verifikation der Objekthypothesen finden sich in [SIEDERSBERGER et al., 2001].

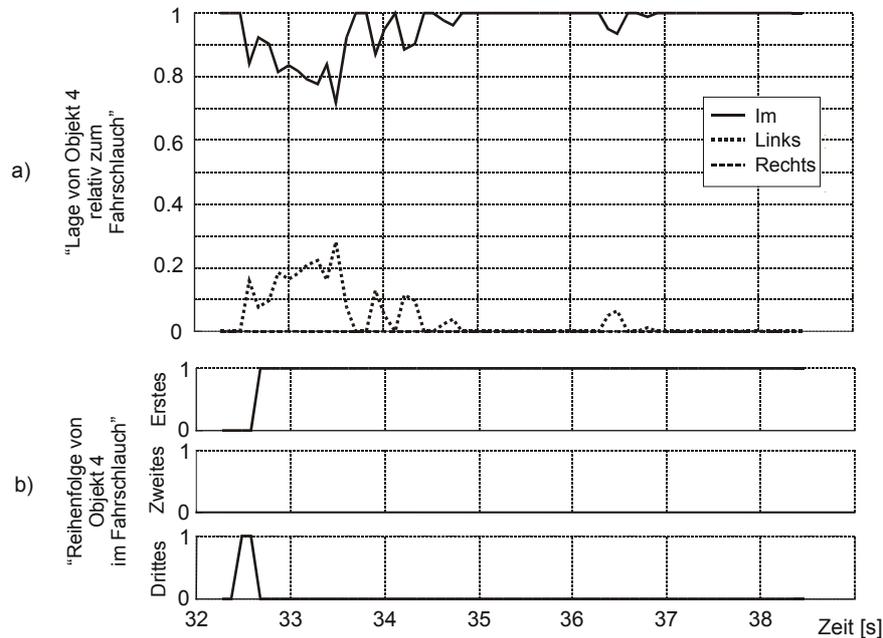


Abb. 6.4: Autonomes Anhalten vor einem Graben

## 6.2 Ausführung einer komplexen Mission

Mit dem Versuchsfahrzeug VAMORS wurden (voll)autonome Testfahrten sowohl auf unmarkierten Straßen niedriger Ordnung durchgeführt, als auch Querfeldeinfahrten. Dieser Abschnitt beschreibt eine domänenübergreifende Mission. In Abb. 6.5 ist der Lageplan dieser komplexen Mission mit den einzelnen Missionselementen dargestellt: Fahren auf einer unmarkierten Straße, Abbiegen nach links in eine Querstraße, Folgen dieser Straße, Verlassen der Straße, Anfahren einzelner GPS-Wegpunkte auf unbefestigtem Gelände, Wiederauffahren auf die Straße, Folgen der Straße, Kreuzung überqueren, Verlassen der Straße, Anfahren einzelner GPS-Wegpunkte solange bis ein wahrgenommener Graben ein Anhalten erzwingt. Diese komplexe Mission wurde bei der Abschlussveranstaltung des Projekts IFF 3 im Oktober 2001 auf dem ehemaligen Flughafengelände nahe der UniBwM einem internationalen Publikum vorgeführt.

Abb. 6.6 zeigt Daten vom ersten Teil dieser komplexen Mission bis zum zweiten GPS-Wegpunkt. Die Missionsplanung generiert eine Liste von Missionselementen. Abb. 6.6 a zeigt das jeweils aktuelle Missionselement während der Ausführung der Mission. In Abb. 6.6 und im folgenden Text sind Missionselemente klein geschrieben und müssen von den groß geschriebenen (schematischen) Fähigkeiten des Systems gedanklich klar getrennt werden. Missionselemente sind abstrakte Aufgabenbeschreibungen ohne Wissen darüber, wie diese vom System ausgeführt werden. Im Gegensatz dazu ist in den Fähigkeiten das Wissen angelagert, wie die mehr oder weniger komplexen Aktionen in den Systemkomponenten anzustoßen und zu koordinieren sind, um ein Missionselement (sicher) auszuführen.

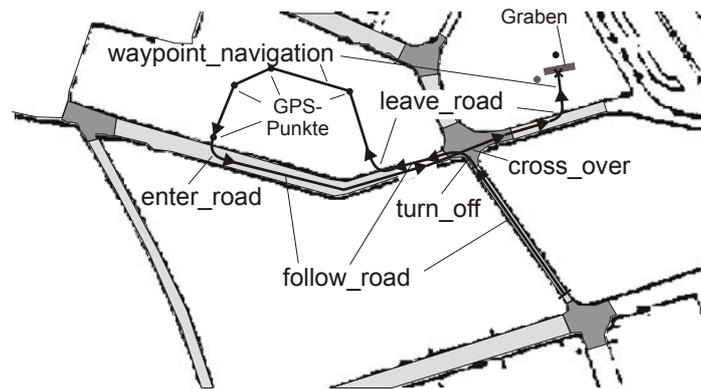


Abb. 6.5: Der Lageplan der komplexen Mission mit den Missionselementen

Nach dem Hochfahren des Systems ist kein Missionselement spezifiziert (vgl. Abb. 6.6 a). Nach 156 Sekunden wird die Mission gestartet und halt ist das aktuelle Missionselement. Dementsprechend ist eine andere Regel der Regelbasis<sup>3</sup> am meisten erfüllt (vgl. Abb. 6.6 d). Die neue gültige Regel spezifiziert andere Parameter für die Blickrichtungsfähigkeit OVB als die vorherige Regel, so dass die Instanz dieser Fähigkeit neue Parameter zugesandt bekommt (vgl. Abb. 6.6 e+f). Das Missionselement halt dauert 20 Sekunden. In Abb. 6.6 c ist der abnehmende zeitliche Abstand zum nächsten Missionselement aufgetragen. Ist das aktuelle Missionselement halt und das Fahrzeug steht, so führt dies zum Start der Fähigkeit Standing. CD kann aber nur Fähigkeiten nutzen, die von ihren Experten als verfügbar markiert sind. Erst nach 167 Sekunden überlässt der Benutzer über das Human Machine Interface dem autonomen System die Kontrolle über die Fortbewegung, so dass die Fortbewegungsfähigkeiten verfügbar werden. Dies führt zum Start der Fähigkeit Standing durch CD (vgl. Abb. 6.6 e+f).

Nach 176 Sekunden ist die Dauer des Missionselements halt abgelaufen und follow\_road wird das aktuelle Missionselement. Auch dieser Wechsel im Missionselement führt zu einem Wechsel der gültigen Regel. Die neu gültige Regel benutzt die Fähigkeiten Follow\_Lane und nicht Standing, so dass Standing gestoppt und gelöscht und Follow\_Lane initialisiert, parametrisiert und gestartet wird (vgl. Abb. 6.6 e+f). Abb. 6.6 g-k zeigt den Organisationsstatus und Operationsstatus einiger Fortbewegungsfähigkeiten, welche in VC ablaufen und von BDL kontrolliert werden (vgl. [SIEDERSBERGER, 2003]). Insbesondere geht aus Abb. 6.6 g hervor, dass nach dem Startkommando in Sekunde 176 eine Instanz der Fähigkeit Follow\_Lane mit der ID 70402 existiert, deren Organisationsstatus ACTIVE und Operationsstatus RUN ist.

CD nutzt für die Ausführung des Missionselements turn\_off die Fortbewegungsfähigkeit Turn\_Off. Um die erwartete Querstraße zu erkennen, wird ein Wahrnehmungsexperte für Straßen aktiv (nicht gezeigt). Zwischen Sekunde 176 und 186 nimmt die Distanz zur erwarteten Position der Querstraße durch die Berücksichtigung der vom Fahrzeug zurückgelegten Strecke kontinuierlich ab. Nach 186 Sekunden detektiert der Wahrnehmungsexperte die Querstraße und schätzt die Entfernung zu dieser auf ca. 35 m. Dementsprechend reduziert sich die räumliche Distanz zum nächsten Missionselement turn\_off auf diesen Wert (vgl. Abb. 6.6 b). Zwischen Sekunde 186 und 207 schätzt der Wahrnehmungsexperte kontinuierlich die Entfernung zur Querstraße.

<sup>3</sup> Die Ausführung der Mission wird von der in Kapitel 3.4.4 beschriebenen und in Abb. 3.22 dargestellten Regelbasis bestimmt.

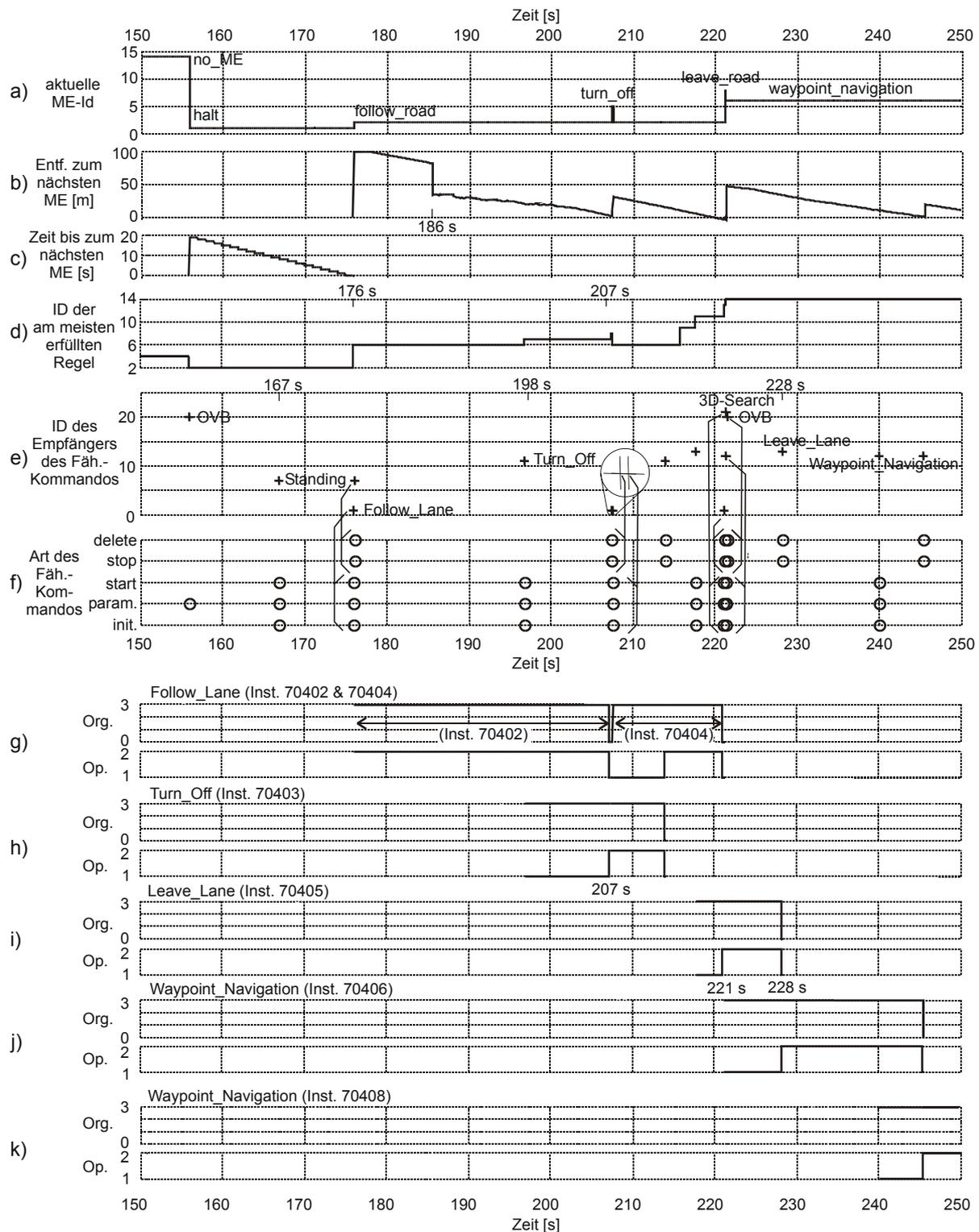


Abb. 6.6: Die Ausführung einer komplexen Mission: aktuelle Missionselemente (ME) mit ihren räumlichen/zeitlichen Abständen zueinander (a-c), die aktuell am meisten erfüllte Regel der Regelbasis (d), die Steuerbefehle an die Fähigkeiten (e-f), Organisationsstatus (Org.) und Operationsstatus (Op.) einiger Fähigkeiteninstanzen (g-k) (Org.: 0 = INIT, 1 = PASSIVE, 2 = READY, 3 = ACTIVE; Op.: 1 = PASSIVE, 2 = RUN);

Nach ca. 198 Sekunden ist die Entfernung so klein geworden, das CD die Fähigkeit Turn\_Off initiiert (vgl. Abb. 6.6 d-f). Wie Abb. 6.6 h zeigt, existiert ab diesem Zeitpunkt eine Instanz der Fähigkeit Turn\_Off mit der ID 70403. Der Organisationsstatus dieser Instanz ist ACTIVE, was bedeutet, dass BDL und VC das Abbiegen vorbereiten. Bis zur Sekunde 207,46

ist jedoch der Operationsstatus dieser Instanz `PASSIVE` und die Instanz hat keinen Durchgriff auf die Aktuatoren des Fahrzeugs<sup>4</sup>.

Nach 207,46 Sekunden wird der Operationsstatus der Instanz 70403 `RUN` und `VC` beginnt mit der Ausführung des Abbiegens (vgl. Abb. 6.6 h). `VC` reduziert dabei die Relevanz der bisher befahrenen Straße. Aufgrund dieser Reduzierung wechselt das aktuelle Missionselement zu `turn_off` und eine Reorganisation des Szenenbaums findet statt. Nach der Reorganisation ist die bisherige Querstraße die neue Eigenstraße und das aktuelle Missionselement wechselt weiter zum Missionselement `follow_road` (vgl. Abb. 6.6 a). Die Instanz der Fähigkeit `Turn_Off` ist zu diesem Zeitpunkt jedoch noch nicht abgeschlossen (vgl. Abb. 6.6 h).

Die Wechsel im aktuellen Missionselement haben Auswirkungen auf die Erfüllung der Regeln in der Regelbasis (vgl. Abb. 6.6 d). Ist das aktuelle Missionselement `turn_off` und der Operationsstatus der Instanz 70402 der Fähigkeit `Follow_Lane` wechselt in den Zustand `PASSIVE`, so stoppt und löscht `CD` diese Instanz (vgl. Abb. 6.6 e-g). Nach 207,59 Sekunden ist das aktuelle Missionselement wieder `follow_road` und dies führt zum Start einer neuen Instanz der Fähigkeit `Follow_Lane`. Der Status dieser neuen Instanz mit der ID 70404 ist ebenfalls in Abb. 6.6 g aufgetragen. Solange `VC` die Fähigkeit `Turn_Off` ausführt, bleibt der Operationsstatus der Instanz 70404 `PASSIVE` (vgl. Abb. 6.6 g+h). Nach 214 Sekunden ist das Abbiegen beendet und der Operationsstatus der Instanz 70404 der Fähigkeit `Follow_Lane` wechselt zu `RUN`. Für `CD` ist der Übergang im Operationsteil der Instanz 70403 der Fähigkeit `Turn_Off` von `RUN` zu `PASSIVE` das Zeichen, diese Instanz zu löschen (vgl. Abb. 6.6 e+f).

Nach dem Fahren von weiteren 30 Metern auf dieser Straße wechselt das aktuelle Missionselement nacheinander auf `leave_road` und `waypoint_navigation`. In analoger Weise wie zuvor startet `CD` eine Instanz der Fähigkeit `Leave_Lane` zu einem ausreichend frühen Zeitpunkt. Beim Wechsel des aktuellen Missionselements zu `leave_road`, erfolgt eine erneute Reorganisation des Szenenbaums, die Instanzen der Fähigkeiten `Follow_Road` und `OVB` werden gelöscht und Instanzen der Fähigkeiten `Waypoint_Navigation` und `3D-Search` gestartet. Zwischen Sekunde 221 bis 228 führt `VC` die Fähigkeit `Leave_Lane` aus (vgl. Abb. 6.6 i). Nach 228 Sekunden wird die Instanz 70406 der Fähigkeit `Waypoint_Navigation` aktiv und `CD` löscht die Instanz 70405 der Fähigkeit `Leave_Lane` (vgl. Abb. 6.6 e+f, i+j).

Während des Annäherns an den ersten GPS-Wegpunkt startet `CD` für den nächsten Wegpunkt eine weitere Instanz der Fähigkeit `Waypoint_Navigation` (bei Sekunde 240 in Abb. 6.6 e+f)<sup>5</sup>. Ab diesem Zeitpunkt gibt es zwei Instanzen der Fähigkeit `Waypoint_Navigation`: 70406 und 70408 (vgl. Abb. 6.6 j+k). Nach 245 Sekunden erreicht das Fahrzeug den ersten Wegpunkt, die Instanz 70406 wird passiv und gelöscht und die Instanz 70408 wird aktiv (vgl. Abb. 6.6 e+f, j-k).

Die Ausführungen in diesem Kapitel zeigen, dass das Fähigkeitsnetz eine flexible Steuerung der Aktivitäten im System erlaubt, so dass eine abstrakte Aufgabenbeschreibung in Form eines Missionsplans ausgeführt werden kann. Die Steuerung der Handlungen geschieht auf einheitliche Weise und lässt sich schnell und unkompliziert um weitere Fähigkeiten erweitern. Zu betonen ist auch, dass das Versuchsfahrzeug keine exakte Karte benötigt, um auf einem

---

<sup>4</sup> Zu einem bestimmten Zeitpunkt kann nur eine Fortbewegungsfähigkeit den Operationsstatus `RUN` einnehmen und damit die Aktuatoren des Fahrzeugs beeinflussen (vgl. [SIEDERSBERGER, 2003]).

<sup>5</sup> Für jeden Wegpunkt wird von `CD` eine eigene Instanz der Fähigkeit `Waypoint_Navigation` gestartet. Eine Instanz dieser Fähigkeit erhält aber nicht nur die Referenz auf den nächsten GPS-Wegpunkt, sondern auf die 5 nächsten Wegpunkte, so dass `BDL` deren Lagen bei der Planung der Fahrtrajektorie berücksichtigen kann (vgl. [SIEDERSBERGER, 2003]).

Wegenetz zu navigieren. Eine topographische Karte in Verbindung mit der Fähigkeit Straßen und Landmarken zu erkennen ist für eine lokale Navigation ausreichend.

### 6.3 Beispiel einer Kompensation systematischer Abweichungen

Wie in den Abschnitten 3.3.3 und 3.3.4 dargestellt wurde, ist es sinnvoll, die Güte der im System ablaufenden Aktionen sowohl zentral in der Verhaltensentscheidung als auch dezentral in den Experten der Fähigkeiten zu beobachten und zu kontrollieren. In Abschnitt 3.3.6.2 wurde am Beispiel eines Ablaufmodells für die Fähigkeit *Anhalten* gezeigt, dass durch die parallele Modellierung von Sollverläufen auf der Situationsebene und auf der Zustandsebene systematische Abweichungen bei der Ausführung kompensiert werden können. Die Funktionalität dieses Ablaufmodells ist entsprechend der organisatorischen Hierarchie von Entscheidungsinstanzen des EMS-VISION Systems nicht der zentralen Entscheidungsinstanz CD zuzuordnen, sondern dem Experten für die Fortbewegung BDL (vgl. Abschnitt 3.5.1). Im folgenden werden experimentelle Ergebnisse der in Kapitel 3.3.6.2 beschriebenen Kompensation systematischer Abweichungen gezeigt. Bei den Fahrversuchen wurde darauf geachtet, dass innerhalb der Fähigkeit *Anhalten* (Stop) des Prozesses Vehicle Control keine entsprechende Kompensation erfolgt, weil dann die Kompensation durch das Ablaufmodell keine Wirkung gezeigt hätte.

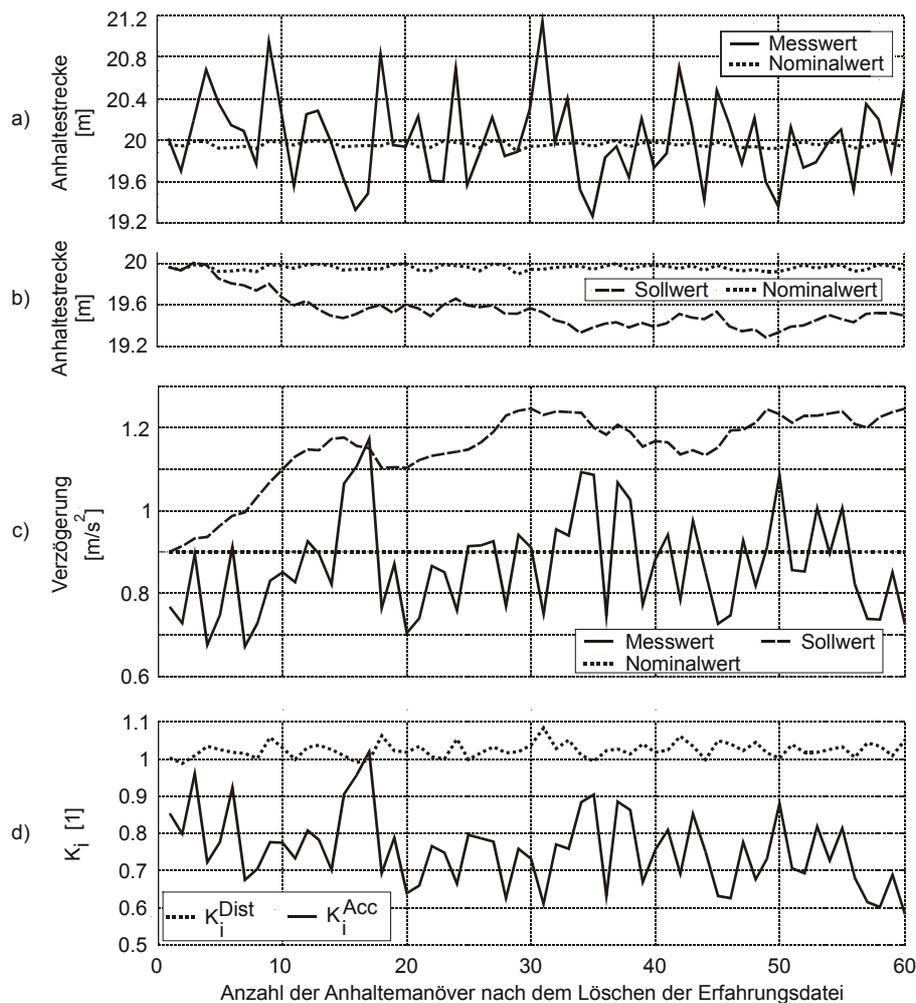


Abb. 6.7: Adaption der kommandierten Anhaltestrecke und Anhalteverzögerung

Zu Beginn des Experiments wird die Erfahrungsdatei für alle Fähigkeiten gelöscht; d. h. dem System liegen keine Informationen über systematische Abweichungen vor und die Ablaufmodelle nehmen deshalb ein ideales Übertragungsverhalten ohne Abweichungen an ( $\bar{K}_i = 1$  in Gl 3.9 auf Seite 76).

In der Abszisse von Abb. 6.7 ist die Anzahl der Anhaltenmanöver nach dem Löschen der Erfahrungsdatei aufgetragen. In Abb. 6.7 a+b sind die Nominalwerte, die kommandierten Sollwerte und die gemessenen Istwerte für die Anhaltestrecke dargestellt. Abb. 6.7 d zeigt die daraus errechneten  $K_i$ -Werte (gepunktete Linie). Aus Abb. 6.7 a+d geht hervor, dass die Anhaltestrecke relativ gut eingehalten wird; es kommt lediglich zu Schwankungen im Bereich 19,6 – 20,8 m und zu einer mittleren Abweichung von ca. + 3 % (d. h. das Fahrzeug fährt etwas zu weit).

Zu Beginn des Experiments ist aufgrund des fehlenden Wissens über diese systematische Abweichung der kommandierte Sollwert gleich dem Nominalwert (Abb. 6.7 b). Nach dem Neuaufbau der Erfahrung wird der Sollwert um die systematische Abweichung zurückgenommen. Dies hat zur Folge, dass nach ca. 20 Anhaltevorgängen die gemessenen Anhaltestrecken symmetrisch um den annähernd konstanten Nominalwert schwanken (Abb. 6.7 a).

Bei der Anhalteverzögerung ist die systematische Abweichung größer (vgl. Abb. 6.7 c+d): Das Fahrzeug bremst durchschnittlich mit einer um ca. 25 % zu kleinen Verzögerung. Auch die Streuung der einzelnen gemessenen Bremsverzögerungen ist sehr groß; die Bremsverzögerungen schwanken zwischen 0,7 und 1,1 m/s<sup>2</sup>. Wegen dieser großen systematischen Abweichung ist nach dem Aufbau der Erfahrung die Differenz zwischen dem kommandierten Sollwert und dem Nominalwert relativ groß (Abb. 6.7 c). Durch die Korrektur des Sollwerts schwanken trotz der großen systematischen Abweichung die gemessenen Istwerte nach ca. 30 Anhaltevorgängen um den konstanten Nominalwert.

In Abb. 6.8 sind einige weitere Details des Experiments dargestellt. Im Missionsplan des Experiments wechseln sich die Missionselemente `follow_road` und `halt` periodisch ab (vgl. Abb. 6.8 a). Für das Missionselement `follow_road` ist eine räumliche Distanz zum nächsten Missionselement definiert und für `halt` eine zeitliche Distanz. Die Verläufe beider Distanzmaße sind in Abb. 6.8 b aufgetragen. Abb. 6.8 c zeigt die am meisten erfüllte Regel der Regelbasis und Abb. 6.8 g die fuzzifizierte Geschwindigkeit. In Abb. 6.8 h ist die zurückgelegte Gesamtstrecke dargestellt. Die Sollgeschwindigkeit des Missionselements `follow_road` war mit 10 km/h bemessen, so dass während des Fahrens die linguistische Aussage „*die Geschwindigkeit ist klein*“ die größte Gültigkeit besaß<sup>6</sup>.

Abb. 6.8 d+e zeigen den Organisationsstatus und Operationsstatus der Instanzen der Fähigkeit *Anhalten*. Die erste Instanz wird in ca. 18 Meter Entfernung vom Missionselement `halt` gestartet (Abb. 6.8 b-d, nach 253 s). Die Fahrzeugführung beginnt nach 259 s mit dem Bremsen. Dieser Übergang führt im Zustandsautomaten des zugehörigen internen Ablaufmodells zum Übergang von „*noch nicht bremsen*“ zu „*bremsen*“ (vgl. Abb. 6.8 f und Abschnitt 3.3.6). Nach 261,3 Sekunden ist das Bremsen beendet und das Fahrzeug steht. Der Operationsstatus und der Organisationsstatus wechseln in den Zustand `PASSIVE` und im Ablaufmodell erfolgt der Zustandsübergang von „*bremsen*“ zu „*Stillstand erreicht*“. Nach dem Beenden des Anhaltens wird die Instanz von `CD` gelöscht. Nach 10 Sekunden fährt das Fahrzeug wieder an (vgl. Abb. 6.8 a+b). Bei Sekunde 277,5 instanziiert `CD` eine neue Instanz der Fähigkeit *Anhalten*, welche nach 284 s aktiv wird.

<sup>6</sup> Neben `zero` und `small` besitzt der Situationsaspekt `Velocity` noch die linguistischen Werte `medium` und `high`.

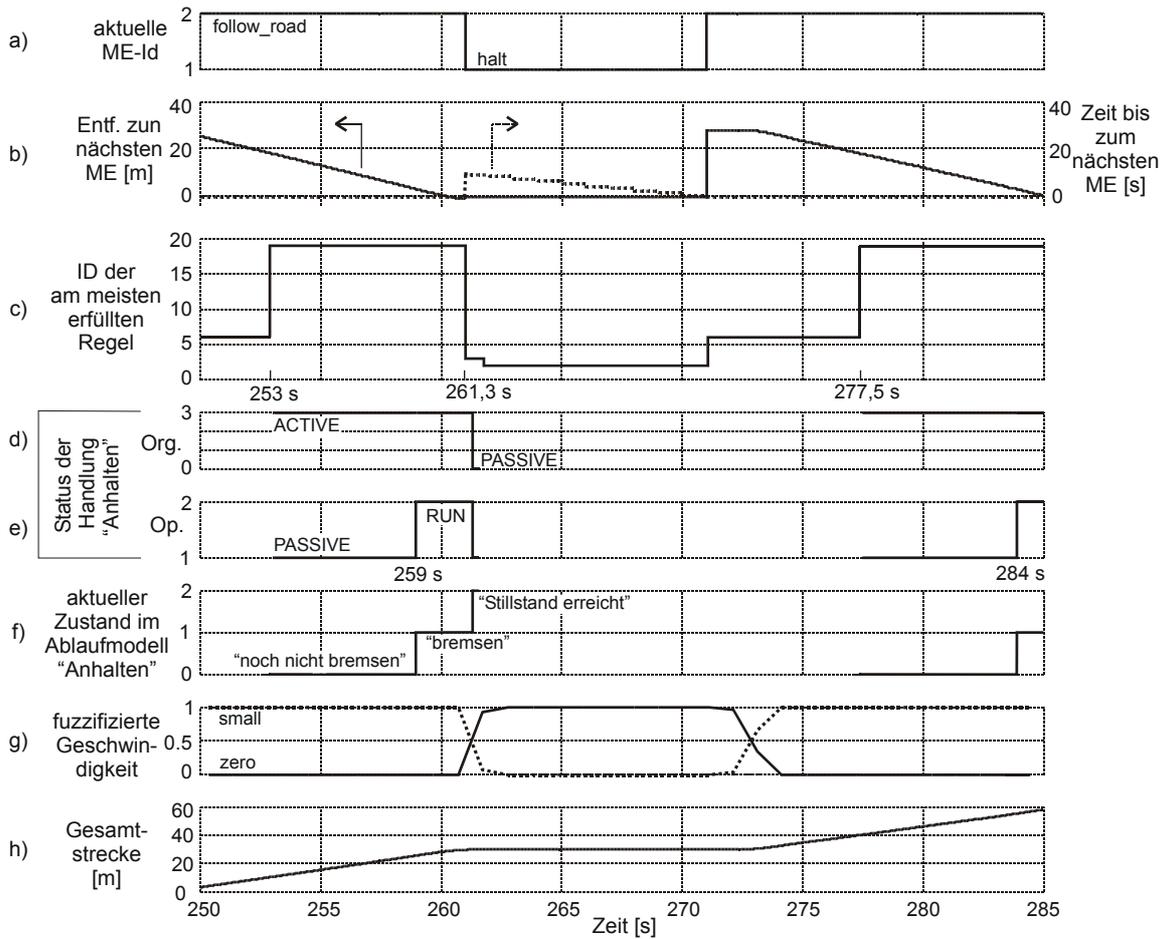


Abb. 6.8: Status der Instanzen der Fähigkeit **Anhalten** im Experten (d+e) und Zustand im zugehörigen internen Ablaufmodell (f)



# Kapitel 7

## Resultate der Blickrichtungssteuerung

Ein in verschiedenen Domänen fahrendes, autonomes Fahrzeug muss Objekte unterschiedlicher Größe, Position und Orientierung zuverlässig wahrnehmen. Die Testfahrzeuge VAMORS und VAMP sind zu diesem Zweck mit einer aktiven Kameraplattform und einer Blickrichtungssteuerung ausgestattet. Die Blickrichtungssteuerung interagiert dynamisch mit den Wahrnehmungsprozessen im System und mit der zentralen Verhaltensentscheidung. Dieses Kapitel zeigt Ergebnisse der Blickrichtungssteuerung, die während verschiedener autonomer Testfahrten gewonnen wurden.

### 7.1 Abbiegen

Mit dem Testfahrzeug VAMORS wurde autonomes Abbiegen sowohl auf unmarkierten, befestigten Straßen als auch auf unbefestigten Schotterstraßen gezeigt. Der in Abschnitt 4.5 beschriebene Optimierungsalgorithmus bestimmte dabei die Blickrichtung der Kameraplattform des Fahrzeugs. Abb. 7.1 zeigt einige Ergebnisse dieses Algorithmus während eines autonomen Abbiegens.

Während des Straßenfolgens sind zwei Objekte für die Blickrichtung relevant (Abb. 7.1 e, bis Sekunde 91): Das nahe Segment der befahrenen Straße<sup>1</sup> mit der Objekt-ID 2355 muss von den Weitwinkelkameras abgebildet werden und das ferne Segment (Objekt-ID 2356) von der milden Telekamera (vgl. Abschnitt 4.4 und Abb. 4.4). Wie aus Abschnitt 4.5 hervorgeht, berechnet der Algorithmus die Sichtbarkeitsbereiche (SB) aller Kombinationen von ROA anhand einer von den Wahrnehmungsexperten formulierten zweistufigen Logik. Wenn diese Sichtbarkeitsbereiche ungleich null sind, werden diese in entsprechende Listen in den zugehörigen VAGA-Objekten eingetragen. Die Abb. 7.1 i+j zeigen das jeweils erste Element dieser Liste für das ferne und nahe Segment der Eigenstraße (nur Gierwinkelintervalle). Für jede Planungsphase des Algorithmus erscheint in den Abb. 7.1 c-e und g-j ein Kreis<sup>2</sup>.

Bis zur Sekunde 91 werden vom Algorithmus zwei GC-Objekte generiert: GC-Objekt 0 beinhaltet die ‚Best Single‘-Lösung und das GC-Objekt 1 die ‚Best Pair‘-Lösung (vgl. Abb. 7.1 c+d). GC-Objekt 0 besitzt einen höheren Informationsgewinn ( $\sim 0,5$ ) als das GC-Objekt 1 ( $\sim 0,09$ ). Aus diesem Grund kommt GC-Objekt 0 zur Anwendung. In Abb. 7.1 a ist der Verlauf des Gierwinkels des Kameraträgers während des gesamten Abbiegevorganges dargestellt.

---

<sup>1</sup> Die vom autonomen Fahrzeug befahrene Straße wird im folgenden auch Eigenstraße genannt.

<sup>2</sup> Der Planungszeitraum ist i. A. 4 Sekunden. Externe Ereignis (z. B. das Auftauchen eines neuen, für die Blickrichtung relevanten Objekts) können jedoch schon vor Ablauf dieses Zeitraums eine Neuplanung anstoßen.

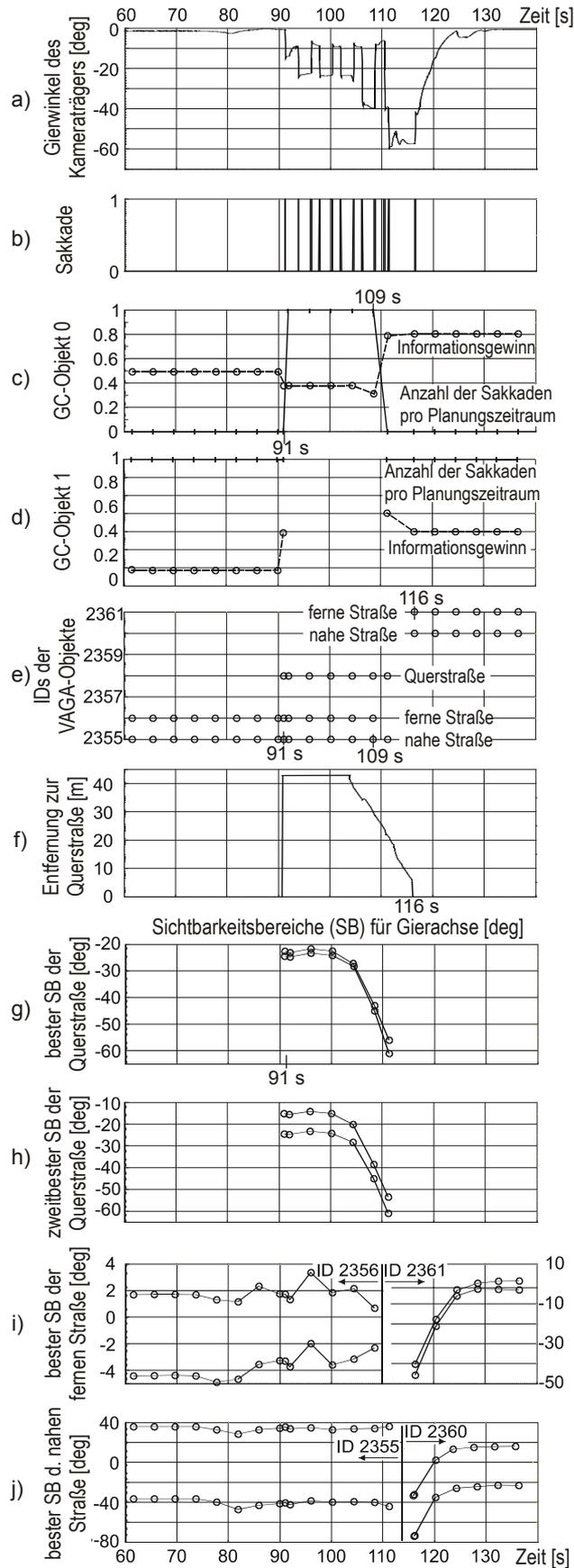


Abb. 7.1: Die Objekte in der Blickrichtungssteuerung während eines autonomen Abbiegens

Nach 91 Sekunden erscheint im Szenenbaum eine Objekthypothese für die Querstraße (ID 2358) (vgl. Abb. 7.1 e), so dass ab diesen Zeitpunkt drei Objekte für die Blickrichtung relevant sind. Abb. 7.1 g+h zeigen den besten und zweitbesten Sichtbarkeitsbereich für die Querstraße. Der beste Sichtbarkeitsbereich resultiert aus der Kombination aller drei ROA der Querstraße; der zweitbeste Sichtbarkeitsbereich ergibt sich aus der Kombination des inneren Paares von ROA relativ zum Abzweig (vgl. Abb. 4.4). Man sieht, dass die zweitbeste Lösung eine größere Variation des Gierwinkels erlaubt. In Abb. 7.1 f ist die geschätzte Entfernung zwischen Fahrzeug und Querstraße aufgetragen. Während des Annäherns an den Abzweig wandern die ROA der Querstraße entlang der Skelettlinie der Querstraße vom Abzweig weg, um die Aspektbedingungen für die Wahrnehmung zu verbessern (vgl. Abschnitt 4.4). Aus diesem Grund verkleinern sich die Gierwinkel der Sichtbarkeitsbereiche für die Querstraße und die Amplituden der Sakkaden (Abb. 7.1 a+g+h) nehmen zu.

Aus Abb. 7.1 g-j ist ersichtlich, dass im Zeitraum 91 - 109 s die Sichtbarkeitsbereiche des fernen Segments der Eigenstraße und die der Querstraße nicht überlappen. Aus diesem Grund existiert keine ‚Best Single‘-Lösung und GC-Objekt 0 beschreibt eine ‚Best Pair‘-Lösung mit einer Sakkade (ohne Anfangssakkade)<sup>3</sup> (vgl. Abb. 7.1 c). In Abb. 7.1 c ist ferner der Informationsgewinn dieser Sequenz aufgetragen. Während der ersten Blickfixierung mit einem Plattform-Gierwinkel von ca. 10° wird das ferne und nahe Segment der Eigenstraße abgebildet (vgl. Abb. 7.1 a). Die zweite Blickfixierung berücksichtigt das nahe Segment der Eigenstraße und die Querstraße. Abb. 7.1 b zeigt das Sakkaden-Bit, welches die Ausführung einer Sakkade signalisiert. Es dient den Wahrnehmungsexperten dazu, während der Ausführung von Sakkaden die Messung zu unterbrechen.

Nach 109 Sekunden ist aufgrund der Nähe der Kreuzung das ferne Segment der Eigenstraße nicht mehr relevant für die Blickrichtung. Ab diesem Zeitpunkt ist nur noch das nahe Segment zusammen mit der Querstraße abzubilden, wofür keine Sakkaden notwendig sind. Nach 116 Sekunden erreicht das autonome Fahrzeug den Abzweig und es erfolgt eine Reorganisation des Szenenbaums. Durch die Reorganisation wird aus der bisherigen Querstraße die neue Eigenstraße. Das nahe und ferne Segment der neuen Eigenstraße (IDs 2360+2361) sind nun relevant für die Blickrichtung. Da die Sichtbarkeitsbereiche dieser beiden Objekte überlappen (Abb. 7.1 i+j), existiert eine ‚Best Single‘-Lösung und es kommt zu keinen Sakkaden (Abb. 7.1 c).

Abb. 7.2 zeigt während des autonomen Abbiegens vom MARVEYE-Fahrzeugauge aufgenommene Bilder mit eingezeichneten Ergebnissen der Straßenerkennung. In der linken und mittleren Spalte von Abb. 7.2 befinden sich die Bilder der linken bzw. rechten Weitwinkelkameras; die rechte Spalte zeigt Bilder der milden Telekamera. In den Telebildern sind die vertikalen Suchpfade entsprechend der erwarteten bzw. gemessenen Position der Querstraße zu erkennen. Das nahe Segment der Eigenstraße wird mithilfe der Weitwinkelkameras unter Verwendung von horizontalen Suchfenstern an den Fahrbahnrändern vermessen. In großer Entfernung zum Abzweig ist eine gleichzeitige Abbildung der Querstraße und des linken Randes der Eigenstraße mit der Telekamera möglich (siehe Zeile a+b in Abb. 7.2). Während der Annäherung an den Abzweig wandern die ROA der Querstraße vom Abzweig weg, so dass der Kameraträger für deren Abbildung weiter gegiert werden muss. Die gleichzeitige Abbildung von Querstraße und Eigenstraße mit der milden Telekamera ist dann nicht mehr möglich und es kommt zur Ausführung von Sakkaden (Zeile c in Abb. 7.2). Die Bilder von Zeile d wurden nach dem Abbiegen aufgenommen. Wie schon vor dem Abbiegen erlaubt die Breite der Eigenstraße eine gleichzeitige Abbildung des linken und rechten Randes. Unab-

<sup>3</sup> Bei einer ‚Best Pair‘-Lösung ergeben sich pro Planungszeitraum zwei Sakkaden. In Abb. 7.1 c+d ist aber nur die Anzahl der (Zwischen-)Sakkaden ohne die Anfangssakkade aufgetragen.

hängig davon, welche Objekte die milde Telekamera fixiert, bleibt das nahe Segment der Eigenstraße immer im Gesichtsfeld der Weitwinkelkameras und kann so kontinuierlich vermessen werden (außer während der Ausführung von Sakkaden). Die Straßenerkennung wurde von LÜTZELER [2002] implementiert und beschrieben.

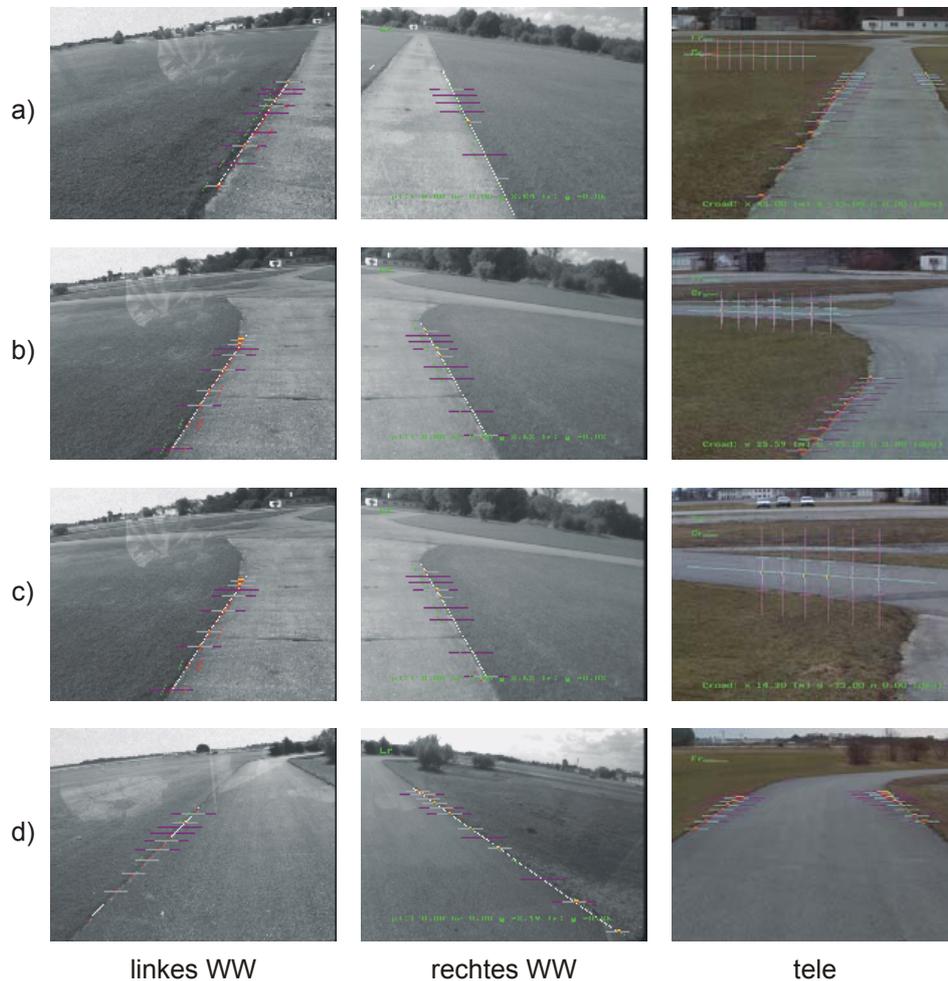


Abb. 7.2: Die Straßenerkennung mit Blickrichtungssteuerung während eines autonomen Abbiegens (nach [PELLKOFER et al., 2001])

## 7.2 Überqueren eines Abzweigs

Beim Überqueren eines Abzweigs konkurrieren vier Objekte um die Blickrichtung. Drei davon sind mit der milden Telekamera abzubilden: das ferne Segment der Eigenstraße, die Querstraße nach rechts (Abb. 7.3 b) und der jenseits der Kreuzung liegende Teil der Eigenstraße (im folgenden Straßenfortsetzung genannt) (Abb. 7.3 a). Das Fahrzeug soll laut dem Missionsplan der Straßenfortsetzung folgen. Die Querstraße und die Straßenfortsetzung haben unterschiedliche Bedeutungen, was sich auch auf ihre Relevanz und ihren Aufmerksamkeitsbedarf auswirkt: Die Straßenfortsetzung hat eine Relevanz für die Fortbewegung und ist von der Blickrichtungssteuerung bis zur Reorganisation des Szenenbaums zu berücksichtigen (Abb. 7.3 d). Die Querstraße dient lediglich zur lokalen Navigation: Durch sie kann die aufgrund von digitalen Karten und dem GPS-Signal nur ungenau bekannte Relativposition des Abzweigs mit einer für die Steuerung des autonomen Fahrzeugs ausreichenden Genauigkeit geschätzt werden (vgl. [SIEDERSBERGER, 2003]). Nach der Positionsbestimmung ist die Querstraße nicht mehr relevant für die Blickrichtung.

Beim Überqueren des Abzweigs kommt es zu einer Reorganisation des Szenenbaums. Aus der flächenbasiert vermessenen Straßenfortsetzung wird die neue, kantenbasiert vermessene Eigenstraße, deren nahes und fernes Segment relevant für die Blickrichtung sind. Vor der Reorganisation war es gerade noch möglich, die zwei entferntesten ROA am linken und rechten Straßenrand der Eigenstraße mit der Telekamera gleichzeitig wahrzunehmen (Abb. 7.3 a). Weil die Straßenerkennung nach dem Wiederaufsetzen der Zustandsschätzung das Formmodell der Straße von Nah nach Fern aufbaut (vgl. [LÜTZELER, 2002]), befinden sich die ROA der Eigenstraße zunächst näher beim Eigenfahrzeug. Eine gleichzeitige Abbildung des linken und rechten Randes des fernen Segments der Eigenstraße mit der Telekamera ist deshalb nicht möglich und die Blickrichtungssteuerung bildet den rechten Rand der Eigenstraße ab (Abb. 7.3 e).

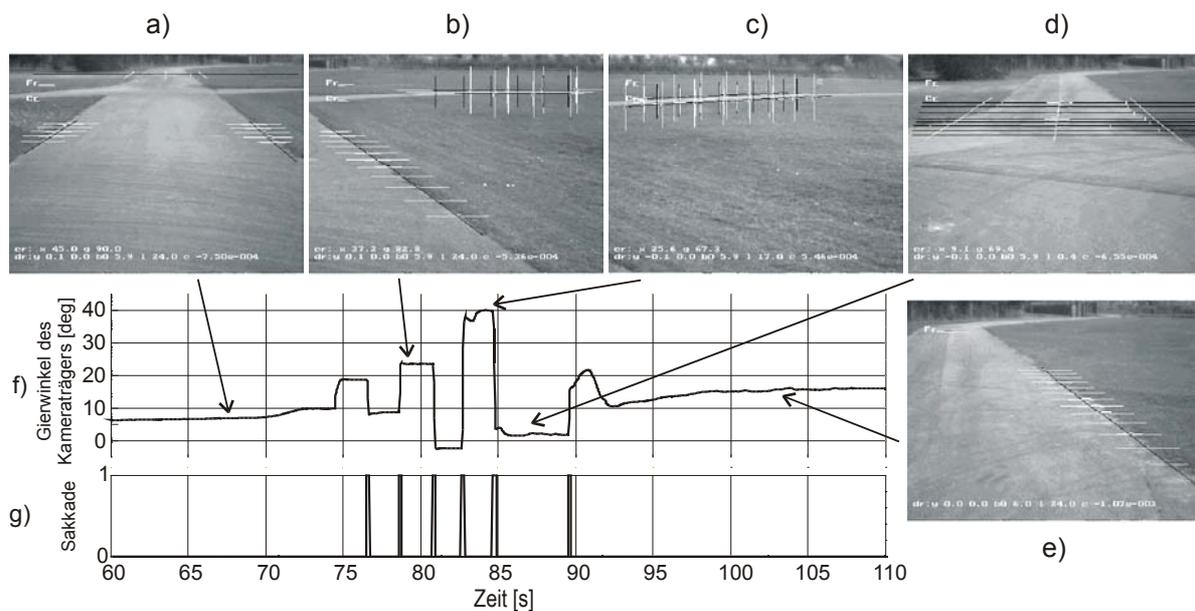


Abb. 7.3: Straßenerkennung mit Blickrichtungssteuerung beim Fahren über eine Kreuzung

### 7.3 Querfeldeinfahren mit Ausweichen vor einem Hindernis

Es ist die Aufgabe der zentralen Entscheidungsinstanz (CD), die Handlungen im System bezüglich Fortbewegung, Blickrichtung und Wahrnehmung zu koordinieren, so dass ihr Zusammenspiel das Fahrzeug dem Missionsziel näher bringt. Dieser Abschnitt zeigt ein Beispiel einer solchen komplexen Interaktion: Das Ausweichen vor einem Hindernis.

Vor der Darstellung der Ergebnisse einer Testfahrt sind einige Vorbemerkungen hinsichtlich des prinzipiellen Ablaufs notwendig. Die Sicherheit des autonomen Fahrzeugs gebietet es, dass während des Navigierens im unbefestigten Gelände durch eine geeignete Wahrnehmung gewährleistet ist, dass mögliche Hindernisse im Fahrschlauch erkannt werden. Zu diesem Zweck startet CD die Fähigkeit „Erkennung negativer Hindernisse“; abgekürzt PNO (Perceive Negative Obstacles). Daneben spezifiziert CD, in welchem Bereich in der Umgebung nach Hindernissen gesucht werden soll. Diesen für die Blickrichtungssteuerung relevanten Bereich definiert CD in Form eines Detektionsbereichs und fügt diesen als zusätzlichen Szenenknoten der Szenenrepräsentation hinzu.

Der Detektionsbereich hat zwei ineinanderliegende, trapezförmige ROA, welche den vom Fahrzeug überfahrenen Bereich großflächig abdecken (vgl. Abb. 7.4). Der Ursprung des durch



Hindernis unterschritten wird, so initiiert CD ein Anhalten vor dem Hindernis. Sobald das Fahrzeug zum Stillstand gekommen ist, entzieht CD dem Detektionsbereich vor dem Fahrzeug den Aufmerksamkeitsbedarf und das Hindernis verbleibt als einziges für die Blickrichtung relevantes Objekt. Parallel dazu beauftragt CD den Wahrnehmungsexperten PercG mit der Vermessung des linken und rechten Randes des Hindernisses (vgl. oben). Nach der erfolgreichen Vermessung startet CD das Ausweichen, und das Fahrzeug fährt aus dem Stand wieder an.

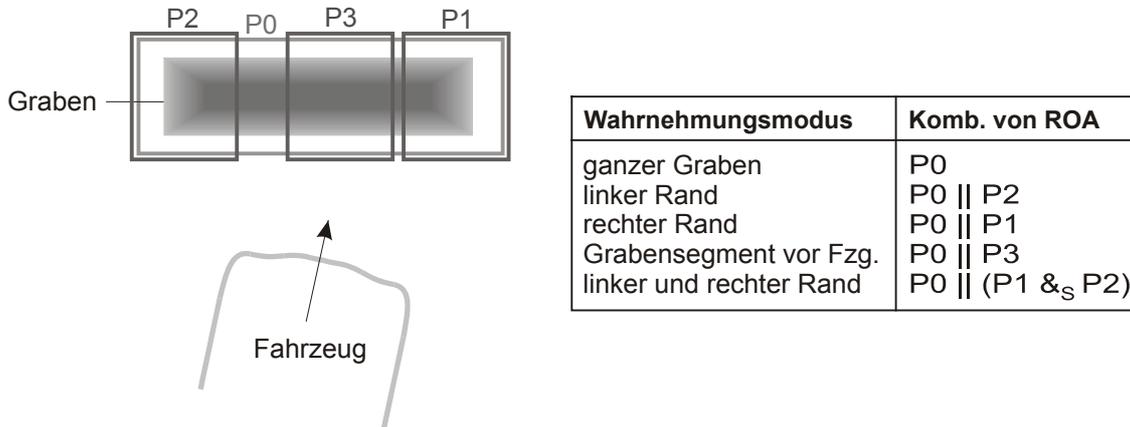


Abb. 7.5: Die Kombinationen der ROA eines rechteckigen Hindernisses in Abhängigkeit vom Wahrnehmungsmodus (vgl. [HOFMANN, 2003])

Nach diesen Vorbemerkungen werden nun die Ergebnisse einer Testfahrt diskutiert. Zu Beginn des in Abb. 7.6 aufgetragenen Zeitfensters bewegt sich das Fahrzeug mit geringer Geschwindigkeit durch das Gelände, wobei kein Hindernis erkannt ist (Abb. 7.6 d). Das einzige für die Blickrichtung relevante Objekt ist der Detektionsbereich vor dem Fahrzeug, dessen ROA  $P0_{DB}$  abgebildet wird<sup>5</sup>. In Abb. 7.7 b resultiert der 1. Sichtbarkeitsbereich (SB) aus  $P0_{DB}$  und der 2. Sichtbarkeitsbereich aus  $P1_{DB}$ , wobei in der Abbildung für jede Planungsphase ein Kreis erscheint. Während einer Fixationsphase müssen die Sichtbarkeitsbereiche der dabei berücksichtigten ROA zyklisch berechnet werden. Die Ergebnisse dieser zyklischen Berechnungen erscheinen in den Abb. 7.7 a+c.

Die vom Fahrzeug anzufahrenden GPS-Wegpunkte sind so positioniert, dass das autonome Fahrzeug auf einen Graben mit der Länge von ca. 6 m und einer Breite von ca. 1,2 m trifft und zu einem Ausweichen gezwungen ist. Nach 97,5 s erkennt die Wahrnehmung den Graben (Abb. 7.6 e-f). Da sich der Graben im Fahrschlauch befindet, wird der Experte der Wahrnehmungsfähigkeit PNO damit beauftragt, den rechten und linken Rand des Hindernisses zu vermessen (Abb. 7.6 b+c). Entsprechend diesem Wahrnehmungsauftrag setzt der Prozess PercG die ROA für den rechten und linken Rand der Grabenhypothese und verknüpft diese mit dem Operator AND\_SEQU (Abb. 7.6 g). Aus Abb. 7.7 e+g ist ersichtlich, dass die sequentielle Berücksichtigung des Grabens aus zwei Teilen besteht, wobei der 1. SB durch den 0. Teil der Sequenz abgebildet wird und der 2. SB durch den 1. Teil.

Zu Beginn der Grabenerkennung besitzt die Hypothese des Grabens einen Wert für die Länge von ca. 5 m und überdeckt damit nur einen Teil des Grabens (Abb. 7.6 f). Dies hat zur Konsequenz, dass nach der Planung der Blickrichtung die Objektgruppen  $\{1. SB_{DB} \wedge 1. SB_{Graben}\}$ <sup>6</sup> und  $\{1. SB_{DB} \wedge 2. SB_{Graben}\}$  nacheinander abgebildet werden sollen (Abb. 7.7

<sup>5</sup>  $Pi_{DB}$  und  $Pi_{Graben}$  stehen für die ROA des Detektionsbereichs (DB) bzw. des Grabens.

<sup>6</sup>  $SB_{DB}$  und  $SB_{Graben}$  bezeichnen die Sichtbarkeitsbereiche des Detektionsbereichs (DB) bzw. des Grabens.

b+d+f, nach 97,5 s). Zuerst fixiert die Kameraplattform die erste Objektgruppe  $\{1. SB_{DB} \wedge 1. SB_{Graben}\}$ , wobei die Sichtbarkeitsbereiche der daran beteiligten ROA zyklisch berechnet werden (Abb. 7.7 a+c).

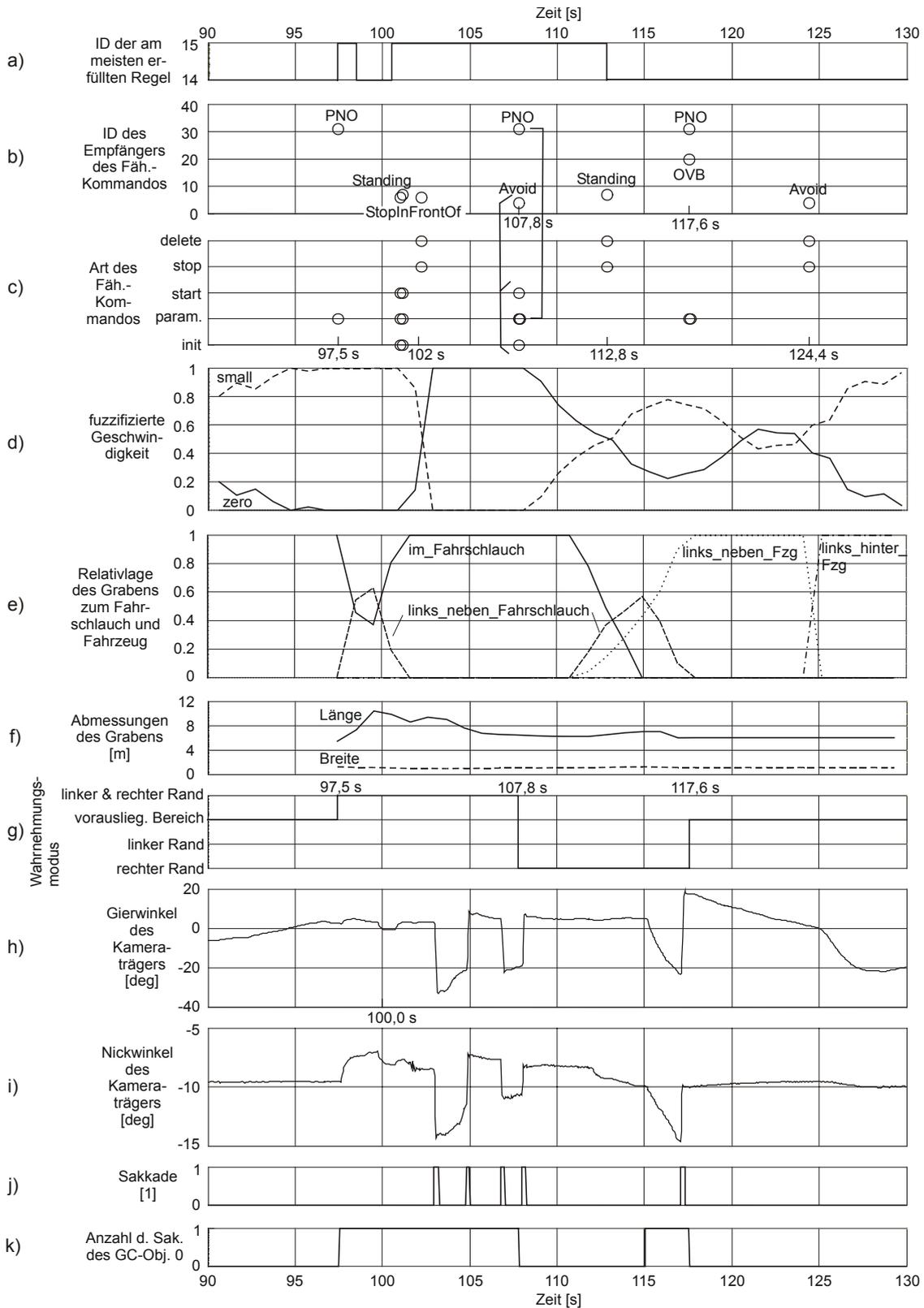


Abb. 7.6: Querfeldeinfahren und Ausweichen vor einem Hindernis

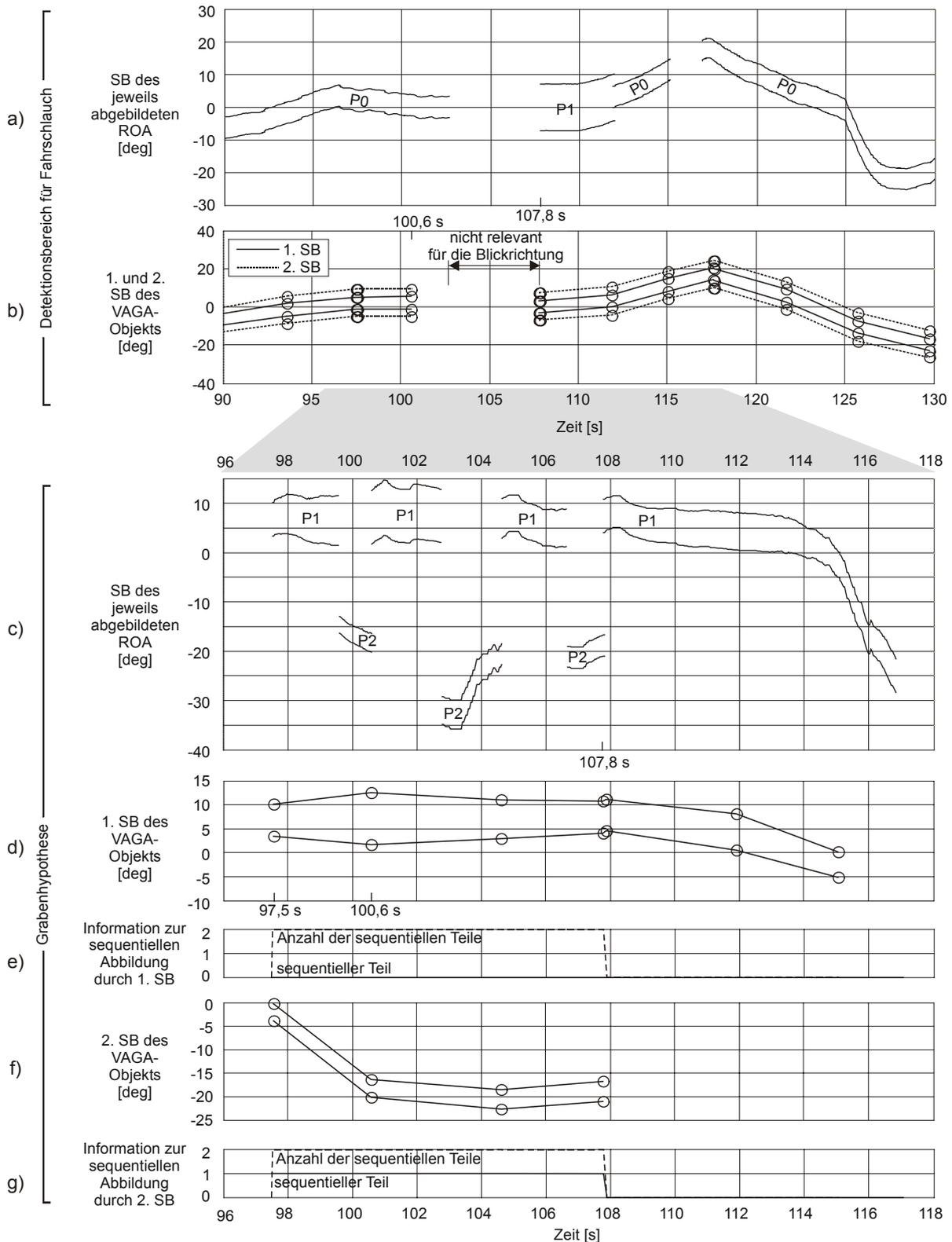


Abb. 7.7: Die Sichtbarkeitsbereiche beim Ausweichen

Während der Ausführung der Sequenz nähert sich das Fahrzeug dem Graben weiter an. Zusätzlich verdoppelt sich infolge störender Einflüsse<sup>7</sup> die Länge der Grabenhypothese auf ca. 10 m (Abb. 7.6 f). Dies hat zur Folge, dass die zum Zeitpunkt 97,5 s geplante zweite Phase der Sequenz mit der Objektgruppe  $\{1. SB_{DB} \wedge 2. SB_{Graben}\}$  nicht mehr ausführbar ist, da sich die Sichtbarkeitsbereiche von  $P0_{DB}$  und  $P2_{Graben}$  nicht mehr überlappen. Wie Abschnitt 4.5

<sup>7</sup> Aufgrund des unebenen Untergrunds kam es während der Fahrt zu Schwankungen des Fahrzeugaufbaus.

beschreibt, verharrt in einem solchen Fall die Kameraplattform für eine gewisse Anzahl von Zyklen<sup>8</sup> in der zuletzt eingenommen Position (Abb. 7.6 h, nach ca. 100,0 s). Nach ca. 100,6 s ist der Schwellwert erreicht und es wird eine Neuplanung angestoßen. Die dabei gefundene Lösung besteht aus den zwei Objektgruppen  $\{1. SB_{DB} \wedge 1. SB_{Graben}\}$  und  $\{2. SB_{Graben}\}$ .

Nach 101,0 s hat sich das Fahrzeug dem Graben so weit genähert, dass der minimale Zeitabstand oder der minimale Abstand unterschritten ist. Daraufhin startet CD die Fähigkeit `StopInFrontOf` (Abb. 7.6 b+c). Diese Fähigkeit wird vom BDL unmittelbar ausgeführt, was wiederum den Start der Fähigkeit `Standing` nach sich zieht. Die (geringe) Geschwindigkeit des Fahrzeugs sinkt sehr schnell ab und das Fahrzeug kommt bereits nach 1,2 s zum Stehen (Abb. 7.6 d, nach 102,2 s). Nach Erreichen des Stillstands wird die Instanz der Fähigkeit `StopInFrontOf` passiv und durch CD gelöscht (Abb. 7.6 b+c) und der Detektionsbereich verliert seine Relevanz für die Blickrichtung (Abb. 7.7 a).

Nun fixiert die Kameraplattform abwechselnd den linken und rechten Rand der Grabenhypothese, wobei Sakkaden zur Anwendung kommen (Abb. 7.6 h-k und Abb. 7.7 c). In Abb. 7.8 ist der Gierwinkel des Kameraträgers für diesen Zeitabschnitt nochmals vergrößert aufgetragen und den dazugehörigen Videobildern gegenübergestellt<sup>9</sup>. Durch die Abbildung der Grabenränder wird dem Wahrnehmungsexperten PercG die Möglichkeit gegeben, das Formmodell der Grabenhypothese der Realität anzupassen. Dies führt dazu, dass PercG die Hypothese an den Enden auf die tatsächliche Länge verkürzt (vgl. Abb. 7.8). Diese Verkürzung hat Auswirkungen auf die darauf positionierten ROA (vgl. Abb. 7.7 c) und die Blickrichtung wird während der Fixationsphase den sich verschiebenden ROA nachgeführt (Abb. 7.8).

Nach 107,8 s signalisiert PercG die erfolgreiche Vermessung des Grabens, worauf CD die Fähigkeit `AvoidObstacle` startet und das Fahrzeug wieder anfährt. Parallel dazu erkennt CD dem Detektionsbereich vor dem Fahrzeug wieder einen Aufmerksamkeitsbedarf zu (Abb. 7.6 b+c und Abb. 7.7 a+b) und ändert den Wahrnehmungsauftrag für PercG: Da im vorliegenden Fall ein Ausweichen nach rechts erfolgt, ist von nun an nur noch der rechte Rand des Grabens wahrzunehmen (Abb. 7.6 g).

Da der Detektionsbereich für die Blickrichtungssteuerung wieder relevant geworden ist, wird innerhalb BDGA eine Planungsphase ausgelöst (Abb. 7.7 b+d), welche eine Lösung ohne Sakkade mit der Objektgruppe  $\{1. SB_{DB} \wedge 1. SB_{Graben}\}$  findet (Abb. 7.7 a+c+k). Nach 112,8 s liegt der Graben aufgrund der Ausweichbewegung des Fahrzeugs nicht mehr im Fahrerschlauch, sondern links neben dem Fahrerschlauch (Abb. 7.6 e). Dieses Ereignis bewirkt einen Wechsel in der Regelbasis (Abb. 7.6 a) und es wird die seit dem Wiederanfahren des Fahrzeugs (nach ca. 108 s) passiv gewordene Instanz der Fähigkeit `Standing` von CD gelöscht (Abb. 7.6 b+c). Während des Umfahrens des Grabens verbessern sich die Aspektbedingungen für die Wahrnehmung des Grabenendes gravierend. Deshalb ist es durchaus möglich, dass während des Umfahrens die Position des Grabenendes korrigiert werden muss. Ferner benutzt die Fahrzeugführung während des Ausweichens das Grabenende zur lokalen Navigation. Aus diesen Gründen ist es sinnvoll, dass der Graben auch nach dem Verlassen des Fahrerschlauchs zunächst seinen Aufmerksamkeitsbedarf behält und damit relevant für die Blickrichtung bleibt. Erst wenn die Winkel aller Eckpunkte des Grabenmodells im Fahrzeugkoordinaten-

<sup>8</sup> In der hier dargestellten Testfahrt war dieser Schwellwert (`CGCObject::m_iNoOfMissingIntersections`) mit 15 belegt (vgl. Abschnitt 4.5). Wenn seitens der Blickrichtungssteuerung schneller auf sich räumlich rasch ändernde Objekthypothesen reagiert werden soll, ist ein kleinerer Wert (z. B. 5-10) zu wählen.

<sup>9</sup> Die Grabenerkennung arbeitet nicht mit den in Abb. 7.8 gezeigten Bildern der 3-Chip-Farbkamera, sondern mit dem monochromen Stereokamerapaar. Das Bild der Farbkamera diente aus technischen Gründen der Visualisierung der Grabenhypothese. Bei der Visualisierung ist zu beachten, dass der (wirksame) Öffnungswinkel der Stereokameras (18° horiz.) kleiner ist als der Öffnungswinkel der Farbkamera (22° horiz.).

system relativ zur x-Achse (Fahrzeuglängsachse) unter einem Winkel größer  $75^\circ$  erscheinen<sup>10</sup>, und damit eine Kollision mit dem Graben ausgeschlossen werden kann, entzieht CD dem Graben den Aufmerksamkeitsbedarf.

Nach 115,1 s ist eine gemeinsame Abbildung von rechtem Grabenrand und Detektionsbereich nicht mehr möglich und es werden Sakkaden zwischen dem 1.  $SB_{DB}$  und 1.  $SB_{Graben}$  notwendig (Abb. 7.6 k). Aus Abb. 7.7 c ist ersichtlich, dass aufgrund der Fahrzeugbewegung die Abbildung des rechten Grabenrandes einen immer größeren Gierwinkel erfordert.

Zum Zeitpunkt 117,6 s wird der oben genannte Winkelschwellwert überschritten und CD entzieht dem Graben den Aufmerksamkeitsbedarf. Nun ist der Detektionsbereich wieder das einzige für die Blickrichtung relevante Objekt, dessen Sichtbarkeitsbereiche sich entsprechend den Lenkbewegungen des Fahrzeugs ändern (Abb. 7.7 a+b). Zudem setzt CD den Wahrnehmungsauftrag für den Prozess PercG zurück und stößt eine Neuplanung<sup>11</sup> des Blickverhaltens an (Abb. 7.6 b+c).

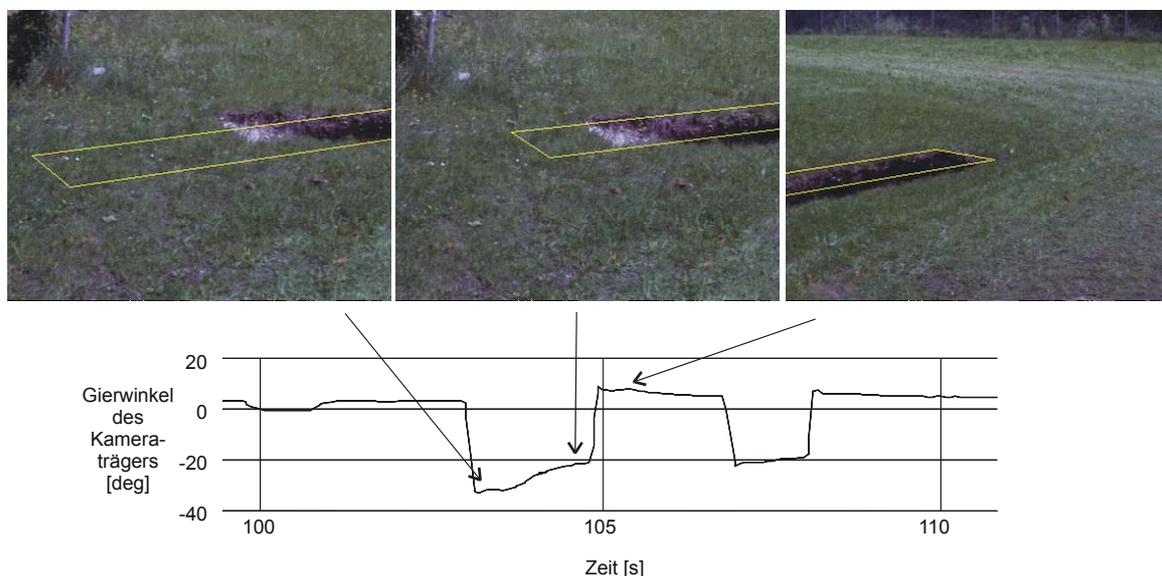


Abb. 7.8: Korrektur der Hypothese und Nachführen der Blickrichtung (vgl. auch [HOFMANN, 2003])

In Abb. 7.6 e ist ersichtlich, wie während des Ausweichens die symbolische Relativlage des Grabens wechselt:

im\_Fahrschlauch  $\rightarrow$  links\_neben\_Fahrschlauch  $\rightarrow$  links\_neben\_Fzg  $\rightarrow$  links\_hinter\_Fzg

Nach 124,4 s ist das Ausweichen beendet und CD löscht die passiv gewordene Instanz der Fähigkeit AvoidObstacle (Abb. 7.6 b+c). Nach dem Ausweichen ist wieder die Fähigkeit Waypoint\_Navigation aktiv.

<sup>10</sup> Dieser Schwellwert hat sich in zahlreichen Testfahrten als geeignet erwiesen.

<sup>11</sup> Verliert ein Objekt seine Relevanz für die Blickrichtung, so löst dieses Ereignis im Prozess BDGA keine Neuplanung aus.



## Kapitel 8

### Zusammenfassung und Ausblick

Die vorliegende Arbeit schlägt eine Brücke zwischen verschiedenen Wissenschaftsgebieten. Die vorgestellten Konzepte und Implementierungen erlauben es, Bildverarbeitung, Fahrzeugführung, Situationsanalyse, Verhaltensentscheidung, Blickrichtungssteuerung und Missionsplanung im „geschlossenen Kreis“ – d. h. mit Rückkopplung über die reale Welt – zu betreiben, zu testen und weiterzuentwickeln.

Der Beginn dieser Arbeit beschäftigt sich mit dem ‚Aktiven Sehen‘ als vielversprechende Strategie für die Entwicklung robuster Sehsysteme für autonome Fahrzeuge und mobile Roboter. Wie dabei ersichtlich wird, kann man nicht davon ausgehen, dass Algorithmen und Vorgehensweisen, die in einer künstlichen oder virtuellen Welt funktionieren, mit einigen Erweiterungen auch in natürlichen Umgebungen tragfähig sind. Die Devise „*erst wahrnehmen – dann agieren*“ führte bei maschinellen Sehsystemen in natürlichen Umgebungen viele Jahre lang nur zu kleinen Fortschritten (vgl. Kapitel 2.1.1). Vielmehr muss die Devise lauten: „*wahrnehmen und agieren*“. Viele Probleme, mit denen Sehsysteme und autonome Fahrzeuge zu kämpfen haben, sind eben nur durch Aktion und durch eine dynamische Interaktion mit den Objekten in der Umgebung zu lösen. Es macht deshalb Sinn, den Wirkungskreis so früh wie möglich zu schließen, um ein reaktives System zu erhalten.

Nach dem aktuellen Stand der Technik bezüglich der Entwicklung aktiv sehender Fahrzeuge und mobiler Roboter folgt eine kurze Gegenüberstellung der Methoden zur Modellierung von Wissen und Verhalten. Die in dieser Arbeit verwendete Modellierungsmethode durch linguistische Aussagen und unscharfe Regeln wird dabei ausführlicher beschrieben. Der Hauptteil dieser Arbeit stellt Konzepte und Implementierungen vor, die für eine geschlossene Verhaltensmodellierung für ein autonomes Fahrzeug notwendig sind. Die Konzepte zeichnen sich durch Transparenz, leichte Erweiterbarkeit und Robustheit aus. Es wird darauf verzichtet, eine explizite Situationsbeschreibung zu erstellen, sondern ein Ansatz aufgegriffen, der mit mehr oder weniger detaillierten Umschreibungen von Situationen arbeitet.

Ein wichtiger Teil der Arbeit war die explizite Beschreibung der Fähigkeiten des autonomen Systems. Dazu implementierte der Autor das in der Arbeitsgruppe VERHALTEN [2000] entworfene Fähigkeitenkonzept. Das daraus entstandene *Fähigkeitennetz* repräsentiert die einzelnen Fähigkeiten der Systemkomponenten mit ihren unterschiedlichen Abstraktionsgraden und ihren Abhängigkeiten. Das Fähigkeitennetz bietet eine umfangreiche Funktionalität an, die es erlaubt, die Aktivitäten im verteilten System situations- und zielgerecht zu steuern und zu koordinieren. Ein mit Statusmeldungen und Fortschrittsmaßen arbeitender Mechanismus ermöglicht es der zentralen Verhaltensentscheidung, auf Mängel während der Ausführung von Aktionen und auf Systemausfälle richtig zu reagieren.

Für die Verhaltensmodellierung wurden Regelbasen mit unscharfen Regeln für unterschiedliche Domänen entwickelt. Die Situationsumschreibungen allgemeiner Regeln decken dabei große Teile des Situationsraumes ab; detaillierter beschreibende Regeln formulieren Ausnahmen davon. Durch diese Vorgehensweise ist die Anzahl der notwendigen Regeln beschränkt. Es werden Konzepte zur effizienten Gestaltung der Regelbasen vorgestellt und eine iterative Ausführung von Situationsanalyse und Verhaltensentscheidung beschrieben, welche eine minimale, aber ausreichende Situationsrepräsentation generiert.

Ein zweiter, wichtiger Punkt dieser Arbeit ist die Entwicklung einer aktiven Blickrichtungssteuerung im EMS-VISION System. Durch sie macht das autonome Fahrzeug den Schritt vom (passiven) Beobachter zum aktiv sehenden Verkehrsteilnehmer, welcher durch sein multifokales Fahrzeugauge und durch seine aktive Kameraplattform bestimmen kann, welche Information mit welcher Auflösung er aus der Umwelt aufnimmt. Dem Autor ist aus der Literatur kein weiteres Kraftfahrzeug bekannt, bei dem durch eine Blickrichtungssteuerung die Bedürfnisse und Eigenschaften verschiedener Wahrnehmungsprozesse und die Geometrien und Dynamiken der wahrzunehmenden Objekte berücksichtigt werden.

Die Implementierung der vorgestellten Konzepte ist möglichst erschöpfend dargestellt. Die Darstellung benützt die allgemein anerkannte Modellierungssprache UML (*Unified Modeling Language*).

Im experimentellen Teil dieser Arbeit wird die Funktionalität der einzelnen Module zur Verhaltensentscheidung und Blickrichtungssteuerung schrittweise nachgewiesen. Das am ISF entwickelte EMS-VISION System dient dabei als experimentelle Grundlage. Höhepunkt der Experimente ist ohne Zweifel die vollautonome Ausführung einer komplexen, abstrakt formulierten Mission durch das Versuchsfahrzeug VAMORS, welche das Fahrzeug in die Domänen „*Fahren auf Straßen niederer Ordnung*“ und *Querfeldeinfahren* führt. Diese Mission wurde einem internationalen Publikum auf dem Testgelände *live* demonstriert.

Ein wichtiges Element dieser Mission stellt das autonome Abbiegen dar, anhand dessen die Abläufe in der Blickrichtungssteuerung detailliert diskutiert werden. Das Ausweichen vor einem Graben bedingt eine komplexe Interaktion zwischen Wahrnehmung, Blickrichtungssteuerung und Verhaltensentscheidung. Auch diese wird mit experimentellen Ergebnissen nachgewiesen.

Aus der Sicht des Autors sind folgende Weiterentwicklungen des Systems möglich:

- Ohne Zweifel ist es erforderlich, dass die Güte der geschätzten systemdynamischen Größen in die Steuerung der Blickrichtung mit einfließt. Zu diesem Zweck müssen die aktuellen Varianzen der Zustandsgrößen und die (für die Fortbewegung) erforderlichen Genauigkeiten miteinander verglichen werden. Verwendbar sind dabei nicht relative Größen und deren Änderungen, sondern absolute Größen mit einer physikalischen, geometrischen Bedeutung. Die Gewinnung solcher absoluten Gütemaße aus der Bildverarbeitung ist eine interessante, komplexe Forschungsaufgabe für die Zukunft.
- Durch das vorgestellte EMS-VISION System sind der Erprobung von KI-Methoden keine Grenzen gesetzt. So ist zu erörtern, auf welche Weise Lernen in das System integriert werden kann. Denkbar ist zum Beispiel, (Bewegungs-)Fertigkeiten zu lernen. Das Fähigkeitsnetz ist dann nicht mehr statisch, sondern es wächst mit den Erfahrungen des Systems. Lernmechanismen führen dazu, dass Subjekte mit ursprünglich gleichem Hardware- und Softwarestand sich mit der Zeit auseinanderentwickeln und zu individuellen Subjekten mit unterschiedlichen Fähigkeiten werden.

- Ferner ist es möglich, die Modularisierung im Rahmen des Fähigkeitenkonzepts konsequent weiterzuentwickeln und ein verteiltes System aufzubauen, in dem die Prozesse keinen bestimmten Aufgaben mehr zugeordnet sind. Vielmehr werden beim Systemstart leere, nicht unterscheidbare Prozesshülsen eingerichtet, welche zunächst keine Funktion erfüllen. Soweit es die Hardwareanbindungen erlauben, ist es der Verhaltensentscheidung dann möglich, eine Fähigkeit in einer dieser Prozesshülsen an jedem beliebigen Ort im verteilten System ausführen zu lassen, oder während des Betriebs Funktionalität im System zu verlagern. Ein solches, „flüssiges“ System erlaubt es, die Rechenlast gleichmäßig zu verteilen und situationsbedingt Ressourcen für Sonderaufgaben zur Verfügung zu stellen. Ein weiterer Vorteil eines solchen Systems ist seine zusätzliche Robustheit: So bleibt das System funktionsfähig, wenn ganze Teile des Systems zerstört werden oder ausfallen, indem die Verhaltensentscheidung die weggefallenen Rechenoperationen auf die noch vorhandenen Rechenknoten verteilt<sup>1</sup>.
- Des Weiteren sollten die Verhaltensentscheidungen verschiedener autonomer Systeme die Möglichkeit besitzen, Modelle, Missionspläne, Landkarten, Datenstrukturen, Regelgesetze und komplette Fähigkeiten mit anderen Subjekten auszutauschen. In einem solchen Fall agiert ein Subjekt nicht mehr für sich alleine, sondern als individueller Teil einer Gemeinschaft von kooperativen Subjekten [DICKMANN, 1998]. Jedes Subjekt kann aus der Koordination seiner Aktionen mit den Aktionen anderer und durch Erfahrungsaustausch Fähigkeiten erlangen, die es als einzelnes Subjekt nicht besitzt und niemals erlernen würde. Der Grund dafür liegt in der Verlagerung der Situationsanalyse und Verhaltensentscheidung auf eine höhere Hierarchiestufe [DICKMANN & FÜRST, 1998]: Diese höhere Ebene fusioniert die von den einzelnen Subjekten wahrgenommene Umgebungsinformation zu einer gemeinsamen, globalen Situationsbeschreibung „aus der Vogelperspektive“. Die Analyse dieser gemeinsamen Situationsbeschreibung bestimmt dann das Verhalten der einzelnen Subjekte.

Hier tritt jedoch das grundsätzliche Problem auf, dass die Ziele des Kollektivs mit den Zielen des Individuums nicht immer übereinstimmen müssen und die Frage zu beantworten ist, welche der zwei Arten von Zielen höher priorisiert wird. Ein Kollektiv, das von der Kooperationsbereitschaft seiner Individuen abhängt, kann nicht funktionieren, wenn die Individuen die Prioritätenverteilung nicht akzeptieren. Diese Art von Problemen, welche man innerhalb menschlicher Gemeinschaften oft mit den Begriffen soziales Verhalten und Moral in Verbindung bringt, stellen sich auch für technische autonome Systeme.

Nach WITTIG & SCHULTE [1993] werden etwa 90 % aller relevanten Information durch die visuelle Sinnesmodalität aufgenommen. Die visuelle Sinnesmodalität ist auch am stärksten beim Aufbau einer internen Repräsentation der Umwelt beteiligt. Deshalb erscheint es vielversprechend, das aktive maschinelle Sehen in zukünftigen Automobilen anzuwenden. In der Automobilindustrie ist man skeptisch, ob sich „teure“ Kameratechnik in Automobilen aus Kostengründen durchsetzen wird, insbesondere wenn diese aktiv ansteuerbar ist. Sicher ist jedoch, dass Farbbildverarbeitung und ein gewisser Brennweitenbereich zwingend notwendig sind, wenn man die Sehleistung des Menschen erreichen will. Ein komplexes und robustes Sehsystem wird sich aus finanziellen Gründen nicht durchsetzen, wenn es nur eine Funktion erfüllt. Dies gilt insbesondere dann, wenn der Käufer diese Funktion im Normalfall gar nicht wahrnimmt, wie dies z. B. bei Warnsystemen der Fall ist. Wenn aber die Flexibilität und An-

---

<sup>1</sup> Diese Art der Funktionsverlagerung tritt auch bei Gehirnschäden auf, wo Funktionen der geschädigten Gehirnparten durch andere, intakte Gehirnteile übernommen werden.

passungsfähigkeit dieses Sehsystems in vielen erfahrbaren Applikationen zum Tragen kommt und eine echte Entlastung des Fahrers nach sich zieht, werden sich Käufer finden.

## Literatur

- [ALOIMONOS, 1993] Aloimonos Y.: *Introduction: Active Vision Revisited*, in: Active Perception, Aloimonos Y. (Ed.), S. 1-18, Erlbaum, Hillsdale (NJ), USA, 1993.
- [ARKIN, 1998] Arkin R. C.: *Behavior-Based Robotics*, The MIT Press, Cambridge, Massachusetts, 1998.
- [BAJCSY, 1988] Bajcsy R.: *Active Perception*, Proceedings of the IEEE **76**, S. 996-1005, August 1988.
- [BALLARD & BROWN, 1982] Ballard D. H, Brown Ch. M.: *Computer Vision*, Prentice-Hall, Englewood Cliffs (NJ), USA, 1993.
- [BALLARD & BROWN, 1993] Ballard D. H, Brown Ch. M.: *Principles of Animate Vision*, in: Active Perception, Y. Aloimonos (Ed.), S. 245-282, Erlbaum, Hillsdale, 1993.
- [BANDER & GOTTWALD, 1993] Bander H., Gottwald S.: *Einführung in die Fuzzy-Methoden*, 4. Auflage, Akademik Verlag, Berlin, 1993.
- [BARATOFF et al., 1999] Baratoff G., Ahrns I., Toepfer Ch., Neumann H.: *Ortsvariantes aktives Sehen: Von biologischer Motivation zu technischer Realisierung*, in: Künstliche Intelligenz, 1/1999, S. 33-35, arenDTaP Verlag, Bremen, 1999.
- [BASTISTA et al, 1997] Bastista J., Peixoto P., Araújo H.: *Real-Time Vergence and Binocular Gaze Control*, Intern. Conf. on Intelligent Robots (IROS '97), Vol. **3**, S. 1348-1354, New York (NY), USA, 1997.
- [BERNASCH, 1997] Bernasch J.: *Konstruktion und Anwendung einer maschinellen Aufmerksamkeitssteuerung für komplexe Bildfolgen*, Dissertation, Universität Dortmund, 1997.
- [BOLLMANN, 1999] Bollmann M.: *Entwicklung einer Aufmerksamkeitssteuerung für ein aktives Sehsystem*, Dissertation, Universität Hamburg, 1999.
- [BRILL et al., 1995] Brill F. Z., Martin W. N., Olson T. J.: *Markers Elucidated and Applied in Local 3-space*, Intern. Symp. on Computer Vision, S. 49-54, Los Alamitos (CA), USA, 1995.
- [BROCKMANN et al., 1999] Brockmann W., Heide W., Kluthe R., Kömpf D., Maehle E., Sprenger A.: *Modellierung der kortikalen Steuerung von Blicksakkaden bei der visuellen Exploration*, in: Künstliche Intelligenz, 1/1999, S. 36-38, arenDTaP Verlag, Bremen, 1999.
- [BROOKS, 1986] Brooks R. A.: *A Robust Layered Control System for a Mobile Robot*, IEEE Journal of Robotics and Automation, RA:2(1), S. 14-23, 1986.
- [BROOKS, 1992] Brooks R. A.: Vorwort zu *Active Vision*, A. Blake und A. Yuille (Eds.), MIT Press, London, 1992.

- [BRUCKHOFF & DAHM, 1998] Bruckhoff C., Dahm P.: *Neural Fields for Local Path Planning*, IEEE Intern. Conf. of Intelligent Robots and Systems, Victoria (B.C.), Canada, Oktober 1998.
- [CHARNIAK & MCDERMOTT, 1985] Charniak E., McDermott D.: *Artificial Intelligence*, Addison-Wesley Publishing, Reading: MA, 1985.
- [CHERFAOUI et al., 2000] Cherfaoui V., Burie J.-Ch., Royère C., Gruyer D.: *Dealing With Uncertainty in Perception Systems for the Characterization of Driving Situations*, IEEE Intelligent Transportation Systems, S. 83-88, Dearborn (MI), USA, 2000.
- [CLARK, 1998] Clark J. J.: *Spatial Attention and Saccadic Camera Motion*, IEEE Intern. Conf. on Robotics & Automation, Leuven (Belgien), S. 3247-3252, Mai 1998.
- [DAVISON & MURRAY, 1998] Davison A. J., Murray D. W.: *Mobile Robot Localisation Using Active Vision*, European Conf. on Computer Vision, Vol. 2, S. 809-825, Springer Verlag, Berlin, 1998.
- [DENZLER, 1997] Denzler J.: *Aktives Sehen zur Echtzeitobjektverfolgung*, Dissertation, Universität Erlangen-Nürnberg, Sankt Augustin (Infix), 1997.
- [DIAS et al., 1998] Dias J., Fonseca I., Lobo J.: *New Perspectives on Mobile Robot Navigation with Visual and Inertial Information*, in: Advanced Motion Control Procs., S. 261-266, 1998.
- [DICKMANN, 1987] Dickmanns E. D.: *4D-Szenenanalyse mit integralen raumzeitlichen Modellen*, Paulus (Ed.), 9. DAGM-Symposium Mustererkennung, Informatik Fachberichte, Nr. 149, Springer Verlag, Braunschweig, 1987.
- [DICKMANN, 1989] Dickmanns E. D.: *Subject-Object Discrimination in 4D-Dynamic Scene Interpretation for Machine Vision*, IEEE-Workshop on Vision Motion, S. 298-304, Irvine, 1989.
- [DICKMANN, 1992] Dickmanns E. D.: *Machine Perception Exploiting High-Level Spatio-Temporal Models*, AGARD Lecture Series 185 'Machine Perception', Hampton (VA), München, Madrid, 1992.
- [DICKMANN, 1994] Dickmanns E. D.: *The 4D-Approach to Dynamic Machine Vision*, 33<sup>rd</sup> IEEE Conf. on Decision and Control, S. 3770-3775, Lake Buena Vista, FL, 1994.
- [DICKMANN, 1998] Dickmanns E. D.: *Dynamic Vision for Intelligent Vehicles*, Lecture Series at MIT/LIDS, Cambridge (MA), USA, Herbst 1998.
- [DICKMANN, 2001] Dickmanns E. D.: *Fahrzeuge lernen sehen: 25 Jahre Forschung und Lehre*, Universität der Bundeswehr, Neubiberg, Oktober 2001.
- [DICKMANN, 2002] Dickmanns E. D.: *The Development of Machine Vision for Road Vehicles in the Last Decade*, IEEE Intelligent Vehicles Symposium, Versailles, Frankreich, Juni 2002.
- [DICKMANN, 2002A] Dickmanns E. D.: *Intelligence Based on Dynamic Vision for Motion Control*, AI-Magazin, in Vorbereitung, 2002.

- [DICKMANN & FÜRST, 1998] Dickmanns E. D., Fürst S.: *Information Processing Architecture for Mission Performance of Autonomous Systems Capable of Dynamic Vision*, RTO Symp. on „The Application of Information Technologies (Computer Science) to Mission Systems“, Ref. 2, Monterey, California, USA, 1998.
- [DICKMANN & WÜNSCHE, 1999] Dickmanns E. D., Wünsche H. J.: *Dynamic Vision for Perception and Control of Motion*, Handbook on Computer Vision and Applications, Jaehne B., Haußenecker H., Geißler P. (Eds.), Vol. 3, S. 569-620, Academic Press, 1999.
- [DICKMANN D., 1997] Dickmanns D.: *Rahmensystem für visuelle Wahrnehmung veränderlicher Szenen durch Computer*, Dissertation, Universität der Bundeswehr München, 1997.
- [FERARIC, 1996] Feraric J. P.: *Echtzeitfähige Modellierung des individuellen Fahrerverhaltens zur Realisierung adaptiver Unterstützungsfunktionen in einem Monitor- und Warnsystem*, Dissertation, Universität der Bundeswehr München, 1996.
- [FEYRER & ZELL, 1999] Feyrer S., Zell A.: *Tracking and Pursuing Persons with a Mobile Robot*, Intern. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, Los Alamitos (CA), USA, 1999.
- [FISCHER, 1996] Fischer B.: *Reflexive und willkürliche Steuerung des optomotorischen Zyklus durch Fixation und Aufmerksamkeit*, in: Aktives Sehen in technischen und biologischen Systemen, Mertsching B. (Hrsg.), S. 75-82, Sankt Augustin (Infix), 1996.
- [FLEMISCH, 2001] Flemisch O. F.: *Pointillistische Analyse der visuellen und nicht-visuellen Interaktionsressourcen am Beispiel Pilot-Assistenzsystem*, Dissertation, Universität der Bundeswehr München, Shaker Verlag, Aachen, 2001.
- [GARCIA-ALEGRE & RECIO, 1998] Garcia-Alegre M. C., Recio F.: *Basic Visual and Motor Agents for Increasingly Complex Behavior Generation on a Mobile Robot*, in: Autonomous Robots, 5, S. 19-28, Kluwer Academic Publisher, Holland, 1998.
- [GODTHELP, 1984] Godthelp H.: *Studies on Human Vehicle Control*, Dissertation am TNO, Soesterberg, Niederlande, 1984.
- [GOMI & IDE, 1999] Gomi T., Ide K.: *The Development of an Autonomous Field Transport Vehicle With an Active Vision System*, Intern. Annual Conference – SICE '99, S. 1221-1226, Tokyo, Japan, 1999.
- [GRASHEY, 1999] Grashey S.: *Ein Klassifikationsansatz zur fertigkeitsbasierten Verhaltensmodellierung beim Autofahren*, Dissertation, Universität der Bundeswehr München, 1999.
- [GREGOR, 2002] Gregor R.: *Fähigkeiten zur Missionsdurchführung und Landmarkennavigation*, Dissertation, Universität der Bundeswehr München, 2002.
- [GREGOR et al., 2000] Gregor R., Lützel M., Pellkofer M., Siedersberger K.-H., Dickmanns E. D.: *EMS-VISION: A Perceptual System for Autonomous Vehicles*, IEEE Intelligent Vehicles Symposium, Detroit (MI), USA, Oktober 2000.
- [GRIBBLE, 1999] Gribble W. S.: *Field of View Control for Closed-loop Visually-guided Motion*, Intern. Conf. on Autonomous Agents, Seattle (WA), USA, S. 384-385, 1999.

- [GVOZDJAK & LI 1998] Gvozdzjak P., Li Z.-N.: *From Nomad to Explorer: Active Objekt Recognition on Mobile Robots*, in: *Pattern Recognition*, **31**(6): S. 773–790, 1998.
- [HAAG, 1998] Haag M.: *Bildfolgenauswertung zur Erkennung der Absichten von Straßenverkehrsteilnehmern*, Dissertation, Universität Karlsruhe, Sankt Augustin (Infix), 1998.
- [HAMDI, 1999] Hamdi M. S.: *Entwurf adaptiver lernender Roboter*, Dissertation, Universität Hamburg, 1999.
- [HAMKER & GROB, 1996] Hamker F. H., Groß H.-M.: *Intentionale Aufmerksamkeit: Ein alternatives Konzept für technische visuo-motorische Systeme*, in: *Aktives Sehen in technischen und biologischen Systemen*, Mertsching B. (Hrsg.), S. 101-108, Sankt Augustin (Infix), 1996.
- [HAMMOND, 1986] Hammond K. J.: *Case-based planning: an integrated theory of planning, learning and memory*, Dissertation, Yale-University, New Haven (CT), USA, 1986.
- [HAREL, 1987] Harel D.: *Statecharts: A Visual Formalism for Complex Systems*, in: *Science of Computer Programming*, Bd. **8**, S. 231-274, North-Holland, 1987.
- [HEIMES, 2000] Heimes T.: *Sichtsystemgestützte Fahrerassistenz im innerstädtischen Straßenverkehr*, Dissertation, Universität Karlsruhe, Akad. Verl.-Ges. Aka, 2000.
- [HERRMANN, 1997] Herrmann J.: *Maschinelles Lernen und Wissensbasierte Systeme*, Springer Verlag, Berlin/Heidelberg, 1997.
- [HOFMANN, 2003] Hoffmann U.: N.N., Dissertation, Universität der Bundeswehr München, (in Vorbereitung).
- [HORSWILL, 1997] Horswill I.: *Real-time Control of Attention and Behavior in a Logical Framework*, First Intern. Conf. on Autonomous Agents, S. 130-137, ACM, NY, USA, 1997.
- [HUBER & KORTENKAMP, 1998] Huber E., Kortenkamp D.: *A Behavior-based Approach to Active Stereo Vision for Mobile Robots*, in: *Engineering Applications of Artificial Intelligence*, Vol. **11**, S. 229-243, 1998.
- [HUMPHREYS et al., 1996] Humphreys G. W., Gilchrist I., Free L.: *Search and Selection in Human Vision: Psychological Evidence and Computational Implications*, in: *Visual Attention and Cognition*, Zangemeister W. H., Stiehl H. S., Freska C. (Eds.), S. 73-93, Elsevier, Amsterdam, 1996.
- [IRMSCHER, 2001] Irmischer M.: *Modellierung von Individualität und Motivation im Fahrerverhalten*, in: *Kraftfahrzeugführung*, Jürgensohn Th., Timpe K.-P. (Hrsg.), Springer Verlag, Berlin/Heidelberg, 2001.
- [JOCHEM et al., 1996] Jochem T. M., Pomerlau D. A., Thorpe C. E.: *Vision Based Intersection Navigation*, IEEE Intelligent Vehicles Symposium, S. 391-396, New York (NY), USA, 1996.
- [JÜRGENSOHN, 1997] Jürgensohn T.: *Hybride Fahrermodelle*, Dissertation, Technische Universität Berlin, ZMMS-Spektrum, Bd. **4**, Pro Universitate Verlag Sinzheim, 1997.

- [KELLY & STENTZ, 1998] Kelly A., Stentz A.: *Rough Terrain Autonomous Mobility – Part 2: An Active Vision, Predictive Control Approach*, in: *Autonomous Robots*, **5**, S. 163-198, Kluwer Academic Publishers, Holland, 1998.
- [KIDONO et al., 2000] Kidono K., Miura J., Shirai Y.: *Autonomous Visual Navigation of a Mobile Robot Using a Human-Guided Experience*, Symp. on Intelligent Autonomous Systems, S. 620-627, 2000.
- [KINZEL, 1994] Kinzel W.: *Präattentive und attentive Bildverarbeitungsschritte zur visuellen Erkennung von Fußgängern*, Dissertation, Universität der Bundeswehr München, 1994.
- [KOPF, 1993] Kopf M.: *Ein Beitrag zur modellbasierten, adaptiven Fahrerunterstützung für das Fahren auf deutschen Autobahnen*, Dissertation, VDI-Verlag, Reihe **12**, Nr. 203, Düsseldorf, 1993.
- [KRAISS, 1985] Kraiss K.-F.: *Fahrzeug- und Prozessführung: Kognitives Verhalten des Menschen und Entscheidungshilfen*, Springer Verlag, Berlin/Heidelberg, 1995.
- [KRÜGER, 1991] Krüger, W.: *Situationsmodellierung in der Bildfolgenauswertung*, Dissertation, Universität Karlsruhe, Springer-Verlag, Berlin, 1992.
- [KRUSE et al., 1993] Kruse R., Gebhardt J., Klawonn F.: *Fuzzy-Systeme*, Teubner Verlag, Stuttgart, 1993.
- [LIU, 1993] Liu Y. M. S.: *Eine auf Fuzzy basierende Methode zur mehrdimensionalen Beurteilung der Straßenverkehrssicherheit*, Dissertation, Universität Friedericiana zu Karlsruhe, 1993.
- [LORENZ, 1973] Lorenz K.: *Die Rückseite des Spiegels*, R. Piper & Co. Verlag, München, 1973.
- [LUTH & WENDT, 1998] Luth H., Wendt W.: *Taschenbuch der Regelungstechnik*, 2. Auflage, Verlag Harri Deutsch, Frankfurt am Main, 1998.
- [LÜTZELER, 2002] Lützel M.: *Fahrbahnerkennung zum Manövrieren auf Wegenetzen mit aktivem Sehen*, Dissertation, Universität der Bundeswehr München, 2002.
- [MARR, 1982] Marr D.: *Vision: A Computational Investigation into the Human Representation of Visual Information*, Freeman, New York, 1982.
- [MARTIN & MCCLURE, 1985] Martin J. A., McClure C.: *Diagramming Techniques for Analysis and Programmers*, Prentice-Hall, Englewood Cliffs (NJ), USA, 1985.
- [MAURER, 2000] Maurer M.: *Flexible Automation von Straßenfahrzeugen mit Rechnersehen*, Dissertation, Universität der Bundeswehr München, 2000.
- [MESTER et al., 1996] Mester R., Hötter M., Pöchmüller W.: *Umwelterfassung mit bewegten Kameras*, in: *Aktives Sehen in technischen und biologischen Systemen*, Mertsching B. (Hrsg.), S. 117-125, Sankt Augustin (Infix), 1996.
- [MIURA et al., 2000] Miura J., Kanda T., Shirai Y.: *An Active Vision System for Real-Time Sign Recognition*, IEEE Intelligent Transportation Systems, S. 52-57, Dearborn (MI), USA, 2000.
- [MOCK-HECKER, 1994] Mock-Hecker R.: *Wissensbasierte Erkennung kritischer Verkehrssituationen – Erkennung von Plankonflikten*, Dissertation, VDI-Verlag, Düsseldorf, 1994.

- [MURRAY et al., 1992] Murray D. W., Fenglei D., McLauchlan P. F., Reid I. D., Sharkey P. M., Brady M.: *Design of Stereo Heads*, in: Active Vision, A. Blake und A. Yuille (Eds.), MIT Press, London, 1992.
- [MURRAY et al., 1995] Murray D. W., Bradshaw K. J., McLauchlan P. F., Reid I. D., Sharkey P. M.: *Driving Saccade to Pursuit Using Image Motion*, Intern. Journal of Computer Vision, **16**, S. 205-228, Niederlande, 1995.
- [MÜSSELER & ASCHERSLEBEN, 96] Müsseler J., Ascherleben G.: *Zur Rolle visueller Aufmerksamkeitsverlagerung bei der Etablierung einer (subjektiv berichtbaren) Raumrepräsentation*, in: Aktives Sehen in technischen und biologischen Systemen, Mertsching B. (Hrsg.), S. 83-92, Sankt Augustin (Infix), 1996.
- [NORLUND & UHLIN, 95] Norlund P., Uhlin T.: *Closing the Loop: Detection and Pursuit of a Moving Object by a Moving Observer*, in: Computer Analysis of Images and Patterns, Crowley J. L., Christensen H. I. (Eds.), S. 400-407, Springer Verlag, 1995.
- [OESTEREICH, 97] Oestereich B.: *Objektorientierte Softwareentwicklung mit der Unified Modeling Language*, 3. Auflage, R. Oldenbourg Verlag München, 1997.
- [OKUBO et al., 1999] Okubo A., Nishikawa A., Miyazaki F.: *Selective Acquisition of 3D Structure with an Active Stereo Vision System*, in: Systems and Computers in Japan, Vol. **30**, No. **12**, S. 1-15, 1999.
- [PAHLAVAN, 1996] Pahlavan K.: *Designing an Anthropomorphic Head-Eye System*, in: Visual Attention and Cognition, Zangemeister W. H., Stiehl H. S., Freska C. (Eds.), S. 73-93, Elsevier, Amsterdam, 1996.
- [PAULI, 1993] Pauli J.: *Erklärungsbasiertes Computer-Sehen von Bildfolgen*, Dissertation, Technische Universität München, Sankt Augustin (Infix), 1993.
- [PELLKOFER & DICKMANN, 2000] Pellkofer M., Dickmanns E. D.: *EMS-VISION: Gaze Control in Autonomous Vehicles*, IEEE Intelligent Vehicles Symposium, Detroit (MI), USA, Oktober 2000.
- [PELLKOFER & DICKMANN, 2002] Pellkofer M., Dickmanns E. D.: *Behavior Decision in Autonomous Vehicles*, IEEE Intelligent Vehicles Symposium, Versailles, Frankreich, Juni 2002.
- [PELLKOFER et al., 2001] Pellkofer M., Lützel M., Dickmanns E. D.: *Interaction of Perception and Gaze Control in Autonomous Vehicles*, SPIE: Intelligent Robots and Computer Vision XX: Algorithms, Techniques and Active Vision, Newton (MA), USA, Oktober 2001.
- [PENG et al., 2000] Peng J., Srikaew A., Wilkes M. Kawamura K., Peters A.: *An Active Vision System for Mobile Robots*, IEEE Intern. Conf. on Systems, Man, and Cybernetics, Vol. **2**, S. 1472-1477, Piscataway (NJ), USA, 2000.
- [PRÉVÔT, 1995] Prévôt T.: *Maschinelles Flugplaner für Verkehrsflugzeuge als autonomes und kooperatives System*, Dissertation, Universität der Bundeswehr München, 1995.
- [PRINZ, 1983] Prinz W.: *Wahrnehmung und Tätigkeitssteuerung*, Springer Verlag, Berlin, 1983.

- [RASMUSSEN, 1983] Rasmussen J.: *Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models*, in: IEEE Transactions on Systems, Man, and Cybernetics, Bd. **SMC-13**, S. 257-266, 1983.
- [REIMER, 1991] Reimer U.: *Einführung in die Wissensrepräsentation*, Teubner Verlag, Stuttgart, 1991.
- [REKERSBRINK, 1994] Rekersbrink A.: *Verkehrsflusssimulation mit Hilfe der Fuzzy-Logic und einem Konzept potentieller Kollisionszeiten*, Dissertation, Universität Friedericiana zu Karlsruhe, 1994.
- [RETZ-SCHMIDT, 1991] Retz-Schmidt G.: *Die Interpretation des Verhaltens mehrerer Akteure in Szenenfolgen*, Dissertation, Universität Hamburg, Springer-Verlag, Berlin, 1992.
- [RICHTER, 1995] Richter S.: *Ein mehrfach adaptierendes, stabiles Modell für die Analyse von Straßenszenen*, Dissertation, Technische Universität München, Sankt Augustin (Infix), 1995.
- [RIEDER, 1998] Rieder A.: *ISF-Kameraplattformen zum aktiven Sehen*, Interner Bericht Nr. UniBwM/LRT/WE13/IB98-3C, Universität der Bundeswehr München, 1998.
- [RIEDER, 2000] Rieder A.: *Fahrzeuge sehen – Multisensorielle Fahrzeuergenerkennung in einem verteilten Rechnersystem für autonome Fahrzeuge*, Dissertation, Universität der Bundeswehr München, 2000.
- [RIMEY & BROWN, 1992] Rimey R., Brown Ch.: *Task-oriented Vision with Multiple Bayes Nets*, in: Active Vision, A. Blake und A. Yuille (Eds.), MIT Press, London, 1992.
- [ROCA et al., 2000] Roca X., Vitria J., Vanrell M., Villanueva J. J.: *Visual Behaviours for Binocular Navigation with Autonomous Systems*, in: Pattern Recognition and Applications, **56**, S. 134-143, Amsterdam, Holland, 2000.
- [RUCKDESCHEL, 1996] Ruckdeschel W.: *Modellierung regelbasierten Pilotenverhaltens mit Petrinetzen*, Dissertation, Universität der Bundeswehr München, 1996.
- [SAFFIOTTI & LEBLANC, 2000] Saffiotti A., LeBlanc K.: *Active Perceptual Anchoring of Robot Behavior in a Dynamic Environment*, IEEE Intern. Conf. on Robotics & Automation, San Francisco (CA), USA, April 2000.
- [SCHAAF, 1998] Schaaf J.: *Über die Suche nach situationsgerechten Fällen im fallbasierten Schließen*, Dissertation, Universität Kaiserslautern, Sankt Augustin (Infix), 1998.
- [SCHÄFER, 1996] Schäfer K. H.: *Unscharfe zeitlogische Modellierung von Situationen und Handlungen in Bildfolgenauswertung und Robotik*, Dissertation, Universität Karlsruhe, Sankt Augustin (Infix), 1996.
- [SCHIEHLEN, 1995] Schiehlen J.: *Kameraplattformen für aktiv sehende Fahrzeuge*, Dissertation, Universität der Bundeswehr München, 1995.
- [SCHMALZ, 2001] Schmalz S.: *Entwurf und Evaluierung von Strategien zur 2D/3D-Objekterkennung in aktiven Sehsystemen*, Dissertation, Universität Hamburg, 2001.

- [SCHNEIDER-FONTAN, 1999] Schneider-Fontan M.: *Plan Execution Based on Active Perception: Adding Hints to Plans*, in: *Autonomous Robots*, **6**, S. 53-68, Kluwer Academic Publisher, Holland, 1999.
- [SCHREINER, 1999] Schreiner F.: *Automatische Führung des Kraftfahrzeugs mit fahreradaptiven Eigenschaften*, Dissertation, Universität der Bundeswehr München, Forschungsberichte VDI Reihe 12, Nr. **404**, 1999.
- [SEELEN et al., 1996] Seelen C. v., Ulf M., Bajscy R.: *Ansätze zur Leistungsbewertung eines aktiven Sehsystems*, in: *Aktives Sehen in technischen und biologischen Systemen*, Mertsching B. (Hrsg.), S. 67-73, Sankt Augustin (Infix), 1996.
- [SEIFERT et al., 2001] Seifert K., Rötting M., Jung R.: *Registrierung von Blickbewegungen im Kraftfahrzeug*, in: *Kraftfahrzeugführung*, Jürgensohn Th., Timpe K.-P. (Hrsg.), Springer Verlag, Berlin/Heidelberg, 2001.
- [SIEDERSBERGER, 2003] Siedersberger K.-H.: *Komponenten zur automatischen Fahrzeugführung in sehenden (semi-)autonomen Fahrzeugen*, Dissertation, Universität der Bundeswehr München, in Vorbereitung.
- [SIEDERSBERGER et al., 2001] Siedersberger K.-H., Pellkofer M., Lützel M., Dickmanns E. D., Rieder A., Mandelbaum R., Bogoni L.: *Combining EMS-VISION and Horopter Stereo for Obstacle Avoidance of Autonomous Vehicles*, Intern. Workshop on Computer Vision (ICVS), Vancouver, Kanada, Juli 2001.
- [SOBOTTKA & WETZEL, 95] Sobottka K., Wetzel D.: *Attention Control Integrated in a System to Autonomous Driving and Collision Avoidance*, Intern. Conf. on Image Processing and its Applications, S. 1796-1800, London, UK, 1995.
- [STARK & CHOI, 1996] Stark L. W., Choi Y. S.: *Experimental Metaphysics: The Scanpath as an Epistemological Mechanism*, in: *Visual Attention and Cognition*, Stelmach G. E., Vroom P. A. (Eds.), Elsevier, Amsterdam, 1996.
- [STROHAL, 1999] Strohal M.: *Pilotenfehler- und Absichtserkennung als Baustein für ein Cockpitassistenzsystem mittels eines halbautomatischen Verfahrens zur Situationsklassifikation*, Dissertation, Universität der Bundeswehr München, 1999.
- [STÜTZ, 2000] Stütz P.: *Adaptive Modellierung des regelbasierten Pilotenverhaltens in Cockpitassistenzsystemen*, Dissertation, Universität der Bundeswehr München, 2000.
- [TRUONG, 2000] Truong H.: *Active Vision Head – A Novel Mechanism for Stereo Active Vision*, final-year honours thesis, Australian National University, 2000.
- [TSOTSOS, 1989] Tsotsos J. K.: *Motion Understanding Systems*, in: *Active Perception and Robot Vision*, Sood A. K., Wechsler H. (Eds.), Springer-Verlag, Heidelberg, 1989.
- [ULMER, 1994] Ulmer B.: *VITA II – Active Collision Avoidance in Real Traffic*, IEEE Intelligent Vehicles Symposium, Paris, Oktober 1994.
- [ULTES, 2002] Ultes H.: *Überblick über neue Erkenntnisse im Forschungsgebiet „Aktives Sehen“*, Studienarbeit, LRT/WE13/S/02-5, Universität der Bundeswehr München, 2002.
- [VERHALTEN, 2000] Siedersberger K.-H., Gregor R., Pellkofer M.: *Diskussionsrunden der Arbeitsgruppe „Verhalten“*, Universität der Bundeswehr, 2000; (vgl. auch [SIEDERSBERGER, 2003] und [GREGOR, 2002]).

- [VIÉVILLE, 1997] Viéville Th.: *A Few Steps Towards 3D Active Vision*, Springer-Verlag, Berlin, Heidelberg, 1997.
- [WASSON et al., 1999] Wasson G., Kortenkamp D., Huber E.: *Integrating Active Perception with an Autonomous Robot Architecture*, in: *Robotics and Autonomous Systems*, Vol. **29**, S. 175-186, Holland, 1999.
- [WEI & LI, 1998] Wie J., Li Z.: *Efficient Disparity-based Gaze Control with Foveate Wavelet Transform*, IEEE Intern. Conf. on Intelligent Robots and Systems, S. 866-871, Victoria (B.C.), Kanada, Oktober 1998.
- [WETZEL, 1995] Wetzel D.: *Wissensbasierte Verkehrsszenenanalyse zur Fahrerunterstützung*, Dissertation, Universität Erlangen-Nürnberg, 1995.
- [WITTIG & SCHULTE, 93] Wittig T., Schulte A.: *Visuelle Situationserfassung von Piloten*, Bericht zu einem Studienauftrag, UniBwM/LRT/WE13/IB/93-6, Universität der Bundeswehr München, September 1993.
- [ZADEH, 1965] Zadeh L.A.: *Fuzzy Sets*, *Information and Control*, Bd. **8**, S. 338-353, 1965.
- [ZISMAN, 1978] Zisman M. D.: *Use of Production Systems for Modeling Asynchronous, Concurrent Processes*, Academic Press, 1978.
- [ZOBEL et al., 2001] Zobel M., Denzler J., Heigl B., Nöth E., Paulus D., Schmidt J., Stemmer G.: *MOBSY: Integration of Vision and Dialogue in Service Robots*, in: *Computer Vision Systems - ICVS 2001*, Schiele B., Sagerer G. (Eds.), S. 50-62, Springer-Verlag, Berlin, Heidelberg, 2001.



# Anhang A

## Basisklassen der Situationsanalyse

### A.1 Die Klasse CSituationAspect

Jeder Situationsaspekt wird von einer Klasse realisiert, welche die Basisklasse CSituationAspect besitzt (siehe Abb. A.1, oben links). In der Klassenvariablen `m_eID` ist die eindeutige ID des Situationsaspekts niedergelegt. Bei der Berechnung der Situationsaspekte kann mit der Variablen `m_iOrder` eine gewisse Reihenfolge vorgegeben werden. Dies ist insbesondere dann notwendig, wenn ein Situationsaspekt die Ergebnisse eines anderen Situationsaspekts weiterverarbeitet. Die Berechnung eines Situationsaspekts mit `m_iOrder = i` erfolgt erst nach der Berechnung aller Situationsaspekte mit `m_iOrder = 0 ... (i-1)`. Die Größen `m_tValencyInterval` und `m_bValencyLogic` bezeichnen das Gültigkeitsintervall bzw. die Berechnungslogik des Situationsaspekts (vgl. Abschnitt 3.2.2). Mit der Klassenvariablen `m_pScene` besitzt der Situationsaspekt eine Referenz auf den Prozessrahmen mit seiner Wissensrepräsentation. In der Liste `m_HandledClasses` sind alle Klassen von Objekten aufgeführt, welchen der Situationsaspekt eine Bedeutung zuordnen kann (= Klassen der möglichen Zuordnungsobjekte) und die Liste `m_ReferenceClasses` beinhaltet alle Objektklassen, die der Situationsaspekt als Referenzobjekte akzeptiert. In der Tabelle `m_MembershipFunctionMap` sind die Zugehörigkeitsfunktionen der einzelnen linguistischen Werte des Situationsaspekts eingetragen (vgl. Abschnitt A.2).

### A.2 Die Klasse CMembershipMatrixEntry

Auf welche Objekte und mit welchen Referenzobjekten der Situationsaspekt angewandt wird, legen die Einträge in der Matrix `m_MembershipMatrix` der Klasse CSituationAspect fest. Ein Eintrag in dieser Matrix ist eine Instanz der Klasse CMembershipMatrixEntry oder einer davon abgeleiteten Klasse (vgl. Abb. A.1, oben rechts). Die Variablen `m_pAssignObject` und `m_apReferenceObject` eines Eintrags spezifizieren das Zuordnungsobjekt und die Referenzobjekte bezüglich derer die Aussage des Eintrags gilt. Die Tabelle `m_ValencyMap` beinhaltet die verschiedenen linguistischen Aussagen mit ihren Zugehörigkeitsgraden. Durch das Verwalten der berechneten Zugehörigkeitsgrade eines Situationsaspekts mithilfe einer Matrix, kann auf die symbolischen Aussage bezüglich eines Objekts durch den Schlüssel

[ID des Zuordnungsobjekts, ID des 1. Referenzobjekts]

effizient zugegriffen werden.

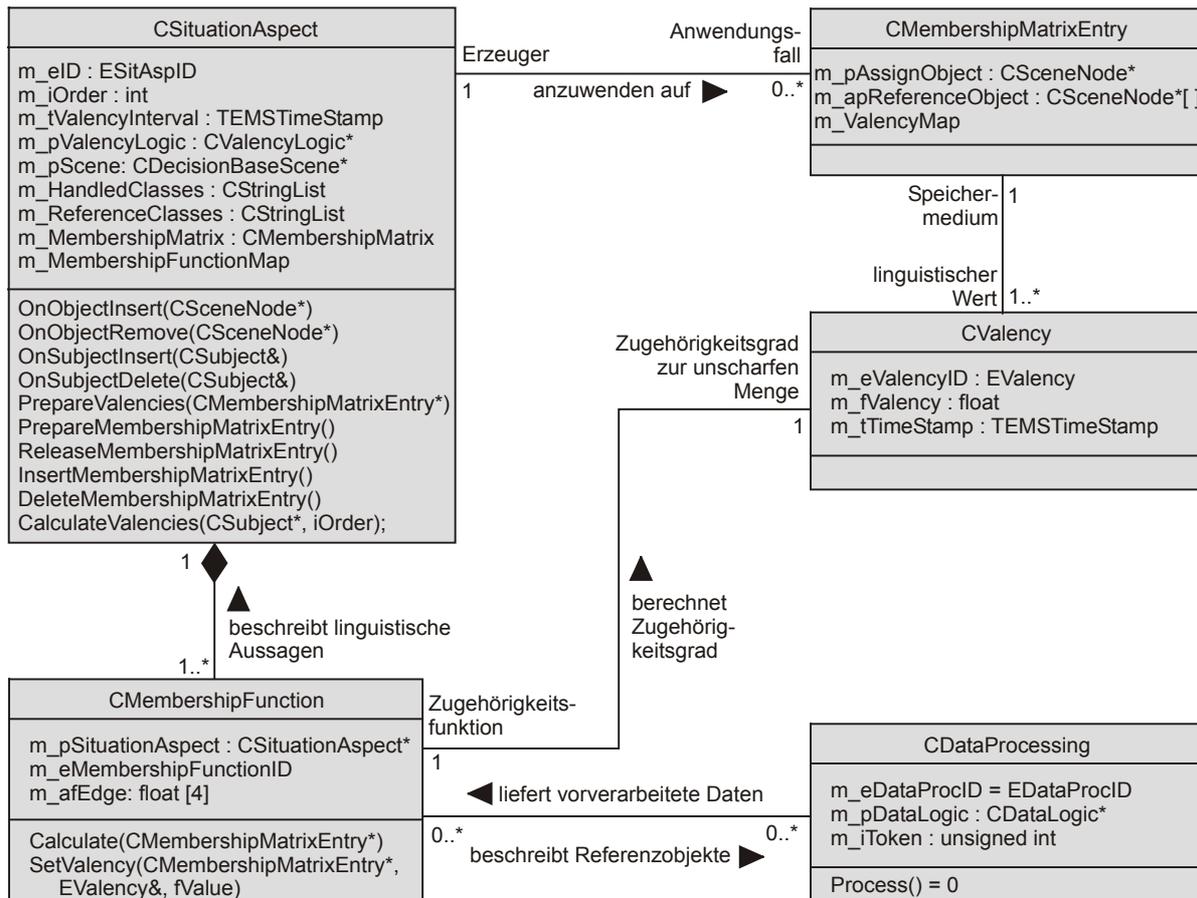


Abb. A.1: Die wichtigsten Klassen bei der Situationsanalyse (Klassendiagramm nach UML)

### A.3 Die Klasse CValency

Die Klasse CValency ist die Basisklasse aller linguistischen Werte (vgl. A.1, rechts Mitte). Auch sie besitzt mit der Variablen `m_eValencyID` eine eindeutige ID. Die Variable `m_fValency` gibt den Zugehörigkeitsgrad zur unscharfen Menge an und die Variable `m_tTimeStamp` dessen Zeitstempel. Der Zeitstempel dient u. a. zur Überprüfung, ob das Gültigkeitsintervall `CSituationAspect::m_pValencyInterval` abgelaufen ist und die linguistischen Werte damit neu berechnet werden müssen. Nach Abschnitt 3.2.2 existieren Situationsaspekte, für deren Beschreibung scharfe Werte sinnvoller sind. Bei diesen Situationsaspekten enthält die Variable `m_fValency` ein scharfes Ergebnis aus einem kontinuierlichen Wertebereich. Die entsprechende Berechnungsfunktion ist in diesem Fall keine Zugehörigkeitsfunktion im strengen Sinn, sondern dient der Ermittlung eines scharfen Ergebnisses. Auf diese Weise können linguistische Aussagen und scharfe Ergebniswerte als situationsbeschreibende Aussagen gemeinsam verwaltet, verknüpft und verarbeitet werden, ohne dass der Prozessrahmen sie unterscheiden muss.

### A.4 Die Klasse CMembershipFunction

Für jeden linguistischen Wert eines Situationsaspekts existiert eine Zugehörigkeitsfunktion, die diesem eine unscharfe Menge zuordnet. Die Klasse CMembershipFunction ist die Basisklasse aller Zugehörigkeitsfunktionen. Die Variable `m_pSituationAspect` beinhaltet einen Zeiger auf den zugehörigen Situationsaspekt und `m_eMembershipFunctionID` eine ID, mit der auf die Funktion in der Tabelle `CSituationAspect::m_MembershipFunctionMap`

effizient zugegriffen werden kann. Die Zugehörigkeitsfunktion beschreibt die Aussage eines linguistischen Wertes durch eine Trapezfunktion, welche durch vier Eckpunkte festgelegt ist (vgl. Abb. 3.3 in Abschnitt 3.2.2). Durch das teilweise Zusammenfassen der Eckpunkte lassen sich auch die anderen in Abbildung 3.3 gezeigten Zugehörigkeitsfunktionen modellieren. Die virtuelle Funktion Calculate() beinhaltet die Rechenvorschrift für die Ermittlung des Zugehörigkeitsgrades zur unscharfen Menge. Welche Objekte dabei zu verwenden sind, ist im übergebenen Matrixelement vom Typ CMembershipMatrixEntry niedergelegt. Das Ergebnis dieser Berechnung wird durch den Aufruf von SetValency() im übergebenen Matrixelement eingetragen.

## A.5 Die Klasse CDataProcessing

Die Zugehörigkeitsfunktion erhält ihre Daten von einer oder mehreren Datenvorverarbeitungen. Jede Datenvorverarbeitung ist von der Klasse CDataProcessing abgeleitet. Die Variable m\_eDataProcID dieser Klasse enthält eine ID, mit der auf die Datenvorverarbeitung effizient zugegriffen werden kann. Die Variable m\_pDataLogic zeigt auf die zugehörige Datenlogik der Datenvorverarbeitung (vgl. Abschnitt 3.2.2). Die Variable m\_iToken zeigt an, von wie vielen Situationsaspekten die Datenverarbeitung aktuell benutzt wird. Fällt m\_iToken auf Null, so löscht der Prozessrahmen die Instanz der Datenvorverarbeitung. Somit ist sichergestellt, dass nur benutzte Daten auch vorverarbeitet werden. In der zyklisch aufgerufenen (virtuellen) Klassenfunktion Process() findet die eigentliche Datenvorverarbeitung statt.

## A.6 Die Klassen CSubject und CEgo

Bei den Objekten im Szenenbaum kann es sich um Subjekte handeln, welche ihr Verhalten eigenmächtig und nach modellierbaren Regeln ändern. In der hier vorgestellten Prozessstruktur wird ein Subjekt – zusätzlich zu seinem Szenenknoten im Szenenbaum – durch eine Instanz der Klasse CSubject repräsentiert (vgl. Abb. A.2). Die Klassenvariable m\_eSubjectType spezifiziert den Typ des Subjekts, wie z. B. PKW, LKW oder Fußgänger. Die Variable m\_pSceneNode zeigt auf dessen Szenenknoten im Szenenbaum und in m\_Actions können die unterstellten oder wahrgenommenen Aktionen des Subjekts eingetragen werden. Ein besonderes Subjekt stellt das autonome Eigenfahrzeug dar, welches durch eine Instanz der Klasse CEgo verwaltet wird. Die Klasse CEgo ist von CSubject abgeleitet. Beim Eigenfahrzeug stehen Fahrzeugdaten zur Verfügung, welche bei Fremdfahrzeugen i. d. R. nicht bekannt sind, wie z. B. Lenkwinkel, Lenkrate, Odometrie und inertielle Zustandsgrößen. Ferner verfügt das Eigenfahrzeug über einen Missionsplan (m\_pMisPlan), den es abzuarbeiten hat.

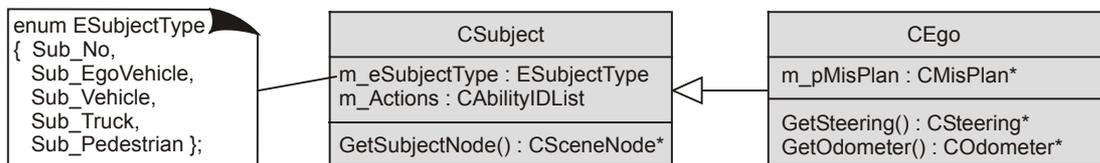


Abb. A.2: Die interne Repräsentation der Subjekte (Klassendiagramm nach UML)

## A.7 Die Behandlung von neuen und verschwindenden Szenenobjekten

Abb. A.3 zeigt den Kontrollfluss, der beim Auftreten oder Verschwinden von Szenenobjekten durchlaufen wird. Hängt z. B. ein Wahrnehmungsexperte einen neuen Szeneknoten in den Szenenbaum ein, so erfolgt zunächst seine Behandlung als Szenenobjekt: Für alle Situationsaspekte wird die Ereignisbehandlungsfunktion `OnObjectInsert()` aufgerufen und ein Zeiger auf das neue Objekt übergeben. Ob und wie der jeweilige Situationsaspekt das neue Objekt berücksichtigt, ist spezifisch für den Situationsaspekt und in der überladenen Funktion `OnObjectInsert()` niedergelegt. Nach der Behandlung als Objekt erfolgt mit der Funktion `CCentralDecisionScene::isSubject(CSceneNode*)` die Überprüfung, ob es sich bei dem Objekt um ein Subjekt handelt. Ist dies der Fall, so wird für dieses Subjekt eine Instanz der Klasse `CSubject` angelegt und im Prozessrahmen eingetragen.

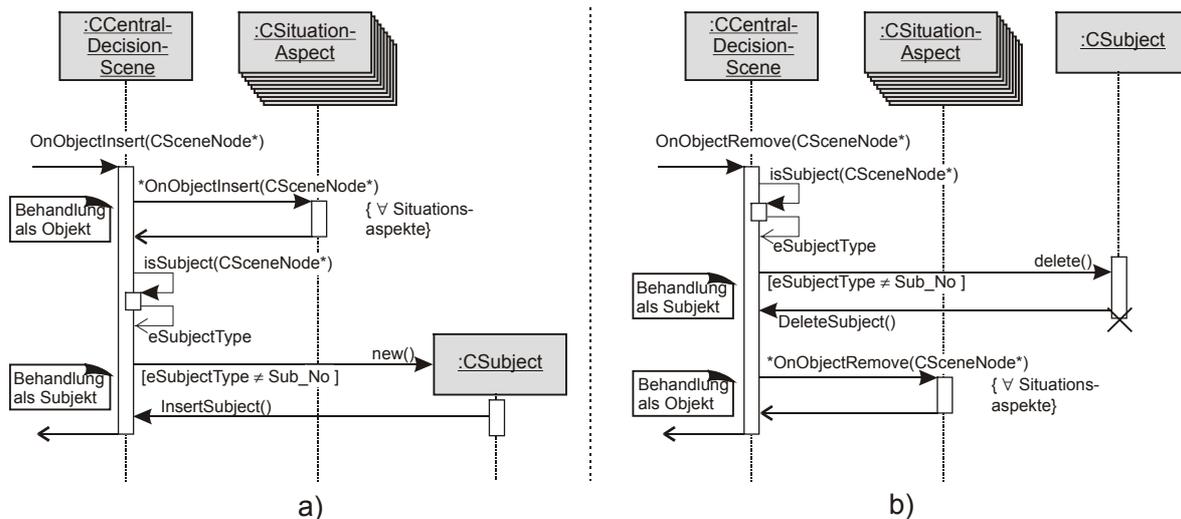


Abb. A.3: Die Behandlung neuer Objekte und Subjekte (a) und verschwindender Objekte und Subjekte (b) (Sequenzdiagramme nach UML)

Beim Aushängen des Objekts durchläuft man den umgekehrten Kontrollfluss (vgl. Abb. A.3 b): Zunächst wird das verschwindende Objekt als Subjekt behandelt und die zugehörige Instanz der Klasse `CSubject` gelöscht. Danach informiert man durch den Aufruf der Funktion `CSituationAspect:OnObjectRemove()` jeden Situationsaspekt über das Verschwinden des Szenenobjekts. Sind z. B. in einem Situationsaspekt für dieses Objekt linguistische Aussagen berechnet oder als zu berechnen markiert, so löscht dieser Aufruf die entsprechenden Einträge in der Matrix `CSituationAspect::m_MembershipMatrix`.

## Anhang B

### Basisklassen des Fähigkeitenkonzepts

Abb. B.1 beinhaltet das Klassendiagramm des Fähigkeitenkonzepts. Die Abbildung zeigt nur die wesentlichen Variablen und Funktionen der Klassen, insoweit sie für das Verständnis der Implementierung wichtig sind.

#### B.1 Die Basisklasse CNetNode

Das Fähigkeitsnetz nutzt die Kommunikationsstrukturen der von RIEDER [2000] implementierten dynamischen Wissensrepräsentation, in welcher Daten objektbezogen kommuniziert und verwaltet werden. So sind Daten immer einem Objekt einer Klasse zugeordnet, welche von der Basisklasse CNode abgeleitet ist. Die von CNode abgeleitete Klasse CNetNode beinhaltet die Möglichkeit, Verknüpfungen zu anderen Objekten dieser Klasse zu repräsentieren. Damit kann man einen netzartigen, gerichteten Graphen aufbauen. Die Klassenvariable `m_iNoOfDependencies` beziffert die Anzahl der gerichteten Verknüpfungen zu anderen Knoten und das Array `m_aDependencies` beinhaltet Zeiger auf diese Knoten. Die verknüpften Knoten sind ebenfalls Objekte der Klasse CNetNode oder einer davon abgeleiteten Klasse. Mit der Klassenfunktion `GetDependency(i)` erhält man einen Zeiger auf den *i*-ten verknüpften Knoten. Die Funktion `IsComplete()` gibt `true` zurück, falls alle verknüpften Knoten auch wirklich im Netz vorhanden und alle Zeiger des Arrays `m_aDependencies` gültig sind. Sie gibt `false` zurück, falls mindestens eine Verknüpfung nicht erstellt werden konnte.

#### B.2 Die Basisklasse CAbilityNode für alle Fähigkeitenknoten

Die Klasse CAbilityNode ist die Basisklasse aller Fähigkeitenknoten. Sie ist von der Klasse CNetNode abgeleitet und erbt damit alle Funktionalitäten, die für den Aufbau eines Fähigkeitsnetzes notwendig sind. Zusätzlich besitzt sie Klassenvariablen, welche die Repräsentation einer Fähigkeit ermöglichen. So beinhaltet die Klassenvariable `m_sExpertName` den Namen des Prozesses, der sich als Experte dieser Fähigkeit eingetragen hat. In der Variablen `m_sComputer` ist niedergelegt, wo sich dieser Experte im verteilten System befindet. Die Klassenvariable `m_eAvailability` zeigt an, wie viele Male die Fähigkeit aktuell verfügbar ist (dynamische Verfügbarkeit). Diese Verfügbarkeit wird periodisch vom Experten im Fähigkeitenknoten gesetzt. Die Klassenvariable `m_tTimeRange` spezifiziert die Zeitspanne, nach welcher spätestens die Verfügbarkeit erneut eingetragen werden muss. Nach Ablauf dieser Zeitspanne ist die Verfügbarkeit ungültig. Damit wird sichergestellt, dass z. B. bei einem Programmabsturz des Experten die zuvor gesetzte, inzwischen aber falsche Verfügbarkeit nicht zeitlich unbegrenzt im Fähigkeitenknoten eingetragen bleibt und dass der Benutzer der Fähigkeit das Fehlen des Experten realisiert. Für den Versand von Parametern vom Benutzer zum Experten der Fähigkeit werden die Klassenvariablen `m_pParameter` und `m_iSizeOfParameter` benötigt. Die Klasse CAbilityNode beinhaltet in ihrer Klassenvariable `m_Instances` auch Informationen über die aktuell ablaufenden Aktionen der Fähigkeit: die Netz-IDs, die Instanz-

IDs, den Status und die relativen und absoluten Fortschrittsmaße der Aktionen (vgl. Abschnitte 3.3.4 und 3.3.7).

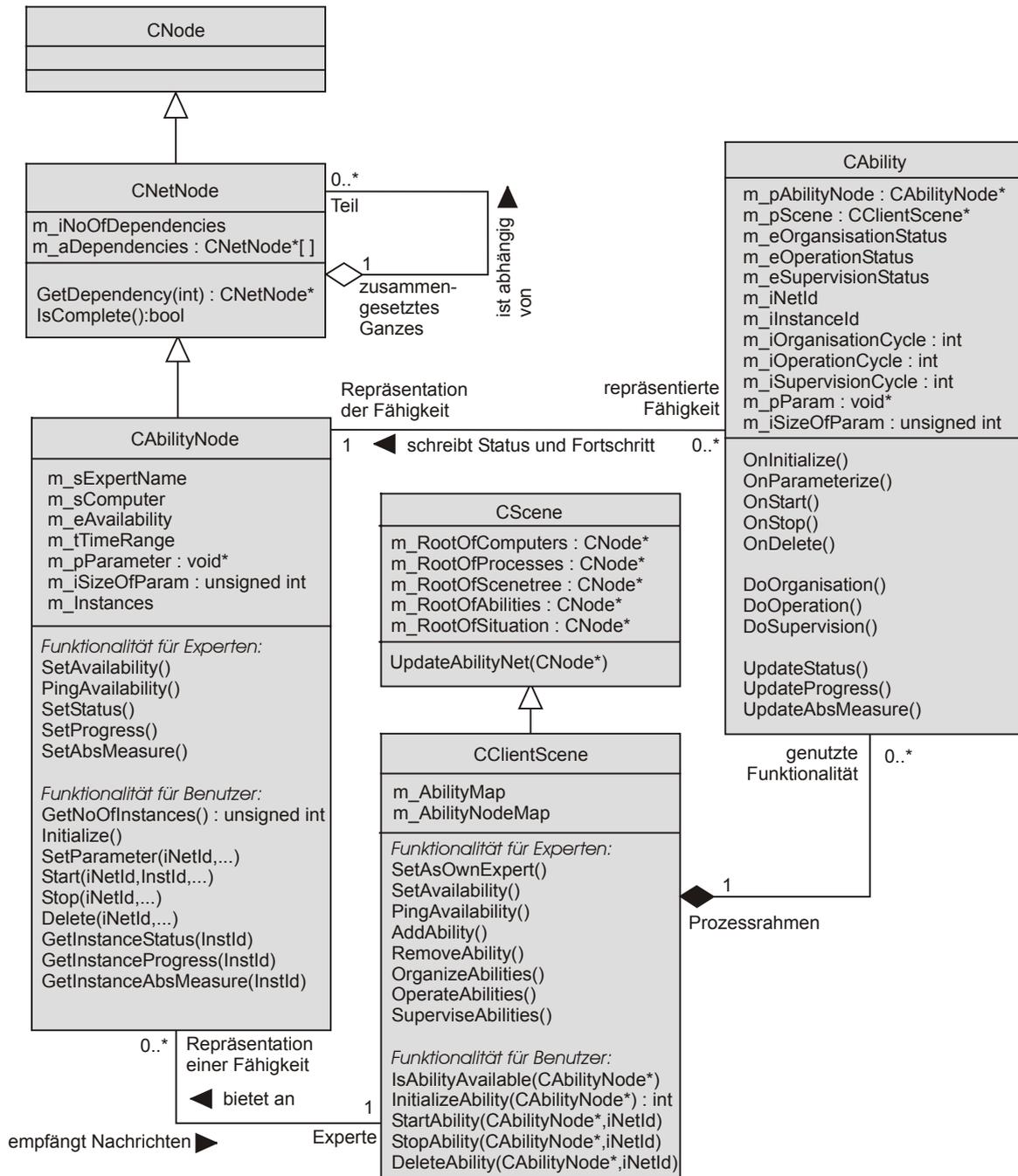


Abb. B.1: Klassendiagramm des Fähigkeitskonzepts (Darstellung nach UML)

Die Klasse **CAbilityNode** bietet Funktionalität sowohl für den Experten als auch für den Benutzer der Fähigkeit an. Mit der Klassenfunktion `SetAvailability()` setzt der Experte die dynamische Verfügbarkeit seiner Fähigkeit. Mit `PingAvailability()` erneuert er den Zeitstempel der Verfügbarkeit, so dass sie nicht ungültig wird. Ferner trägt der Experte den Status und die Fortschrittsmaße der bei ihm ablaufenden Aktionen mit den Funktionen `SetStatus()`, `SetProgress()` und `SetAbsMeasure()` im Fähigkeitsknoten ein.

Dem Benutzer einer Fähigkeit stehen durch die Klasse `CAbilityNode` folgende Funktionen zur Verfügung: Die Funktion `GetNoOfInstances()` gibt die Anzahl der aktuell vorhandenen Instanzen zurück. Mit `Initialize()` bewirkt der Benutzer, dass der Experte der Fähigkeit eine neue Instanz anlegt. Mit `SetParameter()` kann er der neuen Instanz Parameter mitteilen. Mit den Klassenfunktionen `Start()`, `Stop()` und `Delete()` startet, stoppt bzw. löscht der Benutzer die Instanz. Diese Befehle werden durch den Fähigkeitenknoten automatisch an den Experten der Fähigkeit gesandt. Zusätzlich besitzt die Klasse `CAbilityNode` Zugriffsfunktionen, um den Status und die Fortschrittsmaße einer Instanz abzufragen. Da von einer Fähigkeit gleichzeitig mehrere Instanzen vorhanden sein können, muss man bei allen Funktionen die eine Instanz betreffen, die Netz-ID und/oder Instanz-ID angeben (vgl. Abschnitt 3.3.7).

### B.3 Der Prozessrahmen durch die Klasse `CClientScene`

Jeder Prozess im System kann sich durch die Benutzung der Klasse `CClientScene` bei der dynamischen Wissensrepräsentation anmelden. Nach der Anmeldung bietet die Klasse `CClientScene` zusammen mit ihrer Basisklasse `CScene` einen Prozessrahmen, durch den ein skalierbarer Teil des im System vorhandenen Wissens in den Arbeitsbereich des Prozesses gespiegelt wird.

So repräsentieren die am Knoten `m_RootOfComputers` hängenden Knoten - auch Söhne des Knotens genannt - die im System vorhandenen Recheneinheiten. Die Söhne des Knotens `m_RootOfProcesses` sind Objekte der Klasse `CProcessNode`. Mit Hilfe dieser Prozessknoten ist eine Punkt-zu-Punkt-Kommunikation zwischen Prozessen möglich. Der Prozessknoten des Experten einer Fähigkeit wird dazu benutzt, die Steuerbefehle zu diesem zu senden. Am Wurzelknoten `m_RootOfScenetree` hängt die Szenenrepräsentation – auch Szenenbaum genannt – wie sie von RIEDER [2000] realisiert wurde. Über die Söhne des Knotens `m_RootOfSituation` können situationsbeschreibende, symbolische Aussagen kommuniziert werden. Unter dem Knoten `m_RootOfAbilities` befindet sich das Fähigkeitsnetz. Beim Einhängen eines neuen Fähigkeitsknotens ins Fähigkeitsnetz erfolgt ein Aufruf der Klassenfunktion `UpdateAbilityNet()`. Diese Funktion überprüft und aktualisiert die Zeiger zwischen den Fähigkeitsknoten.

Ein Prozess mit dem durch die Klasse `CClientScene` angebotenen Prozessrahmen ist sowohl der Experte seiner eigenen Fähigkeiten, als auch ein möglicher Benutzer einer (fremden) Fähigkeit. So bietet die Klasse `CClientScene` sowohl Experten- als auch Benutzerfunktionalität an. Zunächst wird die Expertenfunktionalität beschrieben. In der Tabelle `m_AbilityMap` befinden sich Referenzen auf alle aktuell im Prozess ablaufenden Instanzen von Fähigkeiten (vgl. unten). In der Tabelle `m_AbilityNodeMap` sind Zeiger auf alle Fähigkeitenknoten eingetragen, von denen der Prozess der Experte ist. Ruft der Prozess die Klassenfunktion `SetAsOwnExpert()` auf, so wird im übergebenen Fähigkeitenknoten der Expertenname gesetzt und der Knoten in die Tabelle `m_AbilityNodeMap` eingetragen. Durch die Funktion `SetAvailability()` setzt der Experte die dynamische Verfügbarkeit seiner Fähigkeit. Mit `PingAvailability()` wird die Verfügbarkeit aller in der Tabelle `m_AbilityNodeMap` befindlichen Fähigkeitenknoten erneuert. Bekommt der Experte eine Nachricht, dass er eine Instanz einer Fähigkeit anlegen soll, so trägt er nach Instanziierung dieser Fähigkeit mit `AddAbility()` einen Zeiger auf die neue Instanz in die Tabelle `m_AbilityMap` ein. Mit der Funktion `RemoveAbility()` löscht der Experte den Eintrag wieder.

In der Echtzeitphase erfolgt der zyklische Aufruf von `OrganizeAbilities()`. Diese Funktion prüft zunächst einzeln für alle in der Tabelle `m_AbilityMap` eingetragenen Fähigkeiteninstanzen, ob deren organisatorische Teile im aktuellen Zyklus aktiv werden müssen. Ist dies der

Fall, so führt dies zum Aufruf der virtuellen, überladenen Funktion `DoOrganisation()` der Fähigkeitenklasse `CAbility`. Für die Klassenfunktionen `OperateAbilities()` und `SuperviseAbilities()` gelten die analogen Aussagen.

Die Klasse `CClientScene` besitzt folgende Benutzer-Funktionalität: Mit `IsAbilityAvailable()` kann der Benutzer die dynamische Verfügbarkeit einer Fähigkeit abfragen. Die Funktion meldet die Fähigkeit als verfügbar, wenn alle Verknüpfungen zu den Teilfähigkeiten vorhanden sind und die Teilfähigkeiten ebenfalls verfügbar sind. Mit der Funktion `InitializeAbility()` kann man eine Instanz einer Fähigkeit beim zugehörigen Experten erzeugen. Ist die Fähigkeit von Teilfähigkeiten abhängig, so wird dieses Kommando automatisch auch zu den Experten der Teilfähigkeiten gesandt und es entsteht ein Netz von Instanzen. Alle Instanzen, welche sich aus dem Aufruf von `InitializeAbility()` ergeben, besitzen die gleiche Netz-ID (vgl. Abschnitt 3.3.7). Die Funktion `InitializeAbility()` gibt diese Netz-ID zurück. Will der Benutzer mit den Klassenvariablen `StartAbility()`, `StopAbility()` und `DeleteAbility()` weitere Befehle an das Netz von Instanzen senden, so muss er die zugehörige Netz-ID mit übergeben. Die Steuerbefehle werden dann parallel zu allen am Instanzen-Netz beteiligten Experten gesandt (vgl. Abschnitt 3.3.7).

#### **B.4 Die Basisklasse `CAbility` für alle Fähigkeitenklassen**

Ein Prozess kann verschiedene Funktionalitäten anbieten. Um sie von außen nutzen zu können, müssen sie in Fähigkeitenklassen gekapselt werden. Die Basisklasse für alle Fähigkeitenklassen heißt `CAbility`. Sie verfügt mit der Variablen `m_pAbilityNode` über einen Zeiger auf den Fähigkeitenknoten, durch den sie im Fähigkeitsnetz repräsentiert ist und durch den sie ihren Status und ihre Fortschrittsmaße nach außen kommuniziert. Ferner besitzt die Klasse `CAbility` einen Zeiger auf den Prozessrahmen (`m_pScene`). Die Variable `m_iOrganisationCycle` gibt an, in welchem Takt der organisatorische Teil der Fähigkeit auszuführen ist. Ist z. B. ihr Wert 10, so wird der organisatorische Teil jeden zehnten Zyklus (alle 400 ms) ausgeführt. Für die Variablen `m_iOperationCycle` und `m_iSupervisionCycle` gelten die analogen Aussagen. Empfängt der Experte Parameter für seine Fähigkeiteninstanz, so kopiert sein Prozessrahmen (`CClientScene`) die Parameter automatisch in den durch die Variablen `m_pParameter` und `m_iSizeOfParam` spezifizierten Speicherbereich der Aktion.

Die Klassenfunktionen `OnInitialize()`, `OnParameter()`, `OnStart()`, `OnStop()` und `OnDelete()` werden durchlaufen, wenn das entsprechende Ereignis eintritt. So erfolgt z. B. ein Aufruf von `OnStart()`, wenn der Benutzer `CClientScene::StartAbility()` für die Instanz der Fähigkeit aufruft. Die Funktionen `DoOrganisation()`, `DoOperation()` und `DoSupervision()` werden durchlaufen, wenn im aktuellen Zyklus der entsprechende Teil der Fähigkeit auszuführen ist. Die von `CAbility` abgeleiteten Fähigkeitenklassen müssen diese Funktionen überladen, um darin die fähigkeitenspezifischen Funktionalitäten aufzunehmen. Mit der Funktion `UpdateStatus()`, `UpdateProgress()` und `UpdateAbsMeasure()` erfolgt ein Übertrag der entsprechenden Information von der Fähigkeiteninstanz in den zugehörigen Fähigkeitenknoten.

# Anhang C

## Basisklassen der Regelbasis

Abb. C.1 zeigt die wichtigsten Basisklassen der Regelbasis. Wie schon in den vorangegangenen Klassendiagrammen, so sind auch hier nur die wesentlichen Klassenfunktionen und Klassenvariablen dargestellt.

### C.1 Die Klasse CFuzzyRule

Jede Regel der Regelbasis ist von der Klasse CFuzzyRule abgeleitet (vgl. Abb. C.1, links). Die Klasse verfügt mit der Klassenvariablen `m_apTerm` über ein Array von (un)scharfen Termen, welche in der Prämisse der Regel durch Operatoren verknüpft sind. Die Variable `m_iNoOfTerms` zeigt die Anzahl der verknüpften Terme an. Die Klassenvariable `m_Logic` spezifiziert, in welcher Reihenfolge und mit welchen Operatoren die Terme verknüpft sind. Die Tabelle `m_ModelMap` spezifiziert alle Fähigkeiten, die im Fall der Gültigkeit der Regel ausgeführt bzw. als Alternativen vorgehalten werden sollen. Die Fähigkeiten sind nach ihrer Priorität geordnet, und die Tabelle erlaubt über die Priorität einen direkten Zugriff auf die Fähigkeiten. Es sind aber nicht die Fähigkeiten selbst in der Tabelle referenziert, sondern ihre internen Ablaufmodelle (vgl. Kapitel 3.3.6).

Die Berechnung der Gültigkeit der Regel geschieht in der Klassenfunktion `Compare()`. Das Klassendesign für die Regelbasis ist so gestaltet, dass man die Regeln parallel auf alle Subjekte anwenden kann. Dementsprechend ist beim Aufruf von `Compare()` die ID des Subjekts zu übergeben, bezüglich dem die Aussage gelten soll. Das Ergebnis der Berechnung wird in der Tabelle `m_MembershipMap` gespeichert. Anschließend kann man in der Tabelle über die ID des Subjekts die Gültigkeit der Regel abfragen. Die Klassenfunktionen `SpecifySituationAspects()`, `OnBegin()`, `OnCycle()` und `OnEnd()` beschreibt Abschnitt 3.5.2.

Die Auswertung der Konklusionen der Regeln erfolgt ausgehend von der Regel mit dem größten Erfülltheitsgrad in Richtung absteigendem Erfülltheitsgrad. Die Fähigkeiten, welche durch Regeln mit geringem Erfülltheitsgrad vorgeschlagen werden, gelangen nur zur Ausführung, wenn sie nicht einer Konklusion einer Regel mit größerem Erfülltheitsgrad widersprechen. Anders ausgedrückt: Widersprechen sich die von zwei Regeln ausgewählten Fähigkeiten, so gewinnt immer die Regel mit dem größeren Erfülltheitsgrad. So hat z. B. eine Regel mit einem Spurwechsel in ihrer Konklusion keinen Einfluss auf die Fähigkeitauswahl, wenn es eine Regel mit größerem Erfülltheitsgrad gibt, die ein Spurfolgen verlangt (vgl. Abschnitt 3.5.3).

Die Parameter der Fähigkeiten sind in den Konklusionen der Regeln ebenfalls spezifiziert. Sind zwei Regeln gültig und schlagen sie die gleiche Fähigkeit aber mit unterschiedlicher Parametrisierung vor, so wird durch ein Defuzzifizierungsverfahren der letztendlich verwendete Parameterwert ermittelt. Ein Defuzzifizierungsverfahren erzeugt aus einer unscharfen

Menge einen scharfen Wert. Es gibt verschiedene Arten von Defuzzifizierungsmethoden. In regelungstechnischen Anwendungen findet man am häufigsten Schwerpunktverfahren (vgl. [LUTH & WENDT, 1998]).

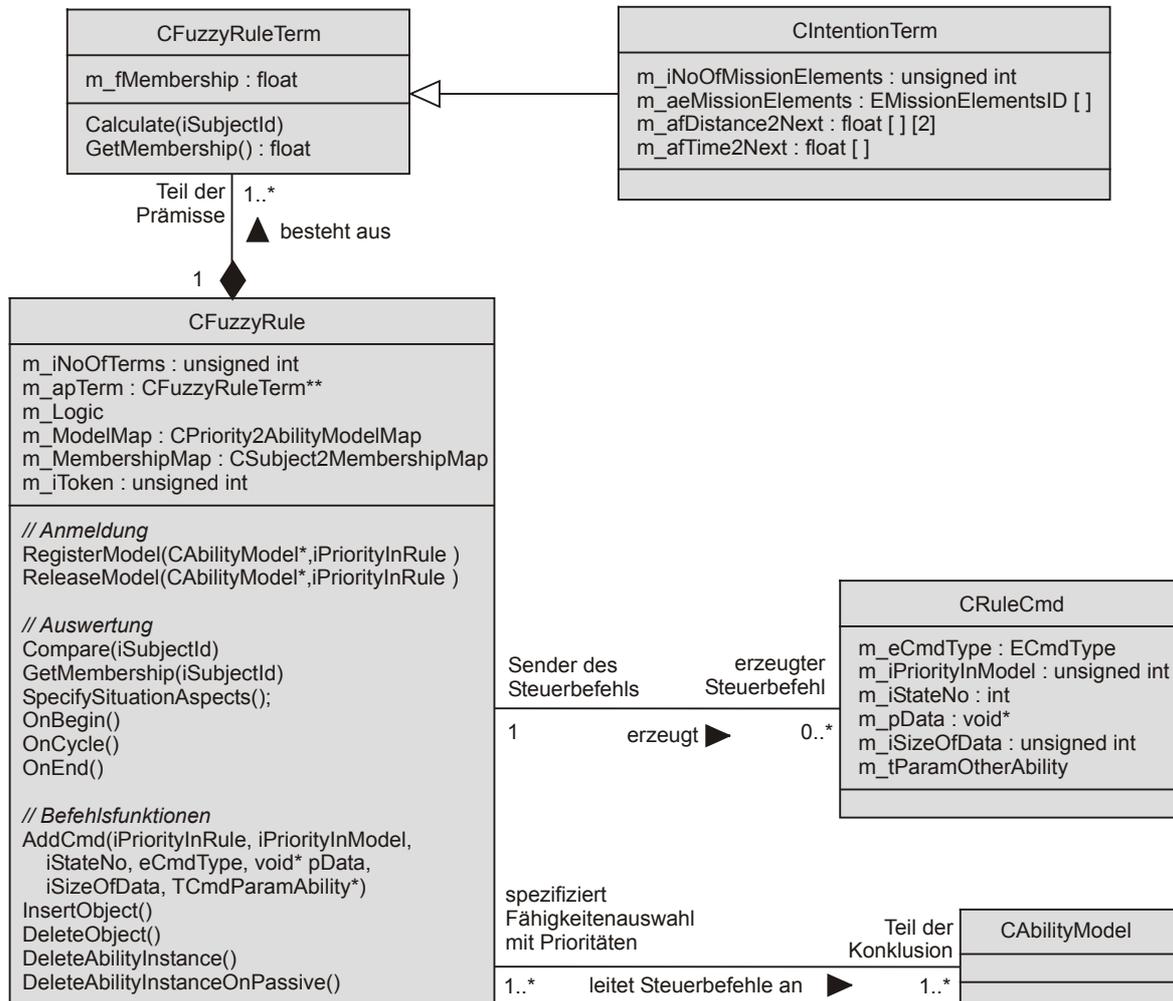


Abb. C.1: Klassendiagramm der Regelbasis (Darstellung nach UML)

Zum Versenden von Steuerbefehlen an die Ablaufmodelle der Fähigkeiten bietet die Basisklasse **CFuzzyRule** die Funktion `AddCmd()` an. Ihr muss man die in der Klasse **CRuleCmd** gespeicherten Informationen übergeben (vgl. Abschnitt C.4). In den Konklusionen der Regeln kann nicht nur die Ausführung der Fähigkeiten gesteuert werden, sondern es ist auch möglich, (virtuelle) Szenenobjekte im Szenenbaum einzuhängen. Solche Objekte sind z. B. Detektionsbereiche zur Steuerung der Wahrnehmung und Blickrichtung (vgl. Kapitel 4 und 7.3). Zum Ein- und Aushängen von Objekten dienen die Funktionen `InsertObject()` und `DeleteObject()`. Die Konklusionen der Regeln beziffern auch den Aufmerksamkeitsbedarf der Szenenobjekte (vgl. Abschnitt 4.5).

## C.2 Die Klasse CFuzzyRuleTerm

Jede Regel der Regelbasis besitzt in ihrer Prämisse eine gewisse Anzahl von unscharfen Termen. Die Basisklasse dieser Terme heißt **CFuzzyRuleTerm** (vgl. Abb. C.1, oben links). Ihre Klassenvariable `m_fMembership` liegt im Intervall  $[0,1]$  und besagt, in welchem Grad die durch den Term formulierte (linguistische) Aussage wahr oder falsch ist. Durch den Aufruf

der Funktion Calculate() wird die Variable m\_fMembership berechnet, wobei die Aussage immer bezüglich des referenzierten Subjekts gilt.

### C.3 Die Klasse CIntentionTerm

Eine besondere Art von Regelterm wird durch die Basisklasse CIntentionTerm realisiert. Solche Regelterme berücksichtigen die gegenwärtigen Missionselemente des autonomen Fahrzeugs (vgl. Abb. C.1, oben rechts). Die Klasse CIntentionTerm ist von CFuzzyRuleTerm abgeleitet. Die Klassenvariable m\_iNoOfMissionElements beschreibt, wie viele Missionselemente durch den Term berücksichtigt werden und m\_aeMissionElements spezifiziert, um welche Missionselemente es sich handelt. Ein Term der Klasse CIntentionTerm ist dann gültig, wenn die berücksichtigten Missionselemente mit dem aktuellen Missionsplan übereinstimmen und die zeitlichen und räumlichen Abstände zwischen den Missionselementen bestimmte Schwellwerte nicht über- oder unterschreiten. Die beiden Arrays m\_afTime2Next und m\_afDistance2Next spezifizieren diese Schwellwerte, wobei man beim letzteren neben einer maximalen auch eine minimale Entfernung angeben kann.

### C.4 Die Klasse CRuleCmd

Möchte eine Regel eine Fähigkeit anwenden, so sendet sie Steuerbefehle an das zugehörige interne Ablaufmodell. Die Basisklasse dieser Steuerbefehle heißt CRuleCmd (vgl. Abb. C.1, Mitte rechts). Die Klassenvariable m\_eCmdType bezeichnet den Typ des Befehls. Die Tabelle C.1 fasst die in der bisherigen Implementierung vorkommenden Befehlstypen zusammen und beschreibt ihre Bedeutung.

Kommandos	Beschreibung
Cmd_Initialize_Ability	<i>Eine Fähigkeit instanziiieren.</i>
Cmd_Parameterize_Ability	<i>Eine Instanz einer Fähigkeit (Aktion) parametrisieren, starten, anhalten bzw. löschen.</i>
Cmd_Start_Ability	
Cmd_Stop_Ability	
Cmd_Delete_Ability	
Cmd_Delete_Ability_OnPassive	<i>Aktion löschen, wenn ihr Operationsteil in den Zustand PASSIVE wechselt.</i>
Cmd_Start_Ability_OnRun	<i>Aktion starten bzw. parametrisieren, wenn der Zustand des Operationsteils einer anderen, referenzierten Aktion RUN erreicht.</i>
Cmd_Parameterize_Ability_OnRun	
Cmd_StartAndStop_Ability_OnRun	<i>Wie Cmd_Start_Ability_OnRun, aber mit Löschen der referenzierten Aktion, wenn deren Operationsteil in den Zustand PASSIVE wechselt.</i>
Cmd_Start_Ability_WithOldParam	<i>Instanz einer Fähigkeit mit den alten Parametern (wieder) starten.</i>

Tab. C.1: Typen von Steuerbefehlen bei der Klasse CRuleCmd

Auf welche Instanz einer Fähigkeit sich ein Steuerbefehl bezieht, wird durch die Klassenvariable m\_iPriorityInModel festgelegt. Wie aus Abschnitt 3.3.6 hervorgeht, beinhaltet ein internes Ablaufmodell einer Fähigkeit einen Zustandsautomaten. Die Variable m\_iStateNo spezifiziert, in welchem Zustand dieses Zustandsautomaten der Steuerbefehl abgesetzt werden soll. Ferner kann man einem Steuerbefehl weitere Parameter mitgeben, indem man diese im

Datenpuffer `m_pData` der Größe `m_iSizeOfData` niederlegt. Bezieht sich der Steuerbefehl auf Ereignisse in einer anderen Instanz einer Fähigkeit, so ist auch diese Instanz mit der Variablen `m_tParamOtherAbility` zu referenzieren. Die Weiterverarbeitung des Steuerbefehls im Ablaufmodell wird in Abschnitt 3.5.3 dargestellt.

## Anhang D

### Der Prozessrahmen CCentralDecisionScene

Der Prozess Central Decision besitzt die Klasse CCentralDecisionScene als Prozessrahmen. Diese Klasse ist von der Klasse CDecisionBaseScene abgeleitet, welche wiederum die Klasse CClientScene als Basisklasse besitzt. Die Funktionalität der Klasse CClientScene hinsichtlich des Fähigkeitenkonzepts wurde bereits im Anhang B.3 dargestellt. Die Klasse CDecisionBaseScene verfügt über die zur Verhaltensentscheidung notwendige Basisfunktionalität. Die Matrix `m_DataProcessingMatrix` beinhaltet alle Datenvorverarbeitungen, die von den Situationsaspekten benutzt werden. Über den Schlüssel

*(ID der Datenvorverarbeitung, ID des Szenenobjekts)*

können die Situationsaspekte in der Matrix auf vorverarbeitete Objektdaten zugreifen. In den Tabellen `m_SituationAspectMap` und `m_AbilityModelMap` sind alle Situationsaspekte bzw. internen Ablaufmodelle eingetragen. Wie Anhang A.6 zeigt, wird für jedes Subjekt in der Szene eine Instanz der Klasse CSubject generiert. Alle diese Instanzen sind in der Tabelle `m_SubjectMap` eingetragen. Die Tabelle `m_FuzzyRuleMap` beinhaltet die unscharfen Regeln der Regelbasis und die Tabelle `m_ObjectCtrlMap` die Referenzen auf die von CD generierten und verwalteten Szenenobjekte (z. B. Detektionsbereiche).

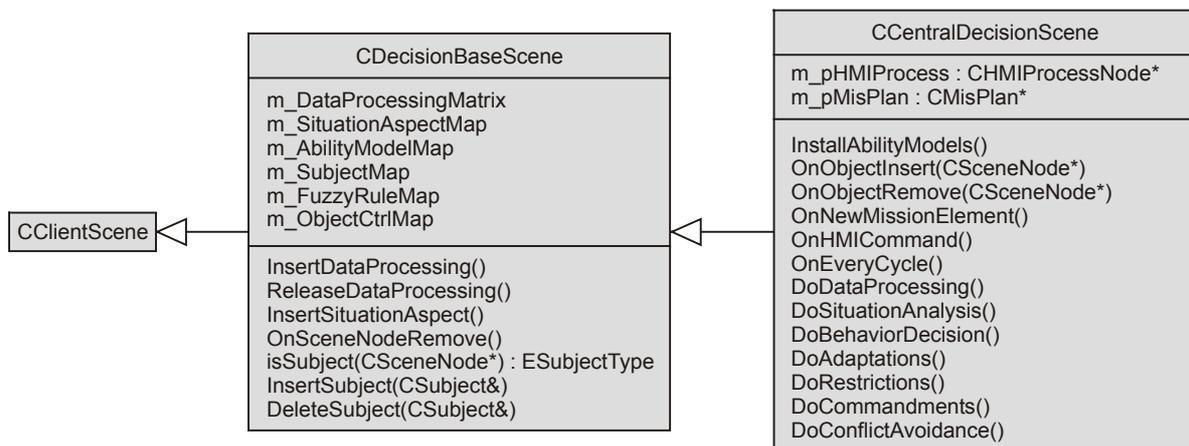


Abb. D.1: Der Prozessrahmen (Klassendiagramm nach UML)

Durch den Aufruf der Klassenfunktion `InsertDataProcessing()` zeigen die Situationsaspekte an, dass sie eine bestimmte Datenvorverarbeitung benötigen. Ist diese bisher in `m_DataProcessingMatrix` noch nicht vorhanden, so wird sie eingerichtet und in die Matrix eingetragen. Mit dem Aufruf der Funktion `ReleaseDataProcessing()` signalisiert ein Situationsaspekt, dass dieser die referenzierte Datenvorverarbeitung nicht länger benötigt. Falls kein anderer Situationsaspekt die Datenvorverarbeitung benutzt, wird diese aus der Tabelle `m_DataProcessingMatrix` entfernt.

Die Klasse CCentralDecisionScene beinhaltet die Funktionalität für die prozedurale Ausführung der Situationsanalyse und Verhaltensentscheidung. Mit den Klassenvariablen m\_pMisPlan und m\_pHMIPProcess besitzt CD eine Anbindung an den Missionsplan und an das Human Machine Interface (HMI). Die Funktion InstallAbilityModels() lädt für jede im Fähigkeitennetz repräsentierte schematische Fähigkeit das dazugehörige interne Ablaufmodell und trägt es in die Tabelle m\_AbilityModelMap ein. Bei einem Ein- und Aushängen eines Szenenobjekts im Szenenbaum erfolgt ein Aufruf der Ereignisbehandlungsfunktion OnObjectInsert() bzw. OnObjectRemove() (vgl. Anhang A.7). Ein Wechsel im aktuellen Missionselement führt zu einem Aufruf der Funktion OnNewMissionElement(). OnEveryCycle() beinhaltet die zyklische Funktionalität, welche im Kapitel 3.5.2 dargestellt ist. Dieses Kapitel erklärt auch die restlichen in Abb. D.1 dargestellten Klassenfunktionen.

## Anhang E

### Die Basisklassen zur Verwaltung der Aktionen

In Abb. E.1 sind die Klassen dargestellt, welche bei der Verwaltung der Aktionen innerhalb der Verhaltensentscheidung eine Rolle spielen. Die Abb. E.1 versteht sich als Fortsetzung von Abb. C.1.

#### E.1 Die Basisklasse CAbilityModel

Die Klasse CAbilityModel ist die Basisklasse der internen Ablaufmodelle der Fähigkeiten. Die Variable m\_pAbilityNode dieser Klasse zeigt auf den zugehörigen Fähigkeitenknoten. Mit m\_pScene besitzt das Ablaufmodell einen Zugang zum Prozessrahmen und zur Wissensrepräsentation. In m\_RuleList sind Verweise auf alle unscharfen Regeln zusammengefasst, welche die Fähigkeit benutzen. Im Array m\_aAbilityCtrl ist für jede mögliche Instanz der Fähigkeit (Aktion) eine Verwaltungsklasse CAbilityCtrl eingerichtet (siehe Abschnitt E.4). Das Array m\_alInstances beinhaltet für jede vorhandene Instanz der Fähigkeit ein Objekt der Klasse CAbilityInstance. Erreicht nun ein Objekt der Klasse CAbilityModel ein Kommando von einer Regel, so leitet dieses das Kommando an das zugehörige Objekt der Klasse CAbilityInstance weiter. Innerhalb eines Ablaufmodells werden die Instanzen anhand ihrer Priorität unterschieden und die Priorität ist gleich der Position im Array m\_alInstances.

#### E.2 Die Klasse CAbilityInstance

Die Klasse CAbilityInstance ist die verwaltende Basisklasse für alle Aktionen innerhalb des internen Ablaufmodells. Die Klassenvariable m\_iPriorityInModel zeigt die Priorität im Ablaufmodell an. Wie in Kapitel 3.3.6 ausführlich dargestellt wurde, besitzt ein internes Ablaufmodell einen Zustandsautomaten. Die Tabelle m\_StateMap beinhaltet alle Zustände dieses Automaten und m\_iNoOfStates zeigt die Anzahl der Zustände an. Mit den Zeigern m\_pAbilityModel und m\_pAbilityCtrl besitzt die Klasse CAbilityInstance Referenzen auf das zugehörige Ablaufmodell bzw. auf die zugehörige Verwaltungsklasse.

#### E.3 Die Klasse CInstanceState

Die Klasse CInstanceState ist die Basisklasse für alle Zustände im Zustandsautomaten des Ablaufmodells. Die Objekte dieser Klasse sind die eigentlichen Empfänger der Steuerbefehle von den unscharfen Regeln. In der Klassenvariablen m\_CommandList sind alle für den Zustand abgesetzten Steuerbefehle gesammelt. Die Variable m\_sName bezeichnet den Namen des Zustands und m\_iNumber die Nummer. Die Variable m\_bIsCurrent gibt an, ob der Zustand der aktuelle Zustand ist. Nur die Steuerbefehle des aktuellen Zustands können zur Anwendung kommen. Die Einträge des Arrays m\_apNextStates zeigen auf die Folgezustände innerhalb des Zustandsautomaten. Mit AddCmd(CRuleCmd&) trägt man einen Steuerbefehl, welcher ausgehend von einer unscharfen Regel über Objekte der Klassen CAbilityModel und CAbilityInstance geleitet wurde, in die Liste m\_CommandList ein. Ist der Zustand der aktuelle

Zustand, so erfolgt der zyklische Aufruf der Funktionen DoCommands() und CheckTransitions(). Innerhalb der ersten Funktion wird die Ausführbarkeit der in m\_CommandList gespeicherten Steuerbefehle geprüft. Ausführbare Befehle kommen zur Anwendung und werden anschließend aus der Liste gelöscht. Durch die Ausführung der Steuerbefehle verändern sich die Markierungen des zugehörigen Verwaltungsobjekts der Klasse CAbilityCtrl. Die Funktion CheckTransitions() überprüft die Übergangsbedingungen zu den Folgezuständen. Bei einem Übergang wird die Variable m\_bIsCurrent des bisherigen Zustands auf FALSE gesetzt und die vom Folgezustand auf TRUE.

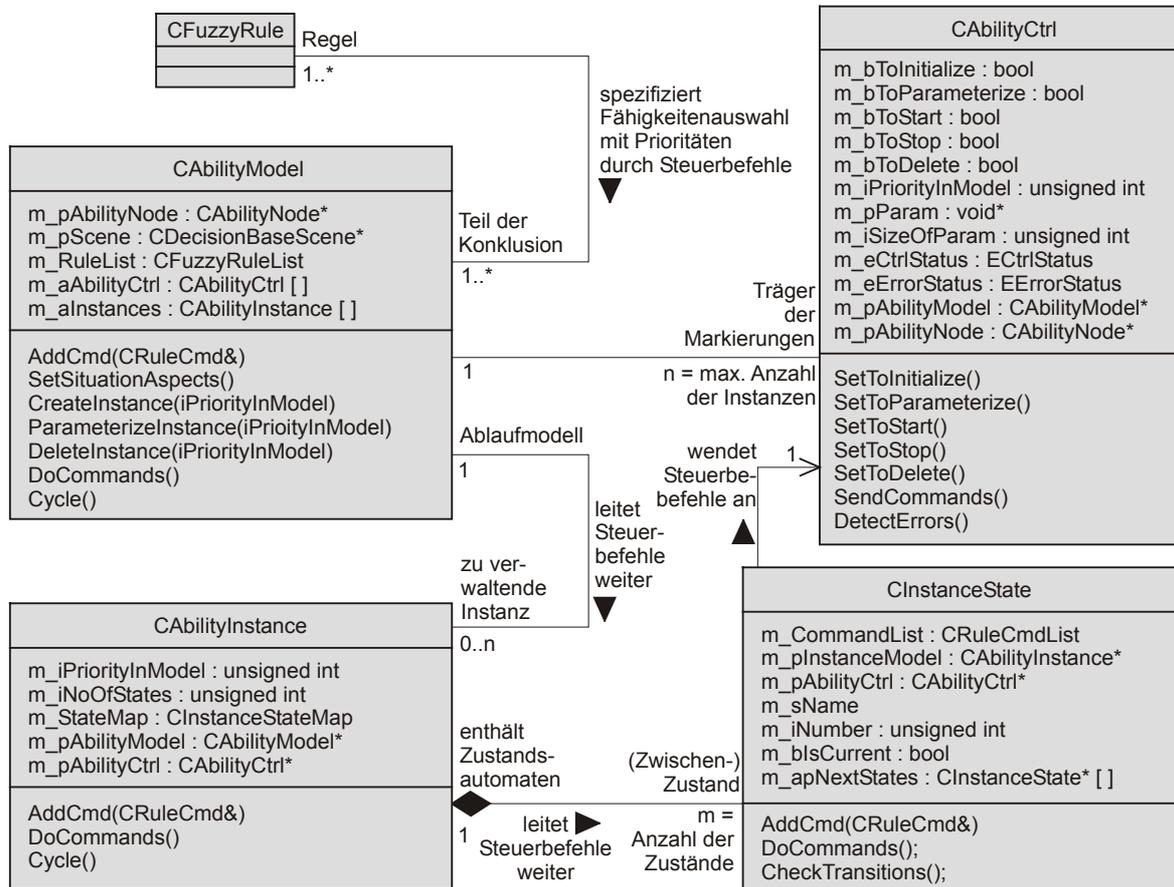


Abb. E.1: Die Verwaltung der Aktionen in CD (Klassendiagramm nach UML)

## E.4 Die Klasse CAbilityCtrl

Ein Objekt der Klasse CAbilityCtrl verwaltet eine mögliche Instanz einer Fähigkeit. Die Klasse beinhaltet mit den Variablen m\_bToInitialize, m\_bToParameterize, m\_bToStart, m\_bToStop und m\_bToDelete die Markierungen, welche durch Steuerbefehle (der Klasse CRuleCmd) verändert werden können. Die Variable m\_iPriorityInModel bezeichnet die Priorität der Instanz innerhalb des internen Ablaufmodells. Der Datenpuffer m\_pParam der Größe m\_iSizeOfParam speichert die vom Steuerbefehl (Klasse CRuleCmd) empfangenen Parameter für die Befehlsausführung. Die Variable m\_eCtrlStatus beschreibt den zuletzt erreichten Steuerungsstatus und m\_eErrorStatus den detektierten Fehlerstatus. Die „Set“-Funktionen der Klasse CAbilityCtrl werden innerhalb der Funktion CInstanceState::DoCommands() benutzt (vgl. Abschnitt E.3). Innerhalb SendCommands() erfolgt die in Abschnitt 3.5.3 gezeigte Auswertung der Markierungen und das Absenden der Kommandos über das Fähigkeitsnetz zu den Experten der Fähigkeiten. Die Funktion DetectErrors() vergleicht den kommandierten Status m\_eCtrlStatus mit dem wirklichen Status der Instanzen und setzt den Fehlerstatus m\_eErrorStatus.