

## Geleitwort des Herausgebers

Die Industrie- und Dienstleistungsgesellschaft ist einem stetigen Wandel unterworfen. Die Herausforderungen an sie werden immer komplexer. Das äußert sich im verstärkten internationalen Wettbewerb, aber auch in dem Bestreben der Gesellschaft, das Erreichte zu sichern. Adäquate Problemlösungen sind daher in zunehmender Weise nur fachübergreifend zu realisieren. Im Heinz Nixdorf Institut leisten wir mit der interdisziplinären Zusammenarbeit vor allem zwischen der Informatik und den Ingenieur- und Wirtschaftswissenschaften unseren Beitrag zur Bewältigung dieser Aufgaben.

Der Sonderforschungsbereich 614 ist ein Musterbeispiel für diese Art interdisziplinären Arbeitens. Unter der Überschrift „Selbstoptimierende Systeme des Maschinenbaus“ werden die Grundlagen der maschinenbaulichen Systeme von morgen erarbeitet. Selbstoptimierung ermöglicht dabei weitgehend autonom handlungsfähige Systeme, die in der Lage sind, flexibel auf veränderte Betriebsbedingungen zu reagieren. Die Vision des SFB 614 ist eine neue Schule des Entwurfs von intelligenten mechatronischen Systemen.

Herr Klöpper greift diese Gedanken in seiner Arbeit auf und identifiziert wesentlichen Handlungsbedarf auf dem Weg zu intelligenten mechatronischen Systemen. Planung als ein wesentlicher Aspekt intelligenten Verhaltens bleibt bisher in mechatronischen Systemen unberücksichtigt. Diese Lücke schließt Herr Klöpper, in dem er die Grundlagen einer Verhaltensplanung für mechatronische Systeme erarbeitet. Hierzu wurden die gängigen Methoden der künstlichen Intelligenz (Planung, probabilistische Inferenz, Multiagenten-Planung) mit den speziellen Anforderungen mechatronischer Systeme verknüpft. Die entwickelten Konzepte sind in Anwendungsbeispielen aus dem Projekt „Neue Bahntechnik Paderborn“ validiert worden.

Paderborn, März 2009

Prof. Dr.-Ing. habil. Wilhelm Dangelmaier



## Vorwort

Diese Arbeit ist im Rahmen meiner Tätigkeit am Lehrstuhl für Wirtschaftsinformatik, insb. CIM entstanden. Neben der sehr interessanten wissenschaftlichen Tätigkeit im SFB 614 wird mir die Zeit am Lehrstuhl vor allem als eins in Erinnerung bleiben: als äußerst lustig und humorvoll. Ohne den regen (nicht immer rein fachlichen) Austausch mit den Kollegen wären die vier Jahre wesentlich schwieriger durchzuhalten gewesen. Stellvertretend für alle **CIMP-SONS** sei mein Bürokollege Carsten genannt. Zwar lag er mir im Sommer 2007 in den Ohren: „Zitierst du mich in deiner Diss?“<sup>1</sup>, aber ansonsten gab es viel zu lachen. Und daher wird sich sicherlich folgender Dialog abspielen, wenn ich erzähle, dass er im Vorwort zitiert wird:

*Benjamin: „Ich habe dich im Vorwort zitiert“.*

*Carsten: „Nein!“*

*Benjamin: „Doch!“*

*Carsten: „Ohhhhhhh!“*

Daneben ist im Rahmen dieses Vorworts natürlich noch eine ganze Menge ernsthafter Dank zu verteilen. Da mechatronische Systeme vor meiner Arbeit im SFB 614 gänzlich unbekannt für mich waren, mussten mir viele der Mitstreiter mit ihren Fachkenntnissen auf die Sprünge helfen. Dafür danke ich besonders Alexander Schmidt, Christoph Romaus und Henner Vöcking. Ebenso danke Christoph Danne, Dietrich Dürksen, Viktor Dück, Daniel Ruth, Christoph Thonemann und Jens Wienstroer für ihre tatkräftige und kreative Unterstützung meiner Arbeit. Meinem Bruder Markus und Herbert Podlogar danke ich für die Übernahme der äußerst mühevollen Aufgabe des Korrekturlesens. Ich hoffe, am Ende ist durchgedrungen, dass Fuzzy Logic nichts mit schlechten Witzen zu tun hat. „Wacker wacker wacker!“

Für die sehr ausführliche Betreuung meiner Arbeit danke ich Prof. Dr.-Ing. habil. Dangelmaier. Auf der Grundlage vieler Zwischenstände Anregung und Kritik zu erhalten, war äußerst hilfreich. Das Einfordern dieser Zwischenstände sicherte einen kontinuierlichen Arbeitsfluss. Herrn Prof. Rammig danke ich für die Übernahme des Zweitgutachtens. Frau Prof. Suhl und Frau Jun.-Prof. Kliewer, die mir als Mitglieder der Promotionskommission wertvolles Feedback gaben.

Der letzte Dank gehört aber natürlich der Familie. Dass ich mich immer in allen kleinen und auch größeren Dingen auf meine Brüder, meine Schwester und natürlich meine Mutter verlassen kann, gibt mir sehr viel Sicherheit.

Nach diesem lockeren Auftakt hoffe ich, dass dem Leser möglichst informative und nützliche Lektüre auf den kommenden etwa 200 Seiten erwartet. Ich fürchte, das reine Lesevergnügen wird aufgrund der etwas trockenen Thematik zurückstehen müssen.

Paderborn, März 2009

Benjamin Klöpfer

---

<sup>1</sup>[CB07] Mündliches Zitat Carsten Böhle im Sommer 2007



# Ein Beitrag zur Verhaltensplanung für interagierende intelligente mechatronische Systeme in nicht-deterministischen Umgebungen

Dissertation  
zur Erlangung der Würde eines  
DOKTORS DER WIRTSCHAFTSWISSENSCHAFTEN  
(Dr. rer. pol.)  
der Universität Paderborn

vorgelegt von  
Dipl.-Wirt.-Inf. Benjamin Klöpper  
33102 Paderborn

Paderborn, Januar 2009

Dekan: Prof. Dr. Peter F. E. Sloane  
Referent: Prof. Dr.-Ing. habil. W. Dangelmaier  
Korreferent: Prof. Dr. F. J. Rammig

---



## Inhaltsverzeichnis

Inhaltsverzeichnis .....	I
Abbildungsverzeichnis .....	III
Verzeichnis der Listings .....	V
Tabellenverzeichnis .....	V
1 Einleitung .....	1
2 Problemstellung .....	7
2.1 Abbildung der Verhaltensplanung auf die Planung in der KI .....	10
2.1.1 Modellierung mithilfe der PDDL .....	11
2.1.2 Modellierung einer mechatronischen Domäne .....	14
2.1.3 Modellierung einer Beispieldomäne .....	17
2.2 Integration kontinuierlicher Prozesse .....	20
2.3 Unsicherheit und unvollständige Information .....	20
2.4 Interaktion zwischen mechatronischen Systemen .....	23
2.5 Aufbau der Arbeit .....	27
3 Integration kontinuierlicher Prozesse in eine Verhaltensplanung auf Basis der PDDL ..	29
3.1 Stand der Technik zur diskret-kontinuierlichen Planung .....	29
3.1.1 Modellierung kontinuierlicher Planungsaspekte .....	29
3.1.2 Planungsverfahren mit quasi-kontinuierlichen Prozessen .....	32
3.1.3 Optimale Steuerung und kontinuierliche Mehrzieloptimierung .....	33
3.1.4 Fazit zum Stand der Technik in diskret-kontinuierlicher Planung .....	34
3.2 Integration kontinuierlicher Verläufe in die Planung .....	35
3.2.1 Modellierung kontinuierlicher Verläufe mithilfe vorgelagerter Optimierung ...	35
3.2.2 Modellierung kontinuierlicher Verläufe mittels Fuzzy-Approximation .....	37
3.2.3 Fazit zur Integration kontinuierlicher Prozesse .....	40
4 Planung unter Unsicherheit .....	43
4.1 Stand der Technik zur Behandlung von Unsicherheit .....	43
4.1.1 Modellierung von Unsicherheit .....	44
4.1.2 Modelle und Verfahren zur Analyse von Unsicherheit .....	46
4.1.3 Planungssprachen für Unsicherheit und unvollständige Information .....	51
4.1.4 Planungsverfahren zur Behandlung von Unsicherheit .....	54
4.1.5 Fazit zum Stand der Technik Planung unter Unsicherheit .....	59
4.2 Planung unter Unsicherheit für mechatronische Systeme .....	60
4.2.1 Planungsverfahren auf Basis bedingter Wahrscheinlichkeiten .....	62
4.2.2 Umplanungsverfahren auf Basis von Echtzeitsuche .....	84
5 Planung von Systeminteraktionen .....	87
5.1 Stand der Technik zur Planung von Systeminteraktionen .....	90
5.1.1 Modellierung der Planung von Systeminteraktionen .....	90
5.1.2 Phasenmodelle zur Planung von Systeminteraktionen .....	91
5.1.3 Verfahren zur Berücksichtigung von Interaktion .....	93
5.1.4 Fazit zum Stand der Technik zur Interaktionsplanung .....	103
5.2 Planung der Interaktion bei gemeinsamer Hauptaufgabe .....	104
5.2.1 Mögliche Koordinationsaktionen .....	105
5.2.2 Blackbox-Modellierung der Wechselwirkungen zwischen Teilsystemen .....	108
5.2.3 Koordination vor der Planung .....	115
5.2.4 Koordination nach der lokalen Planung .....	116
5.2.5 Berücksichtigung gemeinsamer Ressourcen .....	119
5.2.6 Ablauf der Koordination .....	119
5.3 Planung der Interaktion bei individueller Hauptaufgabe .....	121

5.3.1	Mögliche Kooperationsaktionen .....	121
5.3.2	Modellierung von Kooperationsaktionen .....	122
5.3.3	Durchführung der Abgabe von Aufträgen .....	124
5.3.4	Durchführung der Abgabe von Teilfunktionen .....	126
5.3.5	Abstimmung von Aktivitäten .....	129
5.3.6	Gesamtablauf .....	130
6	Integrationsszenario für die Planungsprozesse .....	135
7	Anwendungsbeispiele .....	139
7.1	Integration kontinuierlicher Verläufe für die aktive Federung .....	141
7.2	Planung unter Unsicherheit .....	143
7.2.1	Probabilistische Routenplanung für bedarfsorientierten Schienentransport ...	143
7.2.2	Die Agentenklasse Track-Experte .....	143
7.2.3	Erweiterung des Fahrzeugagenten .....	145
7.2.4	Experimente und Ergebnisse .....	146
7.3	Planung von Systeminteraktionen .....	153
7.3.1	Planung bei gemeinsamer Hauptaufgabe .....	153
7.3.2	Planung bei individueller Hauptaufgabe .....	161
8	Zusammenfassung und Ausblick .....	169
8.1	Zusammenfassung .....	169
8.2	Ausblick .....	171
	Literaturverzeichnis .....	173
A.	Modellierungsmethoden im Entwurf mechatronischer Systeme .....	181
B.	Planungsverfahren der Künstlichen Intelligenz .....	182
C.	XML-Format der Bayes-Netzwerke .....	183
D.	Versuchsaufbau Belief-Dijkstra .....	185



## Abbildungsverzeichnis

Abbildung 1-1: Darstellung einer Aufgabe mithilfe einer Funktionshierarchie .....	2
Abbildung 1-2: Gliederung eines komplexen mechatronischen Systems [LHL+01] .....	5
Abbildung 2-1: Verknüpfung eines mechatronischen Systems mit seinem Umfeld .....	7
Abbildung 2-2: Planungsproblem für mechatronische Systeme als Baum .....	9
Abbildung 2-3: Struktur der Planungsaufgabe für mechatronische Systeme .....	10
Abbildung 2-4: Das PDDL-Planungsproblem als Planbaum .....	13
Abbildung 2-5: Vergleich von Teilfunktionslösung und PDDL-Aktion .....	15
Abbildung 2-6: Funktionshierarchie mit Teilfunktionslösungen .....	16
Abbildung 2-7: Planungsbaum unter Unsicherheit .....	21
Abbildung 2-8: Zwei mechatronische Systeme in einem Umfeld .....	23
Abbildung 2-9: Planungsproblem bei zwei mechatronischen Systemen .....	25
Abbildung 3-1: Modellierung kontinuierlicher Verläufe mit vorgelagerter Optimierung .....	35
Abbildung 3-2: Modellierung kontinuierlicher Verläufe mittels Fuzzy-Approximation .....	37
Abbildung 3-3: Integrationsszenario für Fuzzy-Approximation .....	40
Abbildung 4-1: Skizze des Planungsverfahrens für Unsicherheit .....	61
Abbildung 4-2: Bedingte Planung und reaktive Umplanung im Energiemanagement .....	62
Abbildung 4-3: Allgemeine Struktur eines Bayes-Netzwerks für probabilistische Effekte ....	67
Abbildung 4-4: Beispiel Netzwerk aus der Aktion <i>Drive</i> .....	68
Abbildung 4-5: Planbaum nach der probabilistischen Analyse .....	70
Abbildung 4-6: Struktur der probabilistischen Planrepräsentation .....	71
Abbildung 4-7: Zustandsautomat der erweiterten probabilistischen Analyse .....	73
Abbildung 4-8: Erweiterte probabilistische Analyse .....	75
Abbildung 4-9: Darstellung einer Vorbedingung als Planungsbaum .....	76
Abbildung 4-10: Propagation der Wahrscheinlichkeiten im Vorbedingungsbaum .....	76
Abbildung 4-11: Bewertung zusammengesetzter Ereignisse .....	78
Abbildung 4-12: Zustandstrajektorie zu einer nicht ausführbaren Aktivität .....	80
Abbildung 4-13: Gesamtablauf der bedingten Planung .....	84
Abbildung 5-1: Betrachtete Modi des Antriebs .....	87
Abbildung 5-2: Struktur des Planungsproblems für den Antrieb .....	88
Abbildung 5-3: Wechselwirkung zwischen Antrieb und Luftspaltverstellung .....	89
Abbildung 5-4: Skizze der Interaktionsplanung bei gemeinsamer Hauptaufgabe .....	104
Abbildung 5-5: Mögliche Plananpassung im Planungsproblem des Antriebs .....	106
Abbildung 5-6: Einbettung einer alternativen Teilfunktionslösung .....	107
Abbildung 5-7: Planung beim Verschieben einer Teilaktion aus einem Zeitfenster .....	107
Abbildung 5-8: Planung beim Verschieben einer Teilaktion in ein Zeitfenster .....	107
Abbildung 5-9: Umplanung .....	118
Abbildung 5-10: Ablauf der Interaktionsplanung bei gemeinsamer Hauptaufgabe .....	120
Abbildung 5-11: Einbettung alternativer Pläne als Ersatz für abzugebende Teilfunktionen ..	128
Abbildung 5-12: Hierarchische Beziehung zwischen Kooperationsgegenständen .....	131
Abbildung 5-13: Ablauf der Interaktionsplanung bei individueller Hauptaufgabe .....	132
Abbildung 6-1: Integrationsszenario für die Planungsverfahren .....	137
Abbildung 7-1: Planungshierarchie im RailCab-Beispiel .....	139
Abbildung 7-2: Vereinfachte Darstellung der Abläufe im RailCab-MAS .....	140
Abbildung 7-3: Pareto-Mengen im Bildraum für die Federung [VT08] .....	141
Abbildung 7-4: Zwei mit einem Fahrzeugagenten kommunizierende Track-Experten .....	144
Abbildung 7-5: Klassendiagramm des Packages PNS .....	145
Abbildung 7-6: Ergebnisse bei exakten Prognoseverteilungen .....	148
Abbildung 7-7: Ergebnisse bei einer Variation $\mu$ von -8% .....	148
Abbildung 7-8: Ergebnisse bei einer Variation $\mu$ von +5% .....	149

Abbildung 7-9: Ergebnisse bei einer Variation $\mu$ von +15% .....	149
Abbildung 7-10: Ergebnisse bei einer Variation +5% aller Parameter .....	150
Abbildung 7-11 Ergebnisse bei einer Variation von -10% aller Parameter .....	150
Abbildung 7-12: Fahrzeiterparnis durch den BeliefDijkstra .....	151
Abbildung 7-13: Fahrzeiterparnis bei Variation aller Verteilungsparameter um +10% .....	151
Abbildung 7-14: Ergebnisse bei 70% Anteil ungenauer Streckenabschnitte.....	159
Abbildung 7-15: Ergebnisse in allen Testszenarien .....	159
Abbildung 7-16: Ergebnisse der Umplanung .....	160
Abbildung 7-17: Dauer der Iterationen der Umplanung .....	161
Abbildung 7-18: Beispiele für Wegeinsparungen durch die Abgabe von Aufträgen .....	162

## Verzeichnis der Listings

Listing 2-1: Kopf der RailCab-Domäne .....	17
Listing 2-2: Funktionen der RailCab-Domäne .....	17
Listing 2-3: Fahraktion mit aktiver Federung .....	18
Listing 2-4: Definition des Planungsproblems .....	18
Listing 2-5: Initialisierung in der Problemdatei .....	19
Listing 2-6: Zielzustand des Beispielproblems .....	19
Listing 2-7: Drei alternative Planungsmetriken .....	19
Listing 3-1: Fuzzy Definition der numerischen Variablen.....	38
Listing 4-1: Funktion in der probabilistischen Domänendefinition für das RailCab .....	64
Listing 4-2: Ausschnitt einer probabilistischen Problemdefinition des RailCab-Szenarios ....	69
Listing 4-3: Planmetrik in der probabilistischen Problemdefinition .....	69
Listing 4-4: Probabilistische Analyse eines Plans.....	72
Listing 4-5: Ableitung relevanter Ereignisse aus einer Aktivität.....	77
Listing 4-6: Einfügen proaktiver Verzweigungen in einen Plan.....	80
Listing 5-1: Bedingte Aktion mit zeitabhängigem Nutzenwert .....	113
Listing 5-2: Berechnung der Funktion $req(s_i)$ .....	127
Listing 7-1: Aus der Optimierung abgeleitete Aktionsdefinition.....	142
Listing 7-2: Pseudo-Code des BeliefDijkstra .....	146
Listing 7-3: Planungsalgorithmus für den Antrieb in Pseudo-Code .....	156
Listing 7-4: Anpassungsalgorithmus in Pseudo-Code .....	157
Listing 7-5: Suche nach abzugebenden Transportaufträgen .....	163
Listing 7-6: Fahraktionen mit Berücksichtigung der Kooperationsform "Konvoi" .....	165
Listing C-1: DTD des eXtended Bayesian Network Interchange Format .....	183
Listing C-2: Beispiel einer XBIF Beschreibung eines Experten-Netzwerks .....	184

## Tabellenverzeichnis

Tabelle 4-1: Berechnung der Schrankenwerte für antizipierende Verzweigung .....	79
Tabelle 5-1: Interaktionsplanungsparameter als Bayes-Netzwerke .....	115
Tabelle 7-1: Funktionswerte von <b>XEG</b> (tfl) und <b>XUG</b> (tfl).....	154
Tabelle 7-2: Funktionswerte von <b>XEL</b> (tfl) und <b>XUL</b> (tfl).....	154
Tabelle 7-3: Funktionswerte von <b>Hu<math>\omega</math></b> (tfl tfl') .....	155
Tabelle 7-4: Situationsabhängiger mittlerer Luftspalt auf einem Streckenabschnitt .....	158
Tabelle C-1: Wahrscheinlichkeitsverteilung einer Variablen C mit Elternknoten A und B ..	184
Tabelle D-1: Schablonen für die Bayes-Netzwerke der Track-Agenten.....	185
Tabelle D-2: Weiche Evidenzen für den Knoten Humidity .....	186
Tabelle D-3: Zufallsvariablen der beiden verwendeten Netztypen.....	186
Tabelle D-4: Mittlere Geschwindigkeiten zur Generierung der Erwartungswerte .....	187
Tabelle D-5: Anteile zur Bestimmung der Standardabweichungen .....	187



# 1 Einleitung

Ein RailCab<sup>2</sup>, ein fahrerloses Schienenfahrzeug, hat eine Reihe von Transportaufträgen zu erfüllen. Diese Transportaufträge führen das Fahrzeug zunächst von Kassel nach Paderborn, wo es weitere Fahrgäste für eine Fahrt nach Bielefeld aufsammeln muss. Auf dieser Fahrt hat das RailCab eine Reihe von Entscheidungen zu treffen. Dabei sind diese Entscheidungen so zu fällen, dass das RailCab seine Transportaufträge pünktlich abarbeitet, möglichst geringe Kosten verursacht und den Fahrgästen einen angemessenen Fahrkomfort bietet.

Eine Entscheidung betrifft beispielsweise die Streckenwahl von Paderborn nach Bielefeld. Es besteht dabei die Wahl zwischen zwei unterschiedlichen Streckenverläufen. Ein wichtiges Entscheidungskriterium für das RailCab ist dabei die Möglichkeit einer Konvoibildung, also die Strecke oder einen Teil der Strecke gemeinsam mit anderen Fahrzeugen zurückzulegen, um durch den Windschatten-Effekt Antriebskraft einzusparen.

Eine andere Entscheidung betrifft die Frage, wie schnell einzelne Streckenabschnitte befahren werden sollen, wobei die Abhol- und Ankunftszeiten der einzelnen Fahrgäste eingehalten werden müssen. Diese Entscheidung hat u. a. Auswirkungen auf die Energieeffizienz des RailCabs, da mit steigender Geschwindigkeit das Air Gap Adjustment System (AGAS) des RailCabs nur große Luftspalte für den verwendeten Linearmotor einstellen kann und aus einem großen Luftspalt eine geringere Effizienz des Motors resultiert.

Eine zentrale Entscheidungsgröße für das RailCab stellt die verfügbare Energie dar. Die auf dem Fahrzeug installierten Energiespeicher können einerseits geladen werden, andererseits muss die verfügbare Energie zwischen den einzelnen Verbrauchermodule des RailCabs möglichst Nutzen stiftend verteilt werden. Eine mögliche Form des Nutzens ist der Fahrkomfort, der durch ein aktives Federungsmodul maximiert werden soll. Dabei geht ein hoher Fahrkomfort abhängig von den Eigenschaften des Fahrweges mit hohen Energieverbräuchen einher.

Die gefällten Entscheidungen sind teilweise voneinander zeitlich abhängig, da beispielsweise der Batteriezustand Einfluss auf die Verlustströme und Batterieschädigung hat und einmal verbrauchte Energie zu einem späteren Zeitpunkt nicht mehr zur Verfügung steht.

Um all diese Entscheidungen sinnvoll treffen zu können, muss das RailCab vorausplanen. So muss es beispielsweise seine Fahrwege kennen, um Energieverbrauch und –bereitstellung abschätzen zu können. Ankunftszeiten und Fahrwege müssen ebenfalls bekannt sein, um die Fahrprofile für einzelne Streckenabschnitte auswählen zu können. Um die Entscheidung über den Fahrweg treffen zu können, muss das RailCab Informationen über die Fahrwege anderer Fahrzeuge sammeln, um beispielsweise eine Konvoibildung vor auszuplanen. Planung legt all diese Parameter vor der eigentlichen Ausführung einer Aufgabe fest.

Ein RailCab ist ein Beispiel für ein technisches System, das die Qualität seiner Aufgabenerfüllung (Durchführung von Transporten) mithilfe einer Planung verbessern kann. Technische Systeme werden entwickelt und eingesetzt, um für Menschen Aufgaben<sup>3</sup> zu übernehmen. Mit dem technischen Fortschritt vergrößert sich laufend der Einsatzbereich technischer Systeme und verbessert sich die Qualität, mit der die Aufgaben erfüllt werden. Die Qualität der Aufgabenerfüllung kann dabei in einer Vielzahl unterschiedlicher Dimensionen gemessen werden, die jeweils vom konkreten Anwendungsgebiet des technischen Systems abhängen. Beispiele für diese Dimensionen der Qualität sind:

---

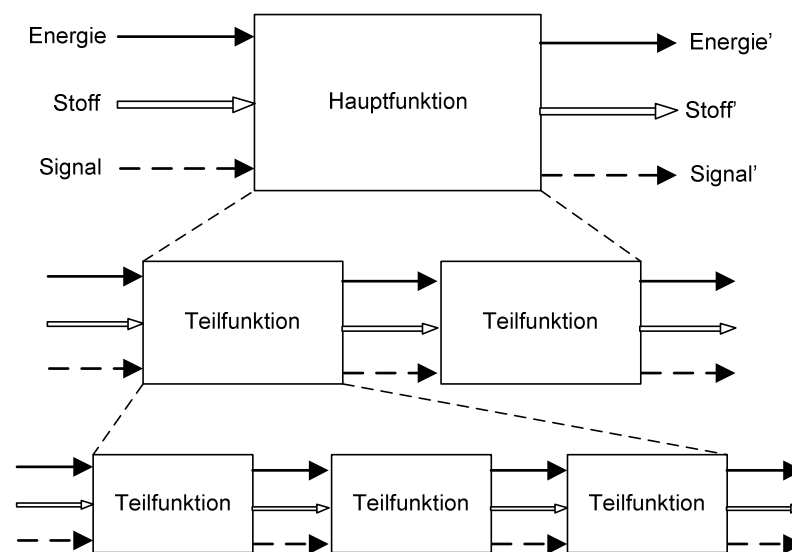
<sup>2</sup> Details über das RailCab-System und die Beziehungen zwischen seinen Teilmodulen können z.B. in [ADG+08] nachgelesen werden.

<sup>3</sup> Nach Krüger „...lassen sich Aufgaben knapp als Verpflichtung, Verrichtungen an Objekten durchzuführen“ (vgl. Frese, 1980) charakterisieren“ [Fres92]. Sie beschreiben als die gewollte Durchführung einer Manipulation an einem materiellen oder immateriellen Objekt.

- Zeit, die zur Aufgabenerfüllung nötig ist,
- Pünktlichkeit des Ergebnisses,
- Ressourcen, die zur Aufgabenerfüllung eingesetzt werden,
- Fertigungspräzision bei Werkzeugmaschinen oder
- Komfortwerte oder Fahrspaß bei Fahrzeugen.

Während die ersten drei Beispiele unabhängig vom konkreten Anwendungsfall interpretiert werden können, ergeben sich die beiden letzten aus einer bestimmten Anwendungsdomäne heraus. Gemeinsam ist allen fünf Beispielen, dass sie zur Beurteilung des Systemverhaltens herangezogen werden können und daher in eine mögliche Zielfunktion eingehen können.

Das technische System erfüllt seine Aufgabe, indem es ein bestimmtes Verhalten umsetzt. Im Rahmen des Entwurfs des technischen Systems wird dieses Verhalten in der Regel in Form eines Ein- und Ausgabeverhaltens betrachtet. Wird die Aufgabe des Systems als die Abbildung der Eingabegrößen auf die gewünschten Ausgabegrößen beschrieben, bezeichnet man diese Abbildung als Hauptfunktion des Systems. Funktionen und deren Umsetzung sind ein zentrales Mittel zur Beschreibung des Verhaltens eines technischen Systems. Dabei steht die Transformation eingehender Energie-, Stoff- und Signalflüsse in ebensolche ausgehende Flüsse im Vordergrund. Durch die Bildung einer Funktionshierarchie wird dabei die Hauptaufgabe schrittweise immer feiner in Teilaufgaben zerlegt, bis ein angemessener Konkretisierungsgrad erreicht ist. Dabei werden die einzelnen Funktionen jeweils als Blackbox aufgefasst, es wird also jeweils nur ihr Ein- und Ausgabeverhalten betrachtet [Rode91]. Abbildung 1-1 stellt dieses Vorgehen dar.



**Abbildung 1-1: Darstellung einer Aufgabe mithilfe einer Funktionshierarchie**

Die Funktionen werden in der Regel zunächst verbal beschrieben – bspw. mithilfe sogenannter technischer Verben aus Konstruktionskatalogen [Roth01], durch die Angabe eines Haupt- und Zeitworts [PB93] oder aus einem dreistufigen Funktionsmodell [Huan01]. Nachdem eine geeignete Funktionshierarchie für das zu entwickelnde Produkt aufgestellt ist, müssen die Teilfunktionen auf mögliche Zusammenhänge und Abhängigkeiten untersucht werden. Insbesondere auf eine zeitliche Reihenfolge der Ausführung der Teilfunktionen wird dabei abgezielt. Daraus ergibt sich eine Funktionsstruktur oder Funktionskette, die das Aufeinanderfolgen der Teilfunktionen beschreibt. In einem nächsten Schritt werden logische Funktionen genutzt, um die Zusammenhänge zwischen Teilaufgaben oder den Ein- und Ausgangsgrößen von Funktionen zu beschreiben. Nachdem die Zusammenhänge zwischen den Teilfunktionen erfasst sind, schließt sich die Suche nach den technischen Lösungen zur Erfüllung der Funkti-

on an. Die möglichen Lösungen zur Erfüllung einer Teilfunktion werden in Form von Wirkprinzipien beschrieben. Auf diese Weise wird das Verhalten des technischen Systems immer konkreter beschrieben. Obwohl durch den Einsatz von logischen Operatoren wie „und“ und „oder“ grundsätzlich auch situationsabhängige Funktionsketten denkbar sind, ist die Abfolge doch relativ unflexibel angegeben. Daraus ergeben sich eine relativ geringe Situationsangemessenheit und mangelnde Möglichkeiten, im Rahmen intelligenter Systeme auf geänderte Zielstellungen zu reagieren. Eine intelligente Entscheidungsfindung, wie sie im Eingangsbeispiel für das RailCab gefordert wurde, ist mithilfe solcher Funktionsketten nicht möglich. Die Anzahl der möglichen Betriebssituationen, die der Entwickler zu berücksichtigen hätte (beispielsweise alle möglichen Streckenabfolgen), ist zu groß.

Ein aktueller Trend in der Entwicklung technischer Systeme ist das zunehmende Zusammenwachsen der Disziplinen Maschinenbau, Elektrotechnik und Informationstechnik. Dabei entstehen integrierte mechanisch-elektronische Systeme. Das Gebiet der integrierten mechanisch-elektronischen Systeme wird mit dem Begriff Mechatronik bezeichnet [Iser99]. Mechatronische Systeme<sup>4</sup> zeichnen sich gegenüber konventionellen technischen Systemen vor allem durch eine höhere Flexibilität aus. Diese höhere Flexibilität erlangen sie durch die Fähigkeiten, ihre Umgebung wahrzunehmen (durch Sensorik) und eine Verhaltensanpassung als Reaktion auf diese Umgebung zu berechnen (durch Informationstechnik). Das mechatronische System ist also in der Lage, seine Funktionen an veränderte Umgebungen anzupassen.

Im Rahmen der Selbstoptimierung [FGK+04] kann das mechatronische System sogar die Ziele, die es verfolgt, also die Gewichtung oder Priorisierung der unterschiedlichen Qualitätsdimension, anpassen. Die höhere Flexibilität der mechatronischen Systeme führt in der Literatur häufig zu der Bezeichnung „intelligentes technisches System“ (z. B. [Rodd02]). Dabei bleibt jedoch weitestgehend unklar, was mit „intelligent“ gemeint ist. Im Bereich der Künstlichen Intelligenz hat sich neben anderen Kriterien das des rationalen Handelns<sup>5</sup> etabliert. Russel und Norvig [RN03] bezeichnen ein System als rationalen Agenten<sup>6</sup>, wenn es den besten Ausgang erreicht oder seine Handlungen am besten erwarteten Ausgang ausrichtet<sup>7</sup>, falls Handlungen mit Unsicherheit verbunden sind. Um ein rationales Verhalten umsetzen zu können, muss das (intelligente) System einige Fähigkeiten und Eigenschaften besitzen. Dazu gehören:

- die Fähigkeit, Wissen zu repräsentieren und zu speichern,
- die Fähigkeit, das Wissen zu verarbeiten,
- die Fähigkeit zu lernen und
- die Fähigkeit zur Wahrnehmung.

---

<sup>4</sup> Mechatronische Systeme sind demnach das Ergebnis der Integration mechanischer und elektronischer Systeme ergänzt um eine dazugehörige Informationsverarbeitung. Isermann [Iser99] führt zusätzlich den Begriff *intelligentes mechatronisches System* ein, das über zusätzliche Fähigkeiten wie Adaption oder Lernen verfügt. Im Rahmen dieser Arbeit soll zusätzlich das Kriterium der Rationalität als notwendige Bedingungen herangezogen werden.

<sup>5</sup> Russel und Norvig [RN03, S 2-5] nennen außerdem *menschlich handeln*, *menschlich denken*, *rational denken*. *Rational handeln* eignet sich insbesondere deshalb, da es im Gegensatz zu den Kriterien *menschlich handeln* und *menschlich denken* überprüfbar ist und im Rahmen der Verbesserung technischer Systeme das Ergebnis (das Handeln) und nicht der kognitive Prozess (das Denken) im Vordergrund stehen soll.

<sup>6</sup> Agent stammt von lateinische *agere* – tun und bezeichnet daher an dieser Stelle lediglich ein System, das etwas tut.

<sup>7</sup> Hier muss beachtet werden, dass dieses Optimalitätskriterium nur für unbeschränkte Rationalität zutrifft. Das bedeutet u.A., dass vollständige Informationen und unbegrenzte Zeit zum Nachdenken bereitstehen. Sind diese idealen Bedingungen nicht erfüllt, spricht man von beschränkter Rationalität. Dann ist nur ein „gutes“ Ergebnis zu fordern.

Heutige mechatronische Systeme verfügen durch den Einsatz von Sensorik und Informationsverarbeitung grundsätzlich über diese Fähigkeiten. So sind regelbasierte Umschaltungen oder Lern- oder Adaptionalgorithmen häufig zu finden. Jedoch sind die sogenannten intelligenten technischen Systeme immer noch *nur* geregelte Systeme, die ihre Reglerparameter an wechselnde Umfeldeinflüsse anpassen oder sogar nur nach bestimmten Regeln operieren, die im Entwurf definiert wurden. Mit dieser Art der Wissens- oder Informationsverarbeitung handelt es sich bei den heutigen mechatronischen Systemen um rein reaktive Systeme, die auf Veränderungen in der Umwelt und ihrem eigenen Zustand nur reagieren können. Ein optimales und damit rationales Verhalten ist so über einen längeren Betrachtungszeitraum nur unter sehr einfachen Bedingungen zu gewährleisten.

Um dagegen das rein reaktive Verhalten zu überwinden, neue Nutzenpotenziale zu realisieren und der Vision von intelligenten technischen Systemen näherzukommen, muss die Wissensverarbeitung auch auf zukünftige Handlungen, Zustände und Einflüsse erweitert werden. Dieser kognitive Prozess der Vorwegnahme zukünftiger Handlungen und Situationen wird im Allgemeinen als Planung bezeichnet: *„Planung kann als gedankliche Vorwegnahme künftigen Geschehens durch systematische Entscheidungsvorbereitung und Entscheidungsfällung verstanden werden. Sie beinhaltet einen Entscheidungsprozess, in dem zur Lösung eines Problems zielorientiert Alternativen zu suchen und beurteilen und auszuwählen sind. Dies geschieht unter Zugrundelegung einer monistischen oder pluralistischen Zielfunktion auf Basis einwertiger oder mehrwertiger Erwartung“* [Hahn96]. Anhand der Definition ist erkennbar, dass Planung einen Beitrag zum Erreichen eines rationalen Verhaltens liefern kann. In Bezug auf mechatronische Systeme bedeutet Planung vor allem die Vorwegnahme der Entscheidung über das zukünftige Verhalten, sprich die Ausführung von Funktionen. Im Gegensatz zu konventionellen mechatronischen Systemen erfolgt diese Entscheidung nicht bereits während der Entwicklung, sondern wird in den Betrieb des Systems verlagert. Daraus resultieren ein stärkerer Situationsbezug und damit Verbesserungen der Funktionserfüllung. Nur durch die Vorwegnahme des künftigen Geschehens und das Durchspielen und die Analyse unterschiedlicher Handlungsalternativen kann für einen gegebenen Auftrag<sup>8</sup> die optimale Abfolge von Funktionsausführungen bestimmt werden.

Voraussetzung für das Durchspielen und die Analyse unterschiedlicher Handlungsalternativen ist ein geeignetes Modell des Systemverhaltens. Es existiert eine Vielzahl an Modellierungstechniken für den Einsatz im Entwurf technischer Systeme<sup>9</sup>. Dabei lassen sich vor allem Ansätze, die kontinuierliche Funktionen darstellen (z. B. Blockdiagramme [Föl94] oder Bondgraphen [Tho90]), und Ansätze auf Basis von Zustandsautomaten (z. B. SDL, Abstrakte Datenmodellierung [CCI89, CCI93], die Mechatronic UML [BTG04] und die SysML [HV05]) unterscheiden. Diese Modelle sind aber nicht für die Verarbeitung in einem Planungsalgorithmus geeignet, da sie keine Information über mögliche und sinnvolle Abfolgen von Teilfunktionen enthalten.

Bei der Konzeption und Umsetzung einer Verhaltensplanung für mechatronische Systeme, die festlegt, wann und wie ein mechatronisches System im Rahmen seiner Aufgabenerfüllung welche Funktionen ausführt, sind einige weitere Aspekte zu beachten.

Zunächst ist die Vorwegnahme zukünftigen Geschehens grundsätzlich mit einer gewissen Unsicherheit verbunden. Dies gilt für mechatronische Systeme in besonderem Maße, da das Umfeld entscheidenden Einfluss auf die Wirkung von Aktivitäten besitzt. Diese Unsicherheit ist im Rahmen der Planung zu berücksichtigen. Dies kann z. B. durch mehrere alternative

---

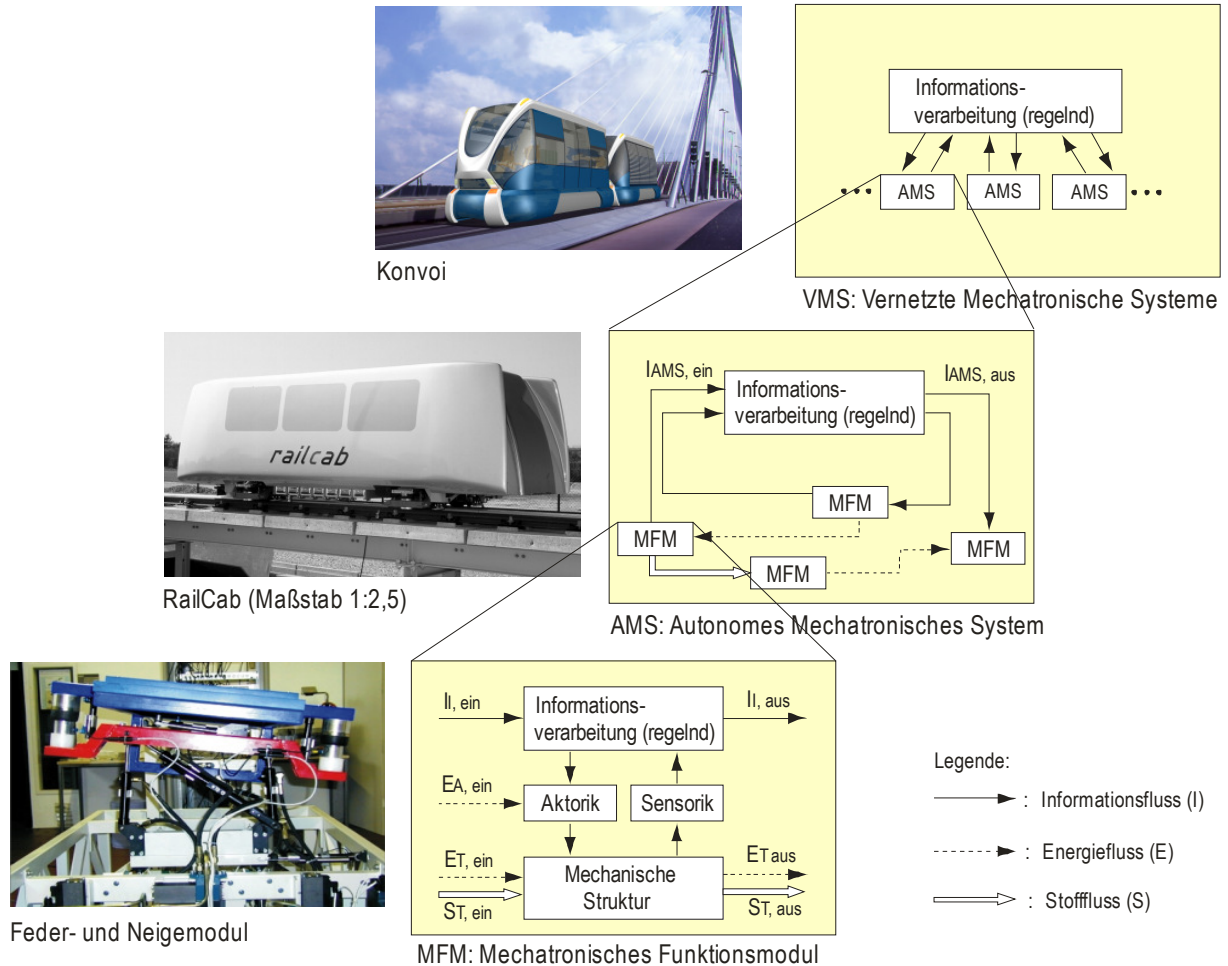
<sup>8</sup> Und damit für einen klar begrenzten zeitlichen Horizont. Unbeschränkte Horizonte werfen bzgl. des Kriteriums „Optimalität“ neue Probleme auf, die nicht Gegenstand dieser Arbeit sind.

<sup>9</sup> Eine ausführlichere Würdigung dieser Ansätze ist in Anhang A zu finden.



Planverläufe geschehen oder indem im Sinne des rationalen Verhaltens der Erwartungswert optimiert wird.

Mit der Mechatronik geht ebenfalls die Tendenz einher, das technische System nicht mehr einzeln und isoliert zu betrachten, sondern zusätzlich mögliche Vernetzungen und Interaktionen zwischen technischen Systemen zu berücksichtigen.



**Abbildung 1-2: Gliederung eines komplexen mechatronischen Systems [LHL+01]**

Dies schlägt sich auch in einem Strukturierungsvorschlag für mechatronische Systeme nieder (vergl. Abbildung 1-2). Auf der untersten Ebene dieses hierarchisch organisierten Strukturierungsvorschlags finden sich die mechatronischen Funktionsmodule (MFM), die die grundlegende Struktur mechatronischer Systeme sind. Basis ist immer eine mechanische Struktur, auf die eine Aktorik wirkt und deren Zustand über eine Sensorik wahrgenommen werden kann. Die Sensorik stellt Messungen und Informationen bereit, die in einer übergelagerten Informationsverarbeitung analysiert werden. Als Ergebnis wirkt die Informationsverarbeitung über die Aktorik auf die mechanische Struktur ein. Auf der nächsten Hierarchieebene fasst ein sogenanntes autonomes mechatronisches System (AMS) mehrere MFM zusammen. Zwischen den MFM und der Informationsverarbeitung des AMS bestehen Informationsflüsse. Schließlich können mehrere AMS zu einem Vernetzten mechatronischen System (VMS) zusammengefasst werden. Es lässt sich anhand dieser Gliederung erkennen, dass vielfältige Beziehungen zwischen den einzelnen Systemen in Form von Informations-, Stoff- und Energieflüssen bestehen. Mechatronische Systeme interagieren also miteinander und beeinflussen sich gegenseitig. Dies gilt zunächst innerhalb des autonomen mechatronischen Systems für die Funktionsmodule. Hier gilt es im Rahmen der Verhaltensplanung das Verhalten der einzelnen mechatronischen Systeme so zu koordinieren, dass ein optimales oder zumindest gutes Verhalten

des Gesamtsystems resultiert. Anders ist der Fall auf Ebene der vernetzten mechatronischen Systeme gelagert. Da jedes System grundsätzlich einem anderen Betreiber gehören kann, ist es hier nicht möglich, ein gemeinsames Zielsystem zu bestimmen. Dennoch können auch die autonomen mechatronischen Systeme im vernetzten mechatronischen System so interagieren, dass sie ihre jeweilige Aufgabe besser erfüllen (z. B. energieeffizienter durch Konvoifahrt).

Um der Vision intelligenter technischer Systeme näherzukommen, stellt damit die Planung des Verhaltens einen notwendigen Baustein dar. Unter Berücksichtigung der besonderen Eigenschaften des Verhaltens mechatronischer Systeme soll daher in dieser Arbeit eine Verhaltensplanung für mechatronische Systeme konzipiert werden, die miteinander interagieren und in nicht-deterministischen Umgebungen handeln.

## 2 Problemstellung

Untersuchungsgegenstand dieser Arbeit ist die Planung des Verhaltens von intelligenten mechatronischen Erzeugnissen, die mit komplexen und dynamischen Umgebungen vernetzt sind. Im folgenden Kapitel soll definiert werden, was unter einer Planung des Verhaltens oder eben Verhaltensplanung zu verstehen ist. Weiterhin soll auf wesentliche Aspekte, die die Konzeption einer solchen Planung beachten muss, näher eingegangen werden.

Um zeigen zu können, wie die Planung im Betrieb mechatronischer Systeme eingesetzt werden kann und welche besonderen Herausforderungen dabei beachtet werden müssen, wird an dieser Stelle ein stark vereinfachtes Modell des Systemverhaltens aufgebaut. Das Modell abstrahiert dabei zunächst von den physikalischen Wirkungen innerhalb eines mechatronischen Systems und in der Wechselwirkung mit seiner Umgebung. Nach Pahl und Beitz [PB03] kann die Gesamtfunktion und damit das Ein- und Ausgabeverhalten eines technischen Systems als Abfolge von Teilfunktionen dargestellt werden. Im konkreten technischen System wird eine Teilfunktion durch eine oder mehrere Lösungen realisiert. Bezeichne:

- $TF$  die Menge der Teilfunktionen eines mechatronischen Systems,
- $L_{tf}$  die Menge der Lösungen<sup>10</sup> von Teilfunktion  $tf \in TF$ , die das mechatronische System realisiert und
- $L = \bigcup_{tf \in TF} L_{tf}$  die Menge der Teilfunktionslösungen, die das mechatronische System realisiert.

Dabei kann  $tf$  als eine Funktion betrachtet werden, die Eingangsgrößen auf Ausgangsgrößen abbildet. Das Ein- und Ausgabeverhalten des Gesamtsystems lässt sich als Tupel aus Eingabewerten aus dem Umfeld, einer Folge von Teilfunktionslösungen und Ausgabewerten beschreiben. Die Gesamtfunktion wiederum ist die Menge aller Folgen von Ein- und Ausgabewerten, die der Aufgabe des technischen Systems entsprechen.

Die Folge der Teilfunktionslösungen ist so zu konstruieren, dass die notwendigen Eingaben für alle enthaltenen Lösungen vor ihrer jeweiligen Anwendung bereitgestellt werden.

Beschränkt man die Betrachtung nicht auf das reine Ein- und Ausgabeverhalten, sondern bezieht den Zustand des Systems und des Umfelds mit ein, ergibt sich eine Darstellung wechselseitiger, zeitlich verzögerter Einflüsse wie in Abbildung 2-1.

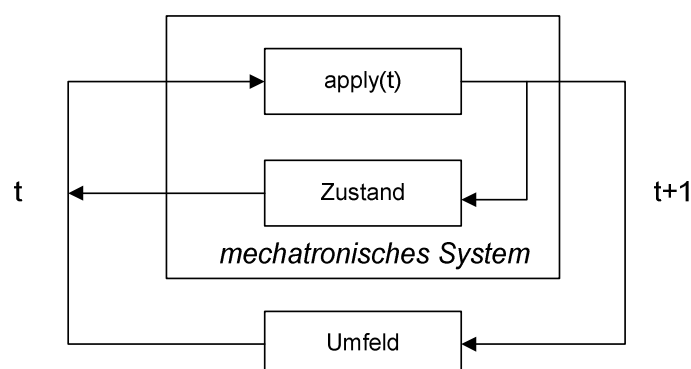


Abbildung 2-1: Verknüpfung eines mechatronischen Systems mit seinem Umfeld

<sup>10</sup> Dabei werden zwei Lösungen, die technisch identisch sind bis auf die Wahl von Parametern, als zwei unterschiedliche Lösungen betrachtet. Beispiel dafür ist das Fahren mit hoher und geringer Geschwindigkeit oder Wahl unterschiedlicher Reglerparameter.

Dabei steht  $apply(t)$  für die zum Zeitpunkt  $t$  angewandten Teilfunktionslösungen. Ferner seien definiert:

- $M$ , die Menge der möglichen Zustände des Systems
- $E$ , die Menge der möglichen Zustände des Umfelds
- $I$ , die Menge der möglichen Eingabewerte
- $O$ , die Menge der Ausgabewerte
- $affect: P(L) \times M \rightarrow O$  <sup>11</sup>eine Funktion, die angibt, welche Ausgabewerte bei gegebenem Zustand und einer Menge angewandter Teilfunktionslösungen erzeugt werden
- $influence: O \times E \rightarrow I$  sind die Einflüsse auf die Ausgaben des mechatronischen Systems
- $M \times E = S$  bezeichnet die Menge möglicher Gesamtzustände des Modells.

Um das dynamische Verhalten eines mechatronischen Systems zu beschreiben, ist die Einführung einer Zeitmenge notwendig. Die übrigen Größen müssen abhängig von der Zeit bestimmbar sein:

- Sei  $T = (\mathbb{N}, \leq, 0)$  die dem Modell zugrunde liegende Zeitmenge<sup>12</sup>
- Bestimme  $i_t$  die Eingabewerte zum Zeitpunkt  $t$
- Bestimme  $o_t$  die Ausgabewerte zum Zeitpunkt  $t$
- Bestimme  $e_t$  den Zustand der Umgebung zum Zeitpunkt  $t$
- Bestimme  $m_t$  den Zustand des mechatronischen Systems zum Zeitpunkt  $t$
- $apply(t)$  bezeichnet die zum Zeitpunkt  $t$  ausgeführten Teilfunktionslösungen.

Die jeweiligen Werte lassen sich durch Funktionen darstellen, die die Gesetzmäßigkeiten innerhalb des Systems und des Umfelds widerspiegeln:

- $m_t = evolve_m(i_{t-1}, o_{t-1}, m_{t-1}), evolve_m: I \times O \times M \rightarrow M$
- $e_t = evolve_e(o_{t-1}, e_{t-1}), evolve_e: O \times E \rightarrow E$
- $o_t = affect(apply(t), m_t), affect: P(L) \times M \rightarrow O$
- $i_t = influence(e_t, o_t), influence: O \times E \rightarrow I$

Das Verhalten des Systems und des Umfelds kann nun beschrieben werden, indem eine Evolutionsfunktion<sup>13</sup> der beiden Zustandsvariablen  $m_t$  und  $e_t$  erstellt wird:

- $m_t = evolve_m(influence(e_{t-1}, o_{t-1}), affect(apply(t-1)m_{t-1}), m_{t-1})$   
 $= evolve_m(influence(evolve_e(o_{t-2}, e_{t-2}), affect(apply(t-1), m_{t-1})),$   
 $affect(apply(t-1)m_{t-1})m_{t-1})$
- $e_t = evolve_e(affect(apply(t-1), e_{t-1}))$

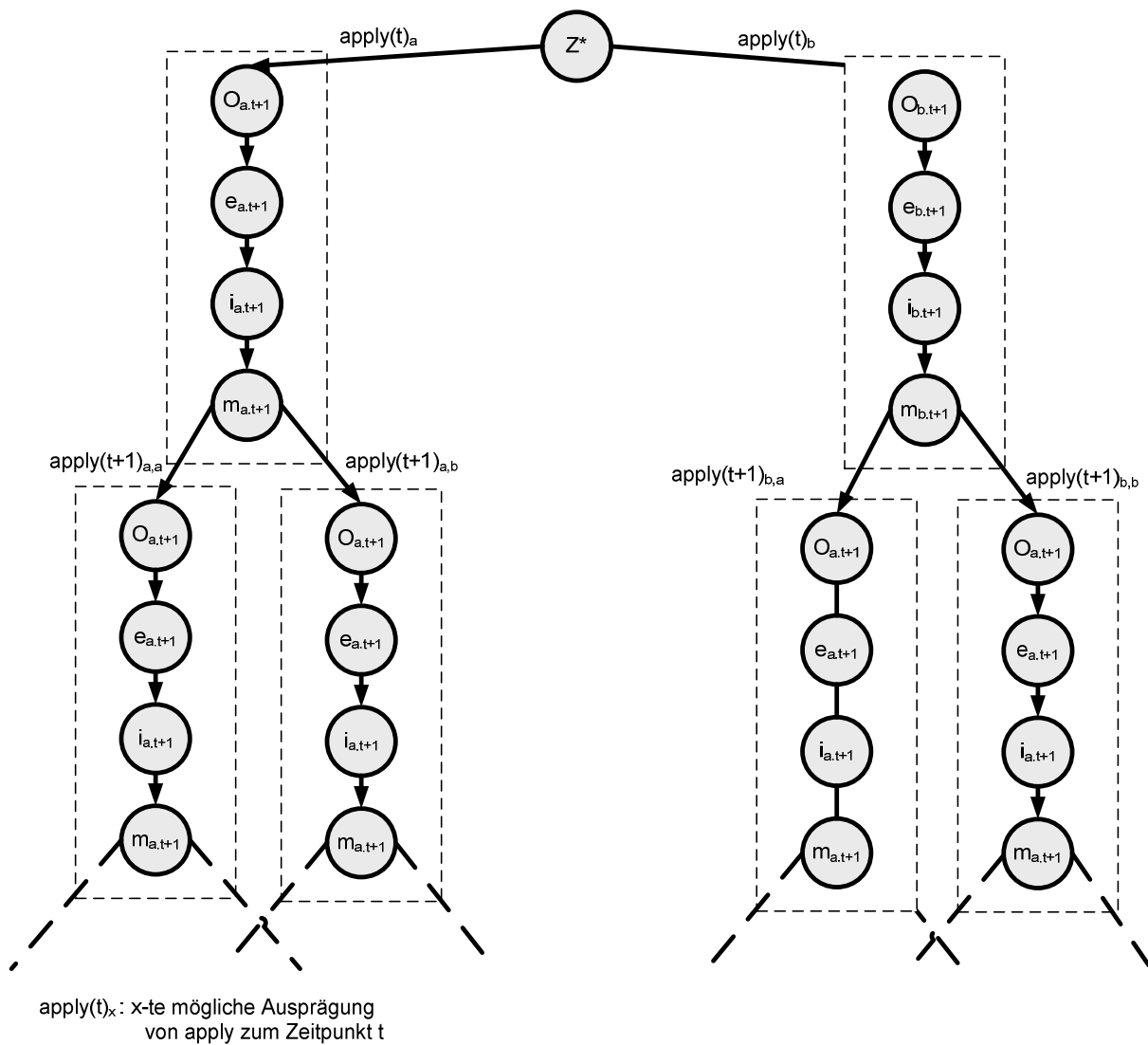
Aus diesen beiden Evolutionsfunktionen wird ersichtlich, dass im Modell die Zustände von Umfeld und mechatronischem System bei gegebenen Anfangszuständen zum Zeitpunkt  $t_0$  nur von den Funktionswerten von  $apply(t)$  abhängen. Das Verhalten eines mechatronischen Systems lässt sich nunmehr als Folge von Zuständen und Anwendungen von Teilfunktionslösungen

<sup>11</sup>  $\mathcal{P}$  bezeichnet dabei die Potenzmenge.

<sup>12</sup> Eine Zeitmenge ist ein Tripel  $(T, \prec, t_0)$ , wobei  $T$  eine Menge,  $\prec$  eine Vollständige Ordnungsrelation über  $T$  und  $T_0$  das minimale Element bzgl. der Ordnungsrelation ist. Hier wird vereinfachend unterstellt, dass die Menge  $T$  die Menge der natürlichen Zahlen ist.

<sup>13</sup> Die Definition einer Evolutionsfunktion geschieht in Anlehnung an Ferber [Ferb01].

gen interpretieren. Um ein Verhalten zu erzielen, das die Hauptaufgabe des Systems erfüllt, gilt es also, die Funktionswerte von  $apply(t)$  festzulegen. Im Falle klassischer technischer Erzeugnisse geschieht dies während des Entwurfs, also lange vor der Inbetriebnahme. Es werden Funktionsketten aufgebaut [PB93]. Die durch die Funktionskette gefundene Lösung wird wiederum bzgl. der Merkmale Qualität und Quantität der Ein- und Ausgabewerte beurteilt. Da die Festlegung der Funktionsketten – wenn auch teilweise mit Varianten – bereits zur Entwurfszeit erfolgt, stellen die Funktionsketten einen Kompromiss dar, da sie für die gesamte Bandbreite an Eingabewerten die Aufgabe erfüllen. Dadurch rückt die Beachtung der qualitativen und quantitativen Merkmale einer Funktionsweise unter Umständen in den Hintergrund. Die Festlegung der Funktionswerte von  $apply(t)$  kann aber auch als Planungsproblem im Sinne der Künstlichen Intelligenz aufgefasst werden.



**Abbildung 2-2: Planungsproblem für mechatronische Systeme als Baum**

Abbildung 2-2 stellt diese Sichtweise dar: Ausgehend vom gegebenen Systemzustand, einem Umfeldzustand (beide zusammengefasst in  $Z^*$ ) und den daraus resultierenden initialen Eingabewerten ist eine Sequenz von Entscheidungen über die Anwendung bestimmter Teilfunktionslösungen zu fällen. Dadurch spannt sich ein Baum auf, in dem die Entscheidungen, die bezüglich der Funktionswerte von  $apply(t)$  getroffen werden, alternative Zustände und Ein- und Ausgabefolgen erzeugen. Die Planungsaufgabe ist die Identifikation eines Pfades inner-

halb dieses Baums, der das gewünschte Ein- und Ausgabeverhalten liefert. Der Vorteil dieses Vorgehens besteht in seiner größeren Flexibilität und seinem stärkeren Situationsbezug. Beim klassischen Vorgehen wird diese Sequenz für alle denkbaren Ausgangssituationen und ausschließlich für eine einzige Zielsetzung festgelegt. Dagegen besteht bei Anwendung der Planung die Möglichkeit, die Entscheidungen für die konkrete Betriebsituation und wechselnde Zielsysteme zu suchen. Dadurch wird zweierlei erreicht:

1. Für Situationen, in denen Funktionsketten mit einer Auslegung auf eine breite Anwendbarkeit die Zweckwirkung nicht erzielen, kann u. U. eine situationspezifische Funktionskette gefunden werden.
2. Die Bewertung der gefundenen Funktionsketten kann ebenfalls von der konkreten Situation und den Präferenzen des jeweiligen Benutzers abhängig gemacht werden.

Für Probleme, die sich auf eine baumartige Datenstruktur abbilden lassen, bietet sich der „*branching factor*“ oder *Verzweigungsgrad* an, um die Komplexität eines Problems abzuschätzen [RN03]. Dieser Faktor gibt an, wie viele Nachfolger ein Knoten minimal, maximal oder durchschnittlich besitzt.

An dieser Stelle lässt sich der Hauptbetrachtungsgegenstand dieser Arbeit definieren, die Planungsaufgabe für den Betrieb mechatronischer Systeme. Abbildung 2-3 zeigt die Struktur der Planungsaufgabe für mechatronische Systeme in Anlehnung an Ferstl und Sinz [FS06]. Die Vor- und Nachereignisse der Planungsaufgabe sind klar umrissen. Die Planung wird ausgelöst durch eine konkrete Aufgabenstellung, die durch das mechatronische System zu erfüllen ist. Zusätzlich liegt der aktuell wahrgenommene System- und Umfeldzustand vor. Das Nachereignis der Planungsaufgabe besteht im Sinne des gedanklichen Modells in einer Festlegung der Funktion *apply(t)*. Formale Aufgabenziele sind mit der Planungsdauer, dem Speicherbedarf des Lösungsverfahrens und der Lösungsqualität angegeben.

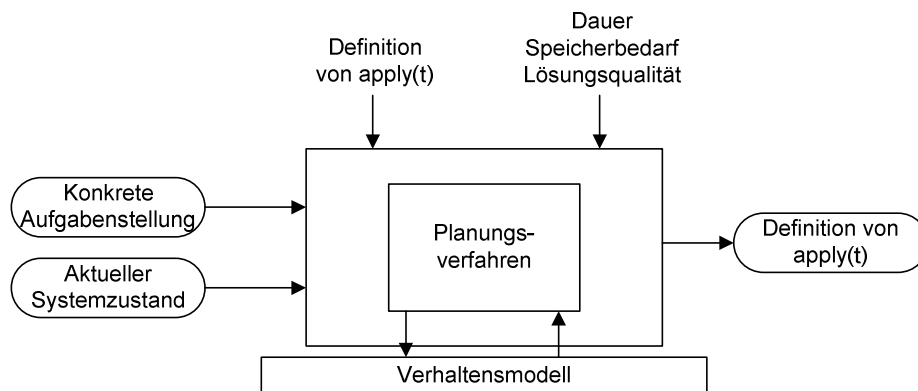


Abbildung 2-3: Struktur der Planungsaufgabe für mechatronische Systeme

Ein Planungsverfahren (also das Lösungsverfahren für eine Planungsaufgabe) arbeitet immer auf einem Modell des zu planenden Systems, da die zukünftige Entwicklung des Systems festgelegt werden soll, was nicht direkt am System geschehen kann. Im nächsten Teilkapitel wird sowohl ein Verhaltensmodell als auch eine Klasse an Lösungsverfahren eingeführt, die in der Lage sind, das eben definierte Problem der Verhaltensplanung zu lösen.

## 2.1 Abbildung der Verhaltensplanung auf die Planung in der KI

Die Künstliche Intelligenz fasst das Planungsproblem als die Suche nach einer Sequenz von Aktivitäten auf, die von einem Ausgangszustand in einen Zielzustand führt. Daher überdeckt sich diese Art von Planungsproblem mit dem Problem der Verhaltensplanung für mechatroni-

sche Systeme. Ihren Ursprung hat diese Teildisziplin der Künstlichen Intelligenz in den späten Siebzigerjahren des letzten Jahrhunderts. Fikes und Nilson stellten einen Planungsalgorithmus mitsamt formalem Modell zur Beschreibung von Planungsproblemen unter dem Namen STRIPS (Stanford Research Institute Problem Solver) [FN71] vor.

In Abgrenzung zum verwandten Problemlösen unterscheidet sich die Planung vor allem in ihrer Skalierung. Diese Skalierung wird vor allem durch die Ausnutzung von Informationen über das Planungsproblem erreicht. Voraussetzung dafür ist die ausdrucksstarke Modellierung des Planungsproblems [RN03, S.375]. Seit frühen Planungssystemen wie GPS oder eben STRIPS sind die meisten KI-Planungssysteme entwickelt worden unter den gleichen weitgehenden Annahmen [Blyt99]:

- Die Umgebung ist vollständig beobachtbar,
- die Umgebung ist deterministisch,
- die Umgebung ist statisch (das geplante System ist die einzige Quelle für Veränderung) und
- die Umgebung ist diskret (hinsichtlich Zeit, Aktivitäten, Objekten und Wirkungen).

Eine weitere Annahme ist die „*Closed World Assumption*“. Diese Annahme ist durch den Modellierungsaufwand motiviert und geht davon aus, dass jede Aussage über die Ausführungsumgebung, die nicht ausdrücklich als *wahr* gekennzeichnet ist, *falsch* ist.

### 2.1.1 Modellierung mithilfe der PDDL

Das Planungsproblem der Künstlichen Intelligenz ist sehr allgemein formuliert. Daher können entsprechende Lösungsverfahren<sup>14</sup> für eine Vielzahl von Planungsproblemen verwendet werden. Dazu ist es allerdings erforderlich, die Umgebung des Planers<sup>15</sup>, seine Fähigkeiten und die konkrete Planungsaufgabe in einer formalen Sprache zu beschreiben, die der Planer verarbeiten kann. Für die Modellierungsmethode, die zur Darstellung des Verhaltensplanungsproblems eingesetzt wird, sollten ausgereifte Lösungsverfahren bereitstehen. Die Planning Domain Description Language (PDDL) erfüllt diese Anforderung am ehesten. Mit Planungssystemen wie z. B. SGPlan<sub>5</sub> [HWHC06] stehen effiziente Verfahren zur Verfügung, die laufend weiterentwickelt werden. Dabei ist allerdings zu beachten, dass je nach gegebener Problemstellung andere Modellierungs- und Lösungsansätze zu einer schnelleren und speichereffizienteren Planung führen können. Die PDDL und die damit verbundene Planungsphilosophie werden vor allem wegen ihrer generischen Anwendbarkeit ausgewählt. Dies ermöglicht es, die Umsetzung der Verhaltensplanung auf Basis eines allgemeinen Modells und daher unabhängig von einer konkreten Anwendung zu beschreiben. Die in den folgenden Kapiteln entwickelten Konzepte sind auf andere Modellierungs- und Lösungskonzepte übertragbar. Im Folgenden wird der Stand der PDDL 2.1 [FL03] genutzt, da in der Fassung PDDL 3.0 [GL05] nur für die Aufgabenstellung unwesentliche Änderungen vorgenommen wurden.

PDDL ist für die Beschreibung dessen konzipiert, was ihre Entwickler die „*Physik*“ einer Domäne nennen. Die *Physik* ist dabei beschränkt auf die Beschreibung von Aktionen, insbesondere wann welche Aktionen ausführbar sind, woraus Aktionen zusammengesetzt sind und was ihre Wirkungen sind. Ein wesentliches Merkmal der PDDL besteht in der Trennung von Domänenbeschreibung und konkreter Problemstellung. Erstere beschreibt, welche Typen in einer Domäne auftreten können, welche Beziehungen zwischen Objekten bestimmter Typen

---

<sup>14</sup> Siehe Anhang B.

<sup>15</sup> Korrekt müsste es heißen: des planausführenden Systems, da der Planer und das ausführende System zwei unterschiedliche Systeme in zwei unterschiedlichen Umgebungen sein können. Einfachheitshalber wird diese Unterscheidung im weiteren Verlauf aber vernachlässigt.

auftreten können und welche numerischen Eigenschaften Typen und Beziehungen haben. Formal ist die Domäne eine Vier-Tupel  $Dom = (Fs, Rs, As, arity)$ . Dabei sind im Einzelnen:

- $Fs$ , eine endliche Menge von Funktionssymbolen
- $Rs$ , eine endliche Menge von Relationensymbolen
- $As$ , eine endliche Menge von Aktionsschemata
- $arity$ , eine Funktion, die allen zuvor genannten Symbolen ihre Stelligkeit<sup>16</sup> zuordnet.

Die Stelligkeit enthält dabei nicht nur Informationen über die Anzahl der Argumente, sondern auch über ihren Typ. Eine konkrete Planungsdomäne gibt zusätzlich die möglichen Typen explizit an (diese Information ist in der formalen Definition nur implizit durch  $arity$  gegeben). Ein konkretes Planungsproblem  $I$  ist ein 3-Tupel  $Prob_I = (Os, Init, Gs)$  bestehend aus:

- $Os$ , der endlichen Menge der Objekte des Planungsproblems
- $Init$ , der Spezifikation des Ausgangszustands des Problems
- $Gs$ , der Spezifikation der Zielzustände des Problems.

Zustände innerhalb eines PDDL-Plans setzen sich aus einer Menge einfacher numerischer Ausdrücke und Atomen zusammen. Die einfachen numerischen Ausdrücke (primitive numeric expression, PNE) sind all diejenigen Ausdrücke, die sich durch Belegung der Funktionssymbole  $Fs$  der Domäne mit den Objekten der Problem Instanz erzeugen lassen. Die Anzahl der unterschiedlichen erzeugbaren Funktionen ergibt die Dimension  $dim$  des Planungsproblems. Der numerische Teil eines Zustands ist damit bestimmt durch einen  $dim$ -stelligen Vektor reeller Zahlen. Der logische Teil eines Zustands ist bestimmt durch eine Teilmenge der Atome des Planungsproblems  $I$   $Atms_i$ . Die Atome  $Atms_i$  ergeben sich dabei durch Anwendung der Relationensymbole auf die Objekte der Problem Instanz.

Pläne bestehen in der PDDL aus Grundaktionen. Zu jedem Aktionsschema  $A \in As$  existiert eine Menge von Grundaktionen  $GA_A$ . Das ist die Menge aller Strukturen, die durch Substitution der Schemavariablen mit Objekten des Planungsproblems erzeugt werden können. Die Komponenten einer Grundaktion  $a$  sind:

- Der Name des Aktionsschemas
- $Pre_a$ , die Menge der propositionalen Vorbedingungen von  $a$  (Die Menge der Atome in  $Pre_a$  wird als  $GPre_a$  bezeichnet)
- $Add_a$ , die Menge der Atome, die sicher als positive Literale Effekt von  $a$  sind
- $Del_a$ , die Menge der Atome, die sicher als negative Literale Effekt von  $a$  sind
- $NP_a$ , die Menge aller Zuordnungspropositionen in den numerischen Effekten von  $a$ .

Dabei ist zu beachten, dass die Vorbedingungen der Aktionen sowohl numerische als auch logische Bestandteile haben. Im Gegensatz zu den Effekten können logische und numerische Bestandteile aber voneinander abhängig sein, beispielsweise durch eine Konjunktion.

Während die Funktionsweise der logischen Effekte  $Add_a$  und  $Del_a$  einfach durch *Wahr-* oder *Falsch-*Setzen der entsprechenden Relationen definiert ist, bedürfen die numerischen Effekte einer genaueren Betrachtung. Ein numerischer Effekt besteht in der Zuordnung eines neuen Werts im numerischen Zustandsvektor. Daher bestehen numerische Effekte aus einem Zuord-

---

<sup>16</sup> Stelligkeit (auch Arität) steht für die Anzahl der Argumente einer Abbildung oder eines Operators.



nungsoperator (assign, increase, decrease, scale-up, scale-down)<sup>17</sup>, einem einfachen numerischen Ausdruck lvalue und einem numerischen Ausdruck rvalue. Der rvalue ist dabei ein arithmetischer Ausdruck, dessen Terme Zahlen und einfache numerische Ausdrücke sind. Der lvalue ist der zu aktualisierende numerische Ausdruck und der rvalue bestimmt den Wert der Zuordnungsoperation. Weiterhin werden innerhalb der numerischen Effekte die Vor- und Nachbedingungswerte gesondert kenntlich gemacht. Statt also einen Effekt (*increase p q*) als eine Funktion  $f(p)=p+q$  zu formulieren, wird die Zuweisung in der Form  $(= p' (+ p q))$  formuliert. Dadurch sind die Werte vor und nach Ausführung der Aktion eindeutig zu unterscheiden.

Zusammenfassend lässt sich das deterministische Planungsproblem, das mithilfe der PDDL formuliert werden kann, wie in Abbildung 2-4 darstellen. Der Planungsprozess ist ein Schließen über Zustände, Aktionen und Effekte.

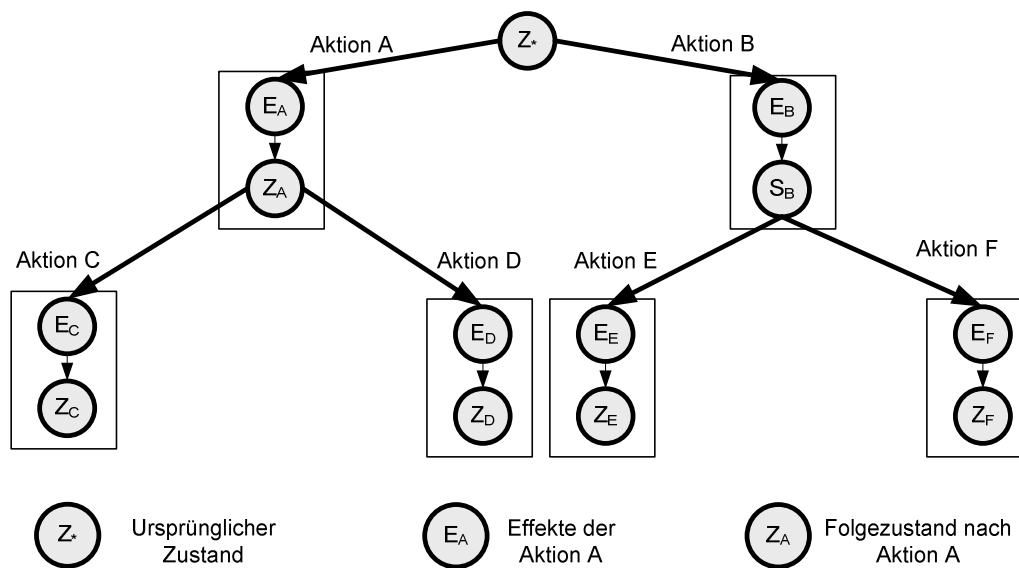


Abbildung 2-4: Das PDDL-Planungsproblem als Planbaum

Ein weiteres wesentliches Sprachelement der PDDL ist die Planmetrik. Eine Planmetrik ist ein arithmetischer Ausdruck, der aus Konstanten und numerischen Zustandsvariablen gebildet werden kann. Zusätzlich muss entschieden werden, ob der Ausdruck minimiert oder maximiert werden soll. Die arithmetischen Ausdrücke müssen nicht zwingend linear sein.

Bei einer Abbildung des eben definierten gedanklichen Modells des Verhaltens mechatronischer Systeme würden die Aktionen der PDDL-Planungsdomäne den Teilfunktionslösungen und die Domänenprädikate den möglichen System- und Umfeldzuständen entsprechen. Die Unterscheidung dieser beiden Zustände wäre allerdings formal nicht mehr erkennbar. Eine explizite Modellierung der möglichen Eingabe- und Ausgabewerte würde entfallen. Die Funktionen *affect* und *influence*, die die Dynamik des gedanklichen Modells beschreiben, würden durch die Effekte der Aktionen beschrieben. Ein gefundener Plan, der eine Sequenz von Aktionen beschreibt, würde der Festlegung der Funktion *apply(t)* entsprechen.

Zur Lösung eines in PDDL formulierten Planungsproblems steht eine ganze Reihe von Planungssystemen zur Verfügung. Aktuelle Beispiele sind SGPlan<sub>5</sub> [HWHC06], MaxPlan [XCZ06] und SatPlan<sub>5</sub> [Kaut06]. Dabei unterstützt SGPlan<sub>5</sub> die meisten Sprachmerkmale, während MaxPlan und SatPlan<sub>5</sub> nur solche Probleme verarbeiten, die sich gut mithilfe von

<sup>17</sup> Assign weist einen neuen Wert zu, increase bezeichnet eine Addition, decrease eine Subtraktion, scale-up eine Multiplikation und scale-down eine Division.

Aussagenlogik darstellen lassen, also insbesondere keine numerischen Eigenschaften beinhalten<sup>18</sup>.

### 2.1.2 Modellierung einer mechatronischen Domäne

Ausgehend von der vollständig spezifizierten Funktionshierarchie und einem Umfeldmodell muss für ein mechatronisches System eine PDDL-Domänendefinition erstellt werden. Dabei werden die relevanten Objekte der mechatronischen Domäne durch Typen repräsentiert. Beispiele für relevante Objekte sind einerseits die Teilsysteme des mechatronischen Systems und andererseits die Umgebungsobjekte, mit denen das mechatronische System interagiert (zum Beispiel Transportgüter, Verkehrswege, Werkstücke, etc.).

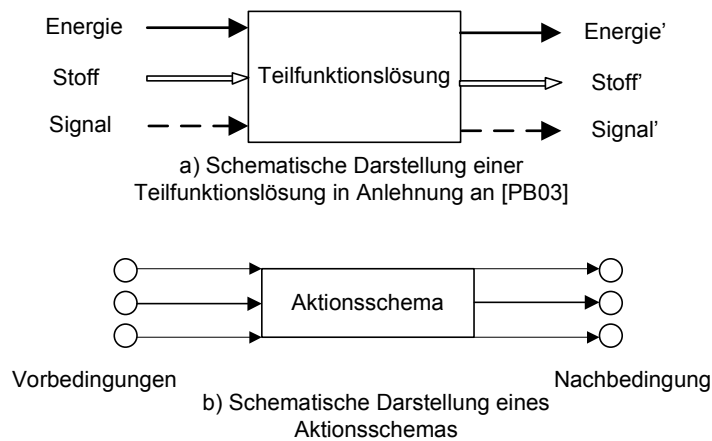
Jedes dieser Objekte besitzt eine Menge von Eigenschaften, die für den Planungsprozess von Bedeutung ist. Entweder zielt die Aufgabe des mechatronischen Systems auf die Manipulation dieser Eigenschaften ab (dies ist vor allem der Fall bei Umgebungsobjekten) oder sie stellen Randbedingungen für Aktivitäten des mechatronischen Systems dar (z. B. der Ladegrad eines Energiespeichers). Derartige Eigenschaften werden entweder über das Mittel der Prädikate (predicates) oder der Funktionen (functions) dargestellt. Die Prädikate dienen zur Beschreibung qualitativer Eigenschaften, deren Ausprägung sich durch Ja oder Nein ausdrücken lässt. Funktionen dienen der Darstellung der quantitativen Eigenschaften der Objekte. Dabei wird in der PDDL jeder numerische Wert grundsätzlich als reelle Zahl dargestellt – weicht eine quantitative Eigenschaft von diesem Wertebereich ab, nimmt also beispielsweise nur Werte aus dem Bereich der natürlichen Zahlen an, so ist dies implizit durch die Konstruktion der Aktionen zu gewährleisten.

Die Dynamik eines mechatronischen Planungsproblems wird wie in jeder PDDL-Domäne durch die Aktionen beschrieben. Dabei bleibt grundsätzlich die Annahme erhalten, dass die Aktionen die einzige Quelle von Veränderungen sind. Da das Verhalten eines mechatronischen Systems geplant wird, ergibt sich daraus die unrealistische Annahme, dass sich die Umgebung nur bei Beeinflussung durch das mechatronische System verändert. Da diese Annahme für mechatronische Systeme nicht realistisch ist, werfen sich neue Probleme auf, die in den Teilkapiteln 2.3 und 2.4 beschrieben werden.

Bereits in der Einleitung wurde dargelegt, dass das Verhalten mechatronischer Systeme üblicherweise in Form von Funktionen bzw. Teilfunktionen sowie Teilfunktionslösungen beschrieben wird. Dabei stellt eine Funktion oder Teilfunktion immer eine Verknüpfung von Eingangsgrößen auf Ausgangsgrößen dar und wird durch die Auswahl einer Teilfunktionslösung genau bestimmt. In der Einleitung wurde dargelegt, wie die Teilfunktionslösungen durch Aufstellen einer Funktionshierarchie hergeleitet werden. Die grundsätzlichen Fähigkeiten eines mechatronischen Systems sind daher definiert durch die Menge der Teilfunktionslösungen, die es implementiert. Im Rahmen der Modellierung einer mechatronischen Planungsdomäne sind die Teilfunktionslösungen daher auf Aktionen abzubilden. Dies geht mit einer Diskretisierung der Beschreibung von Teilfunktionslösungen und damit einem Genauigkeitsverlust einher. Abbildung 2-5 stellt die Begriffe der Teilfunktionslösung und Aktion gegenüber. Nach [PB03] beschreibt eine Teilfunktionslösung eine Funktion, die eingehende Energie-, Stoff- und Signalflüsse in ausgehende Flüsse umwandelt. Die Flüsse können dabei kontinuierlich in die Teilfunktionslösung ein- und austreten oder sie in einem diskreten, also in einzelne „Stücke“ zerlegbaren Fluss durchlaufen. Die Flüsse zerfallen also keinesfalls zeitlich in einen Fluss vor der Ausführung der Teilfunktion und einen nach Ausführung.

---

<sup>18</sup> Dies ist im betrachteten Anwendungsfall eigentlich nie der Fall, da die zu beschreibenden technischen Prozesse in der Regel in Zahlenwerten ausgedrückte Eigenschaften besitzen.



**Abbildung 2-5: Vergleich von Teilfunktionslösung und PDDL-Aktion**

Bei Aktionsschemata lassen sich die Vor- und Nachbedingungen der Aktionen unterscheiden. Der scheinbare Widerspruch zum Flussbegriff der Teilfunktionslösung, den die Begriffe Vor- und Nachbedingung andeuten, existiert dabei nicht. Die PDDL erlaubt es, durch die Definition von dauerhaften und kontinuierlichen Aktionen auch solche Vorbedingungen (Nachbedingungen) zu definieren, die während der Ausführung einer Aktion erfüllt sein müssen (erfüllt sind). Auch bei der Anwendung dieser Modellierungsmöglichkeit liegt der PDDL weiterhin ein diskretes Zeitmodell zugrunde, sodass sich kontinuierliche Flüsse nur näherungsweise darstellen lassen. Allerdings führt die Anwendung von dauerhaften und kontinuierlichen Aktionen zu einem erheblich größeren Zustandsraum und damit zu einer höheren Planungskomplexität. Ihr Einsatz ist daher genau zu überdenken.

In der Regel wird es auch möglich sein, eine Teilfunktionslösung in Form einer konventionellen, diskreten PDDL-Aktion zu beschreiben. Dazu ist es erforderlich, die Anwendung der Lösung zeitlich begrenzt zu betrachten. Im Rahmen dieser zeitlich begrenzten Betrachtung können die Ein- und Ausgabeflüsse summiert betrachtet werden. So wird vereinfacht unterstellt, dass alle notwendigen eingehenden Energien, Stoffe und Informationen vor der Anwendung einer Teilfunktionslösung bereitgestellt werden. Ebenso vereinfacht wird unterstellt, dass alle Ausgabeflüsse nach Abschluss der Teilfunktionslösung bereitgestellt werden. Durch diese Vereinfachung bildet das Planungsmodell das mechatronische System zwar nicht exakt ab, ist aber schneller lösbar.

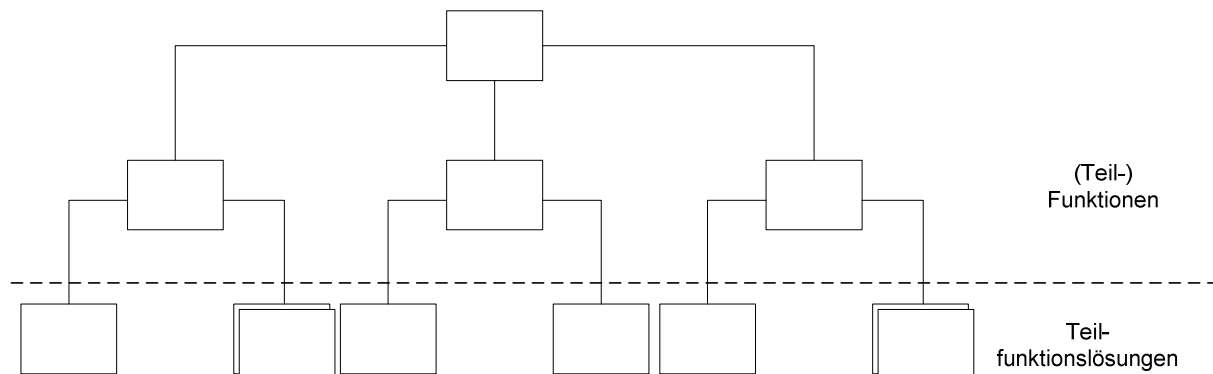
Eine weitere Modellierungsentscheidung, die erhebliche Auswirkungen auf die Komplexität des Planungsproblems hat, besteht in der Entscheidung, was als eine Teilfunktionslösung und damit als Aktion dargestellt wird. Abbildung 2-6 zeigt eine Funktionshierarchie mit Teilfunktionslösungen. Bei Teilfunktionslösungen handelt es sich eigentlich um die Blätter der Funktionshierarchie, da es sich um vollständig spezifizierte und sogar implementierte Funktionen handelt, die nicht weiter in Teilfunktionen zerlegt werden. Allerdings lässt sich jede Teilfunktion als eine Abfolge von Teilfunktionslösungen darstellen und damit in ein Blatt der Funktionshierarchie überführen. Dazu muss für jede als sinnvoll betrachtete Kombination der alternativen Teilfunktionen bzw. Teilfunktionslösungen, die Kind der Teilfunktion sind, eine neue Teilfunktionslösung erzeugt werden. Wie also die Funktionshierarchie in eine Menge von Aktionen in der Repräsentation der PDDL überführt wird, hängt von den Zielen der Planung und der notwendigen Planungsgeschwindigkeit ab. Je nach gewählter Modellierung werden bestimmte alternative Sequenzen von Teilfunktionslösungen ausgeschlossen. Weiterhin geht mit einer eher groben Modellierung (mit der Darstellung von Teilfunktionslösungen weit oben in der Hierarchie) Genauigkeit verloren, sodass der optimale Lösungsweg nicht mehr korrekt erkannt werden kann. Eine allgemeine Handlungsanweisung, wie eine mechatronische Pla-

nungsdomäne modelliert wird, kann hier nicht gegeben werden, da die Anforderungen der Modellierung zu stark von der konkreten Anwendung abhängen.

Grundsätzlich ist aber jede Teilfunktionslösung durch eine Aktion in der PDDL darzustellen. Anhand der Elemente der Vor- und Nachbedingungen, die sich an den Ein- und Ausgängen der Teilfunktionslösungen orientieren, lässt sich dann auch ableiten, welche System- und Umgebungsobjekte mit welchen Eigenschaften in der Domänendefinition aufgenommen werden müssen. Daher eignet sich die Funktionshierarchie als Ausgangspunkt der Modellierung. Weiterhin soll die Information erhalten bleiben, welche allgemeine Funktion eine Teilfunktionslösung implementiert. Im Rahmen der Modellierung einer mechatronischen Planungsdomäne zerfällt damit die Menge der Aktionssymbole  $A_s$  in mehrere disjunkte Teilmengen TFs, die Teilfunktionssymbole. Die Elemente einer solchen Teilmenge TF stellen wechselseitige Surrogate dar, können also in bestimmten Situationen gegeneinander ausgetauscht werden. Diese Informationen sind für einige der in den folgenden Kapiteln vorgestellten Methoden notwendig, um zielgerichtete Manipulationen am Plan zu ermöglichen.

Im weiteren Verlauf dieser Arbeit werden die Aktionen in der Domänendefinition einer mechatronischen Planungsdomäne als Teilfunktionslösungen bezeichnet, während die konkrete Anwendung einer solchen Teilfunktionslösung in einem Plan als Aktivität bezeichnet wird. Dadurch wird die in der PDDL bestehende begriffliche Verwirrung zwischen Aktionen in der Domänendefinition und den Grundaktionen eines konkreten Planungsproblems vermieden.

Mechatronische Systeme werden zur Erfüllung einer Hauptaufgabe konstruiert. Bereits in der Einleitung wurde zwischen der allgemeinen Formulierung dieser Hauptaufgabe im Rahmen der Konstruktion und der konkreten Instanzierung der Hauptaufgabe im Betrieb unterschieden. Das mechatronische System wird im Rahmen seiner Lebensdauer in der Regel viele Instanzierungen der Hauptaufgabe erfüllen müssen. Eine solche Instanzierung soll als Auftrag bezeichnet werden.



**Abbildung 2-6: Funktionshierarchie mit Teilfunktionslösungen**

Auch in der Beschreibung von mechatronischen Planungsdomänen müssen Aufträge berücksichtigt werden. Aufträge werden definiert durch eine Menge von Teilzielen, die zu erreichen sind. Teilziele sind dabei angegeben in Form von Zustandsvariablen, logisch oder numerisch, die innerhalb eines bestimmten Zeitfensters erfüllt sein müssen. Zusätzlich kann die Aufgabe, deren Ausprägung sie sind, spezifiziert werden. Somit kann ein Auftrag modelliert werden als eine zusammengesetzte Datenstruktur, die aus dem Typ der Aufgabe und einem Zielvektor besteht. Der Zielvektor enthält dabei alle Zustände, die im Rahmen des Auftrags zu realisieren sind.

### 2.1.3 Modellierung einer Beispieldomäne

Dieses Kapitel verdeutlicht die Modellierung eines Planungsproblems für mechatronische Systeme anhand eines Beispiels aus dem RailCab-Szenario. Dabei soll grundsätzlich das Fahren des RailCabs mit unterschiedlichen Einstellungen der Federung dargestellt werden.

Im Folgenden wird die Domäne mit ihren Prädikaten, Objekten und Aktionen beschrieben. Listing 2-1 zeigt den Kopf der RailCab-Domäne. Dabei wird der Name der Domäne definiert, die verwendeten Features der PDDL, die in der Domäne vorkommenden Objekttypen und die relevanten Prädikate. Die Beispieldomäne ist einfach gehalten und verwendet nur grundlegende Bestandteile der Sprache. Dazu gehören vor allem die Berücksichtigung unterschiedlicher Typen und die Berücksichtigung numerischer Zustandsvariablen (fluents).

Wie zu erkennen ist, besteht der logische Teil der RailCab-Domäne nur aus dem Prädikat *at*, das den Standort eines RailCab oder eines Passagiers angibt.

```
(define (domain railcab)
  (:requirements :typing :fluent)
  (:types railcab passenger location - object)
  (predicates (at?x (either passenger railcab) ?l location)))
```

**Listing 2-1: Kopf der RailCab-Domäne**

Numerische Zustandsvariablen werden nicht mittels Prädikaten, sondern in Form eines Konstrukts beschrieben, das als Funktion bezeichnet wird (siehe Listing 2-2). Sie geben in der Regel eine numerische Eigenschaft eines Objekts oder den numerischen Wert einer Beziehung zwischen mehreren Objekten an. Eine Ausnahme stellt die Funktion *total-energy-used* dar, die für die gesamte Domäne gültig ist und den Energieverbrauch im gesamten Planungsproblem dokumentiert. Dieser Wert kann beispielsweise innerhalb einer Planungsmetrik als Optimierungsziel genutzt werden.

```
(:functions (:energy_storage ?r -railcab)
  (suspension-active ?r -railcab)
  (suspension-passive ?r -railcab)
  (suspension-compensative ?r -railcab)
  (capacity ?r -railcab)
  (comfort ?p -railcab)
  (total-energy-used)
  (distance ?l1 -location ?l2 -location)
)
```

**Listing 2-2: Funktionen der RailCab-Domäne**

Die Funktion *distance* beschreibt die Länge des Fahrweges zwischen zwei Orten im Transportnetzwerk. Alle übrigen Funktionen beschreiben die Eigenschaften des RailCab Fahrzeugs: die maximale Kapazität des Energiespeichers (*capacity*), den aktuellen Ladestand des Energiespeichers (*energy\_storage*), Verbrauchswerte für die unterschiedlichen Einstellungen der Federung (*suspension-active*, *suspension-passive* und *suspension-compensative*) oder den Komfort, der sich über die Reise des RailCabs abhängig von den gewählten Federungseinstellungen ergibt. Diese Funktionen, die Eigenschaften eines Fahrzeugobjekts beschreiben, werden in der Problemdatei mit konkreten Ausgangswerten belegt.

Das RailCab als Fahrzeug realisiert bei Betrachtung auf der AMS-Ebene folgende Hauptfunktion: Es verkehrt (mit Passagieren oder Transportgütern) zwischen unterschiedlichen Punkten im Transportnetzwerk. Dabei realisiert es diese mittels alternativer Teilfunktionslösungen. Das Beispiel ist aus Gründen des Umfangs auf unterschiedliche Federungseinstellungen beschränkt. Für jede wählbare Federungseinstellung wird eine Aktion definiert. Listing 2-3 zeigt das Beispiel einer Fahraktion mit aktiver Federung. Als Parameter gehen das betrachtete RailCab, der Startort l1 und der Zielort l2 ein. Als Vorbedingung für eine Fahrt muss sich das

RailCab am Startort befinden und mindestens so viel Energie im Speicher sein, dass der Verbrauch auf der Strecke gedeckt wird. Effekt der Aktion ist, dass sich das RailCab nicht mehr am Ort I1 befindet, sondern am Ort I2 und der Ladegrad des Energiespeichers um den entsprechenden Wert reduziert wird, während die Variable *total-energy-used* (die ja quasi Buchhaltungszwecken dient) um den aktuellen Verbrauch erhöht wird. Die Definition von Fahraktionen für die passive Federung und die Federung mit Störgrößenaufschaltung erfolgt analog.

```
(:action drive
:parameters
  (?r railcab ?l1 ?l2 - location)

:precondition
  (and at ?r ?l1)
  (>=(energy_storage ?r)(*distance ?l1 ?l2)(suspension-active ?r))

:effect
  (and (not (at ?r ?l1))
        (at r? ?l2)
        (increase (total-energy-used)(*distance ?l1 ?l2)(suspension
          active ?r)))
        (decrease (energy_storage ?r) (*distance ?l1 ?l2)(suspension-
          active *r)))
)
```

**Listing 2-3: Fahraktion mit aktiver Federung**

Für das Erstellen eines Plans benötigt ein PDDL-Planer neben der Domäne eine entsprechende Problemdatei, die die konkrete Ausprägung des Planungsproblems beschreibt. Die Problemdatei beginnt mit Vergabe eines Problemnamens, der Spezifikation der Problemdomäne und einer Aufzählung der im Problem betrachteten Objekte.

```
(define (problem railcab1)
(:domain railcab)
(:objects
  railcab1 - railcab
  location1 - location
  location2 - location
  location2 - location)
)
```

**Listing 2-4: Definition des Planungsproblems**

Nach der Auflistung aller Objekte müssen die Ausgangszustände mit konkreten Werten initialisiert werden. Dabei werden zunächst die (wahren) Prädikate initialisiert. Anschließend werden die Werte der einzelnen Funktionen initialisiert.

```
(:init
  (at railcab1 location1)
    (= (capacity railcab1) 1000)
    (= (energy-storage railcab1) 1000)
    (= (suspension-active railcab1) 1)
    (= (suspension-passive railcab1) 0.8)
    (= (suspension-compensative railcab1) 1.5)
    (= (comfort railcab1) 0)
    (= (distance location0 location0) 0)
    (= (distance location0 location1) 678)
    (= (distance location0 location2) 775)
    (= (distance location1 location0) 678)
    (= (distance location1 location1) 0)
    (= (distance location1 location2) 810)
    (= (distance location2 location0) 775)
    (= (distance location2 location1) 810)
    (= (distance location2 location2) 0)
    (= (total-energy-used) 0)
  )
)
```

**Listing 2-5: Initialisierung in der Problemdatei**

Letzter Teil der Problemdefinition ist die Angabe des Zielzustands, der durch den zu erzeugenden Plan aus dem initialisierten Ausgangszustand erreicht werden soll.

```
(:goal
  (at railcab1 location2)
)
```

**Listing 2-6: Zielzustand des Beispielproblems**

Zusätzlich kann der Planer angewiesen werden, einen bzgl. einer Zielfunktion optimalen Plan zu erstellen, soweit er diese Funktion unterstützt. Dazu werden die als Metriken bezeichneten Zielfunktionen eingesetzt. Listing 2-7 zeigt den Vergleich von drei alternativen Planungsmetriken. Die erste Metrik definiert das Ziel, einen Plan zu finden, der den minimalen Energieverbrauch erzeugt, die zweite Metrik soll den Komfort maximieren und die dritte Metrik stellt schließlich die gewichtete Summe der beiden Zielgrößen dar.

```
(:metric minimize total-energy-used)
(:metric maximize comfort railcab1)
(:metric minimize (+(* -0.8(comfort))(* 0.2(total-energy-used))))
```

**Listing 2-7: Drei alternative Planungsmetriken**

Die hier beispielhaft angegebene Domänenfunktion bildet das Verhalten des mechatronischen Systems RailCab nur rudimentär ab. Wichtige Eigenschaften des Verhaltens technischer Systeme können mit den Standardsprachmitteln der PDDL nicht dargestellt werden. Ein Beispiel ist die Herleitung der Werte für *comfort*, die sich eigentlich nur bei einer genauen Kenntnis des kontinuierlichen Verlaufs der Fahrt bestimmen lassen. Hier ist also ein Weg zu finden, *kontinuierliche Prozesse* in die Domänen- und Problembeschreibung aufzunehmen. Ebenso handelt es sich um ein *deterministisches Planungsmodell*, das unterstellt, dass die Effekte der Aktion sicher bekannt sind. Da aber auch schwer vorhersagbare Effekte wie „Gegenwind“ auf das Fahrzeug einwirken, ist diese Annahme unzutreffend. Schließlich sieht das Modell keinerlei *Interaktion* des RailCabs mit anderen Systemen vor, wodurch mögliches Optimierungs- und Koordinationspotenzial nicht genutzt werden kann.

Diese drei Punkte werden in den nächsten Abschnitten hinsichtlich der Probleme, die sie im Rahmen einer Verhaltensplanung aufwerfen, genauer untersucht.

## 2.2 Integration kontinuierlicher Prozesse

In der allgemeinen Formulierung des Verhaltensplanungsproblems zu Beginn dieses Kapitels wurde unterstellt, dass die Zeitmenge  $T$  diskret ist. Gleichzeitig werden Teilfunktionslösungen so behandelt, als erzeugten sie genau einen Ausgabevektor zum jeweils nächsten Zeitpunkt. Diese Darstellung erleichtert die Modellierung mithilfe der PDDL<sup>19</sup> und damit letztlich auch die algorithmische Lösung der Planung, ist jedoch nicht zutreffend. Stattdessen kann die Anwendung einer Teilfunktionslösung auch eine zeitliche Abfolge von Ausgabewerten erzeugen, wobei die zugrunde liegende Zeitmenge grundsätzlich kontinuierlich ist. Eine Teilfunktionslösung kann demzufolge nicht mehr beschrieben werden als:

- $l_{t_f}: \text{Eingabewerte} \rightarrow \text{Ausgabewerte}$ ,

sondern bildet für gegebene Eingangswerte auf eine Zeitfunktion ab:

- $af_{l_{t_f}}: T \rightarrow A$ ,

wobei  $AF_{l_f}$  die Menge der Zeitfunktionen des Ausgabeverhaltens der Teilfunktion  $l_f$  bezeichnet. Konsequenterweise können die Eingabewerte ebenfalls als Zeitfunktion interpretiert werden:

- $ef_{l_{t_f}} = T \rightarrow E, ef_{l_{t_f}} \in EF_{l_{t_f}}$ ,

sodass sich die Teilfunktionslösung ergibt als:  $l_{t_f}: EF_{l_{t_f}} \rightarrow AF_{l_f}$ .

Die Betrachtung des Ein- und Ausgabeverhaltens als Zeitfunktion über eine kontinuierliche Zeitmenge erschwert die Analyse des Systemverhaltens erheblich. Die Zustandsbestimmung zu einem Zeitpunkt hängt nicht mehr nur von unmittelbar zuvor angewandten Teilfunktionslösungen ab, die es in dem Sinne nicht mehr gibt. Eine Darstellung des Planungsproblems in Form eines Planbaums ist damit ebenfalls nicht mehr möglich. Dadurch fallen ganze Klassen effizienter Algorithmen zur Lösung des Planungsproblems aus. Dennoch kann dieser Aspekt nicht völlig unberücksichtigt bleiben, da sehr viele Qualitätsgrößen sich nur mit Kenntnis eines kontinuierlichen Verlaufs beschreiben lassen.

Zwar unterstützt die PDDL grundsätzlich bestimmte Formen kontinuierlicher Aktionen, jedoch führt deren Anwendung zu längeren Planungsdauern oder es sind bisher keine Planer verfügbar, die diese Spracheigenschaften der PDDL unterstützen. Daher sind Alternativen zu identifizieren, die die kontinuierliche Betrachtungsweise des Systemverhaltens besser in die (diskrete) Modellierung mittels PDDL integrieren.

## 2.3 Unsicherheit und unvollständige Information

Ein erhebliches Problem bei der Planung für mechatronische Systeme besteht darin, dass sowohl Ein- als auch Ausgabeverhalten des mechatronischen Systems und des Umfelds nicht eindeutig vorhersagbar sind. Mögliche Gründe dafür sind im Wesentlichen:

- Die Eingabegrößen sind nicht vollständig bekannt oder beobachtbar,
- das Umfeld wird von zusätzlichen, nicht vorhersagbaren Faktoren beeinflusst,
- die Funktionen des Modells sind nicht vollständig bekannt oder
- einzelne Modellelemente besitzen ein stochastisches Verhalten.

<sup>19</sup> Und auch mit anderen Planungsformalismen



Die Unterscheidung zwischen unvollständiger Information und Unsicherheit wird bedeutungslos, wenn im Rahmen der Planung nicht die Möglichkeit besteht, die fehlende Information zu erlangen. Letztlich ist der Bildwert einer Funktion zu einem gegebenen (und ggf. teilweise unbekanntem) Urbild nicht eindeutig zu bestimmen. Zur Illustration wird das vereinfachte Modell des Systemverhaltens dahingehend verändert, dass dieser Mangel an Eindeutigkeit in den Funktionsdefinitionen berücksichtigt wird:

- $evolve_m: I \times O \times M \rightarrow \mathcal{P}(M)^{20}$
- $evolve_e: O \times E \rightarrow \mathcal{P}(E)$
- $affect: \mathcal{P}(L) \times M \rightarrow \mathcal{P}(O)$
- $influence: O \times E \rightarrow \mathcal{P}(I)$

Die Bestimmung eindeutiger Werte für  $m_t$ ,  $e_t$ ,  $o_t$  und  $i_t$  ist so vorab nicht mehr möglich. Die Planung, die das Verhalten der Funktion *apply* festlegen soll, müsste nun nach jedem Schritt alle möglichen Folgezustände  $S_t$  ermitteln. Die PDDL ist nicht in der Lage, die skizzierte Art von Planungsproblemen darzustellen. Daher ist auch eine direkte Anwendung der entsprechenden Planungssysteme nicht möglich. Mit jedem Zeitschritt nimmt die Menge der zu berücksichtigenden Folgezustände  $S_t$  exponentiell zu. Abbildung 2-7 stellt diesen Umstand anschaulich dar.

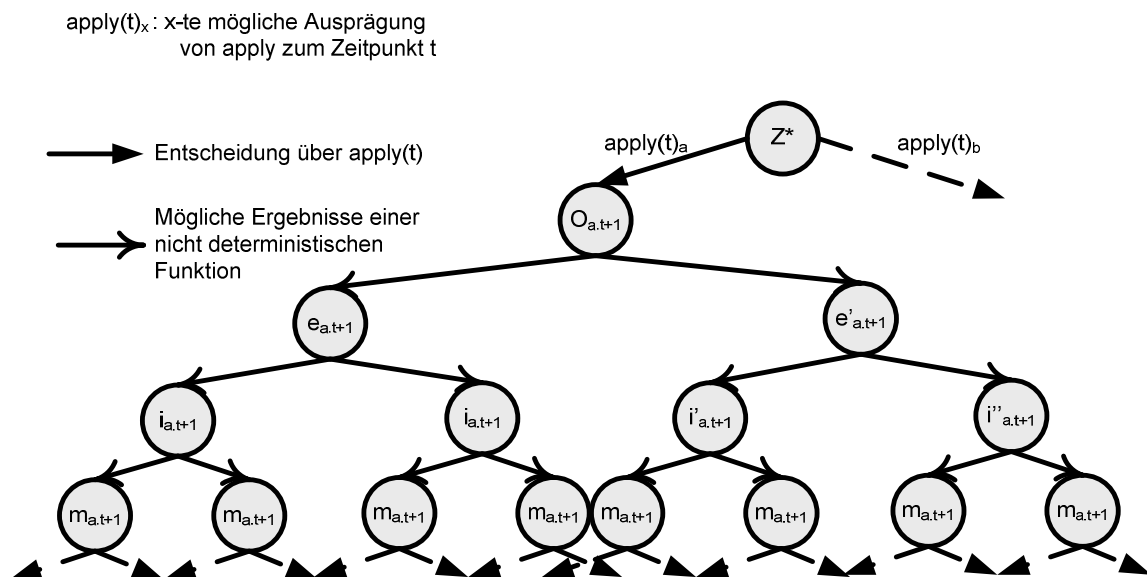


Abbildung 2-7: Planungsbaum unter Unsicherheit

Durch die Unsicherheit steigt der Verzweigungsgrad erheblich an, was direkte Wirkung auf die Komplexität des Planungsproblems hat. Während über das Ein- und Ausgabeverhalten und die möglichen und aktuellen Zustände des mechatronischen Systems viele Informationen vorliegen, ist dies beim Umfeld nicht der Fall. Dieser Umstand ist darin begründet, dass das Umfeld kein genau abgegrenztes System ist, dessen Subsysteme und Funktionen vollständig bekannt sind. Außerdem ist die konkrete Ausprägung der Umgebung in voller Detaillierung erst zur Ausführungszeit bekannt. Die Folge ist, dass der Ausgangsgrad der Knoten  $o$  und  $e$  entweder besonders hoch ist oder die Nachfolger dieser Knoten überhaupt nicht bestimmt werden können.

<sup>20</sup>  $\mathcal{P}$  steht jeweils für die Potenzmenge.

Beides stellt ein massives Hemmnis für die Planung des Verhaltens mechatronischer Systeme dar. Ein Planungsverfahren sollte daher in der Lage sein, Informationen über die Umgebung in die Konstruktion der Pläne einzubeziehen, die erst während des Betriebs gewonnen werden.

Neben fehlenden Informationen zur Konstruktion des Planbaums ist die Planungsdauer ein weiteres Problem bei der Planung in unsicheren Umgebungen. Da nicht unbegrenzt Zeit zur Planung zur Verfügung steht, kann der Planungsbaum nicht vollständig aufgebaut werden und für alle alternativen Pfade die optimale Festlegung für  $apply(t)$  gesucht werden. Um das Problem in gegebener Zeit sinnvoll zu lösen, muss das Planungsverfahren daher in der Lage sein, relevante von irrelevanten Pfaden im Planungsbaum zu unterscheiden. Um diese Unterscheidung treffen zu können, ist ein Maß der Eintrittsmöglichkeit für bestimmte Zustände erforderlich. Zwei Forderungen werden an ein solches Maß der Eintrittsmöglichkeit gestellt:

- Eine situationsabhängige Einbeziehung unsicherer Informationen über die Umgebung muss möglich sein.
- Die Eintrittsmöglichkeit muss abhängig von der Eintrittsmöglichkeit des Elternknotens bemessen werden.

Die erste Forderung ist begründet durch wechselnde Umgebungen, in denen mechatronische Systeme eingesetzt werden können, und dadurch, dass sich Informationen über die Umgebung (erfasst durch Sensoren) im Zeitverlauf verändern. Die zweite Forderung ergibt sich aus der Struktur des Planungsproblems: Ein Zustand kann nur dann eintreten, wenn sein direkter Vorgänger ebenfalls eingetreten ist. Aus diesem Grund macht eine Beurteilung der Eintrittsmöglichkeit unabhängig vom Vorgängerzustand keinen Sinn.

Auf Basis eines Möglichkeitsmaßes können Planungsverfahren so konzipiert werden, dass vor allem in relevanten Teilbäumen nach einer optimalen Wahl für die Funktion  $apply(t)$  gesucht wird. Ein Teilbaum ist dann relevant, wenn er mit besonders hohem Glaubensgrad<sup>21</sup> während der Ausführung des Planes eintritt. Der resultierende Plan würde alternative Verläufe für unterschiedliche Situationen vorsehen.

Auch wenn bevorzugt Alternativpläne für besonders relevante Situationen erstellt werden, ist es nicht möglich zu garantieren, für jede tatsächlich eintretende Situation einen Plan bereitzuhaben. Für diesen Fall stellen sich dem Planungsverfahren besondere Formalziele. Insbesondere die zeitnahe Bereitstellung eines Plans wird dabei zur kritischen Anforderung, andere Ziele (wie die Plangüte) können dafür unter Umständen zurückstehen.

Die Modellierung mittels PDDL muss also durch eine Darstellung mehrdeutiger Zustände und Nachbedingungen von Aktionen erweitert werden. Da aufgrund der Vielzahl von Umgebungseinflüssen eine Abweichung der Ausführung von der Planung eher die Regel als die Ausnahme ist, muss es möglich sein, relevante Abweichungen zu identifizieren. Dazu ist ein Maß der Eintrittsmöglichkeit von Zuständen notwendig, das in der Lage ist, die Vorgänger und Nachfolgerzustände außerdem in Beziehung zu setzen. Auf Basis dieses Maßes ist ein Verfahren zur Bestimmung relevanter Alternativpläne zu entwickeln. Um auch auf unvorhergesehene Ereignisse reagieren zu können, ist außerdem ein schnelles Umplanungsverfahren notwendig. Die unterschiedlichen Klassen von Verfahren müssen schließlich integriert werden.

---

<sup>21</sup> Glaubensgrad soll allgemein ein Maß bezeichnen, das angibt, wie die Möglichkeit eines bestimmten Faktes beurteilt wird. Umgangssprachlich würde Glaubensgrad mit Wahrscheinlichkeit bezeichnet, wobei es jedoch eine Menge alternativer Konzepte zur Beurteilung der Möglichkeit des Wahrheitsgehaltes eines Faktes gibt. Die für diese Arbeit relevanten Glaubensmaße werden in Unterkapitel 4.1.1 vorgestellt.

### 2.4 Interaktion zwischen mechatronischen Systemen

Wie bereits aus dem Vorschlag zur Strukturierung von mechatronischen Systemen (siehe Abbildung 1-2) erkennbar, handeln einzelne mechatronische Systeme in der Regel nicht isoliert in ihrem Umfeld, sondern sie interagieren mit anderen (mechatronischen) Systemen. Dementsprechend ist  $apply(t)$  nicht mehr der einzige Faktor, der den Umfeldzustand beeinflusst. Es seien:

- $MS$  die Menge aller mechatronischen Systeme, die in einer Umgebung agieren;
- $TF^i$  die Menge der Teilfunktionen des mechatronischen Systems  $i$ ;
- $L_{tf}^i$  die Menge der Lösungen von Teilfunktion  $tf \in TF^i$ , die das mechatronische System realisiert;
- $L^i = \bigcup_{tf \in TF^i} L_{tf}^i$  die Menge der Teilfunktionslösungen, die das mechatronische System  $i$  realisiert;
- $M^i$ , die Menge der möglichen Zustände des Systems  $i$ ;
- $I^i$ , die Menge der möglichen Eingabewerte aus dem Umfeld an System  $i$ ;
- $I = \prod_{i \in S} I^i$  die Menge der Eingabewerte aus dem Umfeld in alle Systeme ;
- $O^i$ , die Menge der Ausgabewerte aus System  $i$ ;
- $O = \prod_{i \in S} O^i$  <sup>22</sup> die Menge der Ausgabewerte aller Systeme;
- $affected^i: \mathcal{P}(L^i) \times M^i \rightarrow O^i$  eine Funktion, die angibt, welche Ausgabewerte bei gegebenem Zustand und einer Menge angewandter Teilfunktionslösungen erzeugt werden;
- $influence: O \times E \rightarrow I$  die Einflüsse auf die Eingaben der mechatronischen Systeme,  $influence^i: O \times E \rightarrow I^i$  sind die Einflüsse auf das mechatronische System  $i$  und
- $\prod_{ms \in MS} M^{ms} \times E = S$  bezeichnet den Gesamtzustand des Modells.

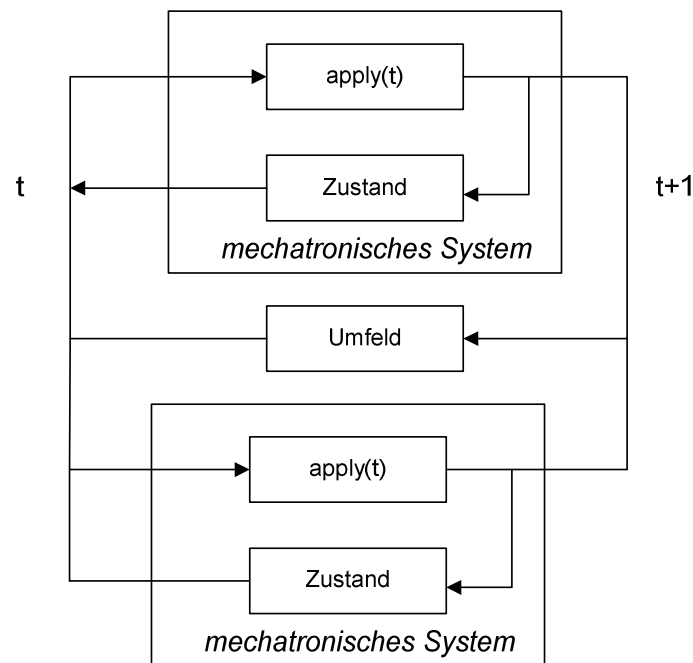


Abbildung 2-8: Zwei mechatronische Systeme in einem Umfeld

<sup>22</sup>  $\prod$  bezeichnet hier die Bildung des kartesischen Produkts über die angegebenen Mengen.

Da sich der Umfeldzustand über

- $m_t = evolve_m(influence(e_{t-1}, o_{t-1}), affect(apply(t-1), m_{t-1}))$

auf das Ein- und Ausgabeverhalten auswirkt, muss dies im Rahmen der Planung berücksichtigt werden. Abbildung 2-8 zeigt ein Umfeld, in dem zwei mechatronische Systeme aktiv sind. Entsprechend ändern sich auch die Formeln für den Umfeldzustand zum Zeitpunkt  $t$ :

- $e_t = evolve_e(o_{1,t-1}, o_{2,t-1}, e_{t-1}), evolve_e: O_1 \times O_2 \times E \rightarrow E$

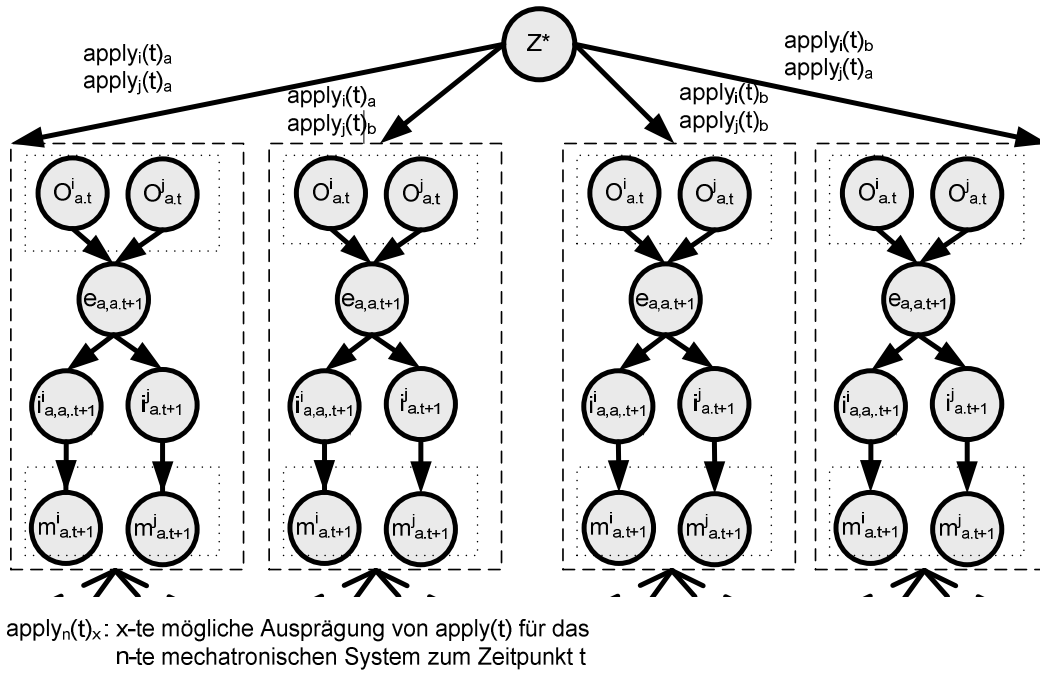
Die Ausgabewerte finden über die Zustandsfunktion Eingang in die Zustandsfunktion der mechatronischen Systeme.

Es ist ersichtlich, dass die beiden Systeme aus Abbildung 2-8 sich gegenseitig beeinflussen. Daher ist es erforderlich, die Funktionswerte von *apply* der beiden Systeme aufeinander abzustimmen, also eine systemübergreifende Planung durchzuführen. Gesucht wird also nicht mehr eine Folge von Teilfunktionslösungen, die das gewünschte Ein- und Ausgabeverhalten eines mechatronischen Systems erfüllt. Stattdessen sind zwei Sequenzen so zu bestimmen, dass zwei mechatronische Systeme jeweils das gewünschte Verhalten zeigen. Im allgemeinen Fall, wenn  $n$  Systeme in der gleichen Umgebung handeln, gilt:

- $e_t = evolve(o_{1,t-1}, \dots, o_{n,t-1}, e_{t-1})$
- $m_{i,t} = evolve_{m_{i,t}}(influence(evolve_e(affect^1 apply_1(t-2)), \dots, affect^n(apply^n(t-2))), affect(apply^i(t-1), m_{i,t-1}))$

weswegen  $n$  Folgen von Teilfunktionslösungen zu bestimmen sind, die ein gewünschtes Ein- und Ausgabeverhalten für  $n$  mechatronische Systeme gewährleisten. Eine Planung, die zur Festlegung einer solchen Menge von Teilfunktionslösungen dient, soll im Weiteren als Interaktionsplanung bezeichnet werden.

Abbildung 2-9 stellt das Planungsproblem für zwei mechatronische Systeme dar. Es ist anschaulich erkennbar, dass die Komplexität der Planung durch die Wechselwirkungen zwischen den Systemen steigt (der Verzweigungsgrad im Baum nimmt zu).



**Abbildung 2-9: Planungsproblem bei zwei mechatronischen Systemen**

Die Situationen, in denen eine Planung der Interaktion von mechatronischen Systemen sinnvoll oder erforderlich ist, unterscheiden sich teilweise erheblich. Daher soll an dieser Stelle ein Klassifikationsschema aufgebaut werden, auf dessen Basis dann die notwendigen Anforderungen an das Lösungsverfahren und das Verhaltensmodell beschrieben werden können. Die Klassifikationsmerkmale, die als relevant betrachtet werden, sind im Einzelnen:

- Hauptaufgabe: gemeinsam oder individuell
- Zusammensetzung der Systeme: statisch oder dynamisch
- Verteilung der Teilaufgaben: bekannt oder unbekannt.

Das wichtigste Unterscheidungsmerkmal ist, ob die beteiligten Systeme eine gemeinsame oder jeweils eine individuelle Hauptaufgabe erfüllen müssen. Für den Fall, dass n mechatronische Systeme eine gemeinsame Hauptaufgabe erfüllen müssen, besteht das Sachziel im Auffinden von n Ausprägungen der Funktion  $apply(t)$ , sodass die Hauptaufgabe erfüllt wird. Beispiele für diese Art der Interaktion finden sich vor allem auf Ebene der mechatronischen Funktionsmodule, die in einem autonomen mechatronischen System zusammengefasst sind, etwa wenn ein Energiemanagement und ein Antriebsmodul innerhalb eines Fahrzeugs zusammenwirken, um die Hauptaufgabe „Transport von Passagieren oder Gütern“ zu erfüllen. Auf Ebene der vernetzten mechatronischen Systeme kann ein Roboterschwarm mit der Aufgabe, Tischtennisbälle farblich zu sortieren, diese Art von Interaktion veranschaulichen.

Bei individuellen Hauptaufgaben kann aufgrund der wechselseitigen Beeinflussung der Systeme der Fall eintreten, dass es nicht möglich ist, n Funktionsfolgen von  $apply(t)$  festzulegen, die alle Hauptaufgaben erfüllen. Weiterhin kann eine wechselseitige Beeinflussung der Plan Güte auftreten. Zusätzlich ist es in der Regel nicht möglich, eine Gewichtung der Hauptaufgaben vorzunehmen oder ein gemeinsames Gütemaß für die n Funktionsfolgen zu finden. Das ist beispielsweise dann der Fall, wenn die mechatronischen Systeme ihre jeweilige Hauptaufgabe für unterschiedliche Nutzer erfüllen müssen. Beispiele für diese Art von Beziehung zwischen mechatronischen Systemen finden sich ausschließlich auf Ebene vernetzter mechatronischer Systeme, da für mechatronische Funktionsmodule durch das überlagerte autonome mechatronische System immer ein gemeinsames Zielsystem gegeben ist.

Auf Basis dieser ersten Unterscheidung ergeben sich zwei völlig unterschiedliche Planungsaufgaben. Der entscheidende Unterschied zwischen den beiden Planungsaufgaben besteht im Sachziel:

- Erstellen eines Plans für  $n$  mechatronische Systeme zur Erreichung von  $n$  konkreten Hauptaufgaben und
- Erstellen eines Plans für ein mechatronisches System zur Erreichung einer konkreten Hauptaufgabe unter Berücksichtigung der Einflüsse von  $n-1$  anderen mechatronischen Systemen.

Die Zusammensetzung der interagierenden Systeme kann vorab bekannt sowie statisch oder dynamisch sein. Je nach Ausprägung muss ein Planungsverfahren also zusätzlich die relevanten mechatronischen Systeme identifizieren bzw. zusammenfassen. Ein ähnliches Problem stellt sich bei der Verteilung der Teilaufgaben. Wenn diese nicht vorab bekannt sind, muss im Verlauf des Planungsprozesses für eine eindeutige Zuordnung von Teilaufgaben zu Systemen gesorgt werden.

Aus der Tatsache, dass zwischen mechatronischen Systemen Wechselwirkungen bestehen, lassen sich spezifischere Zielstellungen für die Arbeit herleiten. Die Möglichkeit der Interaktion ist so in die Darstellung der PDDL zu integrieren, dass sowohl Interaktionen bei gemeinsamer als auch individueller Hauptaufgabe im Rahmen des Planungsprozesses berücksichtigt werden.

Um das gesamte Spektrum möglicher Interaktionen abdecken zu können, ist es außerdem erforderlich, Verfahren zur Identifikation relevanter mechatronischer Systeme zu entwickeln oder zu identifizieren. Ebenso müssen für die Planungssituationen, in denen die Aufgabenverteilungen zwischen mechatronischen Systemen nicht vorab bekannt sind, Verfahren zur Aufteilung der Gesamtaufgabe in Teilaufgaben und zur Verteilung der Teilaufgaben auf die verfügbaren Systeme vorhanden sein. Schließlich müssen die eigentlichen Planungsverfahren mit den Verfahren zur Identifikation von relevanten Systemen sowie zur Aufgabenverteilung sinnvoll verknüpft werden.

Die Formulierung des Planungsproblems in PDDL sollte keine zentrale Lösung des Planungsproblems erfordern. Zunächst ist dies für Systeme mit individueller Hauptaufgabe nicht plausibel (wenn jedes System seine eigenen Ziele verfolgt, gibt es keinen Grund, dass die Systeme sich einer zentralen Instanz unterwerfen). Außerdem hat die Zusammenfassung mehrerer Planungsprobleme negative Auswirkungen auf die Komplexität des Planungsproblems. Am Beispiel wird dies deutlich. Betrachtet werden zwei Systeme A und B, für die ein Planungsproblem zu formulieren ist. Beide Systeme besitzen 2 unterschiedliche Teilfunktionslösungen. Bei einer dezentralen Lösung mit zwei lokalen Planungsmodellen müssen bei einer Suchtiefe von fünf im Planbaum  $2 \cdot 2^5 = 64$  unterschiedliche Verläufe betrachtet werden. Fasst man die Planungsprobleme in einem zentralen Modell zusammen, so erhöht sich der Verzweigungsgrad<sup>23</sup> auf 4. Entsprechend sind  $4^5 = 1024$  unterschiedliche Verläufe zu untersuchen. Das bedeutet, im allgemeinen Fall nimmt die Größe des Planungsproblems bei zentral formulierten Modellen erheblich schneller zu.

Die Möglichkeit der Interaktion muss also in der Modellierung des Planungsproblems mithilfe der PDDL integriert werden. Dabei sind Formen der Interaktion, also bei gemeinsamer und individueller Hauptaufgabe zu berücksichtigen. Um die gesamte Bandbreite an möglichen Kooperationen abzudecken, werden zusätzlich Verfahren zur Identifikation kooperationsfähiger

---

<sup>23</sup> In einer konkreten Anwendung lassen sich zwar vermutlich Kombinationen von Teilfunktionslösungen finden, die gemeinsam nicht sinnvoll sind, dies hebt aber die allgemeine Tendenz des wachsenden Zustandsraums nicht auf.

ger mechatronischer Systeme sowie Verfahren zur Aufgabendeckomposition und –verteilung benötigt. Schließlich sind diese einzelnen Verfahren in einem Gesamtablauf zu integrieren.

## **2.5 Aufbau der Arbeit**

Diese Arbeit widmet sich im Folgenden der Aufgabe, die Verhaltensplanung für mechatronische Systeme zu konzipieren. Dabei dient die PDDL als Grundlage für die Modellierung der Verhaltensplanungsprobleme. Da die PDDL in der Lage ist, generische Planungsprobleme darzustellen, können die Konzepte zur Verhaltensplanung allgemein und losgelöst von einer konkreten Anwendung formuliert werden.

In der Konzeption einer Verhaltensplanung sind natürlich die drei soeben angesprochenen Problemfelder zu berücksichtigen. Da jedes Problemfeld für sich einen erheblichen Umfang besitzt, ist es notwendig, möglichst viele Probleme mithilfe bestehender Arbeiten zu lösen. Da zu jedem Problemfeld ein umfangreicher und relevanter Stand der Technik existiert, aus dem geeignete Modellierungen und Lösungsverfahren auszuwählen sind, orientiert sich die Gliederung der Arbeit zunächst an den drei Problemfeldern *Integration kontinuierlicher Prozesse (Kapitel 3)*, *Planung unter Unsicherheit (Kapitel 4)* und *Planung von Systeminteraktionen (Kapitel 5)*. Ziel der Kapitel ist es jeweils, die Modellierungsmöglichkeiten der PDDL so einzusetzen, dass das jeweilige Teilproblem durch einen PDDL-Planer gelöst werden kann. Wo dies nicht möglich ist, soll die PDDL um zusätzliche Modellierungselemente erweitert werden. Diese Modellierungselemente können dann zur Analyse und Anpassung von PDDL-Plänen eingesetzt werden. In Kapitel 6 erfolgt dann eine Gesamtintegration in einen Planungsprozess auf Basis der PDDL.

Existierende PDDL-Planer unterstützen den Sprachumfang der PDDL nicht vollständig. Insbesondere der Umgang mit Planungsmetriken (Zielfunktionen) wird nicht ausreichend unterstützt. Daher wird in Kapitel 7 (Anwendungsbeispiele) von der Modellierung auf Basis der PDDL abgewichen. Stattdessen werden die Planungsmodelle spezifisch für die Anwendungsbeispiele formuliert und individuelle Lösungsverfahren vorgestellt. Da dieses Vorgehen es ermöglicht, Informationen über Besonderheiten der Anwendungsbeispiele, z. B. zur Formulierung von Heuristiken auszunutzen, sind die Lösungsverfahren schneller und weniger speicherintensiv als PDDL-Planer. Die Laufzeit und Speicherbedarf wesentliche Kriterien im Kontext mechatronischer Systeme sind, empfiehlt sich dieses Vorgehen auch für praktische Anwendungen.

Das Kapitel 8 schließt die Arbeit letztlich mit einer Zusammenfassung der erarbeiteten Konzepte und einem Ausblick.





### 3 Integration kontinuierlicher Prozesse in eine Verhaltensplanung auf Basis der PDDL

Ein Beispiel für die Notwendigkeit, kontinuierliche Prozesse in die Verhaltensplanung zu integrieren, liefert die aktive Federung des RailCab-Systems. Aufgabe der aktiven Federung im RailCab ist die Entkopplung des Aufbaus von Anregungen. Der Aufbau soll also möglichst *ruhig* gehalten werden, was in der Regel als komfortabel bezeichnet wird. Der Komfort, der sich beim Befahren eines Streckenabschnitts ergibt, hängt vom kontinuierlichen Verlauf ab. So hat zu jeder Zeit die gerade wirkende Aufbaubeschleunigung Auswirkung auf das Komfortempfinden der Fahrgäste. Eine solche Größe kann mit der eben beschriebenen Modellierung mit PDDL nicht direkt abgebildet werden, da sie nicht an einen Zustand des Systems oder der Umwelt gebunden ist, sondern von einem kontinuierlichen Verlauf abhängt. Der Komfort lässt sich natürlich in einem Streckenabschnitt in einer Zielgröße fassen. So beschreibt [VT08] den Komfort während einer RailCab-Reise mithilfe der Summe über die Integrale der Aufbaubeschleunigung<sup>24</sup> in den drei Hauptbewegungsachsen:

$$f_1(t) = \frac{1}{t - t_0} \int_{\tau=t_0}^t \sum (W(a_{\text{Aufbau}}(\tau)))^2 d\tau.$$

Diese Zielgröße lässt sich jedoch erst ermitteln, nachdem ein kontinuierlicher Verlauf der Aufbaubewegung über den Streckenabschnitt bestimmt wurde. Zusätzlich kann der kontinuierliche Verlauf ebenfalls im Voraus bestimmt werden (gute Streckeninformationen vorausgesetzt). Diese Informationen sind in sinnvoller Art in die diskrete Planung zu integrieren.

In den folgenden Teilkapiteln werden bestehende Ansätze zur diskret-kontinuierlichen Planung auf ihre Eignung für den Einsatz im Rahmen einer Verhaltensplanung auf Basis der PDDL analysiert. Anschließend wird beschrieben, wie ausgewählte Methoden in das PDDL-Planungsproblem integriert werden können, wobei aus Gründen der Komplexität im Rahmen der Planung die diskrete Darstellung des Verhaltens aus Unterkapitel 2.1.2 beibehalten werden soll.

#### 3.1 Stand der Technik zur diskret-kontinuierlichen Planung

In den folgenden Absätzen werden Modellierungs- und Planungsmethoden vorgestellt, die sich mit der Darstellung und Vorabberechnung solcher Verläufe beschäftigen.

##### 3.1.1 Modellierung kontinuierlicher Planungsaspekte

Um auch Zustandsübergänge in Rahmen der Planung bewerten zu können – sei es hinsichtlich ihrer Qualität oder ihrer Gültigkeit – ist es erforderlich, den Verlauf dieser Aktion über eine kontinuierliche Zeitachse zu beschreiben. Die folgenden Absätze stellen mehrere Ansätze für eine solche Darstellung vor.

###### 3.1.1.1 Darstellung kontinuierlicher Prozesse

Die mathematische Beschreibung des zeitlichen Verhaltens eines dynamischen Systems wird in der Regel mithilfe eines Systems von Differenzialgleichungen angegeben [Möll92]. Das Ziel der Beschreibung durch Differenzialgleichungen ist es, diese in einem Zeitintervall  $[t_0, t_e]$  auszuwerten und dadurch die Systemvariablen zu jedem Zeitpunkt  $t_i \in [t_0, t_e]$  zu bestimmen<sup>25</sup>.

<sup>24</sup> Tatsächlich handelt es sich um gefilterte Aufbaubeschleunigungen, da einige Frequenzbereiche vom Menschen als besonders unangenehm empfunden werden.

<sup>25</sup> Wird dabei von bekannten Anfangsbedingungen ausgegangen, spricht man vom Anfangswertproblem.

In der Regel bleibt die Beschreibung und Analyse dynamischen Systemverhaltens auf lineare Differenzialgleichungssysteme beschränkt, die mathematisch leichter zu handhaben sind. In der Praxis ist daher die Linearisierung um einen Arbeitspunkt sehr weit verbreitet [Rau03, S.22].

Zur Beschreibung nicht-linearer dynamischer Systeme werden z. B. blockorientierte Modelle genutzt. Sie stellen eine Kombination aus statischem nicht-linearem Funktionsanteil und dynamischen linearen Funktionsanteilen dar [Math02a, Math02b]. Da nicht alle nicht-linearen dynamischen Systeme durch blockorientierte Modelle dargestellt werden können, sind oft Differenzialgleichungen notwendig. Die allgemeine Gültigkeit solcher Modelle stellt jedoch hohe Anforderungen an die Messbarkeit der einzelnen Modellparameter, um ein Rechnen mit dem Modell zu ermöglichen [Rau03]. Da im Rahmen der Planung keinerlei Messung möglich ist, sondern nicht exakt bestimmbare zukünftige Werte festgelegt werden, fällt der analytische Ansatz zur Beschreibung des kontinuierlichen Systemsverhaltens aus, wenn nicht auf zuverlässige Vergangenheitsdaten zurückgegriffen werden kann.

Als Alternative für die exakte analytische Bestimmung eines technischen Prozesses kommen approximative Verfahren in Frage, die aus einer Menge von Daten die den Prozess beschreibenden Funktionen näherungsweise bestimmen. Dieser Prozess wird als Identifikation bezeichnet [Möl93]. Zu den approximativen Verfahren gehören algebraische Darstellungen mit Funktionsreihen, Verfahren auf Basis von Stützstellen (Interpolation und Approximation) und konnektionistische Verfahren. Eine weitere Möglichkeit besteht in der Anwendung von neuronalen Netzen [Rau03]. Aber auch neuronale Netze (wenn sie isoliert angewandt werden) liefern exakte Werte zurück, die eine genaue Kenntnis des Prozessablaufs vorspiegeln, die nicht gegeben ist.

Eine Alternative, die zur Beschreibung technischer Prozesse häufig eingesetzt wird, wenn verrauschte oder nur grob zu bestimmende Parameter im Prozess enthalten sind, ist die Modellierung mit Hilfe von Fuzzy-Sets und Fuzzy-Logik. Die Theorie der Fuzzy-Sets (wörtlich: *fransige Mengen*) spezifiziert, wie gut ein Objekt oder Element einer vagen Beschreibung entspricht. Im Gegensatz zur konventionellen Mengentheorie, bei der ein Element einer Menge entweder ganz oder gar nicht angehört, definiert die Theorie der Fuzzy-Sets eine Zugehörigkeitsfunktion. Der Wert dieser Zugehörigkeitsfunktion ist auf das Intervall  $[0,1]$  beschränkt [GCR03]. Ein Wert von 0 bedeutet, dass das Element nicht zu der Menge gehört und ein Wert von 1, dass das Element ganz in der Menge liegt. Die formale Definition eines Fuzzy-Sets  $F$  lautet:

$$\bullet \quad := \{(u, \mu_F(u)) \mid u \in U, \mu_F(u) \in [0,1]\},$$

wobei  $U$  die Grundmenge bzw. das Universum mit den Elementen  $u$  und  $\mu_F(u)$  eine totale Abbildung  $\mu_F: U \rightarrow [0,1]$  vom Universum in das abgeschlossene Intervall  $[0, 1]$  ist.  $\mu_F(u)$  wird auch Zugehörigkeitsfunktion (*membership function*) genannt. Fuzzy-Logik ist ein Formalismus zum Schließen über logische Aussagen, die Zugehörigkeitsfunktionen von Fuzzy-Sets beschreiben. Durch diesen Formalismus können Fuzzy-Sets in einer Vielzahl von Systemen wie z. B. Expertensystemen eingesetzt werden. Der Ablauf des Schließens in einem Fuzzy-Experten- oder -regelsystem ist grundsätzlich immer gleich [Lipp06]:

1. Fuzzifizierung der Eingangsgrößen
2. Inferenz und Regelkomposition
3. Defuzzifizierung der Ausgangsgrößen

Der Schritt der Fuzzifizierung bezeichnet dabei die Auswertung der Zugehörigkeitsfunktionen zu den einzelnen linguistischen Termen für einen gegebenen scharfen Wert. Im Rahmen der Inferenz arbeitet das Fuzzy-Regelsystem auf einer Menge von gegebenen Wenn-Dann-Regeln. Dabei werden im Wenn-Teil der Regel linguistische Terme miteinander verknüpft, die über die Eingangsgrößen definiert sind. Im Dann-Teil werden den Ausgangsgrößen konkrete Ausprägungen der über sie definierten linguistischen Variablen zugeordnet. Ein Beispiel für eine solche Regel ist: *Wenn  $x$  positiv klein und  $y$  positiv klein ist, dann ist  $z$  positiv klein.*

Die Anwendungen der Fuzzy-Logik in technischen Systemen sind vielfältig. Pfeiffer et. al. nennen zum Beispiel [PJK+02a]:

- direkte Fuzzy-Regelung und Steuerung (DFC),
- Fuzzy-Logik zur Parameter-Steuerung oder Adaption,
- Klassifikation/Mustererkennung (KLAS),
- Diagnose (DIAG),
- datenbasierte Modellierung/Identifikation,
- fuzzy-basierte Prozessführung und Optimierung und
- Prädiktivregler mit Fuzzy-Prozessmodell (FPC).

Bei direkter Fuzzy-Regelung und Steuerung handelt es sich um solche Anwendungen, in denen ein klassischer Regler durch einen Fuzzy-Regler ersetzt wird. Der Fuzzy-Regler entspricht dabei einem Fuzzy-Expertensystem, das die Reglereingangsgrößen (Stör- und Führungsgrößen) auf die Ausgangswerte abbildet [Kah195]. Auch Kombinationen mit klassischen regelungstechnischen Ansätzen existieren, wie eine Fuzzy-Störgrößen-Aufschaltung auf einen konventionellen PI-Regler. Beispiele für solche Anwendungen finden sich in [Pfeif00], [Gier98] und [TR95].

Die Parameter-Steuerung oder Adaption bezeichnet all diejenigen Anwendungen, bei denen die Parameter eines konventionellen Reglers durch ein Fuzzy-Expertensystem angepasst werden [SCN99].

Konventionelle Klassifikationsverfahren ordnen einzelne Individuen auf Basis eines Merkmalsvektors genau einer Klasse zu. Die Fuzzy-Klassifizierung dagegen weicht die harten Grenzen zwischen den Klassen auf und ordnet die Individuen etwa zu 80% einer und zu 20% einer anderen Klasse zu (jeweils im Sinne einer Membership-Funktion). Basis ist wieder ein Fuzzy-Regel-System [Kuhn00].

Im Wesentlichen entspricht das Anwendungsgebiet der Diagnose einer Klassifikationsaufgabe, nämlich der Klassifikation einer gegebenen Situation als eine bestimmte Fehlersituation. Fuzzy-Systeme zur Diagnose unterscheiden sich insofern grundsätzlich von Fuzzy-Reglern, als das keine Defuzzifizierung notwendig ist. Eingang des Fuzzy-Systems sind beobachtete Symptome oder Messwerte. Die Grundform der Regeln lautet: „Wenn <Symptom  $i$ > beobachtet, dann liegt <Fehler  $j$ > vor“ [FMB+98].

Die datenbasierte Modellierung (oder Identifikation) mit Fuzzy-Modellen stellt einen Ansatz dar, durch induktives Lernen aus Beispielen Wissen über einen technischen Prozess zu gewinnen und zu modellieren. Der große Vorteil des Einsatzes von Fuzzy-Systemen gegenüber neuronalen Netzen besteht darin, dass konventionelle neuronale Netze reine Blackbox-Modelle darstellen, also keine Informationen über die inneren Wirkzusammenhänge des zu erlernenden Prozesses eingehen. Dagegen lassen sich Fuzzy-Systeme auf linguistischer Ebene interpretieren und sogar modifizieren [PJK+02b]. Zusätzlich besteht bei Fuzzy-Systemen eine intuitive Möglichkeit, die Struktur des zu erlernenden Prozesses vorzugeben, sodass nur die Parameter des Modells, also die Membership-Funktionen und die Regeln des Fuzzy-Systems, erlernt werden müssen [Pfeif93].

### 3.1.1.2 Planungssprachen mit kontinuierlichen Elementen

Auch im Rahmen von Planungssprachen in der Künstlichen Intelligenz wird das Problem kontinuierlicher Aktionen behandelt. So führen Dearden et. al. [BDM+02] zwei unterschiedliche Darstellungsformen für kontinuierliche Aktionen ein, die sich vor allem hinsichtlich ihrer Komplexität unterscheiden. Die weniger komplexe Darstellungsform, die vor allem zur Konstruktion von Plänen eingesetzt wird, behält die übliche Darstellung von Aktionen bei. Die Aktionen erzeugen also bei gegebenen Vorbedingungen bestimmte Nachbedingungen. Es wird lediglich unterstellt, dass jede Anwendung einer Aktion eine individuelle kontinuierliche Dauer hat und dass die Ein- und Ausgangsvariablen ebenfalls kontinuierlich sind. Da diese Repräsentation zwar für die Planung zweckmäßig ist (das Zustandsraumkalkül bleibt grundsätzlich unberührt), ist sie für Diagnosezwecke unzureichend. Daher wird für die Diagnose eine genauere Repräsentation eingeführt [Dear05], die als *Probabilistic Hybrid Automaton* (PHA) bezeichnet wird. Dieser betrachtet das zu diagnostizierende System als eine Menge kontinuierlicher Variablen und eine Menge beobachtbarer Variablen. Das System befindet sich zu jedem Zeitpunkt in einem bestimmten Modus. Jeder Modus besteht aus einer Menge von zeitdiskreten Differenzialfunktionen, die die zeitliche Entwicklung der kontinuierlichen Variablen beschreiben. Zwischen den einzelnen Modi existieren bestimmte Übergangswahrscheinlichkeiten. Dabei bleibt leider unklar, wie diese Differenzialfunktionen bestimmt werden können und wie der Zusammenhang zu den konkreten Umfeldeinflüssen innerhalb eines Planungsschritts hergestellt werden kann.

Beide Ansätze sind zur Lösung des hier beschriebenen Problems ungeeignet, da der einfache Modellierungsansatz keinerlei Aussage über die kontinuierliche Natur einer Aktion ermöglicht und der genauere Ansatz wiederum auf Differenzialgleichungen zurückgreift. Die Probleme der direkten Anwendung dieses Ansatzes in der Planung wurden bereits in Abschnitt 3.1.1.1 dargelegt. Jedoch ist der Ansatz vielversprechend, das Problem der Integration kontinuierlicher Verläufe durch eine Zweiteilung in ein eigentliches Planungsmodell und ein Analysemodell zu bewältigen.

Seit der Version 2.1 unterstützt auch die PDDL numerische Variablen sowie dauerhafte und kontinuierliche Aktionen [FL03]. Der Wertebereich der numerischen Variablen ist dabei die Menge der reellen Zahlen und damit grundsätzlich zur Darstellung kontinuierlicher Variablen geeignet. Bei der Darstellung von kontinuierlichen Aktionen wird zwischen dauerhaften und kontinuierlichen Aktionen unterschieden. Die dauerhaften Aktionen ermöglichen es, den Vor- und Nachbedingungen eine genauere zeitliche Spezifikation zu geben. So kann spezifiziert werden, dass Vorbedingungen zu Beginn der Aktion, während der Ausführung oder am Ende der Aktion erfüllt sein müssen (oder in beliebiger Kombination). Ebenso können Nachbedingungen sofort oder am Ende einer Aktion eintreten. Eine noch genauere Darstellung ist mit kontinuierlichen Aktionen möglich, die Effekte der Aktion in einzelnen Zeitschritten modellieren. Dies entspricht in etwa der Spezifikation der diskreten Differenzialgleichungen bei Dearden. Grundsätzlich ist mit den Methoden der PDDL eine Abbildung zumindest quasi-kontinuierlicher Aktionen möglich. Negativ sind aber die Auswirkungen auf die Planungskomplexität zu bewerten, da der Zustandsraum sprunghaft anwächst. Daher sind auch keine Planer für die höchste Detaillierungsstufe der PDDL verfügbar (siehe auch den folgenden Abschnitt).

### 3.1.2 Planungsverfahren mit quasi-kontinuierlichen Prozessen

Die meisten Planungsverfahren, die über eine quasi-kontinuierliche Repräsentation von Zeit verfügen, bauen auf der Modellierung der PDDL 2.1 auf. Dabei beschränken sich die Lösungsverfahren aktuell auf dauerhafte Aktionen, kontinuierliche Aktionen werden also nicht unterstützt. Dies liegt hauptsächlich an der mit den kontinuierlichen Aktionen verbundenen Vergrößerung des Zustandsraums.

Unterstützen Planer dauerhafte Aktionen, handelt es sich in der Regel um eine Abbildung einer konventionellen Planungstechnik auf das Problem mit dauerhaften Aktionen. Dazu wird meist die verfahrensinterne Repräsentation des Plans um zusätzliche Elemente erweitert, die in Typ und Art aber den bereits vorhandenen Elementen entsprechen. Dadurch wird die ursprüngliche Planungstechnik auf das neue Problem anwendbar.

So bildet der HTN-Planer SHOP2 [NAI+03] dauerhafte Aktionen intern durch Methoden ab. Eine Methode ist eine Verkettung mehrerer primitiver Aktionen<sup>26</sup>, die in einer bestimmten Ausführungsreihenfolge genutzt werden können, um einen komplexeren Task durchzuführen. Der komplexe Task ist dabei wie eine Aktion eine Überführung von Vorbedingungen in Nachbedingungen. Eine Methode, die eine dauerhafte Aktion abbildet, enthält drei Aktionen – jeweils eine für den Start, die Dauer und das Ende der Aktion. Um die restriktive Forderung der PDDL einzuhalten, dass Ressourcen produzierende Aktionen strikt vor Ressourcen konsumierenden Aktionen beendet sind, werden zusätzlich sogenannte Space Pseudo-Aktionen eingeführt, die dies sicherstellen.

Auch der Planer LPG (Local Search for Planning Graphs) unterstützt dauerhafte Aktionen. Intern verwendet der Planer Plangraphen zur Darstellung des Planungsproblems [GSS03]. Auch hier wird eine neue Art von Operationen eingeführt, no-ops, die die unmittelbaren Effekte am Start einer Aktion auslösen und die Einhaltung von Vorbedingungen während der Ausführung sicherstellen. Die no-ops werden beim Einplanen einer dauerhaften Aktion so in den Plangraph integriert, dass die Effekte und Vorbedingungen auf den entsprechenden Ebenen des Plangraphen auftauchen. Zusätzlich wurden für Plangraphen notwendigen *Ordering Constraints* erweitert, da durch die Einführung von dauerhaften Aktionen echte Nebenläufigkeit im Plan auftreten kann. Eine Beurteilung der kontinuierlichen Qualität einer Verhaltensweise ist auf Basis der dauerhaften Aktionen aber nicht möglich.

### 3.1.3 Optimale Steuerung und kontinuierliche Mehrzieloptimierung

Das Forschungsgebiet der Optimalen Steuerung ist das Problem, eine eine Steuerung zu finden, die die Dynamik eines Systems über einem Zeitintervall so beeinflusst, dass ein vorgegebenes Zielfunktional optimiert wird. Diese Probleme umfasst eine Menge von Differenzialgleichungen, die die zeitliche Veränderung der Steuerparameter beschreibt, sodass eine Kostenfunktion minimiert wird [Kirk70]. Dabei bezieht sich die Kostenfunktion einerseits auf die Zustandsvariablen des Systems und andererseits auf die Steuerparameter.

Für einige Fälle kann das Problem, eine optimale Regelung zu finden, als kontinuierliches Mehrzieloptimierungsproblem aufgefasst werden. Im Gegensatz zur klassischen Optimierung betrachtet die Mehrzieloptimierung nicht eine einzige Zielfunktion (die u.U. mehrere Zielgrößen umfasst), sondern mehrere Ziele gleichzeitig. Daraus ergibt sich, dass nicht in jedem Fall eine eindeutige optimale Lösung bestimmt werden kann. Ergebnis der Mehrzieloptimierung ist also eine Menge von Lösungen, die die effizienten Kompromisse zwischen den Zielen darstellen. Diese Menge wird als Pareto-Menge bezeichnet. Ein Punkt in der Pareto-Menge (eine gültige Lösung) zeichnet sich dadurch aus, dass er nicht durch eine andere Lösung dominiert wird. Eine Lösung  $l_a$  wird dann von einer Lösung  $l_b$  dominiert, wenn  $l_b$  hinsichtlich jeden Zielkriteriums mindestens genauso gut ist wie  $l_a$  und in mindestens einem Zielkriterium besser als  $l_b$ .

Im Fall der Mehrzieloptimierung ist das Ergebnis nicht eine einzige Trajektorie von Parametern, sondern eine Menge von Trajektorien. Eine Beispielanwendung auf das Problem, eine optimale Vorgabebahn für die Spurführung eines Schienenfahrzeugs zu ermitteln, findet sich in [GTK+08].

---

<sup>26</sup> Eine primitive Aktion kann nicht mehr in weitere Aktionen aufgespalten werden und ist direkt anwendbar.

Problematisch an der Optimalen Steuerung sind die Anforderungen, die an die Genauigkeit der Modelle gestellt werden. Diese können im Rahmen der Planung, die ja auf zukünftigen und daher ungenauen Werten operiert, nicht gesichert werden. Jedoch zeigt das Beispiel der Vorgebahn-Bestimmung für die Spurführung eines Schienenfahrzeugs, dass die Ergebnisse zumindest einen Anhaltspunkt für den kontinuierlichen Verlauf einer Aktion liefern können, wenn Modell und realer Prozess nur hinreichend ähnlich sind. Dies eröffnet die Möglichkeit, derartige mathematische Problemformulierungen im Rahmen der Verhaltensplanung zu verwenden.

### **3.1.4 Fazit zum Stand der Technik in diskret-kontinuierlicher Planung**

Der Umgang mit kontinuierlichen Aktionen in der Planung ist nach wie vor ein nicht zufriedenstellend gelöstes Problem. Die neueren Ansätze, wie die Modellierung von Dearden oder der PDDL 2.1 ermöglichen es zwar grundsätzlich, Pläne so zu gestalten, dass Randbedingungen für die gesamte Ausführungsdauer einer Aktion erfüllt sind. Sie verwerfen dabei jedoch eine Vielzahl möglicher Lösungen. Weiterhin steigt die Komplexität der Planung durch einen vergrößerten Zustandsraum erheblich an. Für die feinste Modellierungsstufe der PDDL kontinuierlicher Aktionen sind sogar gar keine Planungsansätze bekannt. Die Modellierung geht außerdem von unrealistisch genauen Informationen über die kontinuierlichen Zustandsgrößen im Rahmen der Planung aus (da letztlich sowohl bei Dearden als auch in der PDDL 2.1 diskrete Differenzialgleichungssysteme zur Modellierung des kontinuierlichen Verlaufs genutzt werden). Eine Beurteilung der Qualität einer kontinuierlichen Aktion, die wesentlich über die Überprüfung der Durchführbarkeit hinausgeht, ist gar nicht möglich.

Hier können Fuzzy-Ansätze einen Ausweg aufzeigen, da sie in der Lage sind, mit unpräzisen Informationen umzugehen, wie sie im Rahmen einer Planung ermittelt werden. Zusätzlich ermöglichen die beschriebenen Ansätze zur Identifikation eine experimentelle Ermittlung eines Prozessmodells, was mit den analytischen Modellierungsmethoden der bisherigen Planungssprachen nicht möglich ist. Schließlich zeigt der Einsatz von Fuzzy-Systemen im Bereich der Regelung auch, dass Fuzzy-Systeme auch in schnellen Echtzeitsystemen eingesetzt werden können, sodass eine Auswertung der kontinuierlichen Qualität der Aktionen eines Plans mit geringem zeitlichem Aufwand verbunden ist.

Auf der anderen Seite steht mit der Optimalen Steuerung ein Verfahren zur Verfügung, um mit entsprechenden Informationen die Trajektorie einer kontinuierlichen Aktion (oder die Parameter eines Reglers, der die Trajektorie regelt) im Voraus so festzulegen, dass sie optimal bei einer gegebenen Zielfunktion ist. Insbesondere die Verfahren zur Optimalen Steuerung eignen sich mit ihren Laufzeiten aber nicht, um während des Betriebs des Systems bestimmt zu werden.

Damit lassen sich zwei Lösungsansätze zur Integration kontinuierlicher Prozesse in das PDDL-Planungsproblem identifizieren: eine vorgelagerte Optimierung oder eine Fuzzy-Approximation des Verhaltens. Beide Ansätze besitzen sehr unterschiedliche Eigenschaften und Anforderungen, sodass sie für unterschiedliche Anwendungsgebiete geeignet sind. Optimale Steuerung ermöglicht es zwar, einen kontinuierlichen Verlauf zu bestimmen, der optimal hinsichtlich einer bestimmten Kostenfunktion ist, stellt aber sehr hohe Anforderungen an die Modellierungsgenauigkeit. Die Fuzzy-Approximation bietet auf der anderen Seite zwar eine gute Möglichkeit, um mit ungenauen und unscharfen Informationen umzugehen, jedoch ist eine Optimierung des kontinuierlichen Verlaufs nicht ohne Weiteres möglich. Die folgenden Absätze beschreiben eine Modellierung der kontinuierlichen Verläufe auf Basis der Optimalen Steuerung bzw. kontinuierlicher Mehrzieloptimierung einerseits und auf Basis von Fuzzy-Approximation andererseits.

## 3.2 Integration kontinuierlicher Verläufe in die Planung

Die nächsten beiden Unterkapitel beschreiben zwei alternative Vorgehensweisen zur Integration kontinuierlicher Verläufe in die diskrete Planung. Die erste Alternative geht von einer vorgelagerten Modellierung mithilfe der Optimalen Steuerung oder Mehrzieloptimierung aus. Für andere Anwendungsfälle ist die Darstellung kontinuierlicher Verläufe mithilfe von Fuzzy-Approximation besser geeignet, insbesondere wenn die Optimierungsergebnisse sehr stark von kleinen Veränderungen der Umwelteinflüsse abhängen.

### 3.2.1 Modellierung kontinuierlicher Verläufe mithilfe vorgelagerter Optimierung

Die Methoden der Optimalen Steuerung und der kontinuierlichen Mehrzieloptimierung bieten die Möglichkeit, bei bekannten Umwelteinflüssen einen kontinuierlichen Verlauf von Parametern bzw. eine Trajektorie im Voraus zu bestimmen. Offensichtlich ist es nicht möglich oder sinnvoll, im Voraus für alle denkbaren Umwelteinflüsse (eine möglicherweise unendliche Menge) die optimale Trajektorie zu berechnen und als Teilfunktionslösungen im Planungsproblem aufzunehmen.

Ein möglicher Ansatz besteht darin, die optimalen kontinuierlichen Verläufe nur für einige Kombinationen von Umwelteinflüssen zu bestimmen. Wendet man die optimalen kontinuierlichen Verläufe in Situationen an, die ähnliche Eigenschaften haben, ist zu erwarten, dass auch ähnliche Ergebnisse erzielt werden.

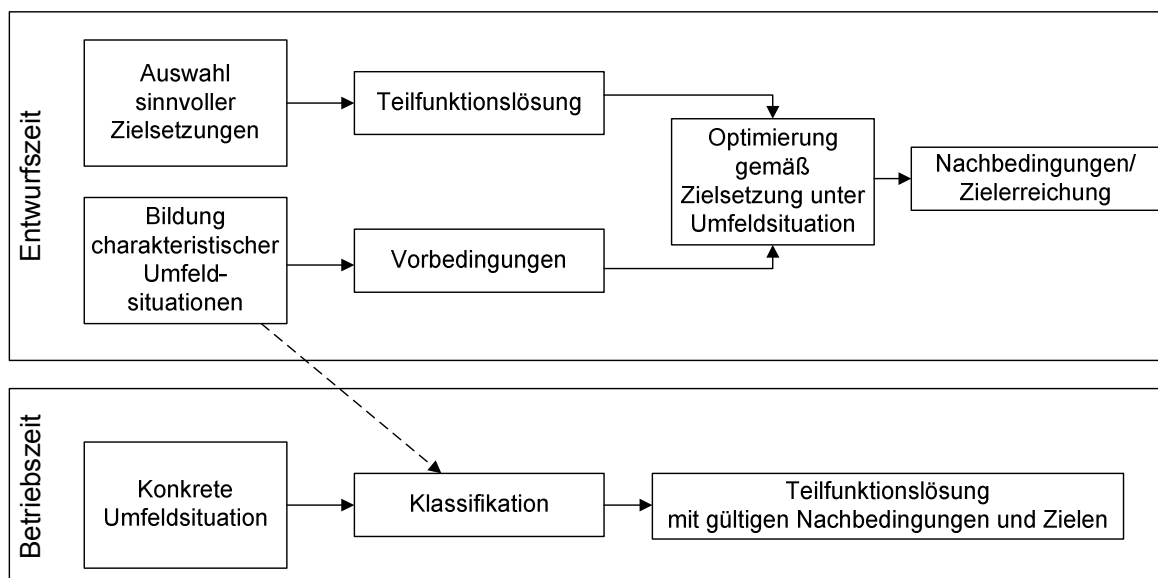


Abbildung 3-1: Modellierung kontinuierlicher Verläufe mit vorgelagerter Optimierung

Abbildung 3-1 veranschaulicht das in den folgenden Absätzen beschriebene Vorgehen. Ausgangsdaten der Integration kontinuierlicher Verläufe sind die Auswahl einer Menge sinnvoller Zielsetzungen (Kostenfunktionen im vorgelagerten Optimierungsproblem) und einer Menge charakteristischer Umfandsituationen. Die sinnvollen Zielsetzungen stellen bei Integration in das Planungsmodell die betrachteten Teilfunktionslösungen dar. Die charakteristischen Umfandsituationen bilden die möglichen Vorbedingungen, die an die Ausführung der Teilfunktionslösung geknüpft werden. Durch Lösung des entsprechenden Optimierungsproblems werden die Eigenschaften der Teilfunktionslösung hinsichtlich ihrer Qualität bestimmt, außerdem ergeben sich die Nachbedingungen (der vollständige technische Prozessablauf ist ja in Form

von Differenzialgleichungen beschrieben, damit lässt sich der Systemzustand am Ende einfach „ablesen“).

Indem eine konkrete Umfeldsituation der charakteristischen Umfeldsituation zugeordnet wird, die ihr am besten entspricht, werden zu jeder möglichen Teilfunktionslösung (also zu jeder sinnvollen Zielkombination) die entsprechenden Nachbedingungen ermittelt. Diese können dann in der Planung genutzt werden, um den Zustandsraum aufzubauen.

### 3.2.1.1 Klassifizierung von Umwelteinflüssen

Während die Bestimmung der sinnvollen Zielsetzung und der charakteristischen Umfeldsituation der eigentlichen Planung vorgelagert ist und damit grundsätzlich im Entwurf des Systems gelöst werden kann, muss die Klassifizierung der konkreten Umfeldsituationen zur Betriebszeit, also durch das System selbst, erfolgen. Um die Zuordnung durchzuführen, bietet sich die Anwendung eines Klassifikators an. Im maschinellen Lernen wird als Klassifizierung die Aufgabe verstanden, Datensätze mit ähnlichen Eigenschaften in Klassen einzuordnen. Es bietet sich an, die charakteristischen Umfeldsituationen als Klassen aufzufassen. Die Aufgabe eines beliebigen Klassifikators ist es damit, einer gegebenen konkreten Umweltsituation die ihr am besten entsprechende charakteristische Umweltsituation zuzuordnen.

Die meisten Klassifikatoren benötigen dafür die Definition eines Distanzmaßes und eine Menge von Trainingsdaten, also eine Menge erfolgreicher und richtiger Klassifikationen [Mit97]. Ein Distanzmaß ist eine Funktion  $d: M \times M \rightarrow \mathbb{R}^+$ , die die folgenden Eigenschaften erfüllt:

- $\forall i, j \in M, i \neq j: d(i, j) > 0$ ,
- $\forall i \in M: d(i, i) = 0$  und
- $\forall i, j \in M: d(i, j) = d(j, i)$ .

Allgemein kann festgehalten werden, dass zur Beschreibung der Umweltsituationen ein Vektor  $v$  zu definieren ist, der die relevanten Umgebungseigenschaften für die Anwendung der Teilfunktionslösung widerspiegelt. Durch Bestimmung einer Abstandsfunktion zwischen den einzelnen Elementen des Vektors und der Gewichtung der Elemente untereinander kann ein Distanzmaß definiert werden.

Ist dies geschehen, kann ein entsprechender Klassifikator zur Abbildung genutzt werden. Ein Beispiel wäre ein *Naive Bayes Classifier* [Mit97] oder Fuzzy-Klassifikator. Da die Eignung eines Klassifikators für ein Problem sehr stark von der Probleminstanz (in diesem Fall, den konkreten Umweltsituationen) und dem gewählten Distanzmaß abhängt, kann keine allgemeine Empfehlung getroffen werden. Es muss für das jeweilige zu beplanende System die beste Konfiguration gewählt werden.

### 3.2.1.2 Einbindung in die diskreten Planungsverfahren

Die Einbindung der Modellierung kontinuierlicher Verläufe mittels vorgelagerter Optimierung ist vergleichsweise einfach. Da die Formulierung von Systemzielen genutzt wurde, um Teilfunktionslösungen als Lösungen eines Optimierungsproblems unter wechselnden Umweltbedingungen (also eigentlich einer Menge von Optimierungsproblemen) darzustellen, kann jeder Teilfunktionslösung abhängig von der Umweltsituation auch ein kontinuierlicher Verlauf zugeordnet werden. Gleichzeitig sind durch die Lösung des Optimierungsproblems die gewünschten Qualitätsmerkmale des kontinuierlichen Prozesses bestimmt.



### 3.2.2 Modellierung kontinuierlicher Verläufe mittels Fuzzy-Approximation

Abbildung 3-2 zeigt das grundsätzliche Schema der Modellierung kontinuierlicher Verläufe mittels Fuzzy-Approximation. Ausgang der Modellierung sind hierbei mögliche technische Realisierungen. Dies können beispielsweise unterschiedliche Reglervarianten (z. B. prädiktive Regelung oder nicht) oder unterschiedliche Betriebsstrategien sein. Die zweite Eingangsgröße stellt ein Umfeldmodell dar, das dem der diskreten Planung entsprechen sollte. Beide Eingangsmengen sind in geeigneter Art zu fuzzifizieren und es ist ein Fuzzy-Regelsystem aufzustellen, das beschreibt, welche kontinuierlichen Eigenschaften und Nachbedingungen sich ergeben, wenn eine bestimmte technische Realisierung unter bestimmten Umgebungsbedingungen ausgeführt wird.

Bei diesem Vorgehen würden also beispielsweise bestimmte Dämpfereinstellungen und Aktoaktivitäten der aktiven Federung als Teilfunktionslösung betrachtet und bzgl. variierender Umwelteinflüsse hinsichtlich des Komforts beurteilt.

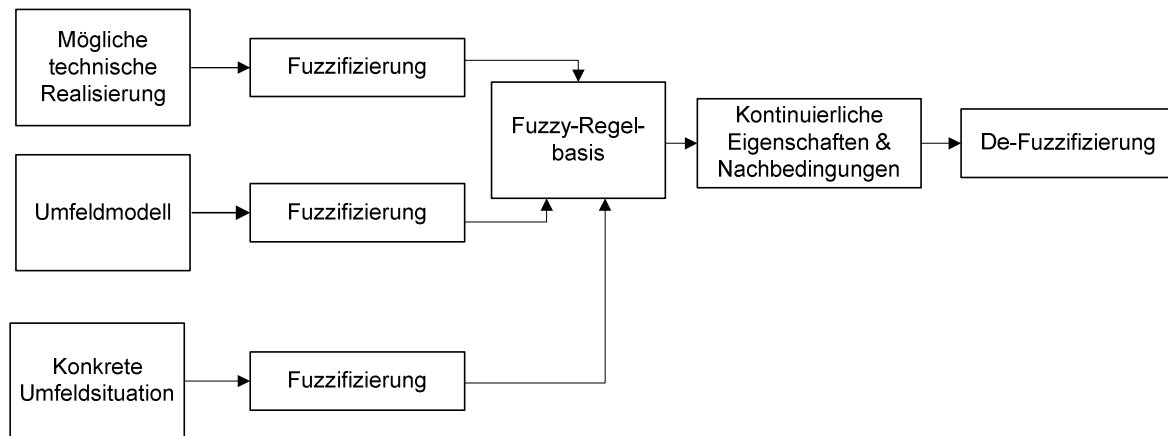


Abbildung 3-2: Modellierung kontinuierlicher Verläufe mittels Fuzzy-Approximation

Der Bezug zum konkreten Planungsproblem wird wie bereits in Unterkapitel 3.2.1 durch eine konkrete Umweltsituation bestimmt, die im Rahmen eines diskreten Planungsprozesses als Zustand in einem Plan bestimmt wird. Um die Verbindung zurück zu diesen Planungsverfahren zu bilden, müssen die Fuzzy-Ausgänge der Fuzzy-Regelbasis wieder auf scharfe Werte abgebildet (defuzzifiziert) werden.

#### 3.2.2.1 Fuzzifizierung der relevanten Planungsparameter

Zur Darstellung der Planungsvariablen als Fuzzy-Variablen wird eine ergänzende Modellierung zur PDDL erstellt. Dabei ist eine Fuzzy-Darstellung nur für numerische Variablen sinnvoll. Logische Zustandsvariablen werden als Singletons<sup>27</sup> behandelt, da sie nur genau einen der Werte *Wahr* oder *Falsch* haben können.

Für jede numerische Funktion der Planungsdomäne ist damit zunächst eine Menge linguistischer Terme aufzustellen, die den gesamten Wertebereich der Funktion abdecken. Dabei ist wiederum der Wertebereich im Rahmen der ergänzenden Modellierung genauer zu modellieren, als das in der PDDL der Fall ist. Als letzten Teil der ergänzenden Domänen sind die den linguistischen Termen entsprechenden Fuzzymengen (also die Zugehörigkeitsfunktionen) zu definieren. Listing 3-1 zeigt die Bestandteile der Ergänzungen der PDDL-Domänendefinition.

<sup>27</sup> Singletons sind Zustandsfunktionen, deren Zugehörigkeitsfunktion nur an einer Stelle 1 und sonst immer 0 ist.

Mit der Ergänzung der Domänendefinition um eine Darstellung der Planungsparameter in Form linguistischer Terme ist die Grundlage zur Anwendung eines Fuzzy-Expertensystems zur Bewertung der kontinuierlichen Aspekte einer Aktion gelegt. Die Menge der Fuzzy-Functions, die aus der Definition der Planungsdomäne abgeleitet wurden, wird im Weiteren als PV (Planvariablen) bezeichnet.

```
(:fuzzyFunctions
  (function1
    (:Universe U1)
    (:linguisticTerms
      (term1,1 μ1,1:U1 → [0,1])
      ...
      (term1,m μ1,m:U1 → [0,1])
    )
  )
  ...
  (functionn
    (:Universe Un)
    (:linguisticTerms
      (termn,1 μn,1:Un → [0,1])
      ...
      (termn,m μn,m:Un → [0,1])
    )
  )
)
```

**Listing 3-1: Fuzzy Definition der numerischen Variablen**

### 3.2.2.2 Fuzzifizierung der Umgebungsvariablen

Die relevanten Eigenschaften der Umgebung sind in das Fuzzy-Domänenmodell aufzunehmen, soweit dies nicht bereits mit der Fuzzy-Definition der numerischen Variablen geschehen ist. Die Notation entspricht dabei vollständig Listing 3-1. Die Menge der lokalen (also umgebungsspezifischen) Fuzzy-Variablen wird in Anlehnung an die fuzzifizierten numerischen Planvariablen als Umgebungsvariablen (UV) bezeichnet.

Es besteht hier die Möglichkeit, aktuelle Informationen aus Umgebungssystemen abzufragen. Dabei bietet sich eine Einbindung des planenden Systems in ein Multiagenten-System an, in dem Expertenagenten Fuzzy-Informationen über die aktuelle Ausführungsumgebung bereitstellen. Dabei können den Expertensystemen Informationen aus der diskreten Planung in Form der fuzzifizierten Planinformationen bereitgestellt werden. Die interne Struktur der Informationsverarbeitung innerhalb des Fuzzy-Expertensystems ist für die Planung bedeutungslos. Jedoch bieten sich insbesondere die in Abschnitt 3.1.1.1 erwähnten Verfahren zur Fuzzy-Identifikation kontinuierlicher Prozesse an, um das Expertenwissen zu akquirieren und aktuell zu halten.

Das Planungssystem kann mit jedem Expertensystem arbeiten, dessen erforderliche Eingabedaten sich auf die Planungsvariablen der diskreten Planung (PV) beschränken und das eine der als entscheidungsrelevant definierten Umgebungsvariablen (UV) zurückliefert.

### 3.2.2.3 Bewertung der kontinuierlichen Qualität einer Aktion

Für jedes mögliche kontinuierliche Qualitätsziel (z. B. Komfort) wird für jede Teilfunktionslösung ein eigenes Fuzzy-Expertensystem (FES) eingeführt. Das FES soll aus der Menge der Planungs- und Umgebungsvariablen auf das kontinuierliche Ziel der Aktivität abgebildet werden. Dazu ist für jedes Ziel, das kontinuierlich über den Verlauf einer Aktion erfasst wird,

eine linguistische Variable zu definieren. Auch hier kann wieder die Notation aus Listing 3-1 genutzt werden.

Jedes Fuzzy-Expertensystem stellt eine Abbildung der Form  $y = f(x)$  dar. Obwohl sich FES grundsätzlich eignen, jede beliebige Funktion abzubilden, kann die Definition der notwendigen Fuzzy-Regeln durch eine Aufteilung auf mehrere Fuzzy-Systeme erheblich erleichtert werden. Zunächst ist es zweckmäßig, die Teilfunktionslösungen des technischen Systems einzeln zu modellieren. Dadurch kann die Menge der betrachteten Plan- und Umfeldvariablen beschränkt werden auf diejenigen, die tatsächlich auf die jeweilige Teilfunktionslösung wirken. Eine weitere solche Unterteilung kann dadurch erfolgen, dass für jedes der  $n$  Ziele ein separates FES aufgestellt wird. Hier gilt wiederum, dass das Aufstellen der Regeln erheblich erleichtert wird und nur solche Ziele in die Inferenz einbezogen werden, die tatsächlich relevant sind. Die Relevanz kann sich dabei aus der aktuellen Gewichtung ergeben (einzelne Ziele haben eine Gewichtung von 0) oder daraus, dass eine Aktivität nicht auf ein bestimmtes Ziel wirkt.

Bei der Definition eines FES für die Bestimmung der Wirkung einer Aktivität hinsichtlich eines bestimmten Ziels sind zunächst die relevanten Eingangsgrößen zu bestimmen. Diese sind eine Teilmenge der Vereinigung aus  $PV \cup UV$ . Aufgrund der Zerteilung der Analyse auf mehrere FES besitzt jedes Expertensystem nur eine einzige Ausgabevariable: Der Grad, mit dem das entsprechende Ziel erreicht wird. Die Definition der Fuzzy-Regeln muss natürlich immer auf Basis des konkreten Anwendungsfalls erfolgen. Die Regeln haben grundsätzlich die folgende Form:

*if  $PV_1$  is term $PV_{1,i}$  and ...  $PV_n$  is term $PV_{n,j}$  and  
 $UV_1$  is term $UV_{1,k}$  and ...  $UV_m$  is term $UV_{m,l}$   
 Then  $y$  is  $O_x$*

Der if-Teil der Regel zerfällt in zwei Bestandteile: Zunächst werden die fuzzyfizzierten Planvariablen referenziert, im zweiten Teil die Umgebungsvariablen. Der Ausgabeteil enthält nur die linguistische Ausgabevariable. Durch Defuzzifizierung wird am Ende eine reelle Zahl aus dem Wertebereich des jeweiligen Ziels bestimmt.

### 3.2.2.4 Einbindung in die diskreten Planungsverfahren

Zur Einbindung der Fuzzy-Approximation bieten sich zwei unterschiedliche Möglichkeiten. Zunächst kann die Fuzzy-Approximation direkt im diskreten Planungsprozess eingesetzt werden, um die kontinuierlichen Eigenschaften einer Aktivität direkt in die Plankonstruktion einzubeziehen. Dies ändert im Wesentlichen nichts am Ablauf der Planung, lediglich die Schritte der Fuzzifizierung und Defuzzifizierung sind in jedem Planungsschritt durchzuführen. Da dieses Vorgehen eine enge technische Verkopplung des Fuzzy-Systems und der Planung vorsieht, kommt dieser Ansatz nicht in Betracht, wenn Standardplanungskomponenten eingesetzt werden sollen.

Ein alternativer Ansatz wäre (ähnlich wie bei [Dear05], siehe Abschnitt 3.1.1.2) ein Analyseprozess, der sich an die eigentliche Planung anschließt. Dies bietet sich insbesondere dann an, wenn in der Analyse aktuellere Informationen über die Ausführungsumgebung einer Aktion vorliegen als zur Planungszeit.

Insgesamt ergibt sich der in Abbildung 3-3 dargestellte Ablauf. Ergebnis der diskreten Planung ist eine Menge zeitlich vollständig geordneter Aktivitäten. Diese Aktivitäten werden unter Berücksichtigung zusätzlicher unscharfer Umgebungsinformation hinsichtlich ihrer kon-

tinuierlichen Eigenschaften beurteilt. Anhand dieser „Fuzzy-Bewertung“ werden die Aktivitäten aus dem Plan gewählt, die hinsichtlich ihrer kontinuierlichen Eigenschaften nicht akzeptabel sind. Sie werden in einer *Tabu-Liste* zusammengefasst, die alle Aktivitäten enthält, die im weiteren Planungsprozess nicht eingesetzt werden dürfen. Diese Aktionen werden mittels einer Anpassungsplanung aus dem Plan entfernt. Dazu wird ein Planungsproblem generiert, das sich an den Schritten vor und nach der betreffenden Aktivität orientiert und die entsprechende Aktivität *tabu* setzt.

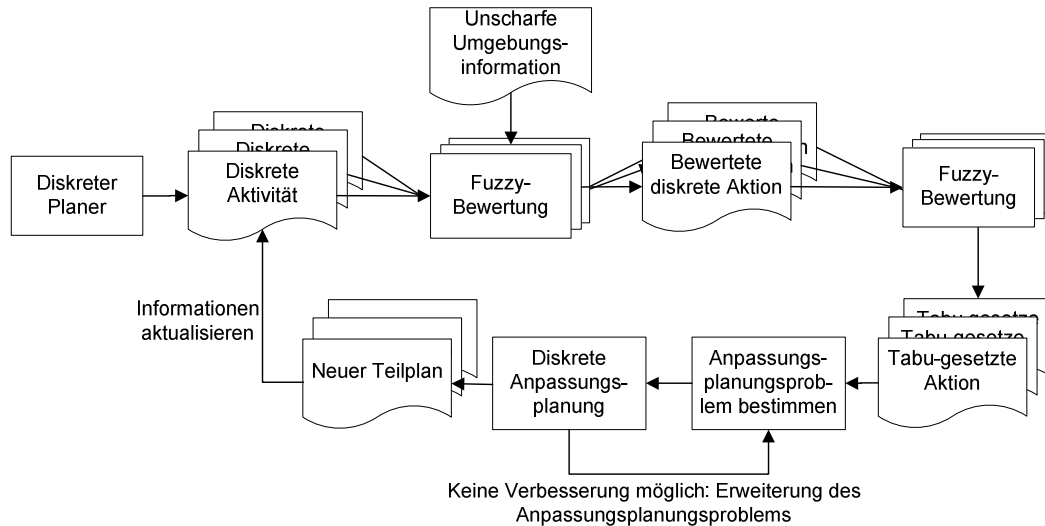


Abbildung 3-3: Integrationsszenario für Fuzzy-Approximation

In einer ersten Iteration kann beispielsweise ein Plan konstruiert werden, der direkt vom Ausgangszustand der Aktivität in den Endzustand der Aktivität führt. Gelingt dies nicht, so ist das Anpassungsplanungsproblem schrittweise auf weitere Vorgänger und Nachfolger im Plan zu erweitern. Der Endzustand im Plan und der Ausgangszustand des Plans stellen dabei Terminierungspunkte für das Verfahren dar.

Schlägt die Konstruktion eines Anpassungsplans völlig fehl, können einzelne Aktivitäten aus der Tabu-Liste entfernt werden, um das Anpassungsplanungsproblem zu vereinfachen. Dadurch kann ggf. doch noch ein gültiger Plan gefunden werden, der besser als der Ausgangsplan ist.

### 3.2.3 Fazit zur Integration kontinuierlicher Prozesse

In diesem Kapitel wurden zwei alternative Möglichkeiten zur Integration kontinuierlicher Verläufe in PDDL-Planungsprobleme vorgeschlagen. Die Modellierung kontinuierlicher Verläufe mithilfe vorgelagerter Optimierung ermöglicht es, kontinuierliche Parameter und Prozesse direkt mit konventionellen PDDL-Methoden zu verknüpfen. Daher verursacht diese Methode vor allem zusätzlichen Aufwand in der Modellierung des Planungsproblems, während in der eigentlichen Planung nur ein geringer Mehraufwand durch die Klassifizierung der Umweltzustände entsteht. Dafür stellt das Vorgehen aber auch hohe Anforderungen an die Vorhersagbarkeit des System- und Umfeldverhaltens.

Die Modellierung mithilfe von Fuzzy-Approximation verursacht deutlich mehr Aufwand in der eigentlichen Planung. Dafür ist die Methode auch für solche Situationen geeignet, in denen das System- und das Umfeldverhalten nur in groben Zügen vorhersagbar sind. In diesem Sinne ergänzen sich die beiden Methoden und bieten sich für unterschiedliche Anwendungsszenarien an.

Beiden Methoden ist gemeinsam, dass im eigentlichen Planungsprozess PDDL-Modelle mit normalen Aktionen verwendet werden können. Weder dauerhafte noch kontinuierliche Aktionen müssen verwendet werden. Dadurch steht zunächst ein breiteres Spektrum an Planungsalgorithmen zur Verfügung. Andererseits kann auch in den Problemfeldern *Planung unter Unsicherheit* und *Interaktionsplanung* die Betrachtung auf normale PDDL-Aktionen beschränkt bleiben.

Da beide Alternativen davon ausgehen, dass die kontinuierliche Modellierung der Prozesse dem Planungsprozess vorgelagert ist, wird die Abweichung des tatsächlichen Ausführungsverlaufs vom geplanten Verlauf die Regel sein. Das muss sich in dem im folgenden Kapitel erarbeiteten Konzept zur Behandlung von Unsicherheit in der Verhaltensplanung für mechatronische Systeme niederschlagen. Das Verfahren muss in der Lage sein, wesentliche und unwesentliche Abweichungen voneinander zu unterscheiden. Trotzdem sollte frühzeitig auf relevante Abweichungen reagiert werden, um Fehlersituationen zu vermeiden.



## 4 Planung unter Unsicherheit

Ein Beispiel für die in der Verhaltensplanung auftretende Unsicherheit liefert das Energiemanagement des RailCabs. Die Planungsaufgabe im Energiemanagement besteht darin zu entscheiden, welche Teilfunktionslösungen (mit resultierendem Leistungsbedarf oder -überschuss) auf welchem Streckenabschnitt gewählt werden können und auf welchen Streckenabschnitten der Energiespeicher geladen oder entladen wird.

Die alternativen Teilfunktionslösungen ergeben sich dabei aus den Fähigkeiten der einzelnen Teilmodule. So können beispielsweise, wie in Unterkapitel 2.1.3 beschrieben, die Federungsarten Passiv, Aktiv und Störgrößenkompensation unterschieden werden. Die Teilfunktionslösungen unterscheiden sich dabei hinsichtlich einer Gütefunktion (Komfort) und ihres Leistungsbedarfs. Tendenziell geht eine hohe Güte mit einem erhöhten Leistungsbedarf einher. Das Verhältnis von erreichter Güte zu Leistungsbedarf hängt erheblich von Streckeneigenschaften ab. Hier liegt das Potenzial der Planung: Leistung auf den Streckenabschnitten einsparen, auf denen mit hohem Leistungsbedarf nur ein geringer Gütegewinn einhergeht, um auf anderen Streckenabschnitten die gespeicherte Leistung effizienter einzusetzen. Weiterhin hilft die Planung, die für den Betrieb notwendige Mindestleistung sicherzustellen.

Der Leistungsbedarf der Module und die aus dem Antrieb übertragbare Leistung hängen von einer Vielzahl von Umwelteinflüssen ab, die im Voraus nicht mit absoluter Sicherheit bekannt sind. Beispiele für Umwelteinflüsse auf den Antrieb und damit auf die maximal ins Bordnetz übertragbare Leistung sind zum Beispiel Gegenwind und vor allem defekte Statoren. Der Leistungsbedarf der aktiven Federungen hängt neben den Streckeneigenschaften (die als verlässlich bekannt betrachtet werden können) vor allem von Witterungseinflüssen ab. So erzeugen Böen unvorhergesehene Anregungen, die ausgeglichen werden müssen. Niederschläge und damit einhergehende rutschige Schienen führen zu erhöhter Wahrscheinlichkeit von Spurkranzanläufen, die ebenfalls abgefedert werden müssen.

Insgesamt ist damit ein deterministisches Modell des Leistungsangebots und des Leistungsbedarfs nicht realistisch. Im Sinne der in Unterkapitel 2.3 beschriebenen Zielsetzung soll der Plan so erstellt werden, dass für besonders wahrscheinliche alternative Zustände eine Verzweigung im Plan vorgesehen wird.

Die nun folgenden Abschnitte betrachten unterschiedliche Ansätze zur Modellierung und Planung derartiger unsicherer Probleme. Auf Basis der gewonnenen Erkenntnisse wird dann ein Planungsverfahren konzipiert, das die in Unterkapitel 2.3 formulierten Anforderungen erfüllt.

### 4.1 *Stand der Technik zur Behandlung von Unsicherheit*

Im Rahmen dieses Kapitels werden unterschiedliche Ansätze verglichen, die zur Erfüllung der in Kapitel 2.3 beschriebenen Zielsetzung in Betracht kommen. Dabei wird zunächst auf die unterschiedlichen Möglichkeiten zur Modellierung von Unsicherheit in Planungsproblemen eingegangen. Die Modellierung wird dabei auf zwei Ebenen betrachtet: Zunächst werden allgemeine Beschreibungsformen zur Beurteilung unsicherer Situationen betrachtet. Auf der zweiten Ebene werden Beschreibungsformen für Pläne und Planungsprobleme untersucht, die eingesetzt werden können, wenn nur unvollständige Informationen über das Umfeld vorliegen und die Wirkung von Handlungen und Aktivitäten nicht eindeutig vorhersagbar ist. Anschließend werden Methoden vorgestellt, die unter diesen Bedingungen Pläne konstruieren können.

### 4.1.1 Modellierung von Unsicherheit

In diesem Kapitel werden unterschiedliche Glaubensmaße betrachtet, um ihre Eignung als Maß zur Beurteilung der Eintrittsmöglichkeit alternativer Situationen während der Planausführung zu bestimmen. Ein Glaubensmaß beschreibt dabei das Zutrauen in die Wahrheit einer getroffenen Aussage. Die Grundlagen der betrachteten Glaubensmaße sind im Einzelnen die Wahrscheinlichkeitstheorie und die Dempster-Shafer-Theorie. Die ebenfalls häufig als Glaubensmaß beschriebene Fuzzy-Theorie beschreibt eher eine Art von Wissen, das zwar ungenau, aber doch sicher ist.<sup>28</sup>

#### 4.1.1.1 Wahrscheinlichkeiten

Das grundlegende Element der Wahrscheinlichkeitstheorie ist die Zufallsvariable. Dies ist eine Variable, die unterschiedliche, sich ausschließende Werte annehmen kann. Sie ist Teil einer Beschreibung einer Umwelt und ihr Wert ist zunächst unbekannt.<sup>29</sup>

Durch die Zuweisung eines konkreten Wertes zur Zufallsvariablen erhält man eine Aussage. Einer solchen Aussage können mithilfe der Wahrscheinlichkeitstheorie Glaubensgrade zugeordnet werden. Ein Elementarereignis erhält man durch Zuweisung eines konkreten Werts zu jeder Variablen des betrachteten Wahrscheinlichkeitsmodells. Ein Elementarereignis entspricht daher genau einem möglichen Zustand des Wahrscheinlichkeitsmodells. Alle Elementarereignisse schließen sich gegenseitig aus. Den Elementarereignissen können ebenso wie den Zufallsvariablen Glaubensgrade zugeordnet werden. Drei nach Kolmogorov benannte Axiome [Kolg50] stellen die mathematische Grundlage der Wahrscheinlichkeitstheorie dar:

1.  $0 \leq P(x) \leq 1$ : Die Wahrscheinlichkeit einer Aussage liegt zwischen 0 und 1,
2.  $P(\text{true}) = 1$  und  $P(\text{false}) = 0$ : Die Wahrscheinlichkeit einer wahren Aussage ist 1 und die Wahrscheinlichkeit einer falschen Aussage ist null,
3.  $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$ : Die Wahrscheinlichkeit für das Eintreten von a oder b ist gleich der Summe der Wahrscheinlichkeiten für a und b minus die Wahrscheinlichkeit für die Schnittmenge der beiden Ereignisse.

Da Elementarereignisse sich ebenso gegenseitig ausschließen wie die möglichen Werte einer Zufallsvariable, lässt sich für alle atomaren Ereignisse (AE) eines Zufallsmodells (M) aus den Kolmogorov-Axiomen folgern:  $\sum_{AE \in M} P(AE) = 1$ .

In der Wahrscheinlichkeitstheorie wird zwischen bedingten und unbedingten Wahrscheinlichkeiten unterschieden. Diese Unterscheidung spiegelt den Umstand wider, dass sich Glaubensgrade bei vorliegender Information verändern. Die unbedingte Wahrscheinlichkeit  $P(A = a)$  gibt an, als wie wahrscheinlich das Ereignis  $A = a$  erachtet wird, wenn keinerlei Information über das übrige Wahrscheinlichkeitsmodell vorliegt. Zusätzliche Information über eine Zufallsvariable B kann den Glaubensgrad signifikant verändern. Ist die zusätzliche Information über die Zufallsvariable B eine veränderte Wahrscheinlichkeitsverteilung, spricht man von einer weichen Evidenz. Ist die Ausprägung von B dagegen sicher bekannt, liegt eine harte Evidenz vor. Mit dem Auftreten einer Evidenz  $B = b$  hat die unbedingte Wahrscheinlichkeit keine Gültigkeit mehr. Sie wird durch eine bedingte Wahrscheinlichkeit der Form  $P(A = a | B = b)$  ersetzt. Mithilfe des Bayes-Theorems können bedingte Wahrscheinlichkeiten durch unbedingte Wahrscheinlichkeiten ausgedrückt werden [CGH97]:

<sup>28</sup> So ist beispielsweise die genaue Gradzahl einer Temperatur nicht bekannt, jedoch ist klar, dass es kalt ist. Zu weiteren Details zur Fuzzy-Theorie und Fuzzy-Logik siehe Abschnitt 3.1.1.1.

<sup>29</sup> Zufallsvariablen werden unterteilt in drei Gruppen: boolesche Variablen, diskrete Variablen und stetige Variablen. Während eine boolesche Zufallsvariable nur die Werte *wahr* oder *falsch* annehmen kann, können diskrete Zufallsvariablen eine begrenzte Menge von Werten und stetige Variablen alle Werte innerhalb eines Intervalls oder aus der gesamten Menge der reellen Zahlen annehmen. [Schw01]



- $P(a|b) = \frac{P(a \wedge b)}{P(b)}$  für  $P(b) > 0$ .

Durch Umstellung der Gleichung erhält man die Produktregel [RN03]:  $P(a \wedge b) = P(a|b)P(b)$ . Damit ist es ebenfalls möglich, unbedingte Wahrscheinlichkeiten auf bedingte Wahrscheinlichkeiten zurückzuführen.

Die Wahrscheinlichkeitstheorie stellt einen brauchbaren Ansatz zur Darstellung eines Glaubensmaßes bzgl. des Eintritts alternativer Planverläufe dar. Die bedingten Wahrscheinlichkeiten können die in Kapitel 2.3 aufgestellte Anforderung erfüllen, dass die Eintrittsmöglichkeit eines Zustands abhängig von seinem Vorgänger bestimmt werden muss. Die Einbeziehung von Informationen über die Umgebung ist ebenfalls möglich.

#### 4.1.1.2 Dempster-Shafer-Theorie

Die Dempster-Shafer-Theorie behandelt die Unterscheidung von Unsicherheit und Unkenntnis. Sie unterscheidet zwischen zwei korrespondierenden Maßen, dem Glaubwürdigkeitsmaß und dem Plausibilitätsmaß [RN03]. Beide Maße hängen in der Form zusammen, dass die Glaubwürdigkeit für die Richtigkeit einer Aussage und die Plausibilität der Komplementäraussage in Summe Eins ergeben müssen. Die konventionelle Wahrscheinlichkeit ist in diesem Sinne ein spezielles Glaubwürdigkeitsmaß, bei dem Wahrscheinlichkeit, Glaubwürdigkeit und Plausibilität einer Aussage immer gleich groß sind.

Ähnlich wie die Wahrscheinlichkeitstheorie geht auch die Dempster-Shafer-Theorie von einer Grundmenge an Möglichkeiten aus (*frame of discernment*, deutsch etwa *Rahmen des Urteilsvermögens*). Es handelt sich um eine Menge  $\Theta$  sich gegenseitig ausschließender Elemente. Je nach Domäne werden sie als Hypothesen, Aussagen, Diagnosen oder Label interpretiert. Die Menge der Aussagen muss vollständig sein. Die Potenzmenge stellt ein hierarchisches System von Aussageklassen dar, wobei eine Aussagenklasse eine disjunktive Verknüpfung feinerer bzw. kleiner Aussageklassen ist. Über dieses System von Aussagenklassen wird eine Massenfunktion definiert. Eine Funktion  $2^\Theta \rightarrow [0,1]$  ist eine Massenfunktion, wenn gilt:

$$\begin{aligned} m(\emptyset) &= 0 \\ \sum_{x \in 2^\Theta} m(x) &= 1 \end{aligned}$$

In der Dempster-Shafer-Theorie gibt eine solche Massenfunktion die *Unterstützung* für die Aussage  $X$  an. Ein wesentlicher Unterschied zwischen den Massenfunktionen der klassischen Wahrscheinlichkeitstheorie und der Dempster-Shafer-Theorie besteht in dem Umstand, dass bei der Dempster-Shafer-Theorie nicht die gesamte „Masse“ der Unterstützung auf die Elemente von  $\Theta$  bzw.  $2^\Theta$  verteilt werden muss. Ein Teil der Menge kann als Unwissen dargestellt werden und wird  $\Theta$  zugewiesen.

Aufbauend auf den Massenfunktionen werden zwei weitere Arten von Funktionen definiert. Die erste Funktion ist die *Glaubensfunktion* (*belief function*). Eine Funktion  $bel: 2^\Theta$  ist eine Glaubensfunktion, wenn  $Bel(X) = \sum_{A \subseteq X} m(A)$  gilt. Eine Glaubensfunktion misst gewissermaßen die Unterstützung für  $X$  durch Addition der Werte von  $m$  für alle Teilmengen von  $X$ . Die zweite Art von Funktionen sind die Plausibilitätsfunktionen. Eine Funktion  $pl: 2^\Theta \rightarrow [0,1]$  ist eine Plausibilitätsfunktion, wenn  $Pl(x) = \sum_{A \cap X \neq \emptyset} m(A)$  gilt. Sie bemisst das Maß an Glauben, das  $X$  maximal zugewiesen werden könnte (unabhängig von den konkreten Umständen). Mithilfe einer konkreten Belegung der Massenfunktion  $m$  kann das Intervall  $[Bel, Pl]$  berechnet werden. Dabei gilt immer  $0 \leq Bel \leq Pl \leq 1$ . Die Intervallbreite  $Pl - Bel$  repräsentiert das Unwissen (*uncommitted belief*, *second order uncertainty*). Je breiter das Inter-

vall ist, desto weniger vertrauenswürdig ist eine Aussage, eine Intervallbreite von 1 steht dabei für vollständige Unwissenheit.

Dempster definierte eine wesentliche Rechenregel für das Rechnen mit Massenfunktionen. Dempsters Kombinationsregel [BCM00] verrechnet zwei Massenfunktionen  $m_1$  und  $m_2$  zu einer neuen Massenfunktion:

$$m_1 \oplus m_2(A) = \begin{cases} 0 & \text{wenn } A = \emptyset \\ \frac{\sum_{A_i \cap B_j = A} m_1(A_i) \cdot m_2(B_j)}{\sum_{A_i \cap B_j = \emptyset} m_1(A_i) \cdot m_2(B_j)} & \text{sonst} \end{cases}$$

Mit dieser Regel ist es möglich, zwei oder mehr Wissensquellen normalisiert zusammenzutragen. Problematisch an dieser Kombinationsregel ist allerdings, dass sie anders als die Wahrscheinlichkeitsrechnung nicht aus einer Menge von zuvor definierten Axiomen abgeleitet wurde. Für einige Anwendungsbereiche lässt sich diese Regel aber leicht interpretieren. Sie stellt in gewisser Weise eine Verallgemeinerung des Bayes-Theorem dar und fügt der Dempster-Shafer-Theorie eine dynamische Komponente hinzu. Die Beurteilung der Möglichkeit eines Ereignisses oder einer Aussage verändert sich abhängig vom aktuell vorliegenden Wissen.

Mit der Kombinationsregel von Dempster ist es auch in der Dempster-Shafer-Theorie möglich, flexibel neue Informationsquellen zur Bestimmung des Glaubensmaßes heranzuziehen. Ebenso ist die Beurteilung der Eintrittsmöglichkeit eines Zustands abhängig von der Eintrittsmöglichkeit der Vorgängerezustände. Obwohl die Dempster-Shafer-Theorie mit der Einführung von Unwissen (uncommitted belief) und dessen Quantifizierung eine interessante Eigenschaft besitzt, fehlen im Vergleich zur Wahrscheinlichkeitstheorie Konzepte und Methoden, um diese Darstellung von Unsicherheit effizient zu modellieren und zu verarbeiten.

## 4.1.2 Modelle und Verfahren zur Analyse von Unsicherheit

Die Darstellung und Quantifizierung von Unsicherheit mithilfe von Glaubensmaßen ist eine wichtige Voraussetzung für eine Planung unter Unsicherheit. Die Quantifizierung der Unsicherheit kann genutzt werden, um den Raum möglicher Ausführungsverläufe einzuschränken und relevante und irrelevante Alternativen zu differenzieren. Um dieses Ziel effektiv zu erreichen, muss der Glaubensgrad aber soviel Informationen wie möglich in die Quantifizierung der Unsicherheit mit einbeziehen. In diesem Kapitel werden Methoden beschrieben, die diese Einbeziehung aktueller Information zur Aktualisierung von Glaubensgraden ermöglichen. Dabei zeigt sich, dass nur für die Wahrscheinlichkeitstheorie derartige Methoden zur Verfügung stehen.

### 4.1.2.1 Bayes-Netzwerke

Ein Bayes-Netzwerk ist ein gerichteter Graph, der die Abhängigkeiten zwischen Zufallsvariablen einer modellierten Domäne darstellt. Nach Jensen [Jens01] ist ein Bayes-Netzwerk wie folgt aufgebaut:

- Die Menge der Knoten des Netzwerks bilden die Variablen ab.
- Zwei Knoten können durch eine gerichtete Kante verbunden sein.
- Knoten A ist Elternknoten von Knoten B, wenn eine gerichtete Kante von A nach B existiert.
- Zu jeder Variable A existiert eine bedingte Wahrscheinlichkeitsverteilung  $P(A|Eltern(A))$ .
- Der Graph enthält keine Zyklen, ist also ein gerichteter azyklischer Graph (DAG).

Ein wesentlicher Vorzug der Bayes-Netze besteht darin, dass sie auch stetige Zufallsvariablen berücksichtigen können. Grundsätzlich gibt es zwei Möglichkeiten, stetige Variablen zu behandeln. Sie können, unabhängig von der ihnen zugrunde liegenden Verteilung, diskretisiert werden. Dazu werden sie durch Einteilung in Intervalle in diskrete Variablen verwandelt. Der Nachteil dieses Vorgehens liegt in dem Verlust von Genauigkeit. Über die Wahrscheinlichkeit von Werten innerhalb eines Intervalls ist keine vernünftige Aussage mehr möglich. Versucht man, diesen Nachteil durch eine sehr feine Diskretisierung (also Einteilung in sehr kleine Intervalle) auszugleichen, führt das zu sehr großen diskreten Wahrscheinlichkeitsverteilungen. Cobb et. al. stellen eine Inferenzprozedur für hybride Netzwerke vor, die beliebige kontinuierliche Verteilungen approximieren kann [CRS07]. Basis der Inferenzprozedur ist ein Schließen über Exponentialverteilungen. Indem die Autoren nahezu beliebige kontinuierliche Verteilungen mittels einer Schar aus Exponentialverteilungen approximieren, heben sie viele Einschränkungen für die Modellierung mittels Bayes-Netzwerken auf (siehe z. B. [Jens01]). Die Approximation der kontinuierlichen Verteilungen kann dabei anhand von Trainingsdaten berechnet werden, sodass noch nicht einmal eine Kenntnis der exakten kontinuierlichen Verteilung notwendig ist.

Mithilfe eines Bayes-Netzwerkes kann die Wahrscheinlichkeit jedes atomaren Ereignisses in einer vollständigen gemeinsamen Wahrscheinlichkeitsverteilung bestimmt werden. Die Wahrscheinlichkeit berechnet sich nach der Formel  $P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{Eltern}(x_i))$ . Die bedingten Wahrscheinlichkeitsverteilungen an den Knoten des Netzwerks bilden damit eine verteilte Repräsentation der vollständigen gemeinsamen Wahrscheinlichkeitsverteilung. In der Regel werden Bayes-Netzwerke eingesetzt, um die a-posteriori-Wahrscheinlichkeitsverteilung einer Zufallsvariablen  $X$  in Abhängigkeit von den bekannten Evidenzen innerhalb der Domäne zu bestimmen.

Ein wichtiges Konzept für die Entwicklung effizienter Algorithmen zur Bestimmung der vollständigen gemeinsamen Wahrscheinlichkeitsverteilung in Bayes-Netzwerken ist die bedingte Unabhängigkeit. Anhand der bedingten Unabhängigkeit lässt sich ermitteln, welche Evidenzen  $R \subset E$  bei der Auswertung einer Abfrage  $P(X|E)$  berücksichtigt werden müssen. Pearl [Pearl88] definiert die bedingte Unabhängigkeit wie folgt:

- Sei  $U = \{\alpha, \beta, \dots\}$  eine endliche Menge diskreter Variablen. Sei  $P(\bullet)$  eine gemeinsame Wahrscheinlichkeitsfunktion über die Variablen  $U$  und seien  $X$ ,  $Y$  und  $Z$  drei Teilmengen der Variablen von  $U$ .  $X$  und  $Y$  werden dann bedingt unabhängig gegeben  $Z$  genannt, wenn  $P(x | y, z) = P(x | z)$  und  $P(y, z) > 0$ .

Um die bedingte Unabhängigkeit aus der Struktur eines Bayes-Netzwerks ableiten zu können, wird das graphentheoretische Kriterium der d-Separiertheit genutzt (siehe [Pearl88, S.127ff], [Nea04]). Durch die Einführung der bedingten Unabhängigkeit und der d-Separiertheit lässt sich die Berechnung der vollständigen gemeinsamen Wahrscheinlichkeitsverteilung umschreiben zu  $P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1)$ . Diese Formel wird als die Kettenregel für Bayes-Netze bezeichnet und bietet die Möglichkeit, durch Konzepte der Dynamischen Programmierung erheblich Laufzeit bei der Berechnung der vollständigen gemeinsamen Wahrscheinlichkeitsverteilung einzusparen.

Russel und Norvig [RN03] unterscheiden drei grundsätzliche Formen der Inferenz in Bayes-Netzwerken:

1. Diagnostische Inferenz versucht, von Symptomen oder Effekten auf die Ursachen zu schließen.

2. Kausale Inferenz versucht, bei Vorliegen bestimmter Ursachen das Auftreten damit verbundener Effekte vorherzusagen.
3. Interkausale Inferenz untersucht den Zusammenhang zwischen Ursachen, die den gleichen Effekt auslösen.
4. Gemischte Inferenz bezeichnet jede Kombination aus den drei zuvor genannten Inferenzen.

Gegenstand der Inferenz ist dabei oft nur eine einzelne Query-Variable, deren a-posteriori-Wahrscheinlichkeit ermittelt werden soll. Die Inferenz ist dabei grundsätzlich auf zwei unterschiedliche Arten möglich: exakte und approximative Inferenz. Eine wichtige Einschränkung für den Einsatz von exakter Inferenz besteht in der Laufzeit: Inferenz für Bayes-Netze ist im Allgemeinen NP-hart. Für den Sonderfall der Singly Connected Networks ist eine exakte Inferenz in Bayes-Netzwerken in linearer Zeit ausführbar. Ein Singly Connected Network ist ein Graph, in dem zwischen jedem Knotenpaar höchstens ein ungerichteter Weg existiert. Wesentliche Algorithmen zur exakten Inferenz sind der Message-Passing-Algorithmus von Pearl [Pearl88], dessen Erweiterung zur Inferenz in mehrfach verbundenen Netzwerken ([Nea04] oder [CGH97]) oder der Algorithmus zur Variableneliminierung [RN03]. Approximative Inferenz bestimmt eine bedingte Wahrscheinlichkeit näherungsweise mithilfe wiederholter Stichproben im Zufallsraum [CDL+03].

Eine wichtige Erweiterung von Bayes-Netzwerken sind die Dynamischen Bayes-Netzwerke (DBN). Diese Erweiterung ermöglicht die explizite Repräsentation von Zeit im probabilistischen Modell einer Domäne. Dem Betrachtungsbereich der Verhaltensplanung ist es inhärent, dass sich Variablen und Parameter im Zeitverlauf verändern. Die Veränderung der Zeit wird durch aufeinanderfolgende Zeitaufnahmen, Zeitscheiben, dargestellt [Jens01]. In jeder Zeitscheibe ist eine Menge an Zufallsvariablen enthalten, die entweder beobachtbar oder nicht beobachtbar sind. Die Menge der nicht beobachtbaren Zufallsvariablen in der t-ten Zeitscheibe wird in der Regel als  $X_t$  und die Menge der beobachtbaren Evidenzvariablen als  $E_t$  bezeichnet. Die Notation  $X_{i;j}$  bezeichnet alle Variablen in den Zeitscheiben  $i$  bis  $j$ .

Um einen Prozess in Form eines DBN darstellen zu können, sind eine Reihe von Annahmen zu treffen bzw. zu erfüllen. Zunächst muss der Prozess stationär sein: Er verändert die Zustände in jeder Zeitscheibe  $t$  nach bestimmten Gesetzmäßigkeiten, die über die Zeitdauer nicht veränderlich sind. Bei einem nicht stationären Prozess ließe sich ein DBN nicht konstruieren. Zusätzlich muss es sich bei dem modellierten Prozess um einen Markov-Prozess handeln. Ein Markov-Prozess ist ein Prozess, bei dem der aktuelle Zustand nur von einer endlichen Anzahl von Vorgängerzuständen abhängt. Man unterscheidet Markov-Prozesse unter anderem anhand der Anzahl relevanter Vorgängerzustände. Hängt der aktuelle Zustand von  $n$  Vorgängerzuständen ab, spricht man von einem Markov-Prozess  $n$ -ter Ordnung. So gilt beispielsweise für einen Markov-Prozess erster Ordnung:  $P(X_t|X_{0:t}) = P(X_t|X_{t-1})$ . Die Wahrscheinlichkeit  $P(X_t|X_{t-1})$  wird auch als Übergangswahrscheinlichkeit erster Ordnung bezeichnet. Auch die Evidenzvariablen  $E_t$  müssen hinsichtlich ihrer Verknüpfung mit dem Modell beschränkt werden. Es wird die Annahme getroffen, dass  $E_t$  nur von den Werten in der aktuellen Zeitscheibe abhängt, sodass  $P(E_t|X_{0:t}) = P(E_t|X_t)$  gegeben ist.  $P(E_t|X_t)$  wird als Sensormodell des Prozesses bezeichnet. Insgesamt ergibt sich folgende vollständige Wahrscheinlichkeitsverteilung für ein DBN:  $P(X_0, X_1, \dots, X_t, E_0, E_1, \dots, E_t)$ . Russel und Norvig [RN03] unterscheiden vier Formen der Inferenz in DBN:

1. Filtern: Berechnung der bedingten Wahrscheinlichkeit  $P(X_t|e_{1:t})$ , also die Bestimmung des aktuellen Glaubenszustands für die aktuelle Zeitscheibe;
2. Vorhersage: Bestimmung der Verteilung  $P(X_{t+k}|e_{1:t})$  mit  $k>0$ , also die Bestimmung des Glaubensgrads für eine in der Zukunft liegende Zeitscheibe;

3. Glättung: Bestimmung der bedingten Verteilung  $P(X_{t-k}|e_{1:t})$  mit  $k>0$ , also die Bestimmung eines Glaubensgrads für eine in der Vergangenheit liegende Zeitscheibe und
4. Wahrscheinlichste Erklärung: Bestimmung der wahrscheinlichsten Sequenz von Zuständen, die zum aktuellen Zustand geführt hat.

Im Folgenden werden die Probleme des Filterns und der Vorhersage näher erläutert, da sie für die prädiktive Bestimmung von Eintrittsmöglichkeiten in Plänen genutzt werden können. Der Prozess des Filterns geschieht für den Zustand zum Zeitpunkt  $t+1$  durch folgende rekursive Berechnung:  $P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})$ . Der erste Term der rechten Seite der Gleichung kann direkt dem Sensormodell entnommen werden. Der erste Term innerhalb der Summe ist durch das Übergangsmodell gegeben. Der zweite Term in der Summe ist die Zustandsverteilung des Vorgängerzustands  $t$ ,  $\alpha$  steht für eine Normalisierungskonstante.<sup>30</sup>

Vorhersagen können in DBNs einfach berechnet werden, indem gefiltert wird, ohne dass neue Evidenzen einbezogen werden. Eine Vorhersage für einen Zustand  $t+k+1$  lässt sich rekursiv aus der Zustandsverteilung für  $t+k$  berechnen. Führt man die Vorhersage in einem DBN immer weiter, so stellt man fest, dass irgendwann ein Zustand erreicht ist, in dem sich die vorausgesagten Zustandsverteilungen nicht mehr ändern. Diese Verteilung wird als stationäre Verteilung des Markov-Prozesses bezeichnet. Aus dieser Konvergenzeigenschaft von Markov-Prozessen folgt, dass eine Vorhersage nur für eine begrenzte Anzahl von Zeitscheiben sinnvoll möglich ist.

Eine weitere nützliche Eigenschaft der Methode der Bayes-Netzwerke ist das Vorliegen mehrerer Lernverfahren. Die Lernverfahren lassen sich in das Lernen von Parametern mit vollständigen und mit versteckten Daten sowie dem Strukturlernen einteilen.

Beim Lernen von Parametern bei vollständigen Informationen geht es darum, die bedingten Wahrscheinlichkeitsfunktionen in einem gegebenen Bayes-Netzwerk anhand einer Menge historischer Trainingsdaten zu bestimmen, die alle Variablen des Netzes umfassen. Zu dieser Klasse von Lernverfahren gehören u. a. die Maximum Likelihood Estimation [Mitt97, S.157], Naive Bayes-Klassifikatoren und das Parameter-Lernen mit Beta- bzw. Dirichlet-Verteilung [Heck95]. Die Maximum Likelihood Estimation bestimmt die Parameter des Netzwerks als die wahrscheinlichste Erklärung für Trainingsdaten. Nach dem gleichen Prinzip funktioniert der naive Bayes-Klassifikator, wobei hier vor allem die Bestimmung der Ausprägung einer Variablen (z. B. ist die E-Mail *Spam* oder nicht) anhand einer Menge von Kriterien im Vordergrund steht. Daher sind die Netzwerke grundsätzlich sternförmig aufgebaut. Beide Ansätze neigen bei nur wenigen Trainingsdaten zu einem sogenannten *Overfitting*, orientieren sich also zu stark an der Ausprägung der Trainingsdaten und lassen andere Informationen außer Acht. Dies ist der große Vorteil der Lernverfahren mit Beta- und Dirichlet-Verteilungen. Hier kann vor dem Lernen den Verfahren eine Abschätzung über die Verteilungen mitgegeben werden. Solange die beispielweise durch Experten geschätzten Verteilungen den tatsächlichen Verteilungen näher sind als die Gleichverteilung, konvergieren die Verfahren schneller gegen die tatsächliche Verteilung. Der Unterschied der beiden Verfahren liegt in ihrer Stelligkeit, die Dirichlet-Verteilungen sind eine multinomiale Verallgemeinerung der binomialen Beta-Verteilung. Daher können mit der Beta-Verteilung nur Parameter von binomialen Variablen erlernt werden, während die Dirichlet-Verteilung für beliebig diskrete Variablen funktioniert. Beim Lernen mit versteckten Daten ist das Ziel, die Parameter anhand von Trainingsdaten zu ermitteln, die nicht alle Variablen des Netzwerkes umfassen. Zu dieser Klasse gehören u.A.

---

<sup>30</sup> Es gilt :  $\alpha = \frac{1}{x_{t+1}}$

der Expectation-Maximization-Algorithmus (EM-Algorithmus) [DLR77] und der APN-Algorithmus (Adaptive Probabilistic Network) [BKR+97]. Beim EM-Algorithmus werden die fehlenden Daten zunächst durch eine Schätzung ermittelt, um dann gemeinsam mit den Trainingsdaten die übrigen Parameter des Problems zu bestimmen (z. B. mithilfe des ML-Algorithmus). Anschließend erfolgt eine erneute Schätzung der fehlenden Daten. Durch wiederholte Iteration dieser Schritte konvergiert das Verfahren schließlich gegen die tatsächlichen Parameter. Das Konvergenzverhalten ist zwar eher schlecht, jedoch ist das Verfahren auf nahezu alle Domänen anwendbar und sehr robust. Der APN-Algorithmus erweitert den EM-Algorithmus um ein Hill-Climbing, das zur Bestimmung besserer Schätzwerte verwendet wird.

Da in struktursuchenden Lernverfahren neben den Parametern auch noch die Abhängigkeiten zwischen den Variablen ermittelt werden müssen, ist der Such- bzw. Lernraum bei diesem Problem ungleich größer als beim reinen Parameterlernen. Die meisten Verfahren basieren auf lokalen Suchstrategien (vergleiche z. B. [Heck95], [TAB06] oder [Frie98]). Jedoch ist das Lernverhalten aktueller Algorithmen für praktische Anwendungen nicht zufriedenstellend. Da die Beschreibung der Netzwerkstruktur durch einen menschlichen Experten meist sehr gut möglich ist, ist die Anwendung des Strukturlernens im Kontext dieser Arbeit kaum lohnenswert.

Bayes-Netzwerke stellen eine geeignete Modellierungsmethode dar, um die Eintrittswahrscheinlichkeit von Planungsverläufen zu beurteilen. Als Darstellungsform für komplexe bedingte Wahrscheinlichkeitsverteilungen können sie Umgebungsinformationen darstellen. Mit Dynamischen Bayes-Netzwerken ist auch eine Darstellung zeitlicher Abläufe möglich, wodurch sich kausale Zusammenhänge zwischen Planungsschritten darstellen lassen.

#### 4.1.2.2 Markov-Netzwerke

Markov-Netzwerke, in der Literatur auch häufig als Markov Random Field [KN80] bezeichnet, stellen die Abhängigkeit zwischen Zufallsvariablen in Form ungerichteter Graphen dar. Ein Markov Netzwerk stellt eine vollständige gemeinsame Zufallsverteilung einer Menge von Zufallsvariablen dar. Sie bestehen aus einem ungerichteten Graphen  $(V, E)$  ( $V$  bezeichnet die Menge der Knoten,  $E$  die Menge der ungerichteten Kanten) und einer Potenzialfunktion  $\varphi_c$  für jede maximale Clique  $c$ <sup>31</sup> im Graphen. Jeder Knoten in  $V$  repräsentiert genau eine Zufallsvariable.

Eine Potenzialfunktion  $\varphi_c$  ist eine Abbildung von allen möglichen Zuweisungskombinationen der Zufallsvariablen in der Clique  $c$  mit dem Wertebereich auf die Menge der positiven reellen Zahlen:  $\varphi_c = D(c) \rightarrow \mathbb{R}^+$ . Offensichtlich stellen Potenzialfunktionen keine Wahrscheinlichkeiten im Sinne der Kolmogorov-Axiome dar, da sie Werte größer als 1 annehmen können. Je höher der Wert der Potenzialfunktion für eine bestimmte Variablenzuweisung ist, desto besser (wahrscheinlicher) ist diese Kombination der Werte der Variablen. Mit einer Potenzialfunktion können die verschiedenen Variablenzuweisungen innerhalb einer Clique bewertet und gewichtet werden. Diese sehr weit gefasste Definition einer Potenzialfunktion erlaubt die Konstruktion komplexer Funktionen.

Nach Clifford und Hammersley [CH71] kann die vollständige gemeinsame Wahrscheinlichkeitsverteilung dargestellt werden durch:

$$P(x_1, \dots, x_n) = \frac{1}{Z} \prod_{c \in C} \varphi_c(x_c)$$

---

<sup>31</sup> Ein Teilmenge  $V' \subset V$  bildet eine Clique  $C$ , wenn alle Knoten mit allen anderen Knoten der Teilmenge verbunden sind.

Dabei ist  $Z$  eine Normalisierungskonstante:  $z = \sum_{x \in X} \prod_{c \in C} \varphi_c(x_c)$ . Die Belegung der Variablen der Clique wird als  $x_c$  angegeben. Die Normalisierungskonstante bewirkt, dass  $P(x_1 \dots x_n)$  im Intervall  $[0,1]$  liegt und damit den Anforderungen an eine Wahrscheinlichkeit genügt. Innerhalb von Markov-Netzwerken gilt die Markov-Annahme erfüllt, dass also Zufallsvariable  $x_i$  nur von ihren direkten Nachbarn abhängig ist:  $P(x_i | x_{X - \{x_i\}}) = P(x_i | X_{Ni})$ .

Bei der Erstellung oder genauer der Modellierung eines Markov-Netzwerks müssen diese formalen Anforderungen beachtet werden. Um auf dem Netzwerk so eine Inferenz durchführen zu können, muss das Netzwerk konsistent und vollständig sein. Ein konsistentes Netzwerk ist nicht mit überflüssigen Variablen überladen, während ein vollständiges Netzwerk alle notwendigen Informationen enthält, um die gewünschte Inferenz durchzuführen. Pearl [Pear88, S. 105] beschreibt eine Vorgehensweise zur Modellierung eines vollständigen und konsistenten Netzwerks. Sie umfasst die folgenden Schritte:

1. Festlegen der Graphenstruktur,
2. Identifikation der maximalen Cliques des Graphen.
3. Bilden einer Potenzialfunktion  $\varphi_c = D(c) \rightarrow \mathbb{R}^+$  für jede Clique  $c$  und den dazugehörigen Wertebereich sowie
4. Bilden einer vollständigen gemeinsamen Wahrscheinlichkeitsverteilung.

Ein möglicher Algorithmus zur Inferenz in Markov-Netzwerken (also zur Bestimmung von a-posteriori-Wahrscheinlichkeiten) ist der Clustering-Algorithmus, der von Lauritzen und Spiegelhalter [LS88] entwickelt wurde und dessen Anwendung in Markov-Netzwerken in [CGH97] vorgestellt wird. Dieser Algorithmus stellt jedoch weitere Anforderungen an die Struktur des Netzwerks:

- Jede Schleife innerhalb des Netzwerks mit Kantenlänge  $>4$  muss eine Sehne<sup>32</sup> besitzen, die die Schleife in zwei kleinere Schleifen unterteilt.
- Die maximalen Cliques  $c_1, \dots, c_m$  des Graphen müssen so angeordnet sein, dass gilt: Die Menge  $c_i \cap \{c_1 \cup \dots \cup c_{i-1}\}$  ist in mindestens einer der Cliques  $c_1, \dots, c_{i-1}$  für alle  $i=1 \dots m$  vorhanden.

Die Einschränkung des Inferenzalgorithmus für Markov-Netzwerke erschwert auch die Einsetzbarkeit dieses probabilistischen Modells, da grundsätzlich nicht alle darstellbaren Modelle auch einsetzbar sind. Zusätzlich konnte in der Literatur kein methodisches Vorgehen zur Konstruktion von Potenzialfunktionen identifiziert werden, sodass dieser wichtige Bestandteil des Modells durch Experimentieren am Modell bestimmt werden muss. Verfahren, um die Potenzialfunktionen, Einzelwahrscheinlichkeiten oder die vollständige gemeinsame Wahrscheinlichkeitsverteilung zu erlernen, sind ebenfalls nicht bekannt. Das spricht insgesamt gegen den Einsatz von Markov-Netzwerken zur aktuellen Bestimmung der Eintrittsmöglichkeit bestimmter Planverläufe.

### 4.1.3 Planungssprachen für Unsicherheit und unvollständige Information

Die folgenden Absätze beschreiben zwei Erweiterungen der PDDL, die die Darstellung von Planungsproblemen unter Unsicherheit beschreiben. Neben diesen beiden Ansätzen, die direkt die PDDL erweitern, existieren auch andere Planungsformalismen zur Darstellung von unsicheren Planungsproblemen. Dazu gehört auch der PDDL-Vorläufer ADL [Pedn86], eine sogenannte Aktionssprache. Zwar existieren noch einige Erweiterungen der ADL  $A_K$  [Lobo97] oder  $AR$  [CGGT97,EFL+04], die sich jedoch durch eine umständliche Formulierung der Pla-

<sup>32</sup> In einem Graphen  $G$  wird eine Kante von  $G$  als Sehne bezeichnet, wenn sie zwei Knoten eines Kreises in  $G$  verbindet, selbst jedoch nicht Teil des Kreises ist.

nungsprobleme auszeichnen. So besteht beispielsweise nicht die Möglichkeit, numerische Variablen auszudrücken. Dies ist aber notwendig, um die in Kapitel 3 in die Planung integrierten kontinuierlichen Eigenschaften der Aktionen zu modellieren.

#### 4.1.3.1 NPDDL

NPDDL stellt eine Erweiterung der Planungssprache PDDL für nicht-deterministische Umgebungen dar [BCLP03]. NPDDL kann drei Eigenschaften einer nicht-deterministischen Umgebung beschreiben: Unsicherheit im Anfangszustand, nichtdeterministische Aktionen (z. B. nicht eindeutige Effekte) und partielle Beobachtbarkeit von Zuständen. Die Darstellung von Plänen in NPDDL unterstützt Strukturen wie Verzweigungen (branching) und Iterationen, sodass ein NPDDL-Planer Pläne erstellt, die dem Code von Computerprogrammen ähnlich sehen.

Als Erweiterung von PDDL 2.1 [FL03] übernimmt NPDDL wichtige Eigenschaften wie die Darstellung numerischer Funktionen. Auch die Instanzierung der Operatoren und Prädikate mittels Objekten erfolgt analog wie in PDDL 2.1.

Die wesentliche Neuerung in der NPDDL ist ein Konstrukt namens *oneof*. Es wird genutzt, um nicht eindeutige Initialzustände und nicht eindeutige Effekte einer Aktion zu definieren. In beiden Fällen basiert die Darstellung auf einer Disjunktion der entsprechenden Ausdrücke aus der PDDL.

Das bedeutet, dass ein nicht eindeutiger Initialzustand durch eine Menge alternativer Zustände beschrieben wird, von denen nur einer tatsächlich zutrifft. Dabei gelten für die einzelnen Alternativzustände die gleiche Syntax und Schlussregeln (insbesondere gilt für jede der Alternativen die Closed-World-Assumption) wie in PDDL 2.1.

Für den Fall nicht eindeutiger Aktionen findet das Konstrukt *oneof* Anwendung im Effektteil der Aktion. Analog zum nicht eindeutigen Initialzustand ist dort eine Menge von alternativen Effekten beschrieben, die jeder für sich in der Syntax und in der Verarbeitung während der Plankonstruktion dem Vorgehen bei der PDDL entspricht.

Zusätzlich zu nicht eindeutigen Initialzuständen und Aktionen kann NPDDL unzuverlässige Wahrnehmung des Umweltzustands darstellen. Derartige Wahrnehmungen werden modelliert durch eine Relation zwischen den möglichen Umweltzuständen und Observationsvariablen, die gewissermaßen die Sensoren des planausführenden Systems darstellen. Die Relation zwischen den Umweltzuständen und den Observationsvariablen wird mithilfe des Konstrukts *observation* in Form einer logischen Formel dargestellt.

Die Darstellung von Plänen geht in der NPDDL über die übliche Modellierung in Form einer Menge partiell geordneter Aktionen hinaus. Es werden zusätzliche Konstruktionen eingeführt, die angeben, wie oft, in welcher Weise und unter welchen Bedingungen Aktionen ausgeführt werden sollen. Dazu gehören das sequenzielle Ausführen von Aktivitäten, die bedingte Verzweigung, die iterative Ausführung in Schleifen und sogar Sprunganweisungen, mit deren Hilfe die Ausführung an einer gänzlich anderen Stelle fortgeführt wird.

Gegenüber der PDDL ist die NPDDL wesentlich besser geeignet, Planungsprobleme in nicht deterministischen Umgebungen darzustellen. Dieser Möglichkeit sind allerdings auch enge Grenzen gesteckt. Die Modellierung des Nicht-Determinismus mithilfe des *oneof* Konstrukts kann nur eine endliche Menge von Initialzuständen und alternativen Aktionseffekten darstellen. Die Modellierung sehr großer Mengen von Initialzuständen und Aktionseffekten ist sehr aufwendig, da auch Zustände und Effekte, die sich nur in einigen Punkten voneinander unterscheiden, vollständig modelliert werden müssen. Dies führt bereits in mäßig komplexen Umgebungen zu einer sehr großen Modellgröße. Unterstellt man beispielsweise, dass der Energieverbrauch einer Fahraktion aus der mechatronischen Beispieldomäne (siehe 2.1.3) unab-



hängig von der gefahrenen Strecke um bis zu  $\pm 10\%$  vom eigentlich erwarteten Verbrauch abweicht, muss diese mögliche Abweichung auf eine endliche Menge möglicher Ausgänge abgebildet (also diskretisiert) werden. Entscheidet man sich für eine Diskretisierung in 5%-Schritten, ergeben sich bereits fünf alternative Ausgänge<sup>33</sup> für die Aktion. Folgen zwei derartige Aktionen mit 5 alternativen Ausgängen, müssen nach der zweiten Aktion bereits  $5^2$  mögliche Zustände berücksichtigt werden. Dabei fehlt jede Möglichkeit, die Alternativen hinsichtlich ihrer Relevanz zu unterscheiden. Alle möglichen Initialzustände und Aktionseffekte müssen als gleichwertig behandelt werden. Daher ist NPDDL zur Modellierung des Planungsproblems für mechatronische Systeme nicht geeignet.

#### 4.1.3.2 PPDDL

Genau wie die im vorherigen Abschnitt vorgestellte NPDDL ist auch die Probabilistic Planning Domain Language (PPDDL) [YL04] eine Erweiterung des Standards PDDL. Die PPDDL führt zusätzlich zur PDDL 2.1 probabilistische Effekte ein. Die PPDDL 1.0 stellt einen Schritt in Richtung einer allgemein anwendbaren Planungssprache zur Beschreibung probabilistischer und entscheidungstheoretischer Planungsprobleme dar. In ihrer ersten Version umfasst die PPDDL keine Nebenläufigkeit, (beschränkte) Beobachtbarkeit und Berücksichtigung von Zeit.

Die Syntax der Erweiterung der PPDDL ermöglicht die Spezifikation eines Markov-Entscheidungsprozesses (vergleiche 4.1.1.1). Die wichtigste Erweiterung dabei sind probabilistische Effekte in Aktionsdefinitionen. Die Syntax eines probabilistischen Effekts ist (*probabilistic*  $p_1e_1 \dots p_k e_k$ ), wobei die alternativen Effekte  $e_i$  mit der Wahrscheinlichkeit  $p_i$  eintreten. Gemäß den Regeln der Wahrscheinlichkeitsrechnung muss dabei  $p_i \leq 0$  und  $\sum_{i=1}^k p_i = 1$  gelten. Im Wesentlichen entspricht diese Definition von Aktionseffekten der Definition nicht eindeutiger Aktionen in der NPDDL mithilfe des Konstrukts *oneof*, wobei zusätzlich die Eintrittsmöglichkeit der Nachbedingung angegeben wird.

Hinsichtlich der Zieldefinition innerhalb des Planungsproblems unterscheidet die PPDDL zwei Möglichkeiten. Eine Zieldefinition in Form der in PDDL üblichen Form eines Zielzustands wird so interpretiert, dass der Plan die Wahrscheinlichkeit maximieren soll, diesen Zustand zu erreichen. Alternativ kann eine Optimierungsmetrik definiert werden, die maximiert oder minimiert werden soll. Eine weitere Möglichkeit besteht darin, statt eines Planungsproblems einen Markov-Entscheidungsprozess anzunehmen. Dazu ist die Definition einer Reward-Funktion notwendig.

Die Reward-Funktion ist eine Variable, auf die innerhalb der Aktionseffekte zugegriffen werden kann. Dabei ist der Zugriff darauf beschränkt, zum Reward den aktuellen Wert bestimmter numerischer Funktionen aus der Planungsdomäne zu addieren oder zu subtrahieren. Damit stellt die Reward-Funktion den kumulierten Reward für alle Zustandsübergänge dar.

Younes und Littmann zeigen schließlich, dass Pläne der PPDDL konvertiert werden können in Dynamische Bayes-Netzwerke (DBN) (vergleiche Kapitel 4.1.1.1.). Da DBN letztlich eine graphische Repräsentation von Markov-Prozessen sind, eignet sich die PPDDL zum Einsatz in entscheidungstheoretischen Planern.

PPDDL besitzt mit den probabilistischen Effekten etwa die gleiche Ausdrucksstärke wie die NPDDL mit ihrem *oneof* Konstrukt. Zusätzlich ermöglicht die PPDDL die Berücksichtigung der Eintrittsmöglichkeit von bestimmten Situationen durch die Angabe von Wahrscheinlichkeit. Durch die Konvertierung in DBNs besteht zusätzlich die Möglichkeit, auf bewährte Methoden zur Berechnung der Eintrittswahrscheinlichkeiten über einen gesamten Plan hinweg zurückzugreifen.

Die Modellierung des Nicht-Determinismus bzw. der probabilistischen Effekte in PPDDL ist ebenso grob wie bei der NPDDL. Damit fällt der gleiche große Modellierungsaufwand zur

<sup>33</sup> Abweichung um -10%, -5%, 0%, +5%, +10%

Darstellung komplexer Domänen an wie bei der NPDDL. Außerdem unterstützt die PPDDL den Modellierer nicht bei der Definition der Eintrittswahrscheinlichkeiten, da kein semantischer Zusammenhang zwischen den Elementen der Vorbedingung und den Effekten besteht. Solche Zusammenhänge müssen durch den Modellierer direkt bestimmt werden oder mittels zusätzlicher Methoden (z. B. durch Modellierung mittels Bayes-Netzwerken, vergleiche 4.1.2.1) ermittelt werden. Der letzte Kritikpunkt an der PPDDL besteht darin, dass alle Informationen a priori, also im Vorfeld der Planung angegeben werden müssen und eine flexible Berücksichtigung von externen Informationen nicht möglich ist.

#### 4.1.4 Planungsverfahren zur Behandlung von Unsicherheit

Planungsverfahren müssen mit der speziellen Struktur umgehen können, die Pläne und Planungsprobleme unter Unsicherheit besitzen. In den folgenden Kapiteln werden grundlegende Planungsprinzipien vorgestellt, die angewandt werden können, wenn unvollständige Informationen oder nicht-deterministische Aktionen vorliegen. Neben einer Vorstellung der Grundidee des Planungsprinzips werden auch jeweils beispielhafte Realisierungen beschrieben.

##### 4.1.4.1 Sensorloses Planen

Das sensorlose Planen konstruiert Pläne für nicht-deterministische Umgebungen, ohne dass eine Überwachung der Planausführung mit Sensoren notwendig wird [GB96]. Zu diesem Zweck führt das sensorlose Planen sogenannte *Belief States* ein, bei denen es sich um eine Menge von Zuständen handelt, die nach der Ausführung einer Reihe von Planschritten erreicht sein können. Auf Basis dieser *Belief States* setzt sensorloses Planen ein neues Kriterium zur Anwendbarkeit von Aktionen ein. Eine Aktion ist genau dann in einem *Belief State* anwendbar, wenn sie im Sinne der klassischen Planung in jedem enthaltenen Zustand anwendbar ist. Die Nachbedingung einer in einem *Belief State* b ausgeführten Aktion a ergibt sich daher aus der Kombination aus a und b. Dadurch wird der Umfang und die Komplexität der *Belief States* mit steigender Planlänge immer größer. Dies erfordert Algorithmen, die mit diesem Zustandsraum effektiv umgehen können. So stellen Cimatti und Roveri einen sensorlosen Planer auf Basis von Symbol Model Checking vor [CR99], der Planungsprobleme verarbeiten kann, die in der Aktionssprache AR [CGGT97] formuliert sind.

Andererseits ergibt sich aus der sehr strengen Anforderung an die Ausführbarkeit von Aktionen, dass ein Plan nur für sehr einfache Planungsprobleme gefunden werden kann. Außerdem gibt das Planungsprinzip des sensorlosen Planens einen der wesentlichen Vorteile der Verhaltensplanung auf: Die Möglichkeit, potenziell eintretende Situationen zu berücksichtigen. Daher kommt die sensorlose Planung für die Verhaltensplanung für mechatronische Systeme nicht in Frage.

##### 4.1.4.2 Bedingte Planung

Die Grundidee des bedingten Planens besteht darin, für alternative Situationen, die während der Planausführung auftreten können, jeweils einen passenden Plan bereitzuhalten. Ein wesentliches Konzept dieser Planungsform sind die Sensoraktionen [SW98]. Diese Sensoraktionen dienen dazu, Informationen zu sammeln, die zur Planungszeit noch nicht zur Verfügung standen [OG90]. Abhängig von den gesammelten Informationen wird entschieden, welcher alternative Plan im Weiteren zur Ausführung gebracht wird.

Die Grundidee des bedingten Planens geht zurück auf Warren [Warr76]. Das vorgestellte Planungssystem WARPLAN-C erweitert die klassische Modellierung von Planungsaktionen um bedingte Aktionen, auf ganz ähnliche Weise, wie das beispielsweise bei der in 4.1.3.1 beschriebenen Planungssprache NPDDL der Fall ist. Dabei ist WARPLAN-C auf zwei alternative Ausgänge je bedingter Aktion beschränkt. Diese Ausgänge werden als P und  $\neg P$  bezeichnet. Die Konstruktion des bedingten Plans geschieht in zwei Phasen. In der ersten Phase

wird ein unbedingter Plan erzeugt, indem angenommen wird, dass für jede bedingte Aktion der Ausgang P eintritt. In der zweiten Phase wird ein Plan für jede Verzweigung, also jeden Ausgang  $\neg P$  konstruiert.

Peot und Smith [PS92] erweiterten das Konzept von WARPLAN-C in ihrem Planer CNLP (Conditional Non Linear Planning), indem sie einerseits eine beliebige endliche Anzahl von alternativen Ausgängen für bedingte Aktionen zulassen, und andererseits nicht-lineare bedingte Pläne erstellen. Zur Definition bedingter Aktionen erweitern Peot und Smith die STRIPS Notation, die keine numerischen Variablen unterstützt, um bedingte Aktionen. Diese Art Variablen ist wichtig für die Verhaltensplanung für mechatronische Systeme. Ein wichtiges Element zur Erstellung nicht vollständig geordneter bedingter Pläne sind die sogenannten Kontexte. Ein Kontext fasst mehrere – nicht widersprüchliche – Ausgänge verschiedener Aktionen zusammen. Eine Aktion, für deren Ausführung ein bestimmter Kontext erfüllt sein muss, kann nur dann ausgeführt werden, wenn alle Ausgänge, die in dem Kontext enthalten sind, genauso eingetreten sind. Während CNLP auf dem POP-Planer SNLP basiert, stellten Weld, Anderson und Smith einen bedingten Planer auf Basis von Graphplan vor [WAS98]. Auch Betroli, Cimatti und Roveri stellen einen bedingten Planer vor [BCR01]. Ihr Planer ist zusätzlich zur Berücksichtigung bedingter Aktionen auch in der Lage, mit einer beschränkten Beobachtbarkeit während der Planausführung umzugehen. Dies führt zu einer Kombination der Ansätze sensorloses Planen und bedingtes Planen. Zu diesem Zweck kann der in [BCR01] vorgestellte Planer ebenfalls *Belief States* verarbeiten.

Das bedingte Planen ist ein für die Verhaltensplanung sehr interessantes Planungsprinzip. Es ermöglicht, nicht nur für die gegebene konkrete Aufgabe eine angepasste Sequenz von Teilfunktionslösungen zu ermitteln, sondern dieses Prinzip auch auf alternative Zustände auszuweiten, die sich während der Planausführung einstellen könnten.

Problematisch am bedingten Planen ist jedoch die durch die zu planenden Verzweigungen steigende Komplexität. Wie auch bei der Modellierung mittels der NPDDL (vergleiche 4.1.3.1) steigt die Anzahl der zu berücksichtigenden Zustände durch die Abfolge nicht deterministischer Zustände exponentiell. Eine Berücksichtigung aller möglichen Alternativen in einer komplexen Domäne und bei relativ langen Plänen wird in einer sehr langen Planungs-dauer resultieren, die zu einer faktischen Unlösbarkeit des Problems führt. Als alleiniger Planungsansatz kommt das bedingte Planen somit nicht in Betracht.

#### 4.1.4.3 Ausführungsüberwachung und Neuplanung

Das dritte grundsätzliche Planungsprinzip zur Planung in nicht-deterministischen Umgebungen ist die *Ausführungsüberwachung und Neuplanung*. Genau wie beim bedingten Planen spielen Sensoren und Wahrnehmungen hier eine entscheidende Rolle.

Die erste Anwendung dieses Planungsprinzips stellt der berühmte Roboter Shakey dar. Ein Planausführer namens PLANEX1 führte Pläne aus, die das Planungssystem STRIPS für den Roboter Shakey konstruiert hatte [Fike71]. Das PLANEX1-System erweiterte das STRIPS-Planungssystem um wichtige Fähigkeiten, die zum Handeln in physischen Umgebungen notwendig sind. PLANEX1 ist in der Lage, über Sensoren während der Planausführung Informationen zu sammeln. In Abhängigkeit von den gesammelten Informationen wird der ursprüngliche STRIPS-Plan dann angepasst. Wenn die gesammelte Information impliziert, dass einige Teile des Plans überflüssig sind, werden diese nicht ausgeführt. Falls aber Teile des Plans nicht die gewünschte Wirkung haben, werden die Aktionen zunächst noch einmal ausgeführt oder der STRIPS-Planungsalgorithmus wird erneut aufgerufen. Dazu wird dem Planungsalgorithmus die aktuelle Situation als Initialzustand übergeben. Der Zielzustand wird als Disjunktion aller im ursprünglichen Plan enthaltenen Zustände definiert. Dadurch besteht die Möglichkeit, in dieser Neuplanungsphase keinen vollständig neuen Plan aufbauen zu müssen, sondern zum alten Plan zurückzukehren.

Wilkins et. al. [WMLW94] stellen mit Cypress ein Planungs- und Ausführungsframework für unsichere und dynamische Umgebungen vor. Anders als beim PLANEX1-System wird die Ausführung der Pläne nicht angehalten, bis die Neuplanung durchgeführt wird. Die Autoren bezeichnen dies als asynchrone Neuplanung. Das Cypress-Framework besteht im Wesentlichen aus zwei Systemen: PRS-CL und SIPE-2. PRS-CL dient zur Implementierung von Ausführungscontrollern, die komplexe Aufgaben in dynamischen Umgebungen ausführen können. Dazu überwachen sie kontinuierlich einkommende Informationen und sich ändernde Zielvorgaben. Die Controller sind außerdem in der Lage, einfache Umplanungen, die im Rahmen der Ausführung notwendig werden, selbstständig durchzuführen.

Nur bei umfassenderen Umplanungen wird das eigentliche Planungssystem SIPE-2 eingesetzt. Dieses System ist in der Lage, Pläne mit parallelen und bedingten Aktionen zu erstellen. Die mit PRS-CL erstellten Controller rufen den SIPE-2 Planer auf, wenn weitreichende Planänderungen notwendig sind. In der asynchronen Neuplanung setzt der Controller dabei die Ausführung von solchen Teilen des Plans fort, die nicht von der Planänderung betroffen sind. Die Neuplanung umfasst dabei die folgenden Schritte:

1. PRS-CL entdeckt während der Ausführung eines Plans einen Fehler, den es nicht selbst beheben kann.
2. PRS-CL übermittelt den aktuellen Zustand an SIPE-2 und setzt dann die Ausführung der Teile des Plans fort, die nicht durch den Fehler berührt sind.
3. SIPE-2 ruft seinen Neuplanungsalgorithmus auf und erstellt einen alternativen Plan.
4. Der geänderte Plan wird an PRS-CL übermittelt.
5. PRS-CL mischt den geänderten Plan mit seinen aktuellen Aktivitäten und setzt die Ausführung fort.

Der von SIPE-2 aufgerufene Planungsalgorithmus ist im eigentlichen Sinne kein Neuplanungsalgorithmus, da auf dem fehlgeschlagenen Plan aufgebaut wird. Wilkins et. al. wählen dieses Vorgehen, um die Planungsdauer zu reduzieren.

Die Planungsdauer ist bei der Ausführungsüberwachung und Neuplanung ein entscheidender Faktor. Nicht in jedem Plan werden parallel ausführbare und daher unabhängige Teilpläne vorliegen. Damit das ausführende System nicht unnötig lange inaktiv wird, sind Planungsalgorithmen gefragt, die schnell einen Plan konstruieren können. Dieses Problem ist Gegenstand der Echtzeitplanung<sup>34</sup> oder Onlineplanung.

Sapena und Onaindía [SO02] stellen ein Onlineplanungsverfahren auf Basis der Planungssprache PDDL 2.1 vor. Das Onlineplanungsverfahren verzichtet darauf, einen vollständigen Plan zu konstruieren sondern beschränkt sich stattdessen darauf, dem ausführenden System jeweils eine einzige Aktion vorzugeben, die sofort ausgeführt wird. Die resultierende Situation stellt die Ausgangssituation für die Festlegung der nächsten Aktion dar. Aus diesem Grund entspricht das vorgestellte Verfahren SimPlanner einer Planung mithilfe einer Tiefensuche, weswegen der Planungsalgorithmus nicht vollständig<sup>35</sup> ist. Es besteht auch die Gefahr, eine Aktionsschleife zu erzeugen, die nicht verlassen werden kann. Um diese Gefahren möglichst gering zu halten, werden heuristische Auswahlmechanismen genutzt, um die nächste auszuführende Aktion auszuwählen. SimPlanner löst dazu ein relaxiertes Planungsproblem, in dem ähnlich wie beim FastForward-Planer [HN01] ein Planungsgraph ohne Berücksichtigung der Delete-Liste der Aktionen gelöst wird. Dabei wird ein relaxierter Plan für jedes Teilziel erzeugt. Die gefundenen relaxierten Pläne werden in Gruppen zusammengefasst, die eine gemeinsame erste Aktion besitzen. Diejenige Gruppe, deren erste Aktion die wenigsten Konflik-

<sup>34</sup> *Echtzeit* hat hier im Gegensatz zur üblichen Verwendung des Begriffs tatsächlich die Bedeutung „besonders schnell“, da die Verfahren keine deterministischen Schranken für Lösungszeiten liefern

<sup>35</sup> Ein Verfahren ist vollständig, wenn es garantiert eine Lösung findet, falls eine existiert

te mit anderen relaxierten Plänen aufweist, wird zur Ausführung gewählt. Damit folgt SimPlanner dem Prinzip des Least Commitments.

Koenig [Koen01] stellt ein Verfahren vor, dass dem ausführenden System mehr als eine einzelne Aktion zur Ausführung vorgibt. Aber auch die *Echtzeit Heuristische Suche* verzahnt die Planung und Planausführung. Im Gegensatz zum Verfahren von Sapena und Onaindía setzt Koenig auf dem vollständigen heuristischen Suchverfahren A\* auf. Das Verfahren Learning Real Time A\* (LRTA\*) [Korf90] kann eine beliebige Anzahl von zukünftigen Schritten berücksichtigen. Es handelt sich bei LRTA\* um einen Any-Time-Algorithmus [Zilb96], sodass der LRTA\* jederzeit abgebrochen werden kann und ein (Teil-)ergebnis zurückliefert, das unter Umständen aber nicht optimal ist. Koenig erweitert LRTA\* zu einem Planungsverfahren für nicht-deterministische Umgebungen namens Min-Max LRTA\* [KS95]. Min-Max LRTA\* ist in der Lage, nicht-deterministische Aktionen zu berücksichtigen und unterstellt dazu im Rahmen der Planung, dass von allen möglichen Folgezuständen einer ausgeführten Aktion immer der ungünstigste eintritt. Leider lässt Koenig die Frage unbeantwortet, auf welche Art und Weise die heuristischen Werte für Zustände ermittelt werden können.

Das Planungsprinzip *Ausführungsüberwachung und Neuplanung* stellt einen intuitiven Ansatz zur Behandlung von Unsicherheit dar. Es eröffnet insbesondere die Möglichkeit, im Rahmen der Planung die Annahmen der klassischen Planung aufrechtzuerhalten und somit Standardplaner einzusetzen (wie beispielsweise im Fall von PLANEX1 geschehen).

Diesen Vorteilen steht jedoch gegenüber, dass das ausführende System im Falle einer Neuplanung seine Aktivität unter Umständen einstellen muss. Um das zu vermeiden, muss die Neuplanung unter Zeitdruck ausgeführt werden. Dies kann zwar mit Verfahren zur Online- oder Echtzeitplanung geschehen, die entsprechenden Planungsverfahren sind aber nicht vollständig und garantieren kein gutes Planungsergebnis. Daher stellt die Anwendung einer Neuplanung eine Möglichkeit dar, die in der Regel vermieden werden sollte.

#### 4.1.4.4 Probabilistische Planungsverfahren

Eine eigene Klasse von Planern für nicht-deterministische Probleme stellen die sogenannten probabilistischen Planer dar. Probabilistische Planer können verschiedene der in den vorigen Kapiteln beschriebenen Planungsprinzipien realisieren. Dabei setzen sie das Glaubensmaß der Wahrscheinlichkeit ein, um zu unterscheiden, welche Verzweigungen im Plan wichtiger sind, und bauen den Baum der möglichen Planverläufe nicht mehr vollständig auf. In der Regel bauen die probabilistischen Planer dabei auf einer Planungssprache auf, die der in 4.1.3.2 beschriebenen PPDDL im prinzipiellen Aufbau sehr ähnlich ist.

Ein probabilistischer Planer, der dem Prinzip des sensorlosen Planens entspricht, ist BURIDAN [HKW95]. BURIDAN basiert auf dem partiell ordnenden Planen<sup>36</sup>. Die entscheidende Erweiterung von BURIDAN besteht darin, dass er mehrere Aktionen als Auslöser für ein bestimmtes Fluent im Planverlauf einfügen kann. Durch das Einfügen solcher redundanter Aktionen wird die Wahrscheinlichkeit erhöht, dass ein Fluent eintritt. Wurde ein Plan gefunden, der die Ziele erreichen kann, wird überprüft, ob die Durchführungswahrscheinlichkeit des Planes einen Schwellenwert überschreitet. Ist dies nicht der Fall, verfeinert BURIDAN den Plan, indem zusätzliche Aktionen eingeführt werden, die die Wahrscheinlichkeit des Plans erhöhen, andernfalls terminiert der Algorithmus. Unklar bleibt, woher BURIDAN die Wahrscheinlichkeiten erhält, die im Rahmen der Planung verwendet werden. Außerdem bleiben mögliche Kosten der redundanten Aktionen unberücksichtigt. C-BURIDAN erweitert BURIDAN um die Fähigkeit zum bedingten Planen [DHW94]. Dazu greift es auf den bedingten Planer CNLP (vergleiche 4.1.4.2) zurück. C-BURIDAN modelliert die Effekte einer Aktion und die Wahrnehmung des sich ergebenden Zustands separat und stellt dadurch teilweise Be-

---

<sup>36</sup> Siehe Anhang A

obachtbarkeit dar. C-BURIDAN führt neue Verzweigungen in den Plan ein, um die Erfolgswahrscheinlichkeit zu erhöhen. Daher ist C-BURIDAN grundsätzlich effizienter als BURIDAN, da redundante Aktionen nur ausgeführt werden, wenn der Erfolg der Planausführung tatsächlich gefährdet ist.

Blythe stellt mit Weaver einen Planer vor, der genau wie BURIDAN die Erfolgswahrscheinlichkeit eines Plans maximiert [Blyt97]. Eine Besonderheit von Weaver liegt in der sehr umfangreichen Modellierung der Unsicherheit in der Planung. Dies äußert sich in der Fähigkeit, exogene Ereignisse zu berücksichtigen. Ein exogenes Ereignis ist ein Ereignis, das außerhalb des direkten Einflusses des Planers liegt. Exogene Ereignisse werden mit einer bedingten Wahrscheinlichkeit belegt und verändern beim Eintreten die im Plan auftretenden Ergebnisse. Der Planer kann exogene Ereignisse nur indirekt durch Aktionen beeinflussen, deren Nachbedingungen Einfluss auf die bedingte Eintrittswahrscheinlichkeit haben. Der Planerstellungsprozess in Weaver erfolgt in zwei Schritten. Zunächst wird ein Plan ohne Berücksichtigung möglicher exogener Ereignisse entwickelt. Im zweiten Schritt werden die relevanten exogenen Ereignisse erfasst. Ein exogenes Ereignis ist relevant, wenn es negative Auswirkungen auf den Plan haben kann. Anschließend werden zusätzliche Aktionen und Abhängigkeiten in den Plan eingefügt, die die Wahrscheinlichkeit eines unerwünschten Ereignisses reduzieren. Um die Wahrscheinlichkeiten im Plan ermitteln zu können, modelliert Blythe die Aktionen für Weaver zusätzlich zur Darstellung in einer konventionellen Planungssprache als Bayes-Netzwerke. Anhand dieses probabilistischen Modells für Aktionen kann durch den Planungsalgorithmus für einen bestehenden Plan ein gesamtes probabilistisches Modell erstellt werden. Ein weiterer Planer, der die Erfolgswahrscheinlichkeit eines Plans maximiert, wird von Little und Thiébaux [LT06] vorgestellt. Der Algorithmus Paragraph, der auf dem Graphplan-Verfahren für deterministische Planungsprobleme basiert, kann parallele Teilpläne konstruieren. In Erweiterung zum klassischen Plangraphen<sup>37</sup> enthält der Plangraph in Paragraph einen weiteren Knotentyp, um probabilistische Planung zu ermöglichen. Für jeden Effekt, den eine Aktion auslösen kann, wird ein Effektknoten in den Planungsgraphen eingefügt. Aufgrund des neuen Knotentyps müssen auch die Mutex-Relationen erweitert werden: Mögliche Effekte einer Aktion schließen sich gegenseitig aus, da nur ein Effekt wahr werden kann. Paragraph versucht alle Aktionsfolgen zu finden, die das Planungsproblem lösen können. In einem weiteren Schritt werden Aktionsfolgen zusammengefasst, um die Erfolgswahrscheinlichkeit des Plans zu erhöhen. Dabei können zwei Aktionsfolgen miteinander kombiniert werden, wenn sie während der Ausführung den gleichen Zustand erreichen können. Als Ergebnis der Planung wird eine Kombination von Aktionsfolgen als gültiger Plan zurückgeliefert.

Alle bisher vorgestellten probabilistischen Planer versuchen, die Erfolgswahrscheinlichkeit der Zielerreichung zu maximieren. Ein anderes Ziel verfolgt der Planer DRIPS [HS94]. Es handelt sich um einen nutzenbasierten Planer. DRIPS verwendet eine Nutzenfunktion, um alternative Pläne miteinander zu vergleichen. Für unterschiedliche partielle Pläne wird ein Nutzenintervall berechnet. Ein Plan  $p$  dominiert einen anderen Plan  $p'$ , wenn sowohl die untere als auch die obere Grenze des Nutzenintervalls von  $p$  höher ist als die von  $p'$  [Well90]. Um diesen Vergleich effizient durchführen zu können, setzt DRIPS abstrakte Pläne ein. Ein abstrakter Plan besteht aus Aktionen, die mehrere Aktionen der Planungsdomäne zusammenfassen. Da ein solcher abstrakter Plan schneller zu konstruieren ist als ein konkreter Plan und gleichzeitig mehrere konkrete Pläne umfasst, können auf diese Weise in kurzer Zeit eine größere Anzahl alternativer Pläne als ungeeignet ausgesondert werden. Die abstrakten Pläne können durch Verfeinerung in konkrete und ausführbare Pläne transformiert werden. DRIPS wählt den konkreten Plan aus, dessen obere Grenze des Nutzenintervalls den höchsten Wert aufweist.

---

<sup>37</sup> Siehe Anhang B

Probabilistische Planung weist eine interessante Kombination von Planungstechniken und dem Glaubensmaß der Wahrscheinlichkeitstheorie auf. Die meisten Ansätze weisen jedoch einen entscheidenden Mangel auf: In der Regel besitzt die verwendete Planungssprache die gleiche einfache Struktur wie die PPDDL. Dadurch ergeben sich die in Kapitel 4.1.3.2 beschriebenen Nachteile für die Modellierung und die Berücksichtigung wechselnder Umfelder. Einzig Blythe beschreibt in Weaver eine strukturierte Modellierung der Wahrscheinlichkeiten innerhalb der Planungsdomänen. Insbesondere die Berücksichtigung exogener Ereignisse ist eine herausragende Eigenschaft von Weaver. Allerdings beschränkt sich auch Weaver nur auf Informationen, die bereits während der Modellierung der Planungsdomäne zur Verfügung stehen. Berücksichtigung externer Informationsquellen ist nicht vorgesehen. Genauso beschränkt sich die Planung auf die Nutzung von a-posteriori-Wahrscheinlichkeiten und Informationen, die während einer (teilweisen) Planausführung gesammelt werden können, bleiben unberücksichtigt.

#### 4.1.4.5 Markov-Entscheidungsprozesse

Markov-Entscheidungsprozesse (englisch: Markov Decision Process, abgekürzt MDP) stellen eine Erweiterung von Markov-Prozessen, die auch den Dynamischen Bayes-Netzwerken zugrunde liegen, zu einem Entscheidungswerkzeug dar. Sie werden zur Modellierung und Lösung sequenzieller Entscheidungsprobleme verwendet. Ein häufiges Anwendungsgebiet stellt das Reinforcement Learning dar, bei dem ein Softwareagent aus den Folgen seines Verhaltens lernen soll. Sutton und Barto [SB98] beschreiben ein MDP als einen stochastischen Automaten, der aus folgenden Elementen besteht:

- einer endlichen Menge von Zuständen  $S = \{s_1, \dots, s_n\}$ ;
- einer endlichen Menge von Aktionen  $A(s) = \{a_1, \dots, a_1\}$  für jeden Zustand  $s$ ;
- einer Wahrscheinlichkeitsverteilung  $P_{ss'}^a = P(s_{t+1} = s' | s_t = s, a_t = a)$ , die die Übergangswahrscheinlichkeit für jedes Tupel  $(s, a)$  zu allen möglichen Folgezuständen  $s'$  definiert und
- einer Menge von Erwartungswerten  $R_{ss'}^a = E(r_{t+1} | s_t = s, a_t = a, s_{t+1} = s')$ , die die Belohnung angeben, die erwartet werden kann, wenn in einem Zustand  $s$  die Aktion  $a$  gewählt wird und als Folge der Zustand  $s'$  erreicht wird.

Ziel eines MDP ist die Festlegung einer optimalen Strategie  $\pi^*$ , die zu jedem möglichen Zustand  $s'$  angibt, welche Aktion den Erwartungswert der Belohnungen maximiert.

Markov-Entscheidungsprozesse fokussieren auf die Darstellung eines mit Unsicherheit behafteten Entscheidungsproblems. Dabei bleibt offen, wie die Übergangswahrscheinlichkeiten  $P_{ss'}^a$  hergeleitet werden können. Der Einsatz von MDP im Rahmen der Verhaltensplanung in mechatronischen Systemen wird außerdem durch die vorgegebene Struktur des Entscheidungsprozesses erschwert. Ein MDP maximiert die zu erwartende Belohnung. Die in Kapitel 2 beschriebene Planungsaufgabe beschreibt aber die Suche nach einem Ein- und Ausgabeverhalten, das der Aufgabe des mechatronischen Systems entspricht. Eine solche Aufgabe ist mithilfe eines MDP nur schwer zu beschreiben, da die Bestimmung einer Belohnungsfunktion, die garantiert in einen definierten Zustand führt, nur schwer möglich ist.

#### 4.1.5 Fazit zum Stand der Technik Planung unter Unsicherheit

Betrachtet man den Stand der Technik zur Planung unter Unsicherheit im engeren Sinne, also die Modellierungsmöglichkeiten zur Darstellung von Planungsproblemen und die Lösungsverfahren, wird großer Handlungsbedarf für die Konzeption einer Verhaltensplanung für mechatronische Systeme deutlich.

Die beiden betrachteten Erweiterungen der PDDL, die NPDDL und PPDDL, schränken die Darstellung von Unsicherheit auf eine diskrete Menge alternativer Startzustände und eine diskrete Menge alternativer Nachbedingungen ein. Dies geschieht, damit der Zustandsraum der Planung endlich bleibt. Jedoch steigt die Komplexität der Planung exponentiell im Vergleich zum deterministischen Fall an, ohne die Ausdrucksstärke der Planungssprache auf ein praxistaugliches Maß zu erhöhen.

Daher erscheint es vielversprechender, den Prozess der Planung unter Unsicherheit zweistufig zu gestalten. Einige der beschriebenen probabilistischen Planer (z. B. Weaver) setzen ein derartiges Vorgehen ein: Zuerst wird ein deterministischer Plan erzeugt, der dann anschließend mithilfe von Wahrscheinlichkeitsmodellen analysiert wird. Jedoch erzeugen die untersuchten probabilistischen Planer keine Alternativpläne in dem Sinne, dass sie im Falle einer auftretenden Planabweichung ausgeführt werden können. Stattdessen wird ein Erwartungsnutzen oder eine Erfolgswahrscheinlichkeit maximiert. Dies entspricht nicht dem in Unterkapitel 2.3 angestrebten Planungsprozess. Zusammenfassend mit den allgemeinen Modellen zur Beschreibung von Unsicherheit und den Analysemethoden stellt jedoch die Integration von bedingten Wahrscheinlichkeiten auf Basis von Bayes-Netzwerken in den Planungsprozess einen vielversprechenden Ansatz dar.

Von den allgemeinen Planungsansätzen entspricht das bedingte Planen am besten der in Unterkapitel 2.3 beschriebenen Verhaltensplanung. In dieses Planungsprinzip soll der Analyseprozess auf Basis bedingter Wahrscheinlichkeiten integriert werden. Dabei soll aus der *sensorlosen Planung* die Idee eines Belief States aufgegriffen werden und im Rahmen der Analyse sollen grundsätzlich alle möglichen Zustände betrachtet werden, aus denen für das bedingte Planungsverfahren dann die relevanten Verzweigungen extrahiert werden. Als schnelle Umplanungsverfahren im Sinne des Unterkapitels 2.3 sind die unter *Ausführungsüberwachung und Neuplanung* beschriebenen Echtzeitplanungsverfahren geeignet.

## 4.2 Planung unter Unsicherheit für mechatronische Systeme

Abbildung 4-1 zeigt eine schematische Darstellung des angestrebten Planungsverfahrens zur Berücksichtigung von Unsicherheiten. Es kombiniert die *bedingte Planung* mit der Planausführung und *Ausführungsüberwachung und Neuplanung*. Die Grundidee besteht dabei darin, die bedingte Planung als Standardplanungsverfahren einzusetzen und auf die Neuplanung nur fallweise zurückzugreifen. Damit stellt die Neuplanung eine Rückfallebene dar, die genutzt werden kann, wenn während der Planausführung Situationen auftreten, die im bedingten Plan nicht berücksichtigt wurden. Da an die Neuplanung besondere Anforderungen hinsichtlich der Planungsgeschwindigkeit gestellt werden müssen, die sich zu Lasten der Planqualität auswirken können, ist die Anwendung der Rückfallebene soweit wie möglich zu vermeiden.

Durch die Verzahnung mit der Ausführung ergibt sich der Vorteil, dass auch Informationen, die während der Planausführung gewonnen werden, zur Auswahl der besonders relevanten alternativen Planverläufe genutzt werden können.

Abbildung 4-1 stellt eine zeitliche Aufteilung des angestrebten Verfahrens dar. Dabei bewegt sich das Planungsverfahren zwischen zwei sich verschiebenden Horizonten. Der erste Horizont verschiebt sich mit der Ausführung des Plans und befindet sich immer auf der Höhe der aktuell ausgeführten Aktivität. An dieser Grenze findet logischerweise die Überwachung des Planverlaufs statt. Sie ist zunächst notwendig, um festzustellen, welchen Verzweigungen im bedingten Plan gefolgt werden soll. Andererseits werden durch die Ausführungsüberwachung auch solche Zustände identifiziert, die nicht im bedingten Plan enthalten sind. In einer solchen Situation ist ein echtzeitfähiges Umplanungsverfahren aufzurufen, das versucht, auf einen der alternativen Planverläufe zurückzukehren.

Das bedingte Planungsverfahren erweitert – sofern die Umplanung nicht aktiv ist – den Planbaum laufend um weitere Verzweigungen. Durch diese Erweiterung ergibt sich der zweite



Horizont, der Planungsfortschritt. Um das Auftreten eines nicht vorhergesehenen Zustands und damit den Aufruf der Echtzeitplanung zu vermeiden, muss das bedingte Planungsverfahren den jeweils relevantesten Knoten zur Expansion des Planbaums auswählen. Dabei ist zwischen den Faktoren *Eintrittswahrscheinlichkeit* und der *Zeit bis zum Eintreten der Aktivität* abzuwägen.

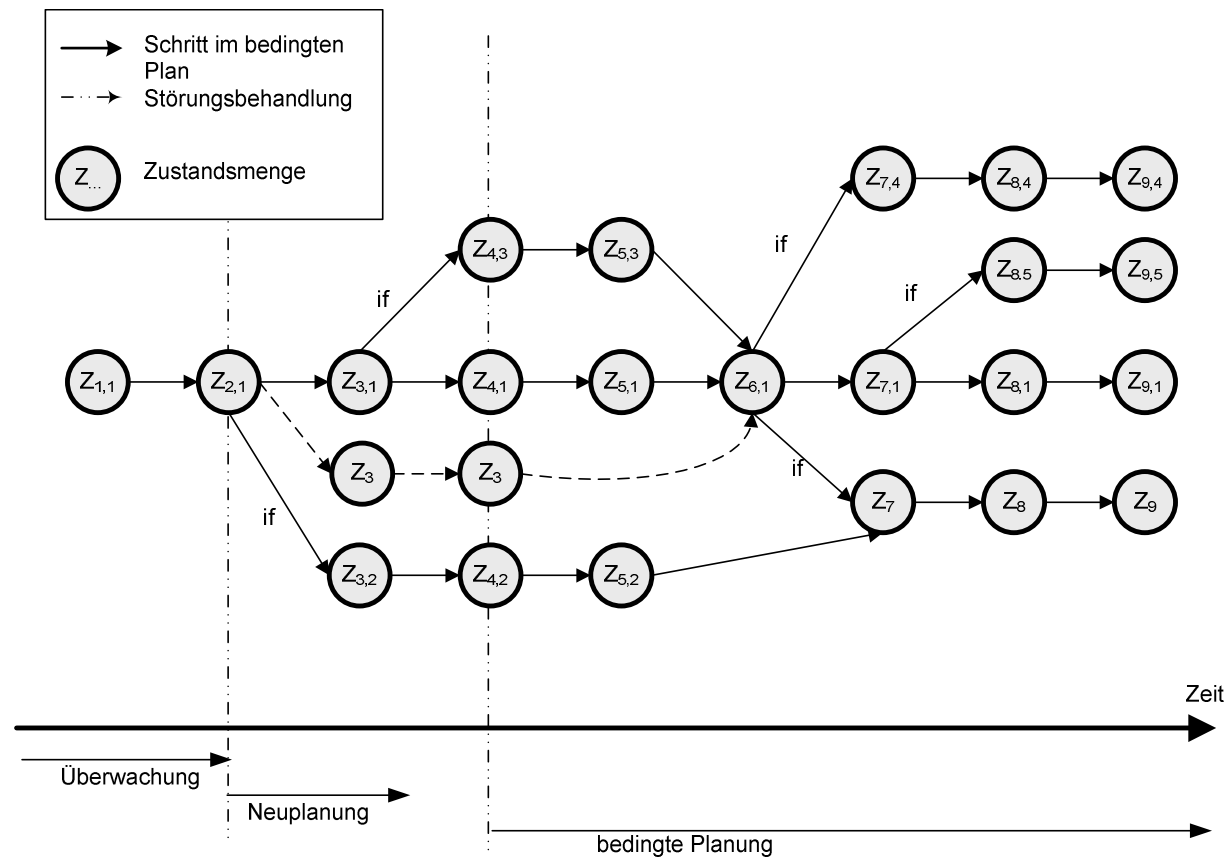
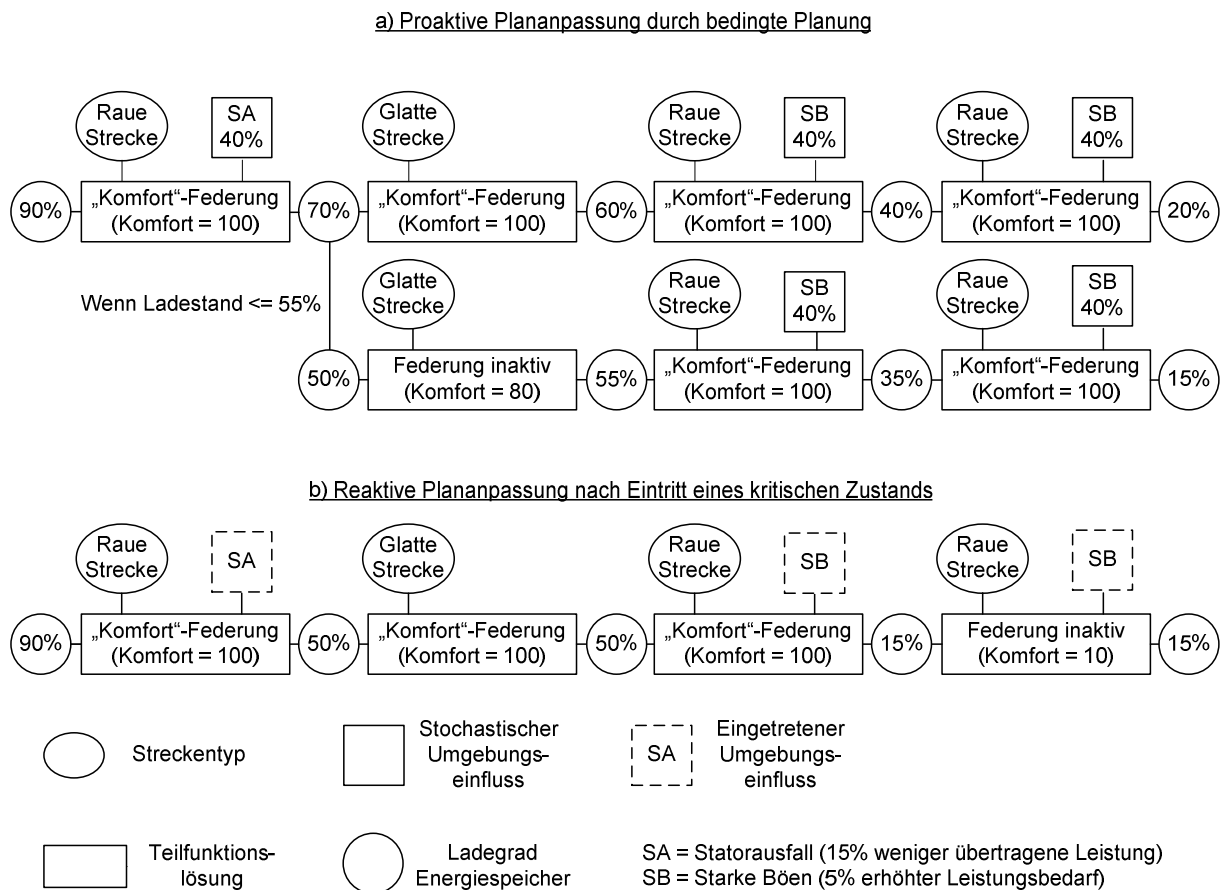


Abbildung 4-1: Skizze des Planungsverfahrens für Unsicherheit

Ein konkretes Beispiel für eine solche Verzweigung im Anwendungsszenario *Energiemanagement des RailCabs* ist ein Zustand, der sich durch einen Statorausfall auf einem der vorher befahrenen Streckenabschnitte ergibt. Sinkt der Ladestand durch die in Folge des Statorausfalls reduzierte Leistungsübertragung so weit ab, dass die Gefahr eines Liegenbleibens besteht, wird ein alternativer Plan erzeugt, der genau so viel Leistung auf dafür geeigneten Streckenabschnitten einspart, dass die Gefahr vermieden wird. Dabei ist es wichtig zu bemerken, dass sich das Verfahren an den Eintrittswahrscheinlichkeiten und den sich aus alternativen Zuständen ergebenden neuen Zustandstrajektorien orientiert und nicht am Auftreten eines einzelnen Ereignisses oder Einflusses. Beispielsweise spielt es keine Rolle, ob ein einmaliger Statorausfall oder wiederholte starke Böen und der dadurch erhöhte Leistungsbedarf der aktiven Federung den geringen Ladestand verursachen.

Abbildung 4-2 veranschaulicht den Vorteil des angestrebten Planungsverfahrens. Das Beispiel beschränkt sich auf die Betrachtung der aktiven Federung und dabei auch auf zwei unterschiedliche Teilfunktionslösungen: „Komfort“-Federung (hoher Komfort bei hohem Energieverbrauch) und inaktive Federung (kein Energieverbrauch und geringerer Komfort). Zwei unterschiedliche Typen von Streckenabschnitten werden unterschieden. Bei *glatter Strecke* verbraucht die Komfortfederung weniger Energie und die inaktive Federung erreicht höheren Komfort als auf der *rauen Strecke*.



**Abbildung 4-2: Bedingte Planung und reaktive Umplanung im Energiemanagement**

Auf die erste Aktivität wirkt mit einer Eintrittswahrscheinlichkeit von 40% der Umwelteinfluss „Statorausfall“. Auf die dritte und vierte Teilfunktionslösung wirkt jeweils der Umwelteinfluss „starke Böen“ mit einer Eintrittswahrscheinlichkeit von 20%. Mithilfe der bedingten Planung kann erkannt werden, dass es zweckmäßig ist, nach Eintreten des Statorausfalls den Plan anzupassen. Im resultierenden alternativen Plan wird die Federung auf dem glatten Streckenabschnitt auf „inaktiv“ gestellt, wodurch am Ende eine Energiereserve von 15% resultiert. Diese Reserve würde auch ausreichen, um auch bei starken Böen auf den letzten beiden Abschnitten im Komfortmodus zu fahren.

Der untere Teil der Abbildung zeigt den Plan, der sich bei einem reaktiven Verhalten einstellen könnte. Da zunächst auch bei einem Statorausfall die Energie ausreicht, um den ursprünglichen Plan zu verfolgen, würde er unverändert beibehalten. Erst wenn zusätzlich auf dem vorletzten Streckenabschnitt starke Böen auftreten, würde das Problem erkannt. Es könnte nur mit einer Abschaltung der Federung auf dem letzten Streckenabschnitt reagiert werden, was größere Komforteinbußen zur Folge hätte.

In den folgenden Kapiteln werden die notwendigen Bausteine des hier umrissenen Verfahrens im Detail beschrieben und konzipiert.

#### 4.2.1 Planungsverfahren auf Basis bedingter Wahrscheinlichkeiten

Ziel dieses Kapitels ist die Definition eines bedingten Planungsverfahrens, das mit der Planausführung verzahnt wird. Das Verfahren soll – abhängig vom tatsächlichen Ausführungsverlauf – möglichst alle eintretenden Situationen berücksichtigen.

Die folgenden Kapitel werden zunächst die Modellierung und Berücksichtigung von bedingten Wahrscheinlichkeiten zur Beurteilung der Eintrittsmöglichkeit unterschiedlicher Situationen vorstellen und anschließend den Aufbau des bedingten Planungsverfahrens erläutern.

#### 4.2.1.1 Modellierung von Planungsproblemen mit bedingten Wahrscheinlichkeiten

Die unsichere Einwirkung des Umfelds mechatronischer Systeme wird in der PDDL und anderen Planungsformalismen nur unzureichend dargestellt. Diese Einwirkung stellt, wie in Kapitel 2.3 dargelegt, die Hauptquelle der Planungsunsicherheit dar. Die Modellierung, die in dem bedingten Planungsverfahren eingesetzt wird, muss zwei Anforderungen erfüllen: Zunächst muss sie für Planungsalgorithmen effizient zu verarbeiten sein, andererseits muss das erstellte Modell reichhaltige und situationsbezogene Informationen über alternative Planverläufe enthalten. Diese Ziele sind zunächst gegenläufig, da die Berücksichtigung alternativer Planverläufe zu einem höheren Verzweigungsgrad führt und die Planungsalgorithmen weniger stringent auf den Zielzustand zustreben können.

Die PDDL muss um die Berücksichtigung ausdrucksstarker Wahrscheinlichkeitsmodelle erweitert werden. Wie in Abschnitt 4.1.3.2 erläutert, ist die existierende Erweiterung durch die PPDDL nicht ausreichend. Um die existierenden Planungssysteme nutzen zu können, soll die PDDL strukturell allerdings nicht verändert werden. Stattdessen wird sie durch eine auf das Planungsproblem angepasste probabilistische Modellierungsmethode ergänzt, die gleichberechtigt neben der PDDL steht. Um die beiden Modellierungsmethoden sinnvoll verzahnen zu können, müssen sie sich für eine gegebene Planungsdomäne die Informationen über die enthaltenen Typen und Objekte teilen.

Auf Basis der formalen Semantik der PDDL (siehe Abschnitt 2.1.1) kann eine ergänzende Form der probabilistischen Modellierung konzipiert werden. Die Berechnung von Eintrittswahrscheinlichkeiten möglicher Planverläufe soll an bestehende Pläne anknüpfen. Daher soll die probabilistische Modellierung sich in ihrer Struktur an die PDDL anlehnen. Dadurch wird es möglich, eine mit Wahrscheinlichkeiten angereicherte Darstellung von PDDL-Plänen zu erhalten, die die Basis für das bedingte Planungsverfahren darstellt. Daraus ergibt sich, dass die probabilistische Modellierung ebenso wie die PDDL in einen allgemeinen Teil zerfällt, der die grundsätzlichen Eigenschaften der Domäne angibt, und in einen speziellen Teil, der ein konkretes Planungsproblem definiert.

Die probabilistische Domänendefinition muss zu jedem Element der PDDL-Domäne eine Entsprechung haben. Jedes Funktionssymbol, jedes Relationssymbol und jedes Aktionsschema muss einen Gegenpart besitzen. Um aus den Grundaktionen eines PDDL-Plans entsprechende Konstruktionen in der probabilistischen Modellierung ableiten zu können, ist es außerdem erforderlich, dass sich die Elemente der PDDL und der probabilistischen Domänendefinition auch in ihrer Stelligkeit entsprechen. In einigen Fällen ist es allerdings notwendig, in der Repräsentation der Elemente abzuweichen. Der Informationsgehalt der probabilistischen Domänendefinition muss teilweise höher sein, da sonst keine sinnvolle Repräsentation der Modellelemente mithilfe von Wahrscheinlichkeitsverteilung möglich ist. In der PDDL-Darstellung von Variablen und Objekten gibt es nur boolesche Variablen (Relationen) und reelle Zahlen (Funktionen). Auf dieser Basis lässt sich im Allgemeinen kein sinnvolles probabilistisches Modell aufbauen.

Der erste und einfachste Schritt in der Definition einer probabilistischen Domänendefinition ist die genauere Spezifikation der Funktionssymbole als Zufallsvariablen. Obwohl die PDDL alle numerischen Werte grundsätzlich als reelle Zahlen modelliert, ist implizit eine genauere Einschränkung des Wertebereichs möglich. Dies ist möglich, da nur durch die numerischen Effekte der Aktionen Veränderungen an den Funktionswerten geschehen. Stellt man die Effekte also so auf, dass zum Beispiel nur ganzzahlige Werte als  $rvalue$ <sup>38</sup> auftreten, und definiert man den Ausgangswert einer Funktion auch als ganzzahligen Wert, modelliert man implizit

---

<sup>38</sup> Ein numerischer Effekt besteht aus einem  $lvalue$  (die zu aktualisierende Variable) und einem  $rvalue$ , der den neuen Wert festlegt (vergleiche Kapitel 2.1.1).

eine ganzzahlige Planungsvariable. Zur sinnvollen Angabe der probabilistischen Domänen-Definition ist es notwendig, diese impliziten Wertebereiche explizit anzugeben. Daher sei:

- $Fs$  die endliche Menge von Funktionssymbolen entsprechend der PDDL-Domäne und
- $range: Fs \rightarrow \mathcal{P}(\mathbb{R})$  eine Funktion, die jedem Funktionssymbol einen möglichen Wertebereich zuordnet.

Listing 4-1 zeigt beispielhaft, wie zusätzlich die Wertebereiche (*ranges*) der Funktionen angegeben werden. Die Stelligkeit der Funktionssymbole kann aus der PDDL-Domänen-Definition entnommen werden.

```
(:function_ranges      (energy_storage → {x ∈ ℝ | 0 ≤ x ≤ 100})
                       (capacity → 100)
                       (comfort → {x ∈ ℕ | 0 ≤ x ≤ 10})
                       (distance → ℝ)
)
```

**Listing 4-1: Funktion in der probabilistischen Domänen-Definition für das RailCab**

Anstatt den gesamten Bereich der reellen Zahlen zu verwenden, wird beispielsweise der Energiespeicher auf den Zahlenbereich zwischen 0 und 100 beschränkt, während die Kapazität des Energiespeichers auf 100 festgesetzt wird und nicht von diesem Wert abweichen kann. Der Komfort wird zur Verdeutlichung in ganzzahligen Schritten von 0 bis 10 bewertet. Durch diese nähere Bestimmung der Zahlenwerte wird die Definition von Wahrscheinlichkeitsverteilungen erleichtert: Es ist beispielsweise möglich, den Komfort mithilfe einer Häufigkeitsverteilung zu beschreiben, die bei Beibehaltung der reellen Zahlen aus dem Wertebereich durch eine kontinuierliche Verteilung nur annähernd hätte beschrieben werden können.

Genau wie bei den Funktionssymbolen bietet sich auch bei der Definition der Relationen im Rahmen der probabilistischen Domänen-Definition eine genauere Modellierung an. Es handelt sich dabei um die logischen Bestandteile eines Zustands. Eine Modellierung als boolesche Zufallsvariable entspräche der PDDL-Darstellung. Damit würden folgende Eigenschaften für jedes Element  $r$  aus  $R_s$  gelten:

- $range(r) = \{\text{Wahr}, \text{Falsch}\}$
- $P(r = \text{Wahr} \vee r = \text{Falsch}) = 1$ .

Obwohl [FL03] dieses Sprachelement als Relation oder logisches Prädikat bezeichnet, hat es oft einen funktionalen Charakter. Ein Beispiel für ein Prädikat mit funktionalem Charakter ist das zweistellige Prädikat (`at ?x railcab ?l location`) aus der mechatronischen Beispieldomäne (siehe 2.1.3). Die Bedeutung dieses Ausdrucks lässt sich vor dem Hintergrund der Anwendung interpretieren: Das Prädikat gibt die Position eines RailCabs an. Derartige Prädikate sind also für einen der beiden Parameter (im Beispiel `?x`) eindeutig. Der Formalismus der PDDL ist allerdings nicht in der Lage, diese Beschränkung darzustellen. Stattdessen ist es wiederum erforderlich, die Beschränkung implizit in den Aktionsschemata zu modellieren. Im Rahmen der deterministischen Dynamik der PDDL ist dies einfach zu realisieren – für Prädikate dieser Art muss nur sichergestellt sein, dass jede Aktion  $Add_a$  maximal ein einziges Prädikat mit gegebenem Parameter dieses Typs enthält und gleichzeitig  $Del_a$  das entsprechende Prädikat aus der Vorbedingung enthält. Die Darstellung innerhalb einer probabilistischen Modellierung der Domäne ist erheblich komplizierter. In der probabilistischen Modellierung muss die Kon-

junktion aller möglichen Belegungen eines Relationensymbols  $r$  mit einem gegebenen lparameter  $l$  die Wahrscheinlichkeit eins haben:

$$P\left(\bigvee_{l \in r} r\right) = 1$$

Diese Bedingung ist kaum darzustellen, wenn alle Relationensymbole gleich modelliert werden. Für die probabilistische Domänendefinition wird die Menge der Relationensymbole  $Rs$  daher in zwei Teilmengen aufgeteilt:

- $Rs_{\text{func}} \subseteq Rs$  enthält diejenigen Relationensymbole, die einen funktionalen Charakter haben und
- $Rs_{\text{rel}} = Rs / Rs_{\text{func}}$  enthält alle übrigen Relationensymbole, die keinen funktionalen Charakter haben.

Durch diese Unterscheidung wird es möglich, bei der Darstellung der Aktionen in der probabilistischen Domänendefinition den unterschiedlichen Charakter der logischen Ausdrücke zu berücksichtigen. Damit lässt sich die Struktur der probabilistischen Domänendefinition bereits angeben. Eine probabilistische Domäne  $\text{dom}_{\text{prob}}$  ist ein Sieben-Tupel  $\text{dom}_{\text{prob}} = (\text{dom}_{\text{PDDL}}, Fs, Rs_{\text{func}}, Rs_{\text{rel}}, As, \text{arity}, \text{range})$ . Dabei bezeichnet  $\text{dom}_{\text{PDDL}}$  die entsprechende PDDL-Domänendefinition. Insgesamt sind folgende Anforderungen an den Zusammenhang der Domänendefinitionen zu stellen:

- $Fs$  ist in beiden Domänendefinitionen die gleiche Menge,
- $Rs_{\text{func}} \cup Rs_{\text{rel}} = Rs$  und  $Rs_{\text{rel}} \cap Rs_{\text{func}} = \{\}$ ,
- $As$  ist in beiden Domänendefinitionen die gleiche Menge und
- $\text{arity}$  ist in beiden Domänendefinitionen die gleiche Stelligkeit

Nachdem die Elemente und die Struktur der probabilistischen Domänendefinition angegeben und definiert wurden, muss nun angegeben werden, wie die dynamischen Zusammenhänge in der Domäne dargestellt werden können. Anknüpfungspunkt hierfür sind Aktionen. Die Struktur der Aktionen weicht dabei in einem Punkt von der Struktur der PDDL-Aktionen ab. In der PDDL werden die positiven und negativen logischen Effekte einer Aktion  $a$   $Add_a$  und  $Del_a$  isoliert modelliert. Diese Darstellung kann bei Relationensymbolen, die keinen funktionalen Charakter haben, beibehalten werden. Für die Relationensymbole mit funktionalem Charakter  $Rs_{\text{func}}$  ist ein eigenes Effektelement  $FP_a$  einzuführen.

In der probabilistischen Modellierung ist die Dynamik zu beschreiben mittels einer gemeinsamen Wahrscheinlichkeitsverteilung über die Effekte und Vorbedingungen der Aktionen:

$$P\langle Add_a, Del_a, NP_a, FP_a, Pre_a \rangle$$

Zur Darstellung dieser Wahrscheinlichkeitsverteilung werden Bayes-Netzwerke (vergleiche 4.1.2.1) verwendet. Dabei wird nicht versucht werden, die Wahrscheinlichkeitsverteilung in einem einzigen Netzwerk darzustellen. Stattdessen wird für jeden einzelnen Effekt einer Aktion  $a$  (für jedes Element aus  $Add_a, Del_a, NP_a$  und  $FP_a$ ) ein eigenes Netzwerk modelliert.

Gewisse Einschränkungen und Vorgaben sind an das Netzwerk zu machen, um einerseits eine effiziente Inferenz durchführen zu können und andererseits den Modellierungsaufwand einzuschränken sowie die notwendigen Informationen aus einem bestehenden PDDL-Plan ableiten zu können. Zunächst ist die Inferenzrichtung für die Anwendung in der Planung eindeutig

festgelegt: Die Inferenz verläuft von den Vorbedingungen einer Aktion zu den Nachbedingungen einer Aktion. Durch diese Überlegung und die Festlegung, dass für jeden probabilistischen Effekt ein eigenes Netzwerk konstruiert wird, kann die Netzwerkstruktur allgemein beschrieben werden. Die allgemeine Beschreibung gibt dabei an, welche Zufallsvariablen an welchen Stellen in einem Bayes-Netzwerk zur Beschreibung probabilistischer Effekte auftreten können und welche Verknüpfungen möglich sind. Abbildung 4-3 stellt diese allgemeine Struktur dar. Die Struktur lässt sich zunächst in drei unterschiedliche Ebenen aufteilen: die Eingabeebene, die Zwischenebene und die Ausgabebene. Auf der Ebene der Ausgabe befindet sich immer nur eine einzige Variable, die modellierte Effektvariable. Die Eingabeebene enthält grundsätzlich nur Variablen, die sich aus den Preconditions der Aktion ableiten lassen. Dies ist erforderlich, um aus einer Grundaktion innerhalb eines PDDL-Plans die für den Inferenzprozess notwendigen Evidenzen ableiten zu können. Die Knotenmengen der einzelnen Ebenen einer Effektvariablen  $e$  einer Aktion  $a$  seien im Weiteren bezeichnet als:

- $Input_{a,e}$  für die Knoten der Eingabeebene,
- $Hidden_{a,e}$  für die Knoten der Zwischenebene,
- $Output_{a,e}$  für den Effektknoten.

Das Netzwerk wird als  $BN_{a,e}$  bezeichnet. Dabei sind nur folgende gerichtete Kanten im Netzwerk einer Aktion  $a$  erlaubt:

- von den Knoten der Eingabeebene zum Effektknoten,
- von den Knoten der Eingabeebene zu den Knoten der Zwischenebene,
- von Knoten der Zwischenebene zu anderen Knoten der Zwischenebene und
- von Knoten der Zwischenebene zum Effektknoten.

Durch die Möglichkeit, Knoten innerhalb der Zwischenebene miteinander zu verbinden, besteht die Gefahr, Zyklen in das Netzwerk einzubauen. Die Bedingung der Zyklenfreiheit muss während der Modellierung der Netzwerke eingehalten werden.

Besondere Erläuterung bedarf die Zwischenebene, die Zufallsvariablen enthalten kann, die weder in PDDL noch in den bisher erläuterten Bestandteilen der probabilistischen Domänendefinition definiert sein müssen. Bei der Zwischenebene handelt es sich um eine Modellierungsvereinfachung, die es bei geschickter Anwendung auch erlaubt, die Komplexität der bedingten Wahrscheinlichkeitsverteilung der Effektvariablen zu begrenzen. Eine derartige verborgene Variable kann eingesetzt werden, wenn der Zusammenhang mehrerer Eingabevariablen auf die Ausgabevariable nicht ohne Weiteres darstellbar ist. Zusätzlich reduziert sich die Stelligkeit der Wahrscheinlichkeitsverteilung der Effektvariablen, wenn die Anzahl der Parameter oder Ausprägungen der verborgenen Variablen kleiner ist als die Summe der Parameter oder Ausprägungen ihrer Elternvariablen.

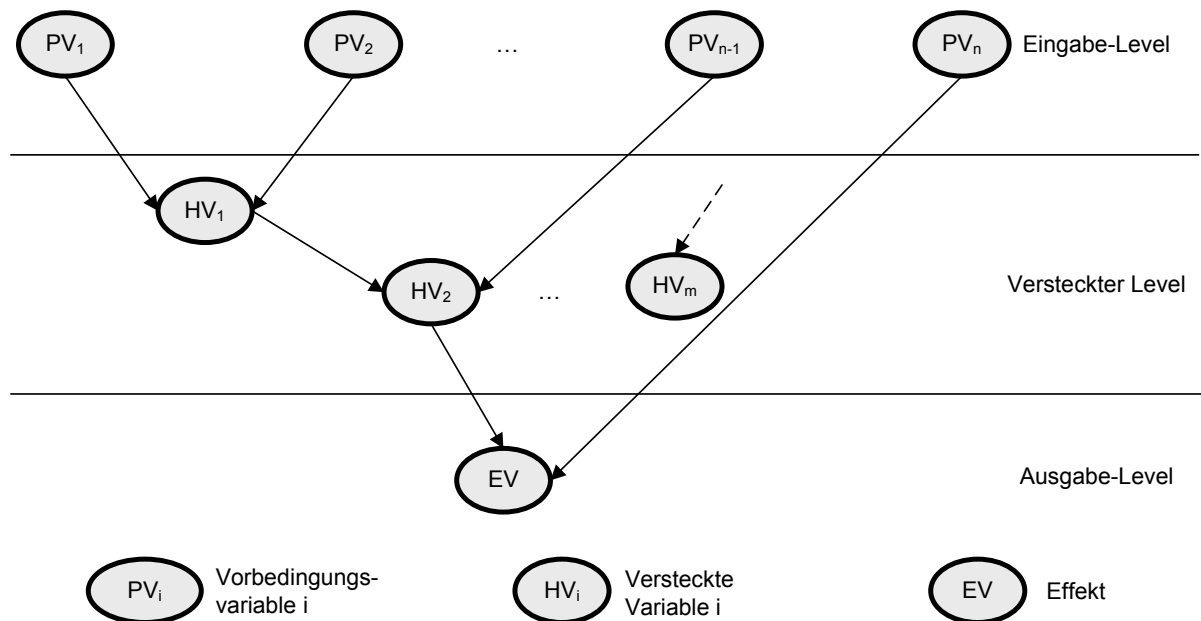


Abbildung 4-3: Allgemeine Struktur eines Bayes-Netzwerks für probabilistische Effekte

Abbildung 4-4 zeigt ein mögliches Netzwerk aus der Aktion Drive. Das Netzwerk beschreibt den Zusammenhang zwischen den Aktionsparametern *time* (Tageszeit), *month* (Monat), *energy storage* (Ladegrad des Energiespeichers), *distance* (Entfernung zwischen Start und Ende der Fahraktion) und der Ausgangsvariable *energy storage*. Während Energiespeicher (also der Ladestand vor der Fahrt) und die Länge der zu fahrenden Strecke unmittelbar auf den Energiespeicher nach der Fahrt wirken, wirken die beiden Einflüsse Tageszeit und Monat nicht direkt auf den Ladestand am Ende der Aktion. Stattdessen werden einige weitere Zufallsvariablen abgeleitet. Es handelt sich um Streckeneigenschaften wie *humidity* (Feuchtigkeit), *ice* (Vereisung) und *leaves* (Blätter). Diese Streckeneigenschaften werden über die physikalische Eigenschaft *friction* (Reibung), die Einfluss (über die nicht modellierte Wirkkette Stärke der Anregung und Energieverbrauch der aktiven Federung) auf den Ladestand am Ende der Fahraktion hat, abgebildet.

Eine wichtige Einschränkung zur Konstruktion der Bayes-Netzwerke wird dahingehend gemacht, dass die Eingabeebene Informationen über logische Atome nur *indirekt* beinhalten darf. *Indirekt* meint dabei, dass nicht die konkreten logischen Atome auftreten dürfen, sondern nur Informationen über deren numerische Eigenschaften. Dabei können aber sehr wohl numerische Eigenschaften vorkommen, die nicht Bestandteil des PDDL-Aktionsschematas sind. Der Grund für diese Einschränkung liegt darin, dass die Aufteilung in Domänendefinition und Problemdefinition den Modellierungsaufwand reduzieren soll. Dazu ist es erforderlich, dass möglichst große Anteile der Modellierung auf die Domänendefinition entfallen. Ein wesentlicher Anteil des Modellierungsaufwands entfällt vor allem auf die Spezifikation und Angabe der bedingten Wahrscheinlichkeitsverteilungen in den Bayes-Netzwerken. In der Domäne liegen aber keinerlei Informationen über Objekte vor, die genutzt werden könnten, um bedingte Wahrscheinlichkeitsverteilungen zu definieren. Würde auf diese Einschränkung verzichtet, müsste die Spezifikation der Verteilungen in die Problemdefinition verschoben werden. Weiterhin ist davon auszugehen, dass die meisten Einflüsse von Elementen der Vorbedingung in der Regel quantifizierbar sind (wie z. B. der räumliche Abstand und die durchschnittliche Steigung zwischen zwei Punkten) und daher auch explizit modelliert werden sollten.

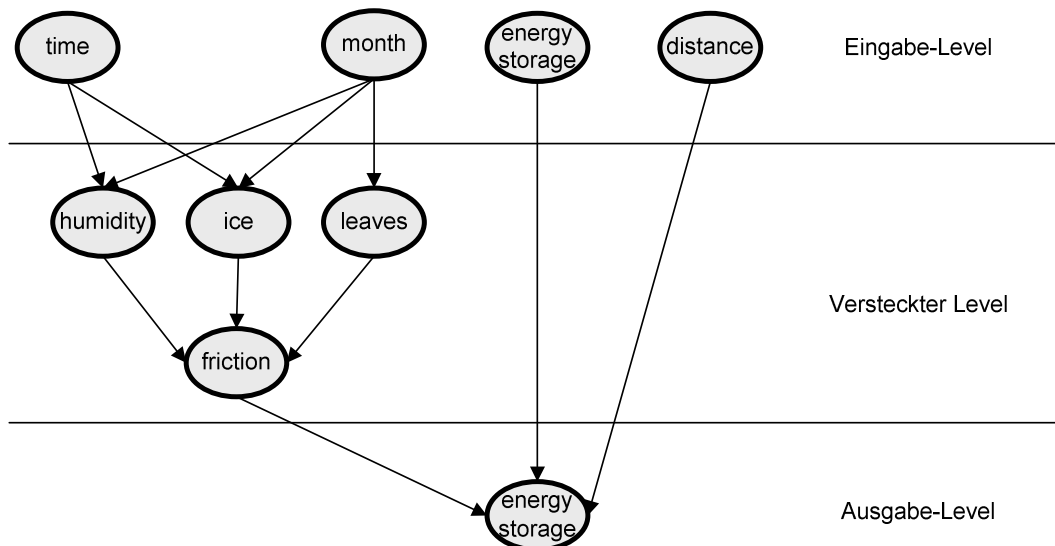


Abbildung 4-4: Beispiel Netzwerk aus der Aktion *Drive*

Neben der Modellierung der Netzwerkstruktur ist auch die innere Modellierung der Knoten von besonderer Bedeutung. Die möglichen Formen der bedingten Wahrscheinlichkeitsverteilungen, die die Knoten letztlich repräsentieren, hängen dabei von den Wertebereichen der Zufallsvariablen ab, für die ein Knoten und dessen Elternknoten stehen. Aufgrund der Zwischenschicht und der Modellierungsfreiheit beim Wertebereich der Funktionssymbole können grundsätzlich sämtliche Kombinationen diskreter und kontinuierlicher Variablen auftreten.<sup>39</sup> Die Darstellungsmöglichkeiten wurden in Abschnitt 4.1.2.1 dargestellt. Die numerischen Funktionssymbole  $F_s$  und die Relationensymbole mit nicht-funktionalem Charakter sind dabei in der Modellierung unproblematisch. Erstere werden entsprechend des angegebenen Wertebereichs als diskrete oder kontinuierliche Zufallsvariablen modelliert, letztere als boolesche Variable. Dabei ist natürlich zu beachten, dass Relationensymbole nur als Effektvariable auftreten können – modelliert wird daher die Wahrscheinlichkeit, dass ein positiver ( $Add_a$ ) oder negativer ( $Del_a$ ) Effekt der modellierten Aktion  $a$  eintritt.

Die Modellierung der Relationensymbole mit funktionalem Charakter hat dagegen keine direkte Auswirkung auf den Wertebereich und die Eintrittswahrscheinlichkeit eines positiven oder negativen Effekts. Stattdessen muss die Modellierung so gewählt sein, dass immer ein konsistenter Glaubenszustand hinsichtlich dieser Variablen gegeben ist. Um sicherzustellen, dass die Konjunktion aller funktionalen Relationensymbole hinsichtlich eines gegebenen  $l$ -parameters immer eine Wahrscheinlichkeit von genau 1 hat, soll eine solche Variable innerhalb des Bayes-Netzwerks nicht als boolesche, sondern als diskrete Variable modelliert werden. Mögliche Ausprägungen sind dabei alle Belegungen des Relationensymbols, die sich aus den Parametern des Aktionsschemas erzeugen lassen. Dabei wird unterstellt, dass alle anderen Relationensymbole die Wahrscheinlichkeit 0 besitzen.

Genau wie in der PDDL ist für ein konkretes Planungsproblem die Angabe einer Probleminstanz notwendig. Da das hier angestrebte Planungsverfahren nicht wie die meisten in Abschnitt 4.1.4.4 vorgestellten Verfahren eine Schwellenwahrscheinlichkeit erreichen soll, ist die probabilistische Definition eines Zielzustandes nicht notwendig. Die Menge der Objekte kann der PDDL-Problemdefinition entnommen werden. Somit muss lediglich der Initialzustand spezifiziert werden. Dieser wird angegeben mit einer Menge von Wahrscheinlichkeitsverteilungen über die numerischen und logischen Ausdrücke der Probleminstanz.

<sup>39</sup> Voraussetzung für die Modellierung einer kontinuierlichen Variable und ihre Verknüpfung mit anderen Variablen ist die Implementierung geeigneter Inferenzmechanismen.



```
(:init
  (at railcab1 location1)
    (= (capacity railcab1) 100)
    (= (energy-storage railcab1) simpson(20,70,50))
    ...
)
```

**Listing 4-2: Ausschnitt einer probabilistischen Problemdefinition des RailCab-Szenarios**

Listing 4-2 zeigt einen Ausschnitt aus einer Problemdefinition aus dem RailCab Szenario. Die ersten beiden Variablen werden als *sicher* modelliert. So steht das RailCab 1 garantiert an Ort location1 und der Energiespeicher hat eine Kapazität von 100. Die Kapazität könnte auch keinen anderen Wert haben, da ihr Wertebereich in der Domänendefinition in Listing 2-2 auf den Wert 100 beschränkt wurde. Die Variable Energiespeicher ist dagegen zu Beginn des Planungsproblems nicht sicher bekannt. Sie ist dreiecksverteilt<sup>40</sup> mit den Parametern 20, 70 und 50.

Auch Planmetriken müssen besondere Berücksichtigung im Rahmen der probabilistischen Modellierung erfahren. Für die Variablen, die in einer Planmetrik angegeben werden, müssen in der probabilistischen Problemdefinition weitere Informationen angegeben werden. Diese Informationen werden genutzt, um mögliche ungünstige Abweichungen vom deterministischen Planverlauf festzustellen. Die dafür notwendigen Informationen bzgl. jeder in der Planmetrik enthaltenen numerischen Zustandsvariablen sind die Richtung, in der sich die Änderung negativ auf die Planmetrik auswirkt, und ab welcher relativen Abweichung eine Behandlung durch einen Alternativplan sinnvoll ist. Listing 4-3 zeigt ein Beispiel für eine solche Planungsmetrik, die aus der dritten PDDL-Planungsmetrik aus Listing 2-7 abgeleitet wurde. Diese Planungsmetrik beschreibt, dass Abweichungen nach unten im Komfort sich negativ auf die Planqualität auswirken und eine Behandlung eines Ereignisses, das zu einer Reduzierung des erwarteten Komforts führt, bei einer Abweichung von mindestens 10% erfolgen soll. Ebenso wirken sich Zuwächse beim Energieverbrauch negativ aus, sollen aber erst bei einer Abweichung von 15% behandelt werden.

```
(: metric
  (Comfort (decrease, 10%)
  (Total-energy-used (increase, 15%)
  )
```

**Listing 4-3: Planmetrik in der probabilistischen Problemdefinition**

#### 4.2.1.2 Probabilistische Analyse von Plänen

Mit dem im vorherigen Kapitel beschriebenen Modellierungsansatz ist die Grundlage für eine probabilistische Analyse von Plänen gelegt. Es ist also möglich, anhand der probabilistischen Domänendefinition, dem Initialzustand und einem vorliegenden PDDL-Plan eine Wahrscheinlichkeitsverteilung für möglicherweise im Verlauf der Planausführung eintretende Zustände zu bestimmen. Die deterministische Sicht des Plans wird überführt in einen Planungsbaum, dessen Knoten anstelle einzelner Zustände Zustandsmengen enthalten und deren Elemente mit einer Eintrittswahrscheinlichkeit bewertet sind. Abbildung 4-5 zeigt den Planbaum, der sich nach der Analyse ergibt.

<sup>40</sup>Die Dreiecksverteilung wird auch als Simpsonverteilung bezeichnet.

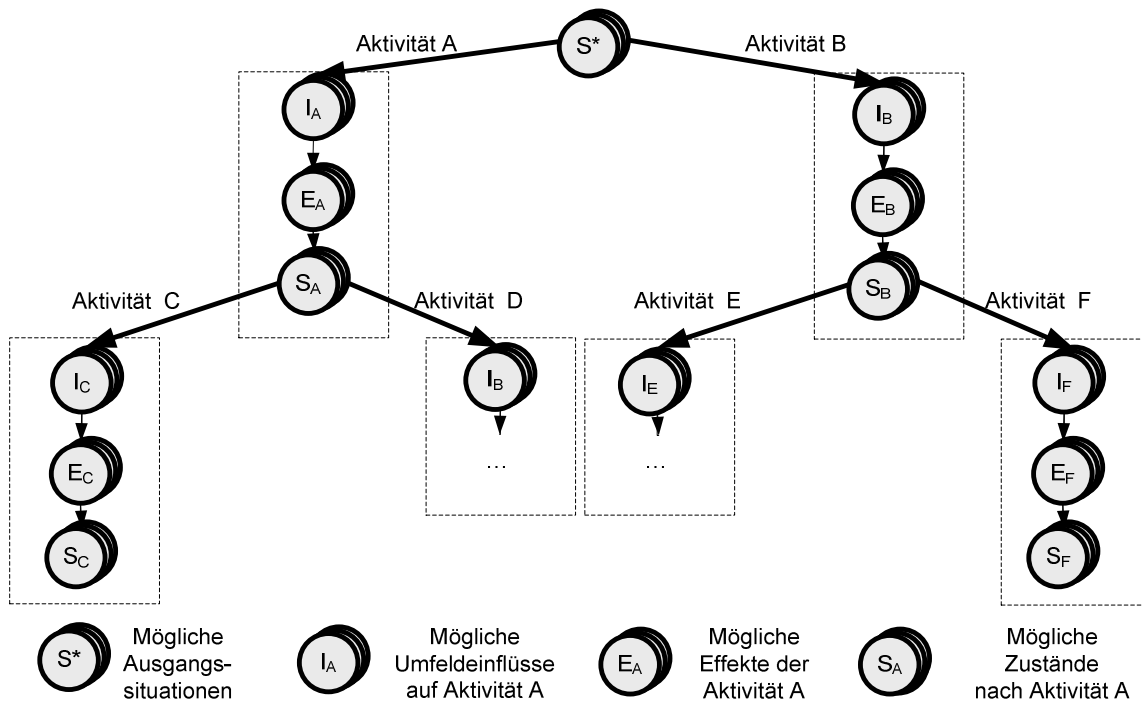


Abbildung 4-5: Planbaum nach der probabilistischen Analyse

Um den Analyseprozess zu definieren, sind zwei Schritte zu beschreiben. Der erste Schritt konstruiert eine probabilistische Repräsentation des deterministischen Plans. Im zweiten Schritt werden mittels eines Inferenzalgorithmus die aktuellen Wahrscheinlichkeitsverteilungen bestimmt.

Abbildung 4-6 stellt die Struktur der probabilistischen Planrepräsentation dar. Sie besteht im Wesentlichen aus zwei einander folgenden Elementen: Zuständen und Aktivitäten. Wie bereits in der PDDL lässt sich ein Zustand dabei in zwei Teilmengen unterteilen: den numerischen und den logischen Teil. Die Elemente der Teilmengen ergeben sich genau wie in der PDDL über die möglichen Belegungen der numerischen Funktionen (z. B. Ladestand des Energiespeichers *energy\_storage*) und Relationensymbole (z. B. Ortsbestimmung mittels des Prädikats *at*) mit den Objekten einer Domäne (also dem Ladestand der Energiespeicher von RailCab 1 und RailCab 2 sowie den Positionen von RailCab 1 und RailCab 2). Elemente des numerischen Teils des Zustands sind Wahrscheinlichkeitsverteilungen über die Wertebereiche der entsprechenden Domänenfunktionen. Hinsichtlich des logischen Bestandteils müssen der funktionale und der relationale Teil unterschieden werden. Für den funktionalen Teil ist für jeden möglichen lparameter eines Relationensymbols ein Element im Zustand enthalten. Der Wertebereich dieses Modells ergibt sich aus allen möglichen Belegungen der Relationen bei gegebenem lparameter. Im relationalen Teil des logischen Zustands sind die Wahrscheinlichkeitsverteilungen der booleschen Variablen enthalten. Die Zustandsübergänge – und damit die Umfeldeinflüsse auf die Aktivitäten und die daraus resultierenden möglichen Effekte – sind durch die Aktivitäten im Plan gegeben, genauer gesagt durch die den entsprechenden Teilfunktionslösungen zugeordneten Bayes-Netzwerke. Implizite Annahme dabei ist, dass Zustandselemente, die in einer Aktion nicht enthalten sind, die Verteilung des Vorgängerzustands beibehalten. Diese implizite Annahme entspricht der Annahme in der klassischen Planung, dass Aktionen im Plan die einzige Quelle von Veränderungen sind. Diese Annahme wird zunächst aufrechterhalten.<sup>41</sup>

<sup>41</sup> Sie wird in Abschnitt 4.2.1.3 abgeschwächt.

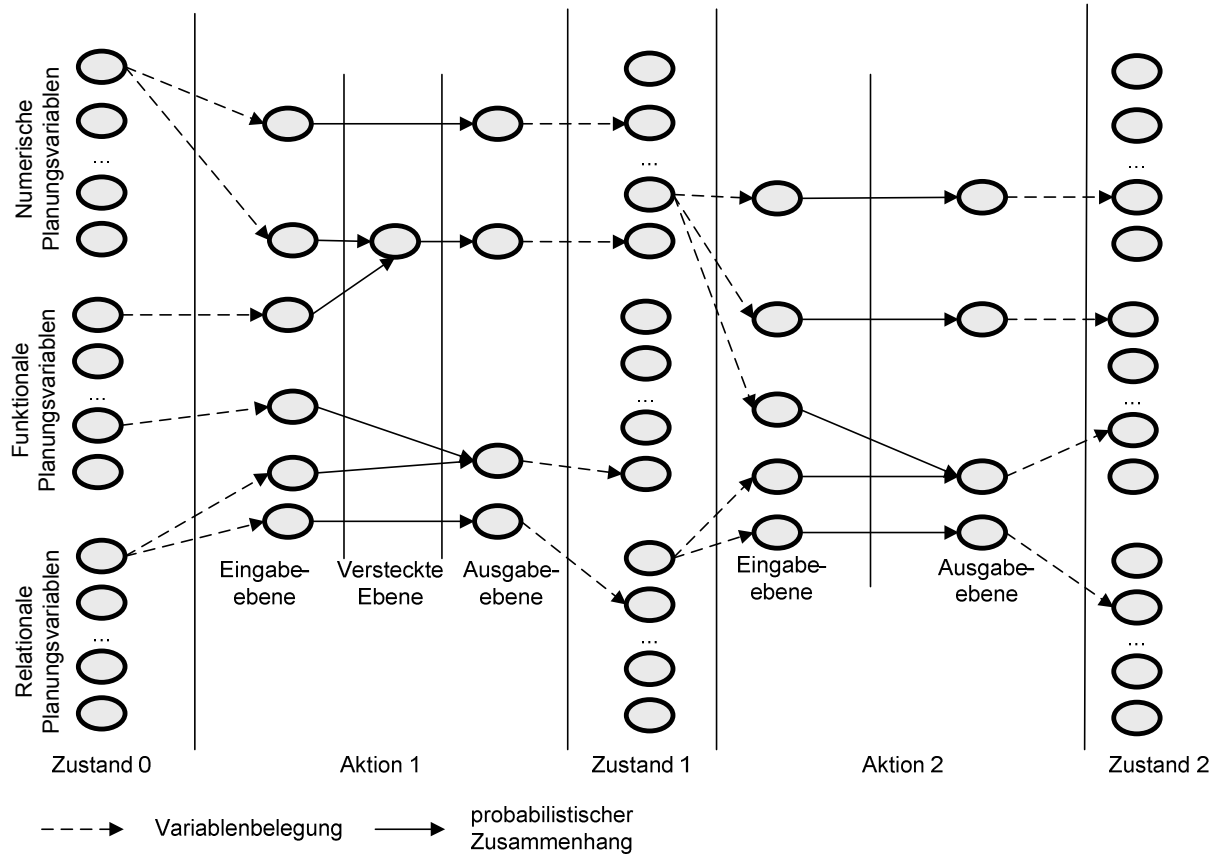


Abbildung 4-6: Struktur der probabilistischen Planrepräsentation

Listing 4-4 zeigt in Form von Pseudo-Code, wie der in Abbildung 4-6 skizzierte Graph konstruiert und ausgewertet wird. Eingangswert für die Analyse stellt ein PDDL-Plan dar. Ausgangspunkt ist der Initialzustand des Planungsproblems. Über die erste Aktivität wird der nächste Zustand ermittelt. Dazu werden die Bayes-Netzwerke aller Effekte der Aktivität ausgewertet. Der erste Schritt dieser Auswertung ist die Belegung der Knoten der Eingabewerte mit den Werten des aktuell betrachteten Zustands. In der Nomenklatur der probabilistischen Inferenz entspricht dies dem Setzen der Evidenzen. Anschließend wird auf dem Bayes-Netzwerk, das zu der entsprechenden Effektvariablen definiert ist, eine Inferenz durchgeführt, deren Ergebnis eine aktualisierte Wahrscheinlichkeitsverteilung der Effektvariablen ist. Diese aktualisierte Verteilung wird in den Nachfolgezustand übertragen. Da die Inferenz nur für die Effektvariablen der jeweiligen Aktion durchgeführt wird, muss nur eine Teilmenge der Zustandsvariablen betrachtet werden, was den Berechnungsaufwand reduziert. Im letzten Schritt der Bestimmung des Nachfolgezustands werden die Verteilungen aller Variablen, die nicht durch die Aktivität betroffen sind, aus dem vorherigen Zustand übertragen.

In Listing 4-4 wird die Inferenz auf den Bayes-Netzwerken der Effekte  $BN_{a,e}$  lediglich als einzelner Aufruf dargestellt. Während der Rahmen von Listing 4-4 aber nur die Konstruktion der probabilistischen Planrepräsentation darstellt und ausschließlich Wertzuweisungen enthält, geschieht die eigentliche Berechnung von Effekten und damit neuer Zustände in der probabilistischen Inferenz.

Durch die im Abschnitt 4.2.1.1 gewählte Modellierung ist sichergestellt, dass jeweils nur die aktuelle Wahrscheinlichkeitsverteilung der Effektvariable  $Eff$  abgefragt wird. Dadurch bietet sich ein Inferenzalgorithmus an, der gezielt eine solche Verteilung  $P(Eff | e)$  unter Vorliegen der Evidenzen  $e$  berechnet, anstatt die vollständige gemeinsame Verteilung des Netzwerks zu berechnen. Ein solcher Inferenzalgorithmus ist der Message-Passing-Algorithmus von Pearl [Pearl88]. In Bezug auf den abgefragten Knoten  $X$  teilt der Message-Passing-Algorithmus die Evidenzen in Evidenzvariablen  $E_X^+$ , die nur über Elternknoten von  $X$  erreichbar sind, und

Evidenzvariablen  $E_x^-$ , die nur über die Kinds-knoten von  $X$  erreichbar sind, ein. Die aktuelle Verteilung von  $X$  wird berechnet, indem der Beitrag von  $E_x^+$  als Wirkung auf die Elternknoten und der Beitrag von  $E_x^-$  als Wirkung auf die Kinds-knoten von  $X$  berechnet wird. Im letzten Schritt der Inferenz wird der Effekt der aktualisierten Verteilung auf  $X$  bestimmt. Da die Effektvariable  $Eff$  die einzige abgefragte Variable und in jedem Fall eine Senke im Netzwerk ist, ist die Menge  $Eff$  immer leer. Dadurch lässt sich der Message-Passing-Algorithmus vereinfachen. Nach der probabilistischen Analyse eines deterministischen Plans steht eine Planrepräsentation zur Verfügung, die keine eindeutigen Zustände, sondern Zustandsmengen enthält. Zusätzlich sind Informationen über die Eintrittswahrscheinlichkeit dieser Zustände verfügbar. Dabei ist zu beachten, dass Informationen über Eintrittswahrscheinlichkeiten nicht direkt, sondern nur indirekt über einzelne Zustandselemente vorliegen. Das gibt jedoch die nötige Freiheit, die Verzweigungen innerhalb des bedingten Plans situationsabhängig zu bestimmen.

#### **AnalysierePlan** (Plan $p$ )

```

Zustand  $s_c$  = erster Zustand
Zustand  $s_n$  = Nachfolger von  $s_c$ 
Aktion  $a$  = erste Aktion im Plan

Tue
  Über alle  $e \in NP_a \cup FP_e \cup Pre_a$ 
    Über alle  $n \in Input_{a,e}$ 
       $n = s_c.n$ 
      inferenz( $BN_{a,e}$ )
       $s_n.e = Ausgabe_{a,e}$ 
    Tue
  Tue
  Über alle  $e \in s_n / (NP_a \cup Pre_a)$ 
     $s_n.e = s_c.e$ 
  Tue
     $s_c = s_n$ 
     $s_n$  = Nachfolger von  $s_c$ 
     $a$  = nächste Aktion
Solange eine Aktion  $a$  im Plan enthalten ist

```

#### **End AnalyzePlan**

**Listing 4-4: Probabilistische Analyse eines Plans**

### **4.2.1.3 Einbeziehen von Experteninformationen über Umfeldeinflüsse**

Einer der wesentlichen Kritikpunkte an bestehenden Ansätzen zur probabilistischen Planung war die unzureichende Verwertung aktueller Informationen über das Ausführungsumfeld eines Plans. In diesem Abschnitt wird ein Ansatz vorgestellt, der es erlaubt, Experteninformationen über das Umfeld einzubeziehen. Abbildung 4-5 zeigt, dass die Modellierung der Teilfunktionslösungen bzw. Aktivitäten mithilfe Bayes-Netzwerke die Berücksichtigung von Umfeldeinflüssen bereits erlaubt. Beschränkt man allerdings die Umfeldeinflüsse auf die Informationen, die in der probabilistischen Domänen-Definition enthalten sind, können nur sehr allgemeine Informationen berücksichtigt werden.

Expertensysteme (z. B. realisiert als Softwareagenten), die über ein detaillierteres Wahrscheinlichkeitsmodell hinsichtlich bestimmter Umfeldeinflüsse verfügen, können genutzt werden, um möglichst genaue Wahrscheinlichkeitsverteilungen erhalten. Die Grundidee besteht dabei darin, dem Planungssystem innerhalb der probabilistischen Analyse die Möglichkeit zu geben, externe Expertensysteme nach einer aktuellen Wahrscheinlichkeitsverteilung einer bestimmten Variablen zu befragen. Dadurch können auch Veränderungen und Entwicklungen, die nicht durch die Planausführung verursacht werden, innerhalb der eigentlichen Planung berücksichtigt werden.

Die Anfragen des Planungssystems an die Expertensysteme beinhalten a posteriori Informationen, die während der Planung erzeugt und gesammelt werden. Zu diesen Informationen gehört vor allem, wann welche Teilfunktionslösung wo ausgeführt werden soll. Abbildung 4-7 stellt den Ablauf der Interaktion des Planungssystems mit den Expertensystemen als Zustandsautomat dar. Nach der Initialisierung des deterministischen Plans durchwandert die probabilistische Analyse die Aktivitäten. Bei jeder Aktivität besteht die Möglichkeit, zu bestimmten Umgebungsvariablen – das können sowohl Variablen aus dem Input-Layer als auch aus dem Hidden-Layer sein – die aktuelle a posteriori Verteilung bei einem Expertensystem zu erfragen. Nachdem alle Variablen zu einer Aktivität abgefragt sind, kann mit dem nächsten Zustands-Aktivitätenpaar fortgefahren werden.

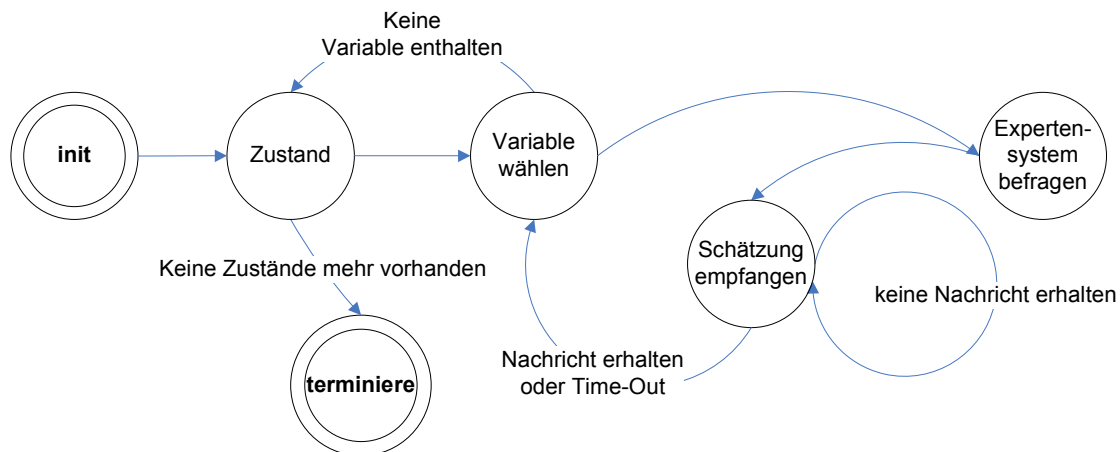


Abbildung 4-7: Zustandsautomat der erweiterten probabilistischen Analyse

Um den Ablauf der erweiterten probabilistischen Inferenz effizient und flexibel zu gestalten, werden Variablen, deren Verteilung von Expertensystemen erfragt wurden, so behandelt, als hätten sie keine weiteren Elternknoten. Das bedeutet, falls das Expertensystem eine Antwort liefert, wird diese im Weiteren als a-posteriori-Wahrscheinlichkeitsverteilung genutzt, unabhängig von dem Ergebnis der Inferenz im Bayes-Netzwerk der Teilfunktionslösung. Für den Fall, dass ein Expertensystem nicht antwortet und es zum Time-Out kommt, kann die a-posteriori-Wahrscheinlichkeit der Variablen weiterhin wie in Abschnitt 4.2.1.2 bestimmt werden. Das bedeutet zwar einen Verlust an Aktualität und Genauigkeit, verhindert aber einen Abbruch der Analyse.

Ein Beispiel für die erweiterte probabilistische Analyse liefert das Netzwerk aus Abbildung 4-4, das die Wahrscheinlichkeitsverteilung über die Ausgangsvariable *energy storage* (Ladegrad Energiespeicher) ermittelt. Dabei werden im Bayes-Netzwerk die Variablen *humidity* (Feuchtigkeit), *ice* (Vereisung) und *leaves* (*Laub auf der Schiene*) benutzt, um auf die physikalische Eigenschaft *friction* (Reibung) zu schließen. Da dieses Bayes-Netzwerk für die Aktion *drive* definiert wurde, sind die Verteilungen allgemein gehalten und gelten über alle möglichen Streckenabschnitte. Entsprechend ungenau sind die Wahrscheinlichkeitsverteilungen. Wird stattdessen ein Expertenagent, der logisch in der Umgebung eingebettet ist, beispielsweise über die Variable *Laub* befragt, kann er spezielles Wissen über die Umgebung einbeziehen. Beim Laub wären dies etwa Informationen über den Baumbestand an der Strecke. Die Modellierung könnte im Extremfall so fein sein, dass für einzelne Baumarten Wahrscheinlichkeitsmodelle über den Zeitpunkt enthalten sind, zu dem sie ihre Blätter fallen lassen.

Das erweiterte Umgebungswissen des Expertensystems wird ebenfalls in Bayes-Netzwerken dargestellt. Dabei verfügt der Experte allerdings über ein detaillierteres Modell der Umgebung und ihm stehen zusätzlich aktuelle Informationen über den Zustand der Umgebung zur

Verfügung, die er im Inferenzprozess nutzen kann. Ebenso wie bei der Modellierung der Bayes-Netzwerke zur Beschreibung der Teilfunktionslösungen kann auch bei den Expertennetzwerken eine einzelne abzufragende Variable als Senke im Graphen modelliert werden. Dadurch muss das Planungssystem die Variablen zwar einzeln abfragen, die Inferenz gestaltet sich jedoch effizienter. Es lassen sich auch hier Eingabeschicht, versteckte Schicht und Ausgabeschicht unterscheiden. Im Unterschied zu den Bayes-Netzwerken der Teilfunktionslösungen kann ein Expertennetzwerk aber Eingabeknoten enthalten, die nicht aus der probabilistischen Repräsentation des Plans stammen. Dies sind Parameter oder Variablen, die dem Expertensystem direkt bekannt sind (z. B. durch Messung) oder bei anderen Systemen erfragt werden können. Da für die Einbindung in den Plan aber letztlich nur die Knoten, die aus dem Planungssystem stammen, und Ausgabeknoten vorab definiert werden können, kann sich die übrige Struktur des Netzwerks an den Eigenschaften des jeweiligen Umfelds orientieren. Dadurch wird eine höhere Flexibilität und Situationsangemessenheit des Planungssystems erreicht.

Abbildung 4-8 zeigt die Struktur der probabilistischen Analyse, nachdem sie um Expertensysteme erweitert wurde. Im Wesentlichen hinzugekommen ist eine Kommunikation zwischen den Planungssystemen und mehreren Expertensystemen. Damit liegt eine Art von verteiltem Planungssystem vor. Eine wesentliche Frage ist, wie das Planungssystem das für eine Abfrage zuständige Expertensystem identifiziert. Der in Abschnitt 5.1.3.1.3 beschriebene Koordinationsmechanismus der Yellow Pages bietet sich zur Lösung dieses Identifikationsproblems an. Jedes Expertensystem, das im System verfügbar ist, kann sich bei den Yellow Pages anmelden und dort hinterlegen, für welche Umfeldvariable es die a-posteriori Verteilung bestimmt und welche Eingabevariablen benötigt werden. Da die Expertensysteme unter Umständen noch mit einer Vielzahl anderer Systeme kommunizieren, um möglichst genaue Informationen bereitzustellen, bietet sich ein Multiagentensystem (MAS) zur Realisierung der Expertensysteme an.

Insgesamt erlaubt die Einführung der Expertensysteme, die komplexe Aufgabe der probabilistischen Modellierung von Umfeldeinflüssen aufzuteilen und dadurch handhabbar zu machen. Weiterhin ermöglichen sie die Berücksichtigung von Umweltereignissen. Da das Bekanntwerden von neuen Ereignissen und Fakten Wirkung auf die a-posteriori-Wahrscheinlichkeitsverteilungen der Planungsvariablen hat, ist eine größere Aktualität der Planinformation gegeben, als dies bei Beschränkung auf Informationen, die zur Modellierungszeit verfügbar sind, möglich wäre.

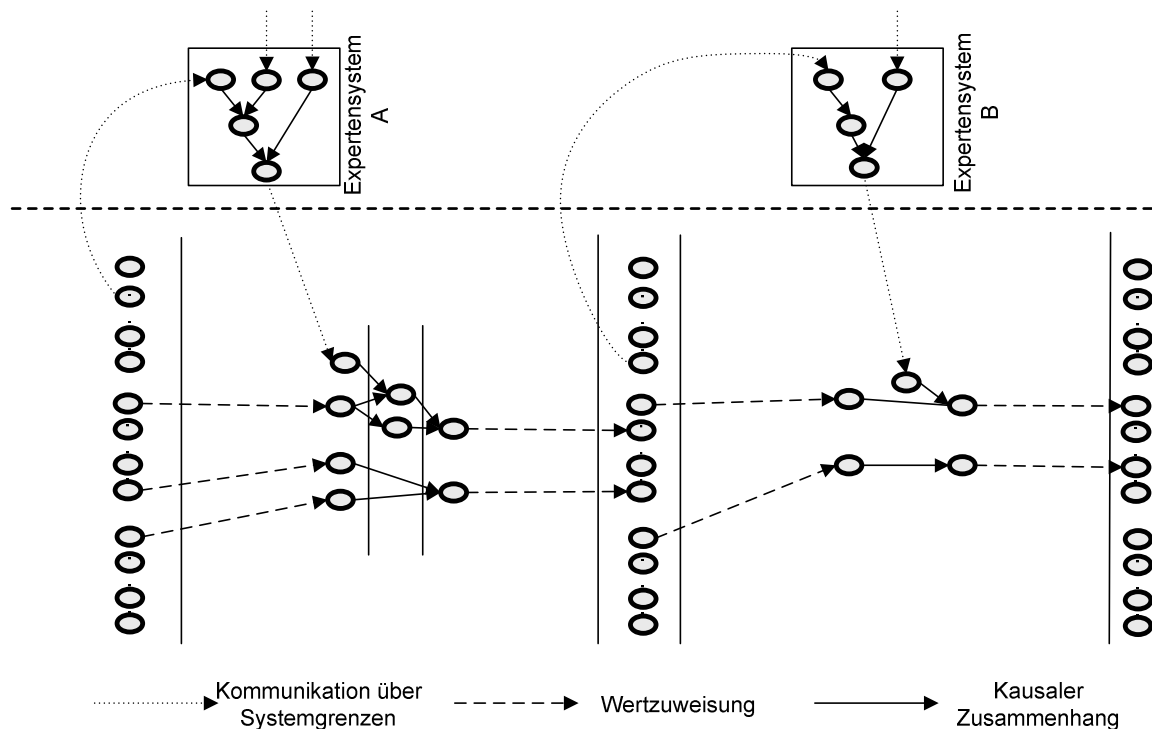


Abbildung 4-8: Erweiterte probabilistische Analyse

#### 4.2.1.4 Bestimmung von Verzweigungsstellen im Plan

Aufgrund der gewählten Modellierung ergeben sich Verzweigungsstellen nicht direkt wie bei einer endlichen Menge alternativer Effekte, wie sie üblicherweise in der bedingten Planung eingesetzt wird (z. B. in der NPDLL oder PPDDL, vergleiche Abschnitte 4.1.3.1 und 4.1.3.2). Stattdessen enthält der Plan nach der probabilistischen Analyse Mengen alternativer Effektelemente. Aus diesen Mengen müssen die Verzweigungsstellen im Plan erst abgeleitet werden. Dabei sind zwei Anforderungen zu stellen, um für eine Abweichung vom deterministischen Planverlauf und einen damit verbundenen alternativen Ausführungsverlauf eine Verzweigungsstelle zu erzeugen:

1. Der alternative Verlauf muss die Qualität oder Durchführbarkeit des bisherigen Plans (spürbar) beeinflussen.
2. Der alternative Verlauf muss eine gewisse Eintrittswahrscheinlichkeit haben.

Diese beiden Kriterien können untersucht werden, während der deterministische Plan vorwärts durchlaufen wird. In der PDDL werden die Vorbedingungen in Form einer aussagenlogischen Formel angegeben. Diese aussagenlogische Formel verknüpft mehrere einfache Ausdrücke, deren Wahrheitswert im Rahmen der deterministischen Planung als sicher bekannt unterstellt wird. Damit sind alle Belegungen der enthaltenen Variablen, die dazu führen, dass die Formel den Wahrheitswert *Falsch* annimmt, grundsätzlich relevante Ereignisse. Aus der Darstellung der Formel in Form eines Baums lassen sich, unter Beachtung der Art der Verknüpfung einzelner Zustandsparameter, die planungsrelevanten Ereignisse ableiten. Abbildung 4-9 zeigt ein Beispiel für eine solche Baumdarstellung. Wurzel und innere Knoten des Baums stellen immer eine Verknüpfung dar, die Blätter des Baums sind einfache Ausdrücke, die genau eine Zustandsvariable enthalten. Für diese lässt sich direkt bestimmen, ob sie wahr oder falsch sind. Ein Ereignis, das dazu führt, dass eine Aktivität nicht durchführbar ist, besteht mindestens aus der Negation eines solchen einfachen Ausdrucks. Aus der Struktur der Vorbedingung lässt sich zusätzlich ableiten, ob eine Verzweigung für das Auftreten eines ein-

zelen Ereignisses (gewissermaßen ein elementares Ereignis) oder einer ganzen Mengen von Ereignissen (ein zusammengesetztes Ereignis) angelegt werden muss.

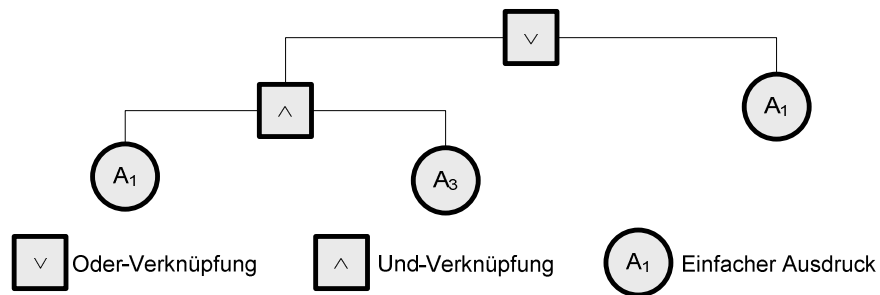


Abbildung 4-9: Darstellung einer Vorbedingung als Planungsbaum

Ein Beispiel für eine solche Vorbedingung aus der Bahntechnik ist ein Fahrzeug, das Energie aus einem Energiespeicher laden kann oder eine Stromschiene benutzt, um die Leistung ins Bordnetz zu übertragen. Das Fahrzeug kann einen Streckenabschnitt befahren, wenn der Energiespeicher ausreichend geladen ist und funktioniert oder der Streckenabschnitt bestromt ist.

Um einem Teilbaum mit einer Disjunktion als Wurzel einen Wahrheitswert *Falsch* zuzuweisen, müssen beide enthaltene Teilbäume *Falsch* zurückliefern. Im Falle einer konjunktiven Verknüpfung in der Wurzel reicht ein Teilbaum, der *Falsch* zurückliefert. Dabei lässt sich jeder Teilbaum letztlich auf eine Menge einfacher Ausdrücke zurückführen. Diese Überlegung kann genutzt werden, um beim Durchlaufen des Vorbedingungs-Baumes alle relevanten Ereignisse zu identifizieren. Um das Durchlaufen zu beschleunigen und das Ergebnis auf die relevanten Ereignisse zu beschränken, müssen die Teilbäume bereits während des Durchlaufens bzgl. ihrer Eintrittswahrscheinlichkeit bewertet werden. Ausgangspunkt der Wahrscheinlichkeitsbewertung stellen dabei die Blätter des Baums dar, deren Eintrittswahrscheinlichkeit sich direkt aus den Wahrscheinlichkeitsverteilungen der Zustandsvariablen ableiten lässt. Abbildung 4-10 zeigt, wie die Wahrscheinlichkeiten unter Beachtung der Verknüpfungsoperatoren von den Blättern bis zur Wurzel propagiert werden können.

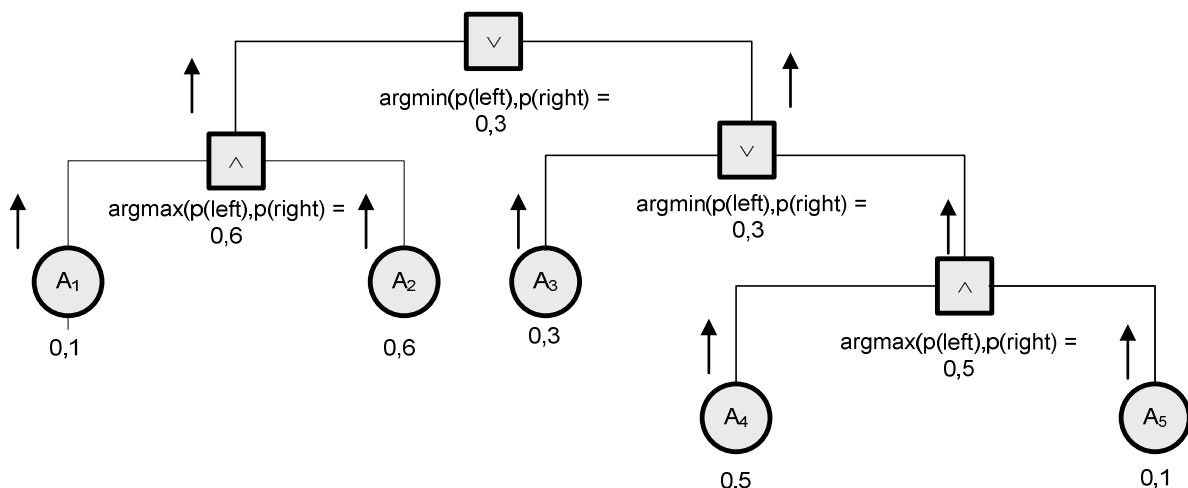


Abbildung 4-10: Propagation der Wahrscheinlichkeiten im Vorbedingungsbaum

Das folgende Listing 4-5 zeigt, wie eine solche Propagation realisiert werden kann und wie dabei zusätzlich beim Durchwandern des Baums eine Menge relevanter zusammengesetzter Ereignisse erzeugt wird. Diese Menge wird dann im weiteren Verlauf der Planung berücksichtigt. Grundlegend bei der Propagation sind dabei die Bildung der Eintrittswahrscheinlichkei-



ten und Mengen relevanter Ereignisse an den Operatorknöten. Die Wahrscheinlichkeit, dass ein Disjunktionknoten den Wert *Falsch* annimmt, ergibt sich dabei als Minimum seiner beiden Kindsknoten, die beide *Falsch* sein müssen. Im Falle eines Konjunktionknoten ist es das Maximum der beiden Kindsknoten, da es ausreicht, wenn einer *Falsch* ist.

Die relevanten Ereignisse, die an einem Knoten anliegen, werden in einem Vektor gehalten. Elemente des Ereignisvektors eines Knotens  $k$  sind dabei Mengen von Blättern, deren Belegung mit *Falsch* insgesamt dazu führt, dass der durch den Baum mit Wurzel  $k$  beschriebene Ausdruck ebenfalls falsch wird. Dabei sind Disjunktion und Konjunktion wiederum unterschiedlich zu behandeln. Im Falle der Disjunktion enthält der Ereignisvektor genau diejenigen Elemente, die sich durch paarweise Verkettung der Elemente der beiden Ereignisvektoren der Kindsknoten erzeugen lassen. Dadurch erhält man alle relevanten zusammengesetzten Ereignisse, die dafür sorgen, dass der durch die Wurzel  $k$  beschriebene Ausdruck *Falsch* wird. Im Falle der Konjunktion muss der Vektor sowohl die durch paarweise Verkettung erzeugbaren Ereignisse als auch die einzelnen Elemente der Ereignisvektoren der Kindsknoten enthalten. Die Anzahl der Elemente der Ereignisvektoren kann durch konjunktive Verknüpfungen sehr schnell steigen. Daher werden nur solche Ereignisse betrachtet, deren Eintrittswahrscheinlichkeit mit einer gewissen Schwellenwahrscheinlichkeit eintritt. Indem diese Schwellenwahrscheinlichkeit bereits beim Durchlaufen des Baumes berücksichtigt wird, muss nur eine kleinere Menge relevanter Ereignisse weiterverarbeitet werden.

**bestimmeRelevanteEreignisse**(Knoten  $k$ )

```

Wenn Typ von  $k$  = 'v' Dann
     $p(k) = \arg\_min(p(k.links), p(k.rechts))$ 
    Wenn ( $p(k) < \text{Schwellenwahrscheinlichkeit}$ )
        Dann liefere zurück ( $p(k)$ , nil)
    Sonst
        liefere zurück ( $p(k)$ ,  $k.links.events \times k.rechts.events$ )

Sonst Wenn Typ von  $k$  == '^'
     $p(k) = \arg\_max(p(k.links), p(k.rechts))$ 
    Wenn ( $p(k) < \text{Schwellenwahrscheinlichkeit}$ )
        liefere zurück ( $p(k)$ , nil)
    Sonst
        liefere zurück ( $p(k)$ ,  $(k.links.events \times k.rechts.events) \cup$ 
             $k.links.events \cup node.rechts.events$ )
Sonst // Knotentyp = "element"
    Wenn ( $p(k) < \text{Schwellenwahrscheinlichkeit}$ )
        liefere zurück ( $p(k)$ , null)
    Sonst
        liefere zurück ( $p(k)$ ,  $k.events$ )

```

**Ende**

#### Listing 4-5: Ableitung relevanter Ereignisse aus einer Aktivität

Der in Listing 4-5 beschriebene Ablauf liefert schließlich einen Vektor von zusammengesetzten Ereignissen zurück, die den Planungsablauf potenziell stören können. Diese zusammengesetzten Ereignisse müssen noch hinsichtlich ihrer Eintrittswahrscheinlichkeit beurteilt werden. Bei vollständig unabhängigen Elementarereignissen ergäbe sich die Wahrscheinlichkeit des gemeinsamen Auftretens durch Multiplikation der Einzelwahrscheinlichkeit. Im Rahmen der Planung kann aber nicht von vollständiger Unabhängigkeit ausgegangen werden – die Zustandsvariablen, die den Elementarereignissen zugrunde liegen, können durch die gleichen Einflüsse beeinflusst werden. Daher würde eine Multiplikation der Ereignisse unter Umständen zu einer Unterschätzung der Eintrittswahrscheinlichkeit führen und zusammengesetzte Ereignisse mit wenigen Elementen würden zu stark bevorzugt. Geht man zusätzlich davon aus, dass ein Alternativplan, der beim Eintreten mehrerer Elementarereignisse gültig ist, häu-

fig auch gültig sein wird, wenn einzelne Elementarereignisse nicht eintreten, ist dieser Effekt nicht wünschenswert. Daher wird zur Bewertung der Eintrittsmöglichkeit die minimale Eintrittswahrscheinlichkeit aller im zusammengesetzten Ereignis enthaltenen Elementarereignisse herangezogen. Abbildung 4-11 veranschaulicht dieses Vorgehen.

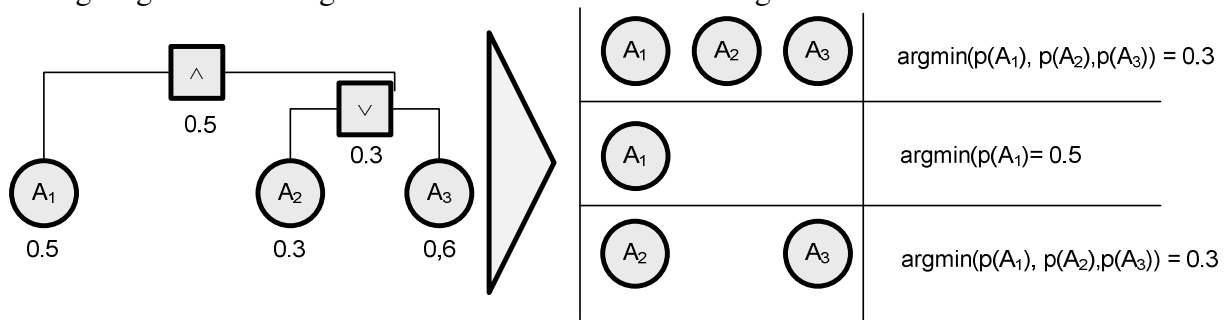


Abbildung 4-11: Bewertung zusammengesetzter Ereignisse

Indem das vorgestellte Vorgehen auf jede Aktivität im Plan angewandt wird, erhält man für jede Aktivität eine Abschätzung, wie wahrscheinlich sie nicht ausführbar wird. Zur Behandlung eines solchen Ereignisses stehen grundsätzlich zwei unterschiedliche Möglichkeiten zur Verfügung:

1. Erstellen eines Alternativplans für den Fall, dass das Ereignis eingetreten ist und die Aktion nicht mehr ausführbar ist.
2. Erstellen eines Alternativplans zur Vermeidung des Ereignisses.

Die Verzweigung liegt im ersten Fall direkt an der Aktivität, an der das Ereignis identifiziert wurde. Da zuvor die Zusammensetzung des Ereignisses bestimmt wurde, kann daraus die Bedingung abgeleitet werden, unter der ein alternativer Plan ausgeführt werden soll.

Der zweite Ansatz ist mit mehr Aufwand und zusätzlicher Analyse des Plans verbunden. Dafür reagiert er antizipativ auf mögliche Ereignisse und kann dadurch Fehler vermeiden. Dabei lässt sich die Analyse des Plans nur für numerische Zustandsvariablen sinnvoll durchführen. Dazu müssen die Aktivitäten, die zeitlich vor  $a$  liegen, dahingehend untersucht werden, ob sie eine negative Wirkung auf die numerische Variable  $v_t$  besitzen. Dazu muss für die Aktivität  $a_{t-1}$ <sup>42</sup> ein Schwellenwert bestimmt werden, ab dem der Alternativplan ausgeführt werden soll. Der erste Schritt in der Bestimmung dieses Schwellenwerts ist die Prüfung, ob  $a_{t-1}$  tatsächlich eine Gefahr für die Ausführbarkeit der Aktivität  $a_t$  darstellt. Die Vorbedingung der Aktivität  $a_t$  bezüglich der Zustandsvariablen  $v_t$  hat dabei grundsätzlich die Form:

$$x_{l,t} \leq v_t \leq x_{u,t}, v \in \mathbb{R}, x_{l,t} \in \mathbb{R} \cup \{-\infty\}, x_{u,t} \in \mathbb{R} \cup \{+\infty\}$$

Für  $x_{l,t} = x_{u,t}$  erfordert die Vorbedingung der Aktivität  $a$ , dass  $v$  genau einen Wert annimmt. Für  $x_{l,t} = -\infty$  ( $x_{u,t} = +\infty$ ) erfordert die Vorbedingung, dass  $v_t \leq x_{u,t}$  ( $v_t \geq x_{l,t}$ ) gilt. Anhand dieser Informationen lässt sich für die als letzte vor  $a$  auf  $v_t$  zugreifende Aktivität  $a_{t-1}$  bestimmen, ab welchen Werten von  $v_{t-1}$  wahrscheinlich ein Wert  $v_t$  erreicht wird, der die Vorbedingung von  $a_t$  nicht erfüllt. Betrachtet man die Struktur eines numerischen Effekts, ist derjenige *lvalue* der Zustandsvariablen  $v_{t-1}$  gesucht, bei dem sich ein neuer Wert für  $v_t$  ergibt, der nicht zwischen  $x_l$  und  $x_u$  liegt. Dabei ist zu beachten, dass der *rvalue* eine Zufallsvariable ist. Für diese Zufallsvariable sollen entsprechend zu  $x_{l,t}$  und  $x_{u,t}$  obere und untere Schranken bestimmt werden, die mit einer gewissen Wahrscheinlichkeit  $\alpha$  eintreten können. Gesucht sind also

<sup>42</sup> Obwohl der Index  $t-1$  nahelegt, dass  $a_{t-1}$  direkter Vorgänger von  $a_t$  ist, muss dies nicht der Fall sein.  $a_{t-1}$  ist die erste Aktion vor  $a$ , die auf die Zustandsvariable  $v$  zugreift.

zwei Werte, die sich aus den in der probabilistischen Analyse bestimmten Werten ableiten lassen:

- $r_{l,t-1}$  mit  $P(r_{t-1} \leq r_{l,t-1}) = \alpha$ ,  $\alpha \in [0,1]$
- $r_{u,t-1}$  mit  $P(r_{t-1} \geq r_{u,t-1}) = \alpha$ ,  $\alpha \in [0,1]$

Mit dem Parameter  $\alpha$  kann gesteuert werden, wie risikofreudig oder vorsichtig das Planungsverfahren vorgeht. Bei einem klein gewählten  $\alpha$  wird bereits sehr frühzeitig auf einen alternativen Plan umgeschwenkt, während bei großem  $\alpha$  erst bei größeren Abweichungen reagiert wird. Durch Verrechnen der Variablen  $x_{l,t}$  und  $x_{u,t}$  sowie  $r_{l,t-1}$  und  $r_{u,t-1}$  lassen sich nun direkt Werte für  $v_{t-1}$  ermitteln, ab deren Über- bzw. Unterschreiten eine Verzweigung im Plan vorzusehen ist. Dabei muss die Art des Zuordnungsoperators beachtet werden. Tabelle 4-1 zeigt, wie die Berechnung der oberen und unteren Schrankenwerte des  $v_{l,t-1}$  und  $v_{u,t-1}$  in Abhängigkeit vom Zuordnungsoperator erfolgt.

Zuordnungsoperator	$v_{l,t-1}$	$v_{u,t-1}$
assign	-	-
increase	$x_l - r_l$	$x_u - r_u$
decrease	$x_l + r_u$	$x_u + r_l$
scale-up	$\frac{x_l}{r_l}$	$\frac{x_u}{r_u}$
scale-down	$x_l \cdot r_u$	$x_u \cdot r_l$

**Tabelle 4-1: Berechnung der Schrankenwerte für antizipierende Verzweigung**

Die ermittelten Werte  $v_{l,t-1}$  und  $v_{u,t-1}$  lassen sich nun anhand der Wahrscheinlichkeitsverteilung der Zustandsvariable  $v_{t-1}$  aus der probabilistischen Zustandsanalyse bzgl. ihrer Eintrittswahrscheinlichkeit beurteilen. Die Konstruktion von möglichen Verzweigungsstellen, die die Durchführbarkeit der Aktivität  $a_t$  absichern, lässt sich durch Kenntnis der Variablen  $v_{l,t-1}$  und  $v_{u,t-1}$  fortführen, indem sie zur Berechnung der Werte  $v_{l,t-2}$  und  $v_{u,t-2}$  herangezogen werden. Der Ablauf der Berechnung ist dabei analog zur Berechnung von  $v_{l,t-1}$  und  $v_{u,t-1}$ , die an die Stellen von  $x_{l,t}$  und  $x_{u,t}$  treten. Indem dieser Ablauf iterativ für alle Aktivitäten, die vor  $a_t$  liegen und auf die Zustandsvariable  $v$  zugreifen, angewendet wird, ergeben sich zwei Zustandstrajektorien (eine für  $v_{l,i}$  und eine für  $v_{u,i}$ ), die die Aktivität  $a_t$  undurchführbar machen würden. Durch die alternativen Pläne kann dann frühzeitig reagiert werden, wenn sich eine der Zustandstrajektorien abzeichnet. Eine solche Zustandstrajektorie bestände im Beispiel des Rail-Cab-Energiemanagements vornehmlich aus den Ladeständen der Batterie. Durch das Durchlaufen des Plans werden die Zustände bestimmt, die beispielsweise mit erhöhter Wahrscheinlichkeit dazu führen, dass nicht mehr ausreichend Energie bereitsteht, um einen Streckenabschnitt in der geplanten Art und Weise zu befahren. Abbildung 4-12 veranschaulicht eine dertartige Zustandstrajektorie.

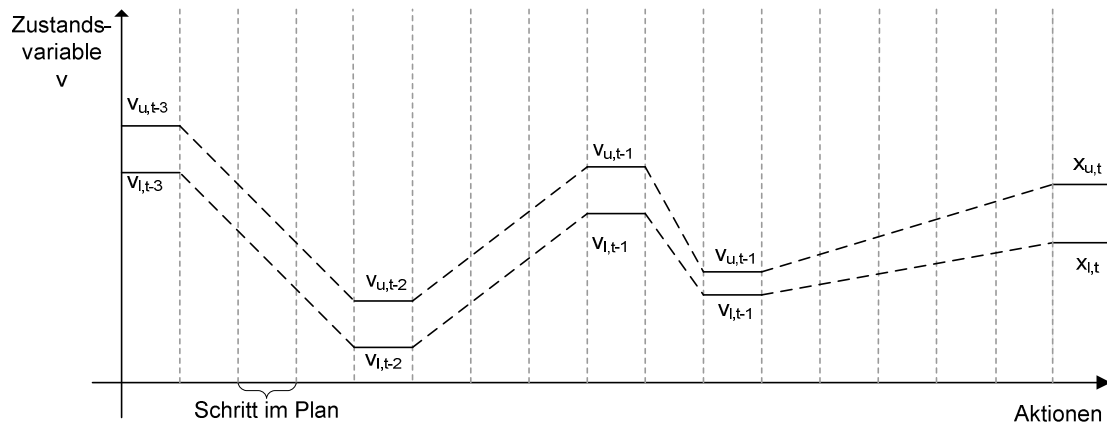


Abbildung 4-12: Zustandstrajektorie zu einer nicht ausführbaren Aktivität

Die isolierte Betrachtung einer solchen Zustandstrajektorie für eine einzige Zustandsvariable ist allerdings nicht ausreichend. Wie zuvor dargelegt, setzen sich planungsrelevante Ereignisse in der Regel aus einer Abweichung mehrerer Zustandsvariablen zusammen. Daher müssen die Zustandstrajektorien mehrerer numerischer Zustandsvariablen gleichzeitig betrachtet werden. Dabei ist zu beachten, dass für diese Menge an Zustandsvariablen nicht die gleiche Menge von Aktivitäten von Bedeutung ist. Daher muss für die gemeinsame Behandlung von der aktivitätszentrierten Betrachtung abgewichen werden, die zur Berechnung der Schwellenwerte eingesetzt wurde. Für einen numerischen Ereignisvektor  $n$ , der Ergebnis der Analyse einer Aktivität  $a_i$  ist, muss für jede Aktivität  $a_u$ , die eine der Zustandsvariablen  $v \in n$  manipuliert, ein entsprechender Vektor an Schwellenwerten betrachtet werden, die gemeinsam eine Verzweigung im Plan darstellen. Da die einzelnen Aktivitäten  $a_u$  aber nur eine Teilmenge von  $n$  manipulieren, muss zur Bestimmung der Schwellenwerte teilweise auf die (in Ausführungsrichtung) nachfolgenden Aktivitäten zurückgegriffen werden. Da diese aber bereits zuvor Gegenstand der Analyse waren, ist dieser Rückgriff unproblematisch. In der gesamten Betrachtung des Plans wird der Vektor mit den Schwellenwerten laufend aktualisiert, während der Plan rückwärts durchlaufen wird (siehe Listing 4-6).

**fügeProaktiveVerzweigungHinzu**(Plan P, Aktion a, Event E)

Verzweigungen b

**Überalle**  $a_v \in P \wedge a_v < a \wedge \arg_{\min_{e \in E}}(P(e)) > \text{Schwellenwahrscheinlichkeit}$

**Überalle**  $e \in E \wedge a$  manipuliert  $e$

Berechne  $x_{l,v}$

Berechne  $x_{u,v}$

Aktualisiere  $E[e]$

$B[a_v] = E$

**Tue**

**Tue**

**Ende**

Listing 4-6: Einfügen proaktiver Verzweigungen in einen Plan

Durch die gleichzeitige Behandlung von mehreren Zustandsvariablen, die jeweils einen oberen und unteren Schwellenwert bestimmen, stellt sich erneut das Problem, dass  $n^2$  Kombinationen bei  $n$  Zustandsvariablen berücksichtigt werden müssten. Diese hohe Anzahl wird dadurch abgemildert, dass Vorbedingungen häufig nur eine der beiden Schranken definieren werden. Ansonsten wird die Verzweigung so erzeugt, dass jeweils die wahrscheinlichere Schranke in die Verzweigung aufgenommen wird.

Um auch unterschiedliche Planqualitäten in die bedingte Planung aufnehmen zu können, müssen die in der Planmetrik enthaltenen Zustandsvariablen betrachtet werden. Ziel der Betrachtung ist es, Situationen zu ermitteln, in denen die Ausführung der aktuellen Aktivitäten vor-

aussichtlich eine negative Wirkung auf die Planmetrik hat. Die Erzeugung eines Alternativplans hat im Falle einer solchen Verzweigung das Ziel, eine Aktivität oder eine Aktivitätsfolge zu finden, die bei der gegebenen Situation ein günstigeres Verhalten bzgl. der Planmetrik hat und den gleichen Folgezustand wie die ursprüngliche Aktivität erzeugt. Um zu bestimmen, für welche Situationen eine solche Alternativplanung sinnvoll ist, werden die Bayes-Netzwerke mit Effekten auf die Parameter der betrachteten Aktivität herangezogen. Dabei wird ausgehend von dem *rvalue*, der auf den betrachteten Parameter wirkt, untersucht, unter welchen Bedingungen sich erhebliche Abweichungen vom *rvalue* der deterministischen Aktivität einstellen. Eine erhebliche Abweichung ergibt sich dabei aus dem *rvalue* und der in der probabilistischen Problemdefinition angegebenen relativen Abweichung (vergleiche Abschnitt 4.2.1.1). Zusätzlich kann anhand der Problemdefinition auch bestimmt werden, in welche Richtung eine Abweichung relevant ist. Je nach Art des *rvalue* werden die diskreten Werte oder Intervalle der kontinuierlichen Variable überprüft, wie wahrscheinlich eine Abweichung für den jeweiligen Wert (diskret) oder Bereich (kontinuierlich) ist. Dabei kann wiederum der Schwellenwert  $\alpha$  genutzt werden, um unwahrscheinliche Ereignisse schnell auszuwählen. Für die Werte, die eine Eintrittswahrscheinlichkeit größer als  $\alpha$  besitzen, wird bestimmt, unter welchen Bedingungen sie mit erhöhter Wahrscheinlichkeit auftreten. Dazu wird das Bayes-Netzwerk von der Ausgabevariablen aus durchlaufen. Anhand der jeweiligen bedingten Wahrscheinlichkeitsfunktionen kann im Rückwärtsdurchlaufen des Netzwerks ermittelt werden, welche Ausprägungen in der Eingabeebene die untersuchte Abweichung begünstigen.

Für jede identifizierte Situation, die zu einer signifikanten Verschlechterung der Ausgangssituation bzgl. eines einzelnen Parameters der Planmetrik führt, muss noch geprüft werden, ob die Situation auch insgesamt zu einer Verschlechterung führt. Dies ist notwendig, da in der gegebenen Situation andere Parameter die Verschlechterung ausgleichen könnten. Dazu wird für jeden Parameter der Planmetrik eine weitere Inferenz auf dem entsprechenden Bayes-Netzwerk durchgeführt, wobei die identifizierte Situation zu Beginn als harte Evidenz gesetzt wird. Am Ende des Inferenzprozesses wird der Erwartungswert für jeden Planungsparameter ermittelt. Auf dieser Basis wird der Erwartungswert der Planmetrik bestimmt, um festzustellen, ob sich tatsächlich eine spürbare Verschlechterung einstellen würde. Ist dies der Fall, ist eine weitere potenzielle Verzweigungsstelle identifiziert.

#### 4.2.1.5 Erstellen alternativer Pläne

Im vorhergegangenen Kapitel wurde erläutert, wie auf Basis der probabilistischen Analyse Verzweigungsstellen in einen Plan eingefügt werden können. Damit ist klar, an welchen Stellen ein Alternativplan sinnvoll angefügt werden kann. Offen ist noch, wie der Alternativplan erzeugt werden kann. Um auf die existierenden Planungsalgorithmen für die PDDL zurückgreifen zu können, soll auch das Erstellen von Alternativplänen auf ein deterministisches Planungsproblem zurückgeführt werden. Mit der bereits vorliegenden Domänenendefinition in der PDDL ist ein wesentlicher Teil dieser Zurückführung bereits geleistet. Weiterhin muss noch ein wohldefinierter deterministischer Ausgangs- und Zielzustand angegeben werden. Zusätzlich kann es für bestimmte Verzweigungen sinnvoll sein, den Alternativplan unter Verwendung einer anderen Planungsmetrik als den Originalplan zu erstellen.

Um aus einer Verzweigung einen deterministischen Ausgangszustand herzuleiten, müssen die Zufallsvariablen, die der Verzweigung zugrunde liegen, in einen skalaren Wert überführt werden. Dabei soll bei diskreten Verteilungen der Wert mit der maximalen Wahrscheinlichkeit als skalarer Wert herangezogen werden. Da bei kontinuierlichen Verteilungen ein solcher Wert nicht vorliegt, werden als skalare Werte die Grenzen  $x_l$  und  $x_u$  herangezogen. Damit kann aus dem im ursprünglichen deterministischen Plan enthaltenen Zustand ein neuer Zu-

stand erzeugt werden, der die planungsrelevanten Abweichungen enthält. Dabei werden die entsprechenden Variablen im Zustandsvektor des deterministischen Plans neu belegt.

Zur Generierung eines deterministischen Zielzustands kann zunächst das ursprüngliche Planungsziel herangezogen werden. Alternativ kann ein bereits im Plan enthaltener Zustand verwendet werden, um ausgehend von diesem den ursprünglichen Plan weiter zu verfolgen. Dies kann als eine Art *Planreparatur* aufgefasst werden. Im Falle der Planreparatur muss abgewogen werden zwischen dem Aufwand der Auswahl eines geeigneten Zielzustands und der Planungsdauer, die durch die Reparatur gegenüber einer Neuplanung eingespart wird. Dabei ist die Auswahl eines geeigneten Zustands hinsichtlich der Komplexität und des Aufwands ein nicht zu unterschätzendes Problem. Zur Lösung muss der Abstand zwischen zwei Zuständen bestimmt oder zumindest abgeschätzt werden. Derartige Abschätzungsfunktionen (Heuristiken) werden auch in deterministischen Planern genutzt, um die Planungsgeschwindigkeit zu erhöhen. In der Regel ist dafür die Lösung eines vereinfachten Planungsproblems notwendig. Indem beim neuen Planaufruf die Disjunktion aller im Plan enthaltenen Zustände übergeben wird, kann die heuristische Schätzung des Standardplaners genutzt werden. Dies entspricht dem Vorgehen, das das Planungssystem PLANEX 1 (vergleiche 4.1.4.3) zur Neuplanung nach einer Störung verwendet. Die Entscheidung, ob nur der originale Zielzustand oder eine Konjunktion aller Zustände dem Planer übergeben wird, hängt von der Implementierung des verwendeten Planers ab. Die Anzahl der Zustände im Plan, die dem Planer übergeben werden, kann dabei ein wesentlicher Entscheidungsparameter sein.

Für den Fall, dass ein Alternativplan für eine proaktive Verzweigungsstelle erzeugt werden soll, kann als Zielzustand der deterministische Vorgängerzustand der Aktivität verwendet werden, die der ursprüngliche Auslöser der Verzweigungsstelle ist. Für diesen Fall bietet sich unter Umständen auch der Einsatz einer modifizierten Planmetrik an. Für jede numerische Variable, die an der Verletzung der Vorbedingung beteiligt ist und deren Verletzung nicht sowohl durch Über- als auch Unterschreitung einer Schranke verursacht wird, kann die Planmetrik einfach durch Minimierung (es liegt nur eine obere Schranke vor) oder Maximierung (es liegt nur eine untere Schranke vor) der entsprechenden Variablen genutzt werden, um einen möglichst robusten Plan zu erhalten. Robust meint dabei, dass der Plan eine hohe Erfolgswahrscheinlichkeit besitzt und hohe Abweichungen von der Vorbedingung sehr unwahrscheinlich sind. Bestehende Metriken können durch Bildung einer gewichteten Summe berücksichtigt werden.

#### **4.2.1.6 Verzahnung mit der Planausführung**

Wie in Abbildung 4-1 dargestellt erfolgt die Auswahl von Verzweigungsstellen und die Erstellung alternativer Pläne verzahnt mit der Planausführung. Dieses Vorgehen hat zwei unmittelbare Vorteile: Mit der Planausführung kann frühzeitiger begonnen werden, wodurch die Zeit bis zum Erfüllen der Aufgabe tendenziell verkürzt wird. Die während der Ausführung gewonnene Information kann außerdem zur besseren Bestimmung und Auswahl relevanter Verzweigungsstellen im Plan genutzt werden.

Dazu sind mit fortschreitender Ausführung die folgenden Schritte iterativ durchzuführen:

1. Aktualisierung der Wahrscheinlichkeiten im Plan,
2. Bestimmung neuer Verzweigungsstellen,
3. Auswahl der nächsten zu berücksichtigenden Verzweigungsstelle und
4. Erstellen eines alternativen Plans für die ausgewählte Verzweigungsstelle.

Für die Aktualisierung der Pläne steht mit dem in 4.2.1.2 vorgestellten Verfahren zur Analyse von Plänen bereits eine Methode zur Verfügung. Im Unterschied zum dort beschriebenen Verfahren wird allerdings nicht der Initialzustand des Plans als Ausgangszustand gewählt, sondern der Zustand, der sich nach der bisherigen Planausführung eingestellt hat. Für diesen Zu-

stand liegen dann auch harte Evidenzen vor, so weit die Ausprägungen der Zufallsvariablen durch Sensoren oder andere Informationsquellen wahrnehmbar sind. Diese harten Evidenzen können genutzt werden, um ausgehend vom aktuellen Zustand eine genauere Verteilung der noch folgenden Zustandsmengen zu bestimmen. Auf den so aktualisierten Plan kann für den zweiten Schritt wiederum das in 4.2.1.4 beschriebene Verfahren zur Bestimmung von Verzweigungen eingesetzt werden.

Die Auswahl der nächsten zu berücksichtigenden Verzweigungsstellen muss vor dem Hintergrund der Verzahnung der Ausführung und dem Formalziel, die Aufrufe der Neuplanung zu minimieren, unter Berücksichtigung zweier Größen getroffen werden. Das sind zunächst die Eintrittswahrscheinlichkeit einer Verzweigungsbedingung und andererseits die Ausführungsdauer, die bis zum Erreichen dieser Verzweigung verstreichen wird. Aufgrund der Verzahnung mit der Planausführung wird Planungszeit zu einer kritischen Ressource. Um möglichst für jede eintretende Situation einen geeigneten Plan verfügbar zu haben, ist es sinnvoll die zur Verfügung stehende Planungszeit zur Erstellung des zeitlich nächsten Verzweigungspunkts einzusetzen. Demgegenüber steht die Überlegung, die Planungszeit nicht für solche Situationen einzusetzen, die wahrscheinlich gar nicht eintreten. Eine Möglichkeit, mit diesen potenziell widersprüchlichen Auswahlkriterien umzugehen, ist die Vereinigung in einer einzigen Kennzahl. Eine mögliche Klasse solcher Kennzahlen, die im Rahmen dieser Arbeit eingesetzt werden soll, ist durch die folgende Formel beschrieben:

$$\text{Selectability}(B) = P(B) \cdot \text{Zeitfaktor}(B).$$

Diese Formel beschreibt mögliche Kennzahlen, bei denen die Eintrittswahrscheinlichkeit einer Planverzweigung B mit einem Zeitfaktor gewichtet wird. Der Zeitfaktor ist dabei eine Funktion, die den zeitlichen Abstand des Ausführungshorizonts bis zur Verzweigung B auf einen Bereich zwischen 0 und 1 normiert. Die Form der Normierung ist dabei im Prinzip unerheblich, steuert aber, wie stark die zeitliche Nähe eines Verzweigungspunkts in die Kennzahl eingeht. Bei der Wahl einer beispielsweise exponentiellen Normierung des Zeitfaktors werden nur solche Verzweigungen ausgewählt, die in unmittelbarer zeitlicher Nähe des Ausführungshorizonts liegen.

Eine effiziente Verwaltung der Verzweigungspunkte ist von großer Bedeutung, um nicht unnötig Planungszeit auf die Auswahl der nächsten zu expandierenden Verzweigung zu verwenden. Für diese Aufgabe bietet sich die Datenstruktur *Prioritätenliste* (engl. Priority Queue, siehe zum Beispiel [CLRS01]). Diese Datenstruktur kann speicher- und laufzeiteffizient eine sortierte Liste der möglichen Verzweigungspunkte vorhalten. Das Sortierkriterium ist dabei die Wählbarkeit der Verzweigungspunkte. Die Prioritätenliste erlaubt die effiziente Änderung, Löschung und Einfügung eines Eintrags, wobei die korrekte Sortierung der Liste immer gewährleistet ist.

Abbildung 4-13 stellt den Gesamtablauf des bedingten Planungsverfahrens bei Verzahnung mit der Planausführung dar. Die Darstellung veranschaulicht dabei den Ablauf der Planung anhand der unterschiedlichen Planversionen, die erstellt werden. Die mit der Ausführung verzahnte bedingte Planung wird in zwei nebenläufigen Prozessen ausgeführt. Der erste und aus Sicht der Planung einfachere Prozess ist die Ausführungsüberwachung.<sup>43</sup> In diesem Prozess wird der aktuelle Zustand der Planausführung bestimmt und dem Planungssystem zur Verfügung gestellt.

Ausgangspunkt für den zweiten, eigentlichen Planungsprozess ist der Aufruf eines deterministischen Planers. Dieser erstellt für deterministische Ausgangs- und Zielzustände einen deterministischen Plan. Dieser Plan wird in den bisherigen Plan integriert, indem er mit der ent-

---

<sup>43</sup> Die Realisierung der Ausführungsüberwachung ist nicht Gegenstand dieser Arbeit.

sprechenden Verzweigungsstelle und dem ggf. im ursprünglichen Plan vorhandenen Zielzustand verknüpft wird. Im nächsten Schritt wird der neue Plan an den aktuellen Zustand angepasst, der durch die Ausführungsüberwachung bereitgestellt wird. Dabei wird der aktuelle Zustand im bedingten Plan gesucht und – so weit über Sensoren verfügbar – harte Evidenzen bzgl. der Zustandsparameter werden gesetzt. Alle Zustände, Verzweigungen und Alternativpläne, die vor dem aktuellen Zustand liegen, werden entfernt. Auf dem so aktualisierten und verkleinerten Plan wird dann die probabilistische Analyse für jeden Alternativplan durchgeführt (wie in Abschnitt 4.2.1.6. beschrieben). Dabei werden auch aktualisierte Verteilungen bei den Expertenagenten abgefragt, wodurch externe Ereignisse, die Einfluss auf den Planverlauf haben können, berücksichtigt werden. Auf Basis der neuen Wahrscheinlichkeitsverteilung werden nun wie in Abschnitt 4.2.1.4 beschrieben neue Verzweigungsstellen erzeugt und unwahrscheinlich gewordene Verzweigungsstellen entfernt. In diesem Prozess wird auch unmittelbar die Selectability der Verzweigungsstellen ermittelt. Dabei kommt die Stärke der Prioritätenliste zum Tragen: Anstatt die Datenstruktur von Grund auf neu aufbauen zu müssen, können die neu hinzugekommenen Verzweigungsstellen hinzugefügt, entfallene gelöscht und solche, bei denen sich die Selectability deutlich geändert hat, aktualisiert werden. Damit ist der letzte Schritt, die Auswahl der nächsten einzuplanenden Verzweigungsstelle, nur noch das Entnehmen des maximalen Elements aus der Prioritätenliste. Nachdem aus dieser Verzweigung wie in 4.2.1.5 beschrieben jeweils ein neuer Ausgangs- und Zielzustand erzeugt wurde, kann der deterministische Planer erneut aufgerufen werden.

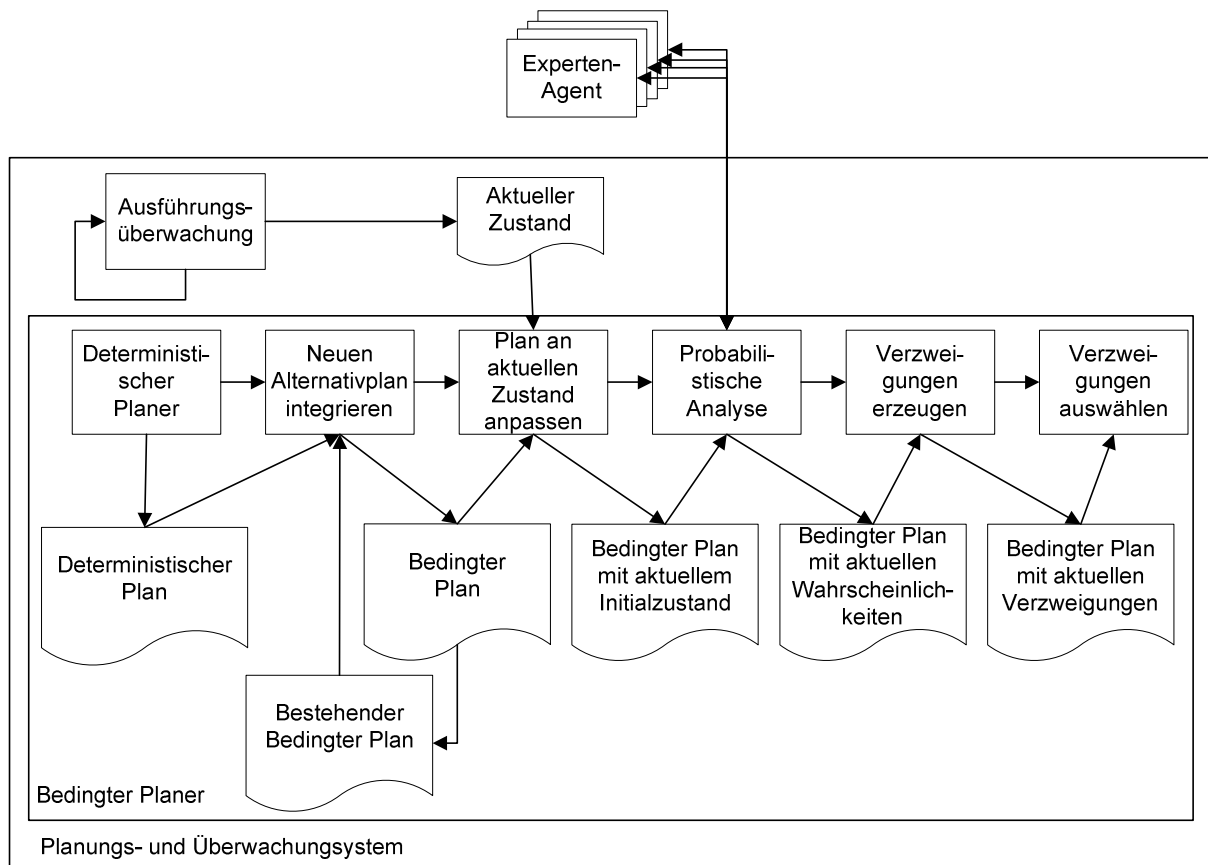


Abbildung 4-13: Gesamtablauf der bedingten Planung

## 4.2.2 Umplanungsverfahren auf Basis von Echtzeitsuche

Wie bereits eingangs Kapitels 4.2 erläutert, kann auch die probabilistische Analyse des Plans nicht sicherstellen, dass alle Eventualitäten berücksichtigt werden können. Daher ist ein Online-Planungsverfahren als Rückfallebene zu integrieren. Das Problem, ein Online-



Planungsverfahren auf Basis der PDDL 2.1 zu entwickeln, ist bereits in [SO02] gelöst worden (siehe Abschnitt 4.1.4.3). Daher ist im Rahmen dieser Arbeit konzeptionell nur noch die Frage nach der Einbindung an das Gesamtplanungsverfahren für Unsicherheit zu beantworten.

Den Anbindungspunkt stellt die in Abbildung 4-13 dargestellte Aktivität „*Plan an aktuellen Zustand anpassen*“ dar. Wird an dieser Stelle festgestellt, dass der aktuelle Zustand nicht im aktuellen Plan enthalten ist, wird das Online-Planungsverfahren aufgerufen. Für die Auswahl eines Zielzustands gelten beim Online-Verfahren die gleichen Aussagen wie bereits bei der Auswahl eines Zielzustands (siehe Abschnitt 4.2.1.5). Im Falle des Umplanungsverfahrens ist das Ziel, zu einen bereits im Plan vorkommenden Zustand wieder zurückzukehren, von besonderer Bedeutung, da in diesem Fall der bedingte Plan weiter verwendet und das bedingte Planungsverfahren wieder aufgenommen werden kann. Die Rückkehr zum bedingten Planungsverfahren ist wünschenswert, da es im Gegensatz zur Onlinevariante vollständig und optimal ist. Daher soll in der Regel eine Konjunktion der im Plan enthaltenen Zustände als Ziel übergeben werden.



## 5 Planung von Systeminteraktionen

Mit den bisher vorgestellten Planungsmodellen und Verfahren werden mechatronische Systeme noch nicht in die Lage versetzt, Interaktion mit anderen Systemen in ihre Planung einzubeziehen. Dieser Baustein für eine Verhaltensplanung für mechatronische Systeme wird in diesem Kapitel erarbeitet. Wie bereits in Unterkapitel 2.4 hergeleitet, sind dabei zwei sehr unterschiedliche Anwendungsszenarien zu behandeln. Im ersten Anwendungsszenario haben die Systeme gemeinsam eine Hauptaufgabe zu erfüllen und besitzen damit grundsätzlich eine gemeinsame Zielfunktion, an der sie ihre Handlungen ausrichten. Bei individuellen Hauptaufgaben besitzt jedes System eine eigene Zielfunktion, die es zu optimieren versucht. Weiterhin sind in diesem Fall die interagierenden Systeme nicht vorab bekannt. Am Beispiel werden die Unterschiede zwischen diesen beiden Planungsaufgaben deutlich.

Als Beispiel der Interaktionsplanung bei gemeinsamer Hauptaufgabe soll die Abstimmung zwischen den Modulen *Antrieb* und *AGAS* des RailCab-Fahrzeugs dienen. Beide Module werden im RailCab eingesetzt, dienen also der Hauptaufgabe, Personen oder Güter in einem Schienennetzwerk zu transportieren.

Das Antriebsmodul steuert dabei das Fahrprofil des RailCabs, also mit welchem Geschwindigkeitsverlauf ein Streckenabschnitt befahren wird. Aufgabe des AGAS ist die aktive Anpassung des Luftspalts, also des Abstands zwischen dem fahrzeugseitigen (Läufer) und streckenseitigen (Stator) Motorteils. Um Abweichungen in der Statorlage auszugleichen (ein Stator liegt höher als der Nächste) ist das AGAS in der Lage, den sekundären Motorteil vertikal zu bewegen. Aus der Größe des Luftspalts resultiert direkt die Energieeffizienz des Fahrzeugs. Für die gleiche Antriebskraft und in das RailCab übertragene Energie muss mit wachsendem Luftspalt erheblich mehr Energie streckenseitig aufgewandt werden.

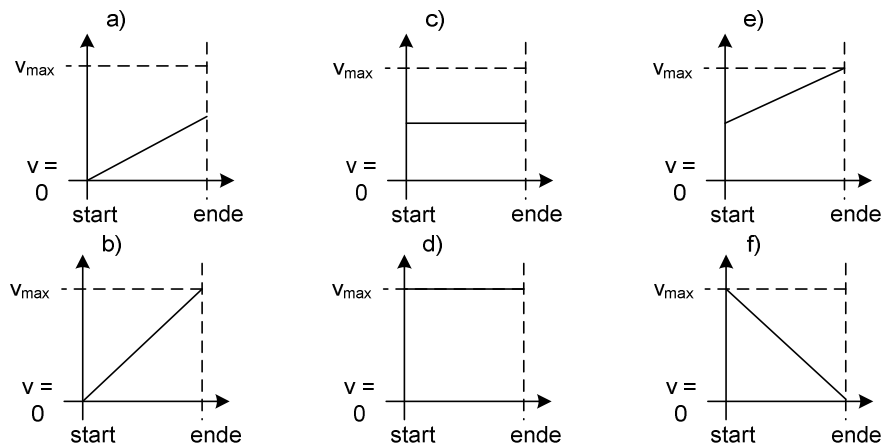
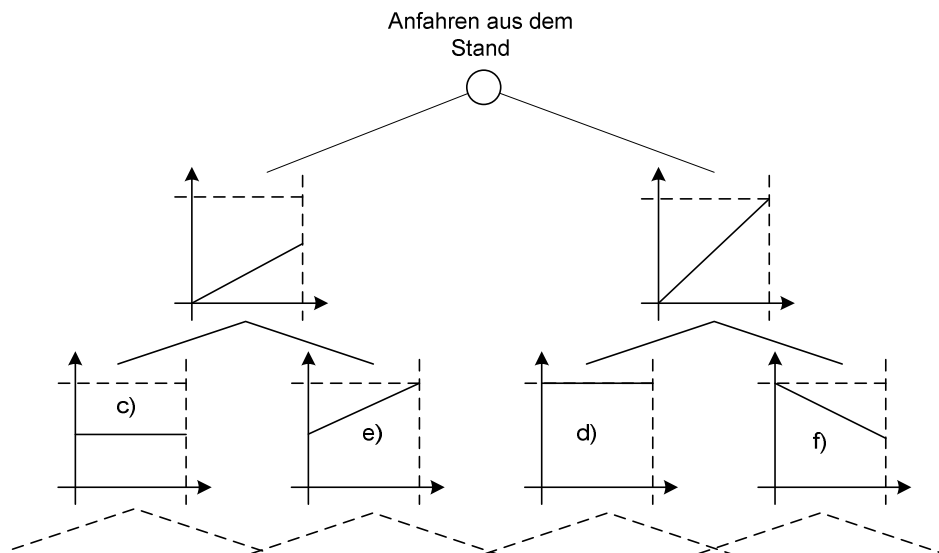


Abbildung 5-1: Betrachtete Modi des Antriebs

Im Falle des Antriebs sind die Teilfunktionslösungen gegeben durch ein bestimmtes Fahrprofil, also einen Geschwindigkeitsverlauf, dem über einen bestimmten Streckenabschnitt gefolgt wird. Abbildung 5-1 zeigt die im Beispiel betrachteten Teilfunktionslösungen des Antriebs. Es handelt sich um Geschwindigkeitsverläufe, die über den gesamten Streckenabschnitt eine gleichmäßige Beschleunigung besitzen. Dabei werden z. B. zwei unterschiedliche Geschwindigkeitsstufen betrachtet: die Höchstgeschwindigkeit  $v_{max}$  (180 km/h) und  $\frac{v_{max}}{2}$  (also 90 km/h). Nimmt man noch das Anfahren aus dem Stand hinzu, ergeben sich somit insgesamt 6 Teilfunktionslösungen.

Es besteht eine Vorgänger-Nachfolger-Beziehung zwischen den Modi, die sich aus den Anfangs- und Endgeschwindigkeiten der jeweiligen Fahrprofile ergibt. Eine Teilfunktion  $tfl'$  kann nur auf eine Teilfunktion  $tfl$  folgen, wenn die Anfangsgeschwindigkeit von  $tfl'$  mit der Endgeschwindigkeit von  $tfl$  übereinstimmt. Dadurch ergibt sich für die Planung des Antriebs das Problem, für eine abzufahrende Sequenz von Streckenabschnitten die optimale Abfolge von Teilfunktionslösungen zu bestimmen. Dabei entspricht das Problem aufgrund der Vorgänger-Nachfolger-Beziehung der Suche nach dem besten Pfad zu einem Knoten (in diesem Fall dem Knoten in der letzten Ebene des Baums mit der höchsten Summe der Gütefunktion über sich und alle seine Vorgänger) in einen Baum (Abbildung 5-2 veranschaulicht dies).



**Abbildung 5-2: Struktur des Planungsproblems für den Antrieb**

Auch das AGAS unterstützt unterschiedliche Teilfunktionslösungen. Diese orientieren sich an der Verstellaktivität, mit der der Luftspalt auf die Streckegegebenheiten angepasst wird. Das Beispiel beschränkt sich auf die beiden Teilfunktionslösungen „einmalige Verstellung“ und „kontinuierliche Verstellung“. Bei der einmaligen Verstellung wird der Sekundärteil des Motors einmalig auf eine Höhe eingestellt, die für den gesamten Streckenabschnitt beibehalten wird. Bei der kontinuierlichen Verstellung wird der Luftspalt über den gesamten Streckenabschnitt im Rahmen der technischen Möglichkeiten laufend angepasst, sodass besser auf unterschiedliche Statorlagen auf dem Streckenabschnitt reagiert werden kann. Das Beispiel umfasst zwei unterschiedliche Typen von Streckenabschnitten: Streckenabschnitte mit exakt verlegten Statorn und Streckenabschnitte mit ungenau verlegten Statorn.

Die Fähigkeit des AGAS, einen kleinen Luftspalt einzustellen und damit den Wirkungsgrad zu optimieren, hängt dabei direkt vom Fahrprofil ab, das der Antrieb verwendet. Die Aktorik des AGAS kann nur beschränkt Kraft aufwenden, um den sekundären Motorteil zu bewegen. Damit kann das AGAS in einer gegebenen Zeit nur einen beschränkten Weg stellen. Folglich hängt der einstellbare Luftspalt unmittelbar von der Geschwindigkeit des RailCabs ab. Da bei hoher Geschwindigkeit weniger Zeit zwischen den einzelnen Statorn bleibt, kann nur ein kleinerer Weg zwischen zwei Statorn zurückgelegt werden. Damit ist bei hohen Geschwindigkeiten tendenziell ein größerer Luftspalt einzustellen. Abbildung 5-3 veranschaulicht diesen Zusammenhang.

Bezieht man noch die für die Verstellung benötigte Energie mit ein, verstärkt sich dieser Zusammenhang weiter. Vereinfacht lässt sich sagen, dass ein Fahrprofil des Antriebs mit hohen Geschwindigkeiten dazu führt, dass das AGAS seine lokalen Planungsziele (Luftspalt) schlechter erreicht. Dabei hängt der einstellbare Luftspalt von Umwelteinflüssen ab. Ist der

Streckenabschnitt beispielsweise unbekannt, kann ohnehin nur ein großer Luftspalt eingestellt werden. Auch Witterungseinflüsse wie etwa vereiste Statoren beeinflussen die möglichen Aktivitäten der Luftspaltverstellung.

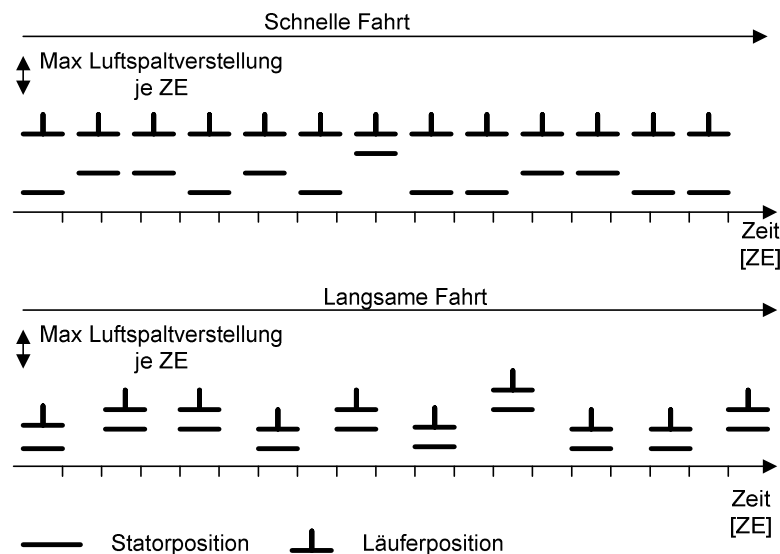


Abbildung 5-3: Wechselwirkung zwischen Antrieb und Luftspaltverstellung

Ein Beispiel für kooperierende Systeme mit individueller Hauptaufgabe sind die RailCabs. RailCabs, die Güter oder Passagiere im Auftrag unterschiedlicher Transportdienstleister befördern, sind ein Beispiel für autonome mechatronische Systeme mit individueller Hauptaufgabe. Sie verfolgen lokale Ziele (z. B. die Maximierung der Erträge ihrer jeweiligen Betreiber), die möglicherweise im Konflikt zueinander stehen.

Eine Besonderheit des RailCab-Konzepts ist die Möglichkeit, mehrere Fahrzeuge in einer Gruppe (im Konvoi) fahren zu lassen. Eine mögliche Form der Kooperation besteht also darin, im Rahmen der Planung bewusst die Routen einer Gruppe von Fahrzeugen so anzupassen, dass ein Konvoi entsteht. Dazu ist in einem dezentral gesteuerten System (die lokalen Zielsetzungen machen die Anwendung eines zentralen Systems unplausibel) zunächst die Erkenntnis eines Fahrzeugs notwendig, dass die Bildung eines Konvois für einen Streckenabschnitt eine Verbesserung der Planqualität bietet. Dies kann erkannt werden anhand von langen Teilrouten, die alleine befahren werden. Im nächsten Schritt müssen solche Fahrzeuge identifiziert werden, deren bisherige Routen gut zum Kooperationsvorhaben passen. Das bedeutet, die Fahrzeuge müssen sich in einem Zeitfenster in der Nähe der Teilroute aufhalten und ähnliche Fahrziele besitzen. Schließlich muss eine Einigung über das Zustandekommen des Konvois erfolgen, die unbedingt notwendigen Haltepunkte der Konvoiroute festgelegt werden und gemeinsam eine neue Route gefunden werden.

Die folgenden Kapitel untersuchen zunächst den bestehenden Stand der Technik aus dem Bereich der Multibody- und Multiagenten-Planung. In beiden Fällen wird versucht, abgestimmte Pläne für mehrere Individuen zu erzeugen. Im Falle der Multibody-Planung geschieht dies eher zentral, im Fall der Multiagenten-Planung eher dezentral. Die identifizierten Ansätze sollen auf ihre Eignung zur Lösung eines der beiden Planungsprobleme untersucht werden. Anschließend werden zwei Planungskonzepte für die beiden Anwendungsszenarien erarbeitet, die die Interaktion zwischen mechatronischen Systemen in die Verhaltensplanung mit PDDL integrieren.

## 5.1 Stand der Technik zur Planung von Systeminteraktionen

Zu den Annahmen der klassischen Handlungsplanung in der Künstlichen Intelligenz gehört auch, dass das geplante System die einzige Quelle von Veränderungen ist (siehe Teilkapitel 2.1). Diese Annahme ist nicht nur für die Verhaltensplanung von mechatronischen Systemen problematisch, sondern für eine Vielzahl von Anwendungsfeldern. Planung unter Berücksichtigung mehrerer aktiver Systeme, die aufeinander abgestimmt werden können, ist Gegenstand der Multiagenten- oder Multibody-Planung.

In den folgenden Kapiteln werden zunächst die Modellierung derartiger Planungsprobleme und zwei Phasenmodelle für die Multiagenten-Planung auf ihre Eignung für den Einsatz in der Verhaltensplanung für mechatronische Systeme untersucht.

Schließlich werden konkrete Verfahren, die sich den Phasen der vorgestellten Phasenmodelle zuordnen lassen, auf ihren Einsatz in der Interaktionsplanung für mechatronische Systeme hin untersucht.

### 5.1.1 Modellierung der Planung von Systeminteraktionen

Die folgenden Abschnitte beschreiben zunächst mögliche Modellierungstechniken für die Darstellung von Multiagenten-Planungsproblemen und untersuchen sie auf ihre Einsatzmöglichkeit im Rahmen der Verhaltensplanung für mechatronische Systeme.

#### 5.1.1.1 Multiagent Planning Language (MAPL)

Die Multiagent Planning Language (MAPL) [Bren03] ist eine Planungssprache, die die PDDL 2.1 [FL03] erweitert und auch partiell geordnete Pläne darstellen kann. Diese Sprache nutzt ein auf Simple Temporal Networks [DMP91] basierendes Modell, um qualitative und quantitative zeitliche Beschränkungen zwischen Aktionen zu beschreiben. MAPL ist also in der Lage, sowohl eine kausale zeitliche Ordnung zwischen zwei Aktionen ( $A_i$  muss vor  $A_j$  ausgeführt werden) als auch quantifizierte zeitliche Abstände (10 Minuten später) darzustellen. MAPL beschreibt Zustände mittels propositionaler und nicht boolescher Zustandsvariablen.

Ein wesentliches Konzept im Rahmen von MAPL ist die sogenannte *Kontrolle*. Jede Aktion in einem MAPL-Plan wird begrenzt durch ein Eingangs- und ein Endereignis. Der Zeitpunkt des Eingangsereignisses einer Aktion wird immer durch den ausführenden Agenten festgelegt, der Zeitpunkt des Endereignisses kann aber auch durch das Umfeld bestimmt werden. Damit werden unterschiedliche Kontrollgrade in Bezug auf Aktionen, Unwissenheit der Agenten und Planungssynchronisationen zwischen ausführenden Agenten darstellbar.<sup>44</sup>

Da in der Regel Koordination und Kommunikation für eine abgestimmte Ausführung eines Plans erforderlich sind, unterstützt MAPL dies durch Bezugs- und Kommunikationsereignisse. Diese Art von Ereignissen kann als Vorbedingung zur Ausführung von Aktionen genutzt werden.

MAPL besitzt eine Vielzahl von Eigenschaften, die für die Planung von Systeminteraktionen sehr nützlich sind. Dazu gehören nebenläufige und gleichzeitige Aktionen, ein ausgefeiltes Zeitkonzept und die Berücksichtigung von Synchronisationsmaßnahmen. Der entscheidende Nachteil ist aber ein Mangel an Planungsverfahren, die auf Basis einer in MAPL formulierten Planungsaufgabe einen Plan erzeugen. Trotz intensiver Recherchen konnte kein einziges MAPL-Planungssystem gefunden werden. PDDL-Standardplaner kommen zur Lösung von MAPL-Planungsproblemen keinesfalls in Betracht.

<sup>44</sup> Brenner unterscheidet zwischen Systemen, die a) Planung durchführen (Planungsagenten), die b) Pläne ausführen (ausführende Agenten) und Systeme, die c) zur Umgebung gehören.

### 5.1.1.2 Aktions-Ressourcen-Formalismus (ARF)

Der Aktions-Ressourcen-Formalismus [VV05] weicht von den bisher vorgestellten Modellierungsmethoden für Planungsprobleme ab. Basis dieser Modellierung ist eine andere Sichtweise auf die Planung. In ARF stehen nicht mehr Zustände und Aktionen im Fokus, sondern Ressourcen.

In einem Planungsframework auf Basis des ARF [WMW05] wird der Initialzustand durch eine Menge von atomaren Ressourcendaten beschrieben. Der Zielzustand wird dargestellt als Ressourcendatum oder mehreren –daten, deren Attribute teilweise mit Variablen belegt sind. Ein Ziel ist erreicht, wenn konkrete Ressourcendaten erzeugt wurden, die eine gültige Instanz des Ziels sind.

Den Aktionenschemata der klassischen Planungssprachen entsprechen im ARF sogenannte Aktionsmodelle, die mögliche Transformationen bzw. Änderungen der Ressourcendaten darstellen. Abhängigkeiten und Reihenfolgebedingungen zwischen Aktionen werden in ARF indirekt über die Ressourcen dargestellt. Dabei kann zu jeder Eingaberessource einer Aktion eine andere Aktion oder der Initialzustand als Erzeuger bestimmt werden.

Im Framework von de Weerdts wird der ARF nicht direkt zur Planung genutzt, sondern um Synergien zwischen den Plänen autonom planender Agenten zu identifizieren und zu realisieren.

Ein wesentlicher Nachteil des ARF ist, dass kein Algorithmus zur Konstruktion von Plänen, sondern nur ein verbesserndes Verfahren verfügbar ist. Ein konstruierendes Verfahren müsste erst konzipiert werden. Darüber hinaus erscheint eine Kombination des ARF mit der PDDL umständlich, da dafür der Aktionsformalismus der PDDL erst auf die Ressourcendarstellung des ARF abgebildet werden müsste.

## 5.1.2 Phasenmodelle zur Planung von Systeminteraktionen

Die Planung von Systeminteraktionen ist ein komplexes Problem, das zu seiner Lösung strukturiert werden muss. Die folgenden beiden Abschnitte stellen zwei bekannte Ansätze vor, die das Problem der Multiagenten-Planung in mehrere Phasen unterteilen, und damit die notwendigen Handlungsschritte strukturieren.

### 5.1.2.1 Phasen-Modell in Anlehnung an Durfee

In Anlehnung an ein Konzept zur Aufgabenverteilung von Durfee [Durf01] unterscheiden de Weerdts et. al. [WMW05] die folgenden Phasen in einem Multiagenten-Planungsproblem:

- 1. Verfeinerung der globalen Aufgaben:**  
Aufteilung der globalen Ziele, bis Teilaufgaben gefunden sind, die einem einzelnen Agenten zugeordnet werden können.
- 2. Aufgabenzuordnung:**  
Zuordnung von Mengen von Teilaufgaben zu einzelnen Agenten.
- 3. Koordination vor der Planung:**  
Definition von Regeln oder Beschränkungen für einzelne Agenten, die verhindern, dass Konflikte in den individuellen Plänen auftreten.
- 4. Individuelle Planung:**  
Jeder Agent erzeugt einen individuellen Plan zur Erfüllung seiner Aufgaben unter Berücksichtigung der Regeln und Beschränkungen.
- 5. Koordination nach der Planung:**  
Koordination der individuellen Pläne.
- 6. Planausführung:**  
Ausführung der Pläne und Synthese der Ergebnisse der Teilaufgaben.

Nicht jede Anwendung von Multiagenten-Planung umfasst notwendigerweise alle 6 Phasen des allgemeinen Prozesses. Wenn beispielsweise keine globalen Ziele oder Aufgaben existieren, sind die Phasen 1 und 2 nicht notwendig. Die Phasen 3 und 5 stehen in Konkurrenz zueinander, sodass eine Koordination auch jeweils ausschließlich vor oder nach der Planung stattfinden kann. Ebenso kann die Abgrenzung zwischen den Phasen 5 und 6 fließend sein. Für die Umsetzung jeder dieser Phasen gibt es eine Reihe von Ansätzen (siehe das nächste Kapitel).

Das in [WMW05] vorgestellte Phasenmodell lässt den für die Planung bei individueller Hauptaufgabe wesentlichen Aspekt außer Acht, wie sich kooperierende Agenten finden können. Daher kommt es für Systeme mit individueller Hauptaufgabe nicht in Betracht, da diese Systeme sich in der Regel nicht vorab kennen. Für die Interaktionsplanung bei gemeinsamer Hauptaufgabe scheinen die Phasen der *Koordination vor der Planung* und der *Koordination nach der Planung* aber besonders interessant. Durch ein derartiges Vorgehen können die lokalen Planungsprobleme erhalten bleiben, was sich wie in Teilkapitel 2.4 beschrieben positiv auf die Komplexität des Gesamtproblems auswirkt.

### 5.1.2.2 Phasenmodell nach Wooldrigde und Jennings

Wooldrigde und Jennings stellen ebenfalls ein Phasenmodell zum kooperativen Planen vor, das sich von dem von Durfee in einigen Punkten unterscheidet. Ihr Modell umfasst vier Phasen:

#### 1. Erkennen:

Der kooperative Problemlösungsprozess beginnt mit der Erkenntnis eines Agenten, dass Potenzial zu einer kooperativen Aktion vorhanden ist. Diese Erkenntnis kann dadurch verursacht sein, dass ein Agent eine Aufgabe allein nicht erfüllen kann oder Unterstützung wünscht.

#### 2. Teambildung:

In dieser Phase sucht der Agent, der das Potenzial zur Kooperation erkannt hat, nach Partnern.

#### 3. Planbildung:

Die Agenten des zuvor gebildeten Teams versuchen, sich auf einen gemeinsamen Plan zu einigen.

#### 4. Teamaktion:

In dieser Phase wird der zuvor ausgehandelte Plan ausgeführt. Die Agenten sind dabei verbunden durch eine gemeinsame Übereinkunft (z. B. eine Joint Intention).

Wooldrigde und Jennings betten ihr Phasenkonzept im Gegensatz zu Durfee in einen formalen Rahmen ein. Darin wird unter anderem ein Fähigkeitsmodell nach Moore [Moo90] verwendet, um zu beschreiben, wie Agenten Kooperationsmöglichkeiten entdecken können. Die Teambildung zwischen mehreren Agenten ist verbunden mit der Einigung auf ein gemeinsames Ziel. Aus dieser Erkenntnis heraus wurde in [LCN90] das Modell der Joint Intentions entwickelt. Es stellt die bewusste Einigung der Agenten über ein gemeinsames Ziel in den Mittelpunkt. Das Modell orientiert sich an dem Belief-Desire-Intention-Ansatz (BDI) [GPP+98], nach dem jeder Agent aus seinen Annahmen über den aktuellen Umfeldzustand (den Beliefs) und seinen Zielen (Desires) konkrete Intentionen (Intentions) ableitet, um daraus konkrete Handlungen zu bestimmen. Eine Intention beschreibt also die Absicht, das Umfeld in einer bestimmten Art und Weise zu verändern. Dieses Modell soll auf eine Gruppe von Agenten übertragen werden, die sich auf gemeinsame Ziele geeinigt hat. Bei der Übertragung ergeben sich jedoch einige Schwierigkeiten. Die Annahmen der Agenten können sich unabhängig voneinander verändern, was sich auf die Desires und Intentionen auswirkt. Dadurch können die sogenannten Joint Intentions gefährdet werden. Daher ist das zentrale Element der



Joint Intentions, dass Agenten nach dem Bilden eines Teams keine Joint Intention aufgrund individueller Erkenntnisse aufgeben dürfen. In [LCN90] wird ein formales Modell für zwei Agenten auf Basis der Logik erster Ordnung beschrieben, die diese Bedingung berücksichtigt. In [Tam97] wird das Modell auf eine beliebige Anzahl von Agenten erweitert. In [SC96] wird das Kommunikationsprotokoll *request-confirm* vorgestellt, das zur Herstellung von Joint Intentions genutzt werden kann.

Aufbauend auf den Joint Intentions etablierte Jennings den Begriff Joint Responsibility [Jenn95]. Dabei zielt er vor allem auf den Umstand ab, dass nicht nur Absichten abgestimmt werden müssen, sondern auch einzelne Aktionen. Damit unterstützt dieser Formalismus die Phasen *Planbildung* und *Teamaktion*. Die Bindung der Agenten an einen konkreten Plan zum Erreichen des gemeinsamen Ziels nennt Jennings *recipe* und formalisiert es zu einem *individual recipe commitment*. Ein Agent führt solange die Aktionen des gemeinsamen Plans aus, bis:

- Das gewünschte Ziel erreicht ist,
- der aktuelle Plan nicht mehr zum Ziel führen kann,
- eine der enthaltenen Aktionen nicht ausgeführt werden kann oder
- eine der Aktionen nicht korrekt ausgeführt wurde.

Analog zu den Joint Intentions wird verlangt, dass ein Agent nicht einfach seine Bindung an ein *Recipe* aufgeben darf, sondern dies und den Grund der Aufgabe dem restlichen Team mitteilen muss. Dies kann daraufhin unter Umständen Gegenmaßnahmen einleiten.

Joint Intentions, Joint Responsibility und das *request-confirm*-Protokoll stellen eine Möglichkeit dar, Systeme zu einer Planung zusammenzufassen, wenn die Zusammensetzung der interagierenden Systeme vorab nicht bekannt ist. Da die formale Modellierung auf Aussagenlogik basiert, ist die Darstellung numerischer Zustandselemente nur schwierig möglich.

Die in den Phasen 1 und 2 eingesetzten Formalismen bedeuten gleichzeitig die wichtigsten Einschränkungen. Da das Erkennen von Kooperationsmöglichkeiten und Kooperationspartnern nur auf den Fähigkeiten der Agenten basiert, werden nur solche Kooperationen erkannt und realisiert, die zur Erreichung von gemeinsamen Teilzielen notwendig sind. Synergien, die zu einer höheren Planqualität (z. B. durch kürzere Ausführungszeit oder geringere Kosten) führen, werden nicht erkannt.

### 5.1.3 Verfahren zur Berücksichtigung von Interaktion

Im vorigen Abschnitt wurden mögliche Ansätze zur Strukturierung der Interaktionsplanung betrachtet. Es sind jedoch noch geeignete Verfahren notwendig, die zur Durchführung der einzelnen Schritte der Phasenmodelle eingesetzt werden können. Die folgenden Abschnitte beschreiben und untersuchen derartige Verfahren. Die erste Klasse von Verfahren widmet sich der Identifikation von Kooperationspartnern und der Aufgabenverteilungen zwischen ihnen. Anschließend werden Verfahren untersucht, die die eigentliche Planung oder Koordination der Systeminteraktionen durchführen können. Da die Einigung über Handlungsalternativen ein wesentliches Teilproblem ist, werden schließlich Verfahren zur Verhandlung zwischen Agenten analysiert.

#### 5.1.3.1 Verfahren zur Identifikation von Partnern und Aufgabenverteilung

Eine der Teilaufgaben, die im Rahmen der Planung von Systeminteraktionen auftreten können, ist die Identifikation der relevanten Systeme. Diese Aufgabe wird im Rahmen der Multiagenten-Forschung als *Matchmaking* [SDW97] bezeichnet. Verfahren zur Lösung des *Matchmaking*-Problems können unter Umständen auch zur Identifikation relevanter Systeme

eingesetzt werden. Zusätzlich wird hier ein Verfahren vorgestellt, das es erlaubt, Multiagentensysteme in Teilmengen von Agenten einzuteilen, die sich gegenseitig beeinflussen.

#### **5.1.3.1.1 Hierarchisches Task Netzwerk Planen**

Im hierarchischen Task Netzwerk (HTN) Planen werden Probleme gelöst, die auf einem sehr hohen Abstraktionsniveau beschrieben werden [RN03]. Pläne entstehen durch eine immer stärkere Verfeinerung des Problems, also seine Zerlegung in Teilprobleme. Der Planungsprozess ist abgeschlossen, wenn die Teilprobleme so stark verfeinert sind, dass ihre Ausführung trivial ist.

Um die Pläne zu konstruieren, nutzt das HTN-Planen eine Planbibliothek, aus der Problemdekompositionen entnommen werden. Eine Dekomposition ist ein Ausdruck der Form *Decompose(a,d)*, der besagt, dass eine Aktion *a* in einen Plan *d* zerlegt werden kann. Dementsprechend kann jede Aktion auf einer Hierarchieebene eines HTN-Plans als Komposition einer Aktionsfolge auf der nächsten Ebene verstanden werden. Somit kann jede mögliche nicht triviale Aktion als ein separates Planungsproblem aufgefasst werden, bei dem die Vorbedingungen der Aktion dem Initialzustand entsprechen und die Nachbedingungen dem Ziel.

Damit eignet sich das HTN-Planen, um für eine Aufgabe eine Dekomposition zu suchen, die anschließend an eine Menge von Systemen verteilt werden kann. Neben der Aufspaltung einer gemeinsamen Aufgabe zur verteilten Ausführung durch verschiedene mechatronische Systeme eignet sich das HTN-Planen ebenfalls zur Aufteilung der Planungsaufgabe.

#### **5.1.3.1.2 Contract Net Protokoll**

Das Contract Net Protokoll (CNP) [Smi80] ermöglicht eine Kooperation durch Delegieren von Aufgaben zwischen kommunizierenden Agenten. Die Agenten delegieren Teile einer Gesamtaufgabe an andere Agenten, wenn sie selbst die Teilaufgabe nicht durchführen können. Im Rahmen des CNP werden zwei unterschiedliche Rollen unterschieden, die die Agenten übernehmen können. Der Manager ist derjenige Agent, der eine Aufgabe zu delegieren versucht. Die zweite Rolle wird einfach als Agent bezeichnet. Der Kommunikationsaufwand, der zur Delegation einer Aufgabe notwendig ist, hängt von den Informationen ab, die dem Manager zur Verfügung stehen. Liegen keine Informationen über Fähigkeiten oder Auslastung der Agenten vor, muss der Manager alle Agenten kontaktieren. Je mehr Informationen vorliegen, desto gezielter kann die Kommunikation erfolgen.

Agenten, die eine Aufgabe zur Ausführung angeboten bekommen, überprüfen, ob sie die Aufgabe erfüllen können. Falls ein Agent sich zur Annahme einer Aufgabe entschließt, gibt er ein Angebot mit Angabe seiner Fähigkeiten und den Bedingungen beim Manager ab. Der Manager wählt aus allen Angeboten das Beste aus. Das *Extended Contract Net Protocol* (ECNP) erweitert das Contract Net Protocol um die Möglichkeit, komplexe Aufgaben rekursiv an weitere Agenten zu verteilen, falls die Aufgabe die Fähigkeit einzelner Agenten übersteigt. [ZHWZ04]

Das Contract Net Protokoll und auch seine Erweiterung stellen stark auf eine hierarchische Aufgabenstellung ab, wie sie für mechatronische Systeme vor allem innerhalb von autonomen mechatronischen Systemen zu finden ist. Allerdings ist für dieses Anwendungsgebiet eine Aufgabenverteilung bereits bekannt und muss nicht mehr hergestellt werden. Auf Ebene der vernetzten mechatronischen Systeme werden sich in der Regel eher heterarchische Kooperationsformen finden lassen.

### 5.1.3.1.3 *Middle-Agents*

Als Middle-Agents werden Vermittlungs-Agenten bezeichnet, die Informationen über Anbieter und Konsumenten im System haben und so eine Lösung des Matchmaking-Problems ermöglichen. Sycara et. al. [SDW97] unterscheiden drei Arten von Middle-Agents:

- Gelbe Seiten (yellow pages),
- Schwarze Bretter (blackboards) und
- Broker.

Die mögliche Suchgeschwindigkeit und der Umfang des Informationsaustauschs sind dabei die wesentlichen Unterscheidungsmerkmale. In den „Gelben Seiten“ veröffentlichen Anbieter von Leistungen ihre Fähigkeiten, die dann von Konsumenten durchsucht werden können. Das Prinzip eines „Schwarzen Bretts“ ist genau umgekehrt: Konsumenten veröffentlichen ihre Anfragen, die von Anbietern durchsucht werden können [DWS96].

Ein Broker besitzt sowohl über Anbieter als auch über Konsumenten Informationen und versucht, Gruppen mit maximalen Übereinstimmungen zu finden und zwischen den Agenten zu vermitteln. Problematisch ist, dass der Broker im Vergleich zu den Gelben Seiten und den Schwarzen Brettern komplexe Operationen durchführt und daher schnell zu einem Engpass wird [DWS96].

Ströbel und Stolze [SS01] stellen eine Matchmaking-Komponente vor, die nicht nur exakte Übereinstimmungen findet, sondern auch Raum für Verhandlungen berücksichtigen kann. Dies geschieht auf Basis der Operatoren  $<$ ,  $\leq$ ,  $=$ ,  $\neq$ ,  $\geq$ ,  $>$ , AND und OR.

Jang et. al. [JMA04] stellen einen Middle-Agent vor, der keine Kriterien zur Suche vom Konsumenten oder Anbieter als Eingabe erhält, sondern einen Suchalgorithmus, den er auf seine Daten anwendet. Dadurch kann die Suche noch flexibler an die Bedürfnisse der suchenden Agenten angepasst werden.

Mit einigen Modifikationen können Middle-Agents genutzt werden, um mechatronische Systeme zu identifizieren, die ihre Interaktionen planen können oder sollten. Zwei Voraussetzungen müssen dafür aber erfüllt werden: Die möglichen Interaktionen zwischen den Systemen müssen in einer Form beschrieben werden, die ablegbar und durchsuchbar ist. Außerdem muss das Anwendungsszenario der mechatronischen Systeme eine Infrastruktur bieten, die Middle-Agents anbieten kann.

### 5.1.3.1.4 *Verteiltes Matchmaking*

Das verteilte Matchmaking kommt ohne spezielle Infrastruktur aus. Um passende Partner zu finden, können die Agenten nur mit Agenten aus ihrer Nachbarschaft direkt kommunizieren. Diese Nachbarn werden im ersten Schritt befragt, ob eine Kooperation in Frage kommt. Kann eine Kooperation gebildet werden, bilden die beteiligten Agenten einen Cluster. Aufgaben, die noch nicht vergeben werden können, werden nun an die Nachbarn des Clusters weitergegeben. Dadurch werden die Cluster laufend erweitert.

Ogston und Vassiliadis untersuchten diese Form des verteilten, zufälligen Matchmakings in großen Multitagentensystemen. Dabei vollzog sich der oben beschriebene Prozess rundenweise. Es konnte gezeigt werden, dass nach spätestens 1000 Runden 90% der möglichen Matchings gefunden wurden [OV01]. In einem weiteren Experiment konnte gezeigt werden, dass kein Unterschied zwischen einem rundenweisen Vorgehen und einem Vorgehen bei kontinuierlicher Zeit besteht [OV02].

Das verteilte Matchmaking kommt in Frage, wenn das Anwendungsszenario der mechatronischen Systeme keine Infrastruktur für das Matchmaking bietet. Nachteilig ist jedoch das vergleichsweise hohe Kommunikationsaufkommen, das aus dem Mangel an nutzbarer Informati-

on resultiert. Abhängig von der Anwendung muss untersucht werden, inwieweit Informationen über die Nachbarn genutzt werden können, um andere Systeme gezielter anzufragen.

#### **5.1.3.1.5 Agenten mit regionaler Synchronisation**

Weyns und Holvoet stellen ein formales Modell für situative Multiagentensysteme vor. Situativ bedeutet, dass die Handlungen der Agenten innerhalb des MAS von ihrer Position, ihrer Wahrnehmung des Zustandes ihrer Umwelt und ihrem internen Zustand abhängen. Situative Agenten führen keine Langzeitplanung durch, sondern entscheiden anhand einfacher Verfahren über auszuführende Aktionen.

In dem Modell von Weyns und Holvoet führen die Agenten ihre Aktionen synchronisiert aus. Dazu werden die Agenten mittels eines Verfahrens auf Basis des sog. Two-Phase-Commit-Protokolls und einer logischen Uhr in disjunkte Teilmengen unterteilt, zwischen denen kein Synchronisationsbedarf besteht. Dabei sind in einer solchen Teilmenge (Region genannt) alle Agenten zusammengefasst, die entweder direkt oder indirekt miteinander interagieren.

Das Verfahren von Weyns und Holvoet ist ausdrücklich für situative Agenten konzipiert. Bei der Anwendung auf die Planung besteht das Problem, dass im Zeitverlauf so viele Systeme direkt oder indirekt miteinander interagieren, dass die erzeugten Regionen fast so groß wie das Gesamtsystem werden. Damit verliert diese Form der Synchronisation aber ihren Sinn.

Eine Erweiterung des Modells von Weyns und Holvoet um eine zeitliche Dimension könnte einen wertvollen Beitrag zur Identifikation von Partnern leisten. Offen bleibt allerdings die Frage, wie die Beziehung einer direkten Interaktion entdeckt wird.

#### **5.1.3.2 Verfahren zur Interaktionsplanung**

Wie bereits in den Phasenmodellen aus 5.1.2 zu erkennen, wird im Rahmen der Multiagentenplanung nur ein geringes Gewicht auf die Erstellung des eigentlichen Plans gelegt. Ein größeres Gewicht liegt in der Regel auf den vor- und nachgelagerten Phasen, durch die im Vorfeld Konflikte zwischen den Agenten vermieden werden sollen. Die folgenden Absätze erläutern Verfahren, die entweder für die Generierung gemeinsamer Pläne oder die Koordination vor bzw. nach der Planung eingesetzt werden können.

##### **5.1.3.2.1 Partial-Order-Planning with Concurrent Interacting Actions**

Boutilier und Brafman stellen eine Erweiterung des Konzepts des Partial-Order-Planners<sup>45</sup> vor, die in der Lage ist, nebenläufige und interagierende Aktionen zu berücksichtigen [BB01]. Es handelt sich dabei um einen Ansatz, der ausdrücklich eine zentrale Planung für Agententeams oder verteilte Akteure anstrebt.

Boutilier und Brafman erweitern dazu die Aktionenschemata der Planungssprache STRIPS um ein weiteres Element, die „concurrent action list“ (Liste mit nebenläufigen Aktionen). Taucht in der „concurrent action list“ einer Aktion  $A_1$  eine andere Aktion  $A_2$  als nicht negiertes Element auf, muss diese  $A_2$  gleichzeitig mit  $A_1$  ausgeführt werden, um  $A_1$  erfolgreich durchzuführen. Mit der Erweiterung der Planungssprache STRIPS wird ein Partial-Order-Planner mit dem Namen POMP vorgestellt, der sowohl korrekt als auch vollständig ist.

POMP ist als zentraler Planungsansatz nur für Situationen geeignet, in denen die interagierenden Systeme eine gemeinsame Hauptaufgabe durchführen. Für Situationen mit verschiedenen Hauptaufgaben und damit unterschiedlichen Interessen der Systeme ist die Anwendung einer zentralen Planungseinheit nicht plausibel. Das Problem der tendenziell größeren Komplexität zentraler Planungsmodelle für mehrere Systeme wurde bereits in Unterkapitel 2.4 angesprochen.

---

<sup>45</sup> Siehe Anhang B.

### 5.1.3.2.2 *DSIPE*

DSIPE ist ein Planungssystem, bei dem der Planungsprozess verteilt in einem Multiagentensystem stattfindet [DW99]. Es nutzt dazu die Konzepte des hierarchischen Task Netzwerk Planens (vergleiche 5.1.3.1.1). Das System setzt dabei das Vorhandensein eines zentralen Agenten voraus, der eine Gesamtsicht auf die zu lösende komplexe Aufgabe hat. Dieser Agent, der auch als koordinierende Planungszelle bezeichnet [DDOW99] wird, zerlegt die komplexe Aufgabe in mehrere zu erreichende Teilziele und definiert die Reihenfolgebeziehungen zwischen ihnen. Diese Teilziele werden dann an andere Agenten delegiert, die einen Plan zur Erreichung der ihnen zugeordneten Teilziele entwickeln. Die Agenten haben dabei wiederum die Möglichkeit, die Aufgabe zu dekomponieren oder selbst einen Plan zu ermitteln. Nachdem die Teilpläne durch die Agenten erstellt sind, werden sie durch die koordinierende Planungszelle zu einem vollständigen globalen Plan zusammengefasst.

DSIPE eignet sich ebenso wie POMP nur für Situationen, in denen die interagierenden Systeme eine gemeinsame Hauptaufgabe durchführen. Ein klarer Vorteil gegenüber POMP ist jedoch die verteilte, nebenläufige Ausführung der Planung, wodurch sich eine höhere Planungsgeschwindigkeit erreichen lässt.

### 5.1.3.2.3 *Generalized Partial Global Planning*

Generalized Partial Global Planning (GPGP, [DL92] und [DL94]) ist eine Verallgemeinerung eines Ansatzes zur Koordination mehrerer Agenten im *Vehicle Monitoring* [DL87]. Die Grundidee des GPGP besteht darin, Agenten ihre Pläne austauschen zu lassen, wenn ihre Aufgaben oder Ziele in einer Relation zueinander stehen. Ein Agent, der zusätzlich zu seinem eigenen Plan die Pläne weiterer Agenten kennt, besitzt damit Informationen über einen teilweise globalen Plan (partial global plan). Diese Informationen kann der Agent dann nutzen, um diesen teilweise globalen Plan zu optimieren und beispielsweise Redundanzen zu vermeiden.

Das ursprüngliche Partial Global Planning (PGP) wurde genau auf die Anwendung im *Vehicle Monitoring* zugeschnitten. Das galt insbesondere für mögliche Relationen zwischen Tasks und Zielen und die möglichen Verbesserungen im teilweise globalen Plan. Im GPGP wird nun eine Menge von anwendungsunabhängigen Relationen und Verbesserungsaktionen eingeführt, die den anwendungsspezifischen Tasks und Zielen der Agenten zugeordnet werden können. Dadurch kann ein anwendungsunabhängiges Verfahren zur Koordination genutzt werden.

Zu den im GPGP definierten Relationen gehören unter anderem:

- Basic Domain Relationships: *verhindert, bricht ab, beschränkt, begünstigt, löst aus, ermöglicht, hierarchische Beziehung*
- Graph Relationships: *überlappen, notwendig, ausreichend, erweiternd, zusammenfassen, konkurrieren*
- Temporal Relationship: *vorher, gleichzeitig, anschließen, überlappend, während, abschließen.*

Eine der wichtigsten Relationen zwischen Aufgaben ist *begünstigt* (engl. *facilitates*). Diese Relation besitzt zwei Parameter (*Power-Parameter*), die die Art und den Umfang der Begünstigung genauer beschreiben:  $\varphi_d$  und  $\varphi_q$ . Sie beschreiben die Wirkung auf Dauer ( $\varphi_d$ ) und Qualität ( $\varphi_q$ ). Die Wirkung wird allerdings nicht allein durch die Parameter bestimmt, sondern auch durch die Qualität, die vom begünstigenden Task bereitgestellt wird, wenn der begünstigte Task beginnt. Angenommen,  $T_a$  begünstigt  $T_b$  mit einem bestimmten  $\varphi_d$  (angenommen  $\varphi_q$  sei 0). Dann lässt sich aus der Qualität, mit der  $T_a$  beendet wurde, die Wirkung auf die Dauer von

$T_b$  errechnen als  $\varphi \cdot \frac{\text{quality received}}{\text{max quality}}$  [DGLH92]. Mithilfe dieser Form der Modellierung von Beziehungen wird eine zielgerichtete Koordination der Aufgaben möglich.

Durfee [Durf02] beschreibt den allgemeinen Ablauf der Koordination wie folgt: Während der ersten Phase des GPGP planen die Agenten lokal. Dazu definieren sie ihre Ziele und bestimmen, welche Aktionen möglicherweise zum Erreichen dieser Ziele führen. Dabei können die lokalen Pläne mehrere Ziele und Verzweigungen und alternative Aktionen enthalten. Aufgrund dieser möglichen Komplexität ist eine *Abstraktion der lokalen Pläne (Local Plan Abstraction)* notwendig, um die Koordination der Pläne effizient und korrekt durchzuführen. Das führt zu einem abstrakten Plan, der nur die wesentlichen Schritte enthält, die für andere interessant sein könnten.

Diese abstrakten Pläne sind Gegenstand des *partial global plan construction and modification*, also der Erstellung und Konstruktion globaler Pläne. Dabei werden die lokalen Pläne in einem (teilweise) globalen Plan zusammengefügt und analysiert. Ergebnis der Analyse sind Möglichkeiten zur Verbesserung der lokalen Pläne aus der globalen Sicht, wie zum Beispiel die Vermeidung redundanter Aktivitäten oder eine verbesserte zeitliche Abfolge von Aktivitäten in verschiedenen lokalen Plänen. Aufbauend auf dem (teilweise) globalen Plan können Interaktionen in Form von Kommunikation zwischen den Agenten geplant werden. Auf Basis des (teilweise) globalen Plans kann ein Agent bestimmen, wann ein Task von welchen Agenten abgeschlossen wird und für welche anderen Agenten dieser relevant ist.

Nach dem ein (teilweise) globaler Plan erzeugt wurde und die lokalen Pläne angepasst wurden, werden diese ausgeführt. Bei bestimmten Abweichungen vom ursprünglichen Planverlauf (z. B. durch unvorhergesehene Umwelteinflüsse) besteht die Möglichkeit, die Pläne erneut zu koordinieren. Dabei löst der Agent, der die Abweichung feststellt, die erneute Koordination aus. Dabei muss abgewogen werden zwischen dem Nutzen der erneuten Koordination und dem damit verbundenen Aufwand.

In [DL00] erweiterten Decker und Li das GPGP um die Berücksichtigung exklusiver, nicht verbrauchbarer Ressourcen.

Generalized Partial Global Planning ermöglicht vor allem die Realisierung von Synergien zwischen den Plänen unterschiedlicher Agenten. GPGP definiert zwar eine Menge an Relationstypen, die zwischen Aktionen bestehen können, jedoch müssen diese für jede neue Problemdomäne für jedes Paar von Aktionen neu definiert werden. Dies bedeutet einen erheblichen Modellierungsaufwand und lässt dem System außerdem keine Möglichkeit zur weiteren Adaption. Sowohl die Relationen als auch die Koordinationsmechanismen (Aufzählung siehe [PL96]) berücksichtigen immer nur Paare von Aktionen. Daher sind die Aktionen entweder weitgehend unabhängig (eine Änderung an einem Plan lässt den restlichen Plan unberührt) oder es entstehen durch die Koordinationen notwendige Plananpassungen, die wiederum koordiniert werden müssen.

#### 5.1.3.2.4 Shared Activity Coordination

Shared Activity Coordination ist ein System zur Koordination eigennütziger Agenten durch Verhandlung [CB03]. Für die Erstellung und Ausführung der individuellen Pläne wird das CASPER (Continuous Activity Scheduling Planning Execution and Replanning) Framework eingesetzt [CKS+00]. Dabei hat jeder Agent eine genaue Vorstellung von seinen geplanten Aktionen und deren zeitlicher Abfolge. Da CASPER einen besonderen Schwerpunkt auf die Verzahnung von Planung und Ausführung legt, muss jede Aktion eines Agenten mit einem festen Zeitpuffer vor dem geplanten Start an ein Ausführungsframework (das tatsächlich den Plan ausführende System) übergeben. Ab diesem Zeitpunkt kann die Aktion nicht mehr geändert werden und ist fest eingeplant. Aus diesem Grund verwendet CASPER eine Zeitspanne (commit window) vor Beginn jeder Aktion, innerhalb derer das Planungssystem Konflikte zwischen den eingeplanten Aktionen auflösen muss. Zusätzlich sieht das SHAC-System noch

die Koordination zwischen Agenten vor, die gemeinsame Aktionen (shared activities) durchführen. Eine gemeinsame Aktivität ist definiert als ein Tupel (*Parameter, Rollen, Protokolle, Dekomposition, Beschränkung*). Die Parameter beschreiben die für die Aktivität relevanten Variablen (z. B. Startzeitpunkt, Dauer, Positionen, etc.) und deren aktuelle Werte. Die Rollen sind Platzhalter für die konkreten Akteure, die an der Aktivität beteiligt sind, und ordnen diesen einzelne Aufgaben zu. Um die Aufgabenverteilung etwas flexibler gestalten zu können, werden unterschiedliche Dekompositionen komplexer Aufgaben modelliert, zwischen denen die Systeme auswählen können. Dekompositionen sind vorab definierte Zerlegungen der gemeinsamen Aktivität, aus denen die Agenten auswählen können.

Zur Koordination und Einigung über die Parameter und die generelle Ausführung der gemeinsamen Aktivität werden Verhandlungen eingesetzt, die einem der gemeinsamen Aktivität zugeordneten Protokoll entsprechen müssen. Dabei kann das Protokoll z. B. Rangfolgen oder Master/Slave-Beziehungen zwischen Agenten definieren, um eine schnelle Einigung zu erzielen. Je kürzer das commit window wird, desto einfachere Protokolle werden eingesetzt, um eine Einigung zu erzielen.

Das Modell der Shared Activities Coordination ist relativ statisch und muss für konkrete Anwendungsbereiche maßgeschneidert ausgelegt werden, da die Shared Activities im Vorfeld der Planung definiert werden müssen und sehr viele Informationen enthalten. Das ermöglicht auch das relativ problemlose Einfügen der Shared Activities in die individuellen Pläne der Agenten, da das Ergebnis der Aktivität vorab im Wesentlichen bekannt ist. Auswirkungen auf die individuellen Pläne werden damit beschränkt.

Das Konzept des Commit Windows scheint jedoch geeignet, um die individuelle Planung, die Koordination und die Ausführung der Pläne miteinander zu verzahnen.

#### **5.1.3.2.5 Coordination Before Planning**

Witteveen et al. ([SWMV06] und [BMVW06]) stellen ein System zur Koordination eigennütziger Agenten im Vorfeld der eigentlichen Planung vor. Dabei gehen die Autoren davon aus, dass die Agenten zwar eigennützig sind, aber doch eine gemeinsame Aufgabe erfüllen müssen. Aus der Eigenschaft der Eigennützigkeit wird gefolgert, dass die Agenten sich nicht über ihre individuellen Pläne austauschen.

Die komplexen Aufgaben, die die Agenten erfüllen sollen, sind bereits vor der Koordination in elementare Teilaufgaben (Tasks) zerlegt. Ein Task kann dabei für einen einzelnen Agenten wiederum ein Planungsproblem darstellen. Zusätzlich sind Abhängigkeiten zwischen den elementaren Tasks angegeben, die ähnlich wie die Mutexe im Partial-Order-Planning eine zeitliche Abfolge der Tasks erzwingen. Abhängigkeiten, die zwischen den Tasks eines Agenten bestehen, werden als Intra-Agent-Abhängigkeit bezeichnet. Inter-Agent-Abhängigkeiten sind Abhängigkeiten zwischen Tasks unterschiedlicher Agenten.

Zur Berücksichtigung der Inter-Agenten-Abhängigkeiten mit möglichst geringem Informationsaustausch wird ein dezentraler Koordinationsalgorithmus vorgeschlagen. Der Algorithmus setzt ein Blackboard ein, auf dem alle Inter-Agenten-Abhängigkeiten ausgewählt sind. Jeder Agent beginnt nun mit der Planung und Ausführung einer Teilmenge der ihm zugeordneten Tasks, die von keinem anderen Task abhängen. Nach der Planung und Ausführung eines Tasks  $t$  teilt der Agent dem Blackboard mit, dass  $t$  erfüllt ist, und das Blackboard entfernt alle Inter-Agent-Abhängigkeiten, in denen  $t$  als zuerst auszuführender Task enthalten ist. Damit wächst die Menge der Tasks, die nicht von Inter-Agenten-Abhängigkeiten betroffen sind, stetig an und schließlich können alle Tasks durchgeführt werden.

Im Konzept von Witteveen et al. bleiben wesentliche Fragen unbeantwortet. Es bleibt unklar, wie sich eigennützige Agenten auf eine Zerlegung der gemeinsamen Aufgabe einigen und die Abhängigkeiten innerhalb dieser Zerlegung bestimmen. Auch erscheint die Einschränkung nicht plausibel, dass Agenten, die sich auf eine gemeinsame Aufgabe verständigt haben, auf

jegliche direkte Kommunikation zur Planung dieser Aufgabe verzichten. Sie führt auf jeden Fall dazu, dass mögliche Synergien zwischen den individuellen Plänen nicht genutzt werden.

### 5.1.3.3 Verfahren zur Verhandlung zwischen Systemen

Einige der im vorigen Kapitel vorgestellten Ansätze setzen Verhandlungen zwischen den beteiligten Systemen ein, um Synergien zu realisieren oder Konflikte zu lösen (z. B. Generalized Global Planning oder Shared Activities). Der Vorteil des Verhandlungsansatzes gegenüber anderen Koordinationsmechanismen besteht darin, dass er sich sowohl für kooperative Systeme (also solche, die eine gemeinsame Hauptaufgabe verfolgen) und konkurrierende Systeme (solche mit unterschiedlichen Hauptaufgaben) eignen. Aus diesem Grunde sollen an dieser Stelle Methoden vorgestellt werden, die zur Verhandlung zwischen interagierenden Systemen eingesetzt werden können.

#### 5.1.3.3.1 Spieltheoretische Verhandlungen

Spieltheoretische Ansätze nutzen die Modellierung und die Prinzipien der Spieltheorie. Das bedeutet, ein verhandelnder Agent beachtet bei der Verhandlung auch die Optionen und vermutliche Strategie seiner Verhandlungspartner. Die Verwendung spieltheoretischer Methoden setzt voraus, dass die Präferenzen eines Agenten numerisch durch eine Nutzenfunktion (*utility function*) ausgedrückt oder zumindest in eine ordinale Rangfolge gebracht werden können.

Ein oft genutztes Verhandlungsprotokoll ist das „*abwechselnde Bieten*“ [FWJ05]. Dabei bieten sich die Agenten abwechselnd einen Kompromiss an. Ein Verhandlungspartner kann einem Angebot zustimmen oder ein neues Angebot abgeben, das nicht schlechter als das vorherige sein darf. Das Protokoll endet mit einem Kompromiss oder dem Überschreiten einer zeitlichen Deadline. Die Verhandlungsprotokolle können unterschieden werden anhand der zur Verfügung stehenden Information (Deadlines, Präferenzen und Strategien der übrigen Agenten), ob der Gegenstand ein oder mehrere Attribute besitzt (z. B. Preis, Qualität, Termin) und ob zwei oder mehrere Agenten beteiligt sind.

Weitere Verhandlungsprotokolle für Situationen mit gemeinsamer Information finden sich beispielsweise bei Rosenschein und Zlotkin [RZ94], Fershtman und Seidmann [FS93] und Calabuig et. al. [CCO02]. Ein Protokoll für private Informationen stellen Fatima et. al. vor [FWJ05]. Verfahren für Verhandlungen über mehrere Attribute werden u. a. in [FWJ04], [FSJ98], [CJ04], [Lin04] und [BL00] vorgestellt. Die Verfahren unterscheiden sich hinsichtlich der Arten von Verhandlungsattributen (feste und optionale Attribute), Garantie eines Pareto-Optimums, Verhandlungsdauern und möglichen Rollen der Verhandlungspartner. Alle diese Eigenschaften sind sehr anwendungsspezifisch und werden daher an dieser Stelle nicht erläutert.

Die spieltheoretische Modellierung der Verhandlung impliziert von eigennützigen Systemen aus. Daher werden spieltheoretische Verhandlungen bei Systemen mit gemeinsamer Hauptaufgabe nicht zu effektiven Lösungen des Verhandlungsproblems führen.

#### 5.1.3.3.2 Argumenten-basierte Verhandlungen

Eine Alternative zu spieltheoretisch orientierten Verhandlungen sind argumenten-basierte Verhandlungssysteme. In der Regel sind sie wesentlich komplexer als spieltheoretische Verhandlungen, da in argumenten-basierten Verhandlungen neben Angeboten und Gegenangeboten auch Meta-Informationen in Form von Erklärungen und Kritik ausgetauscht werden [PJ96]. Die Meta-Information hat zwei wichtige Funktionen: Zunächst sollen Erklärungen zu Absagen dem Verhandlungspartner ermöglichen, weitere nicht akzeptable Angebote zu vermeiden. Die Kritik an Argumenten in Form von Gegenargumenten dient dazu, fehlerhafte



Annahmen des Verhandlungspartners zu korrigieren und somit weitere nicht akzeptable Angebote auszuschließen.

In [PJ96] wird ein Ansatz zur automatischen Argumentation vorgestellt, bei dem Argumente logische Schlussfolgerungen für oder gegen einen Vorschlag sind. Dabei werden zwei Arten von Argumenten unterschieden:

- Nicht-triviale Argumente beruhen auf einer konsistenten Menge von Annahmen.
- Tautologische Argumente beruhen auf keinerlei weiteren Annahmen (es handelt sich also gewissermaßen um die Axiome der Argumentation).

Argumente können auf zweierlei Arten widerlegt werden: Die Schlussfolgerung eines Arguments wird durch ein Gegenargument direkt entkräftet (rebutting argument) oder eine Grundannahme eines Arguments wird durch ein Gegenargument widerlegt, sodass die Argumentationsgrundlage für die Schlussfolgerung zerstört ist (undercutting argument).

Wie bereits bei den spieltheoretischen Ansätzen werden Protokolle zur Steuerung des Verhandlungsablaufs genutzt. Amgoud et. al. [APM01] beschreiben ein Protokoll für argumenten-basierte Dialoge.

Für Systeme, die eine gemeinsame Hauptaufgabe zu erfüllen haben, stellen die argumenten-basierten Verhandlungen eine interessante Alternative zu spieltheoretischen Verhandlungen dar. In Form der Argumentationen können diese Systeme, die mit der Hauptaufgabe auch ein gemeinsames übergeordnetes Zielsystem besitzen, sich über die Wechselwirkungen bestimmter Handlungsoptionen hinsichtlich des gemeinsamen Zielsystems austauschen.

### 5.1.3.3 Satisficing Game Theory

Stirling beschreibt mit der Satisficing Game Theory [Stir03] einen formalen Ansatz zur Koordination von Agenten.

Die Grundidee hinter der Satisficing Game Theory besteht darin, jedem Agenten zwei Rollen zuzuordnen: Die eine konzentriert sich auf das Erreichen des fundamentalen Ziels eines Entscheidungsproblems unabhängig von den anfallenden Kosten. Die zweite Rolle versucht dagegen, genau diese Kosten auf jeden Fall zu senken. Gemeinsam bilden diese beiden Rollen einen Entscheider, der die Waage halten kann zwischen dem Wunsch, ein Ziel zu erreichen, und den Kosten, die damit verbunden sind. Für beide Rollen werden zwei Nutzenfunktionen definiert, die den Entscheidungsalternativen Werte aus der jeweiligen Perspektive zuordnen. Dabei bezeichnet die *Selectability* (*Wählbarkeit*) den Nutzen der Alternativen aus Sicht der Zielerreichung, während die *Rejectability* (*Verwerfbarkeit*) die Kosten darstellt. Dabei werden diese beiden Nutzenfunktionen normalisiert zu Massenfunktionen<sup>46</sup>. In Multiagentenproblemen handelt es sich um multivariate Massenfunktionen. Da diese Nutzenfunktionen als Massenfunktionen die gleichen mathematischen Eigenschaften wie Wahrscheinlichkeitsfunktionen besitzen, können Beziehungen zwischen den Variablen wie Unabhängigkeit und Bedingtheit analog definiert werden.

Hill et. al. bezeichnen die *Selectability* Massenfunktion mit  $p_S$  und die *Rejectability* Massenfunktion mit  $p_R$ . Werden *Selectability* und *Rejectability* gleichzeitig betrachtet, bezeichnet  $p_{SR}$  die vereinigte Massenfunktion. Für ein System mit  $n$  Agenten ist die vereinigte *Selectability/Rejectability* eine Massenfunktion mit zweimal  $n$  Variablen:

$$p_{s_1 s_2 \dots s_n r_1 r_2 \dots r_n} = (u_1, u_2, \dots, u_n, u_1, r_2, \dots, r_n)$$

<sup>46</sup> Beispielsweise sind diskrete Wahrscheinlichkeitsfunktionen Massenfunktionen, da sie die Masse 1 (die Wahrscheinlichkeit des sicheren Ereignisses) auf eine Menge diskreter Punkte verteilen.

Dabei bezeichnen  $S_1$  bis  $S_n$  die Selectability-Rollen und  $R_1$  bis  $R_n$  die Rejectability-Rollen der  $n$  Agenten. Die Variable  $u_i$  ( $i = 1 \dots n$ ) entspricht den Entscheidungsalternativen des  $i$ -ten Agenten aus Sicht der Selectability-Rolle. Entsprechend stellen die Variablen  $v_1 \dots v_n$  die Rejectability der Entscheidungsalternativen dar. Mithilfe von Abhängigkeitsfunktionen lassen sich Abhängigkeiten zwischen den Entscheidungsalternativen der einzelnen Agenten darstellen. Diese Abhängigkeitsfunktionen können ähnlich einer totalen bedingten Wahrscheinlichkeitsfunktion in Form von Graphen dargestellt werden, wobei jeder Knoten des Graphen für eine bedingte Massenfunktion steht. Beispielsweise drückt die bedingte Massenfunktion  $p_{R_1|S_2R_2}(v_1|u_1, v_2)$  die Rejectability der Option  $v_1$  aus, die Agent 1 unter der Annahme bestimmt, dass Agent 2 die Option  $u_2$  wählt (aus Sicht der Zielerreichung) und  $v_2$  dagegen verwirft (unter Betrachtung der Kosten).

Auf Basis derartiger Nutzenfunktionen können die Systeme die Auswirkung ihrer Entscheidung auf andere Systeme bestimmen und zu der Entscheidung kommen, ihre eigenen Ziele zugunsten anderer Systeme zurückzustellen. In [HJA+05] wurde ein derartiges „bedingtes“ altruistisches Verhalten für das Ausweichverhalten von Flugzeugen implementiert.

Die Satisficing Game Theory mit der Definition bedingter Nutzenfunktion stellt eine interessante Basis für Verhandlungen zwischen Systemen mit gemeinsamer Hauptaufgabe dar. Aufgrund der gemeinsamen Hauptaufgabe ist eine Entscheidung für altruistisches Verhalten plausibel und kann die Verhandlungsdauer erheblich abkürzen.

#### 5.1.3.3.4 Auktionen und Voting

Auktionen und Voting stellen in Multiagentensystemen wichtige Werkzeuge z. B. zur Ressourcenallokation bzw. zur Herbeiführung von Einigungen dar.

Ziel einer Auktion ist die Allokation eines Gutes gegenüber einer Menge von Bietern. Dabei sollte diese Allokation effizient sein, es soll der Bieter das Gut erhalten, der es am höchsten einschätzt bzw. den höchsten Nutzen aus ihm zieht [Coni08]. Dabei stehen eine Menge von Auktionsmechanismen mit unterschiedlichen Eigenschaften, Vor- und Nachteilen zur Verfügung. Besondere Bedeutung im Kontext der Multiagenten-Systeme hat die Vickrey Auction, bei der ein verdecktes Gebot abgegeben wird. Es gewinnt derjenige Bieter, der das höchste Gebot abgegeben hat, muss jedoch nur das zweithöchste Gebot zahlen. Durch diesen Mechanismus ist sichergestellt, dass jeder Agent entsprechend seiner tatsächlichen Präferenz bietet, da eine Abweichung keine Vorteile bietet. Auch umgekehrte Auktionen, bei denen mehrere Anbieter um einen Auftrag konkurrieren, sind üblich [PRW01].

Voting dagegen beschreibt eine Situation, in der eine Menge von Agenten sich auf eine Alternative aus einer ganzen Menge von Alternativen einigen muss [PKSA06]. Dazu gibt jeder Agent eine Präferenzbewertung über alle Alternativen an. Ausgewählt wird nach einer zuvor festgelegten Regel. Ein einfaches Beispiel für eine solche Regel ist *Plurality*, bei der eine Alternative einen Punkt erhält, wenn sie von einem Agenten als beste beurteilt wurde. Erheblich komplexere Auswahlregeln sind vorstellbar, bedeuten jedoch auch erheblich mehr Rechenaufwand [Coni08].

Auktionen und insbesondere umgekehrte Auktionen sind geeignet, um im Rahmen einer Aufgabenverteilung einen konkreten Agenten auszuwählen, der die Aufgabe ausführen soll. Der Mechanismus des Votings wäre einsetzbar, wenn sich eine Menge von Agenten zwischen einer Menge von alternativen Plänen entscheiden muss.

#### 5.1.4 Fazit zum Stand der Technik zur Interaktionsplanung

Der Stand der Technik zur Interaktionsplanung liefert Verfahren und Methoden, die zumindest eingeschränkt oder für bestimmte Anwendungsszenarien geeignet sind, um die in Abschnitt 2.4 definierten Anforderungen und Zielsetzungen zu erreichen.

Für die Interaktionsplanung bei gemeinsamer Aufgabe kommt zur Strukturierung der Planung das Phasenmodell von Durfee in Betracht. Zur Verfeinerung der globalen Aufgaben und der Aufgabenzuordnung stehen mit dem *HTN-Planen* und dem *Contract Net Protokoll* geeignete Verfahren zu Verfügung. Für die Schritte *Koordination vor der Planung* und *Koordination nach der Planung* liefert das Generalized Partial Global Planning mit seinen Relationen zwischen den Systemaktivitäten einen interessanten Ansatz. Um die Planung mit der Ausführung zu verzahnen, kommt das Commit-Window aus der GPGP Erweiterung SHAC in Betracht. Dabei können die Koordinationsmechanismen aus GPGP und SHAC nicht direkt zur Anwendung kommen, da diese nur auf einzelnen Aktionen operieren. Sind Aktionen nicht weitgehend unabhängig, kann der Austausch von Aktionen zu einer Inkonsistenz der lokalen Pläne führen. Umplanungen und damit neuer Koordinationsbedarf wäre die Folge. Hier sind Koordinationsaktionen zu identifizieren, die die Konsistenz des lokalen Plans sicherstellen.

Um die individuelle Planung auf Basis der PDDL durchführen zu können, müssen die Wechselwirkungen in das Planungsmodell integriert werden. Dementsprechend ist zu beschreiben, wie eine Koordination vor und nach der Planung für mechatronische Systeme durchgeführt werden kann und wie die dafür zu definierenden Relationen in die PDDL integriert werden können. Hier bietet die Satisficing Game Theory mit ihren bedingten Nutzenfunktionen einen interessanten Ansatz.

Für die Planung bei individueller Hauptaufgabe ist das Phasenmodell nach Wooldridge und Jennings besser geeignet. Da nach Analyse des Stands der Technik offen bleibt, wie das *Erkennen von Kooperationspotenzial* erfolgt, muss dieser Schritt im folgenden Konzept erarbeitet werden. Zur *Teambildung* stehen folgende Methoden zur Verfügung: Je nach Anwendungsgebiet kommt das Contract Net Protokoll, Middle-Agents oder das verteilte Matchmaking in Frage. Hier ist allerdings zu untersuchen, wie im Rahmen dieser Methoden geeignete Kooperationspartner identifiziert werden können. Im Rahmen der *Planbildung* steht die Einigung auf einen Plan für alle beteiligten Systeme im Vordergrund. Hier können die Methoden der Verhandlung oder des Votings eingesetzt werden, um zwischen mehreren alternativen Plänen auszuwählen. Damit müssen im Wesentlichen keine neuen Verfahren entwickelt werden, stattdessen ist die Kooperation zwischen Systemen mit individueller Hauptaufgabe vor allem zu systematisieren und zu strukturieren.

## 5.2 Planung der Interaktion bei gemeinsamer Hauptaufgabe

Bei Vorliegen einer gemeinsamen Hauptaufgabe kann davon ausgegangen werden, dass die Zusammensetzung der interagierenden Systeme vorab bekannt ist, da zwischen den Systemen eine Einigung über Art und Inhalt der Aufgabe stattgefunden haben muss. Im Folgenden soll daher immer die Rede von Teilsystemen sein, um eine Gruppe von Systemen zu bezeichnen, die eine gemeinsame Hauptaufgabe erfüllen. Der Begriff Teilsystem setzt allerdings keine räumliche oder gar bauliche Integration der Systeme voraus. Aufgrund der statischen Zusammensetzung der Teilsysteme scheint das Phasenkonzept nach Durfee (vergleiche 5.1.2.1) geeignet. Daher soll das zu konzipierende Verfahren in die Schritte *Koordination vor der Planung*, *Koordination nach der Planung* und *Planausführung* zergliedert werden, wobei die Koordination nach der Planung und die Planausführung miteinander verzahnt werden. Weiterhin kann in diesem Fall, aufgrund der gemeinsamen Hauptaufgabe, davon ausgegangen werden, dass die Teilsysteme vornehmlich an den gleichen Umgebungsobjekten handeln.

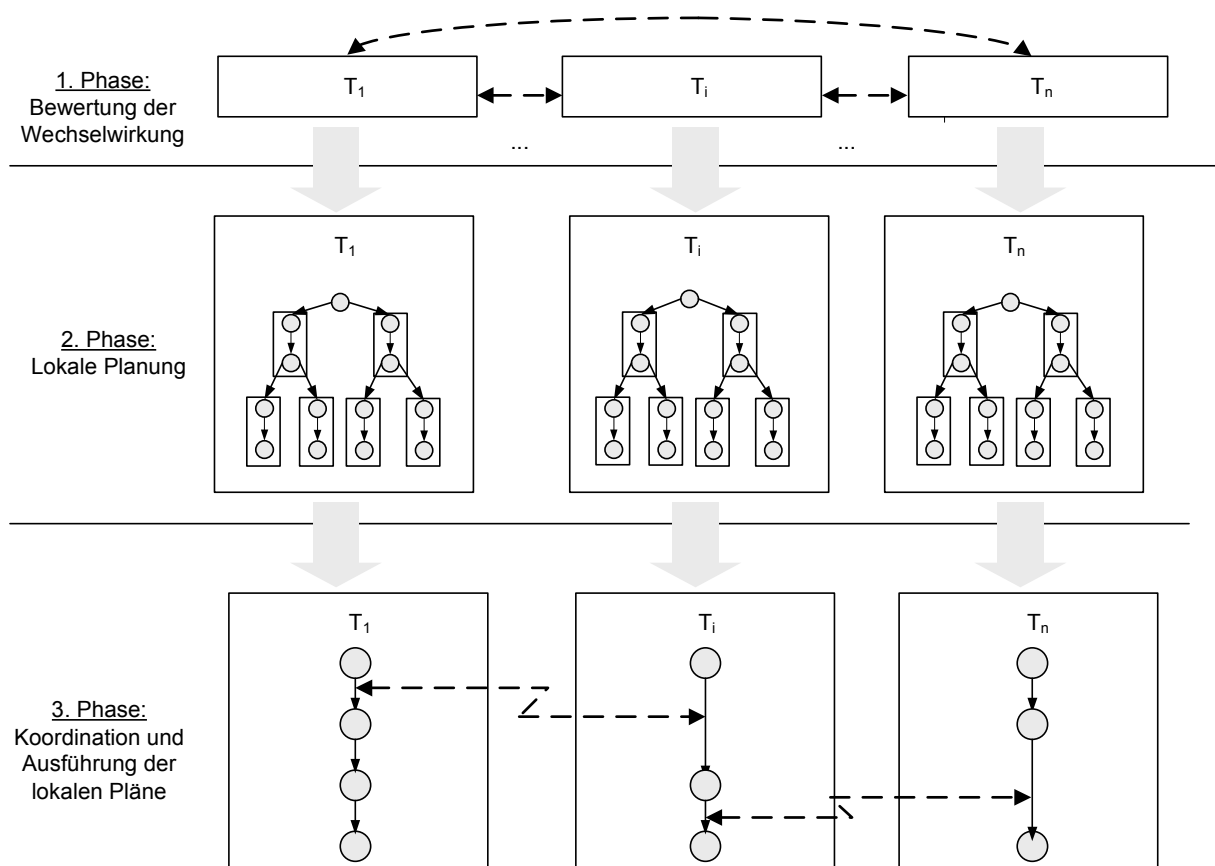


Abbildung 5-4: Skizze der Interaktionsplanung bei gemeinsamer Hauptaufgabe

Aus diesen Gründen fokussieren sich die folgenden Abschnitte auf Verfahren der Planung bei gemeinsamer Hauptaufgabe. Die Probleme der Identifikation von Kooperationspartnern und der Aufgabenverteilung sind nachrangig und durch den Stand der Technik zufriedenstellend gelöst. Dabei wird das Planungsverfahren so konstruiert, dass das Wissen, wie die Teilaufgaben eines Teilsystems erfüllt werden können, nur im jeweiligen System vorliegen muss. Dadurch kann die eigentliche Planung lokal erfolgen. Um dennoch zu einer Menge von Teilfunktionslösungsfolgen zu kommen, die die Wechselwirkungen zwischen den Systemen berücksichtigt und sogar ausnutzt, sodass die Hauptaufgabe in möglichst großer Übereinstimmung mit dem aktuellen Zielsystem erfüllt wird, wird eine Modellierung für die Abhängigkeiten zwischen den Aktivitäten der Systeme entwickelt. Die Modellierung kann im Rahmen einer Abstimmung vor der eigentlichen lokalen Planung sowie einer anschließenden Koordination der Pläne genutzt werden. Dabei ist für beide Fälle keine genaue Kenntnis über die

interne Natur der Teilfunktionslösungen der Teilsysteme notwendig. Abschließend wird gezeigt, wie die gewählte Modellierung auf Lern- und Klassifikationsprobleme abgebildet werden kann, um die Parameter des Modells durch die Systeme selbstständig anhand im Betrieb gesammelter Erfahrungen zu bestimmen und anzupassen. Abbildung 5-4 skizziert das angestrebte Verfahren.

### 5.2.1 Mögliche Koordinationsaktionen

Mechatronische Systeme beeinflussen sich gegenseitig durch die Teilfunktionslösungen, die sie zur Erfüllung ihrer Aufgabe einsetzen. Dabei kann die Ausführung einer Teilfunktionslösung durch das Teilsystem A positiv oder negativ auf die Ausführung einer Teilfunktionslösung durch das Teilsystem B wirken. Offensichtlich können Teilfunktionslösungen nur durch solche Aktivitäten anderer Teilsysteme beeinflusst werden, die ihnen zeitlich vorgelagert oder gleichzeitig ausgeführt werden<sup>47</sup>.

Da ein Plan nur hinsichtlich der ausgewählten Aktivitäten und der zeitlichen Anordnung von Aktivitäten verändert werden kann, ergeben sich drei grundsätzliche Möglichkeiten, die Koordination zwischen zwei mechatronischen Systemen zu verbessern:

1. Einsatz einer alternativen Teilfunktionslösung: Die alternative Teilfunktionslösung hat eine bessere Wechselwirkung mit den Aktivitäten anderer Systeme.
2. Aktivität in ein anderes Zeitfenster verschieben: Die Aktivität hat in dem Zeitfenster eine positive Wechselwirkung auf Aktivitäten anderer Systeme.
3. Aktivität aus einem Zeitfenster schieben: Die Aktivität hat eine negative Wechselwirkung auf eine Aktivität anderer Systeme.

Diese drei grundsätzlichen Koordinationsoperationen beziehen sich jeweils auf genau eine Aktivität, sie können aber dennoch nicht losgelöst vom übrigen lokalen Plan betrachtet werden. Der Austausch von Teilfunktionslösungen oder die zeitliche Verschiebung von Aktivitäten ist nicht ohne Weiteres möglich, wenn:

- Die Voraussetzungen zum Einsatz der alternativen Teilfunktionslösung im bisherigen Plan nicht erfüllt sind.
- Der Einsatz der alternativen Teilfunktionslösungen oder die zeitliche Verschiebung einer Teilfunktionslösung den Plan unausführbar macht, da die Voraussetzungen für die folgende Aktivitäten nicht mehr erfüllt sind.

Dass die Ausführung der Koordinationsoperation im aktuellen Plan nicht möglich ist, bedeutet jedoch nicht, dass kein Plan existiert, der das Koordinationspotenzial ausnutzt. Daher sind Modifikationen am Plan notwendig – das System, das die Koordinationsoperation ausführen soll, muss also umplanen.

Abbildung 5-5 verdeutlicht diesen Sachverhalt am Beispiel des Planungsproblems des Antriebs (Auswahl von Geschwindigkeitsverläufen). Die Abbildung zeigt einen Teilplan, dessen dritte Aktivität einen Konflikt (z. B. eine negative Wechselwirkung) mit einem anderen Teilsystem (z. B. dem AGAS) aufweist. Durch die Einbindungen in den Plan kann die Aktion aber bedingt durch seinen direkten Vorgänger und Nachfolger nicht einfach ausgetauscht werden. Ein möglicher Ansatz besteht nun darin, anstatt eines einfachen Austauschs ein Planungsproblem zu formulieren, das auch die Vorgänger und Nachfolger der Aktivität umfasst. Dabei muss das Ziel des Planungsproblems so formuliert werden, dass der nachfolgende Plan

<sup>47</sup> Das ist zumindest solange der Fall, wie von einem linearen Zeitverlauf ausgegangen wird

ohne Anpassung weiter ausgeführt werden kann. Im Beispiel muss im letzten Schritt also genau die Endgeschwindigkeit im letzten Schritt erreicht werden, die die Ausführung des folgenden Fahrprofils ermöglicht.

Die oben genannten Koordinationsoperationen können auf gewöhnliche PDDL-Planungsprobleme zurückgeführt werden. Durch die im folgenden Kapitel ermittelten detaillierten Nutzenwerte kann bei der Lösung dieser Planungsprobleme die Wechselwirkung mit den übrigen Teilsystemen berücksichtigt werden.

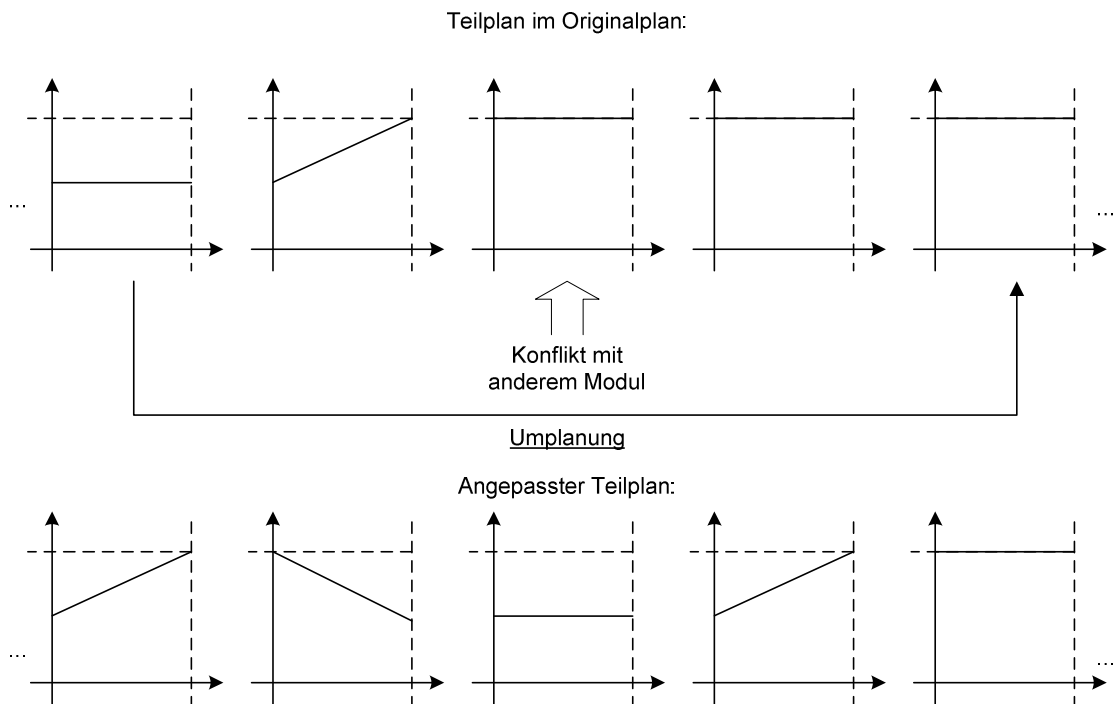


Abbildung 5-5: Mögliche Plananpassung im Planungsproblem des Antriebs

Das Ersetzen einer Teilfunktionslösung  $tfl$  durch eine alternative Teilfunktionslösung  $tfl_a$  ist auf zwei aufeinander aufbauende Planungsprobleme abzubilden. Zunächst muss von einem Vorgängerzustand (ggf. der direkte Vorgänger) ein Plan derartig gestaltet werden, dass die Vorbedingungen von  $tfl_a$  erfüllt sind. Dieser Plan sollte eine kurze Ausführungsdauer besitzen, um den resultierenden koordinierten Gesamtplan nicht unnötig zu verlängern und so einen zeitlichen Verzug beim Erreichen des angestrebten Zustands zu vermeiden. Der so entstandene neue Plan ist durch die Aktivität, die die Teilfunktionslösung  $tfl_a$  ausführt, zu ergänzen. Diese Ergänzung ergibt den Folgezustand von  $tfl_a$ , der den Ausgangszustand des zweiten Planungsproblems darstellt. Der Folgezustand von  $tfl_a$  muss überführt werden in einen Zustand, der bereits im ursprünglichen Plan enthalten ist. So entsteht ein neuer lokaler Plan, der die gewünschte Ersetzung der Teilfunktionslösung berücksichtigt. Abbildung 5-6 veranschaulicht die *Einbettung* der Alternativen der neuen Teilfunktionslösung in den bestehenden Plan.

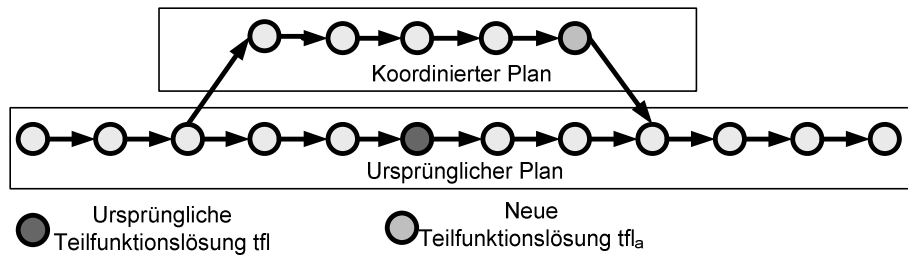


Abbildung 5-6: Einbettung einer alternativen Teilfunktionslösung

Beim Verschieben einer Teilfunktionslösung aus einem Zeitfenster kann das Planungsproblem so modifiziert werden, dass alle Grundaktionen des PDDL-Planungsproblems (vergleiche Unterkapitel 2.1.1) entfernt werden, die einer Anwendung der Teilfunktionslösung in diesem Zeitfenster entsprechen. Eine weitere Alternative wäre die Einführung von Strafkosten (Reduzierung des Nutzens), um eine Verschiebung aus dem Zeitfenster zu erreichen. Lässt sich der Nutzen, den eine Koordinationsaktion erbringt, genau beziffern und in die Formulierung des PDDL-Planungsproblems integrieren, reicht es aus, um den Bereich des Zeitfensters herum eine Neuplanung durchzuführen. Kann die Verschiebung der Aktivität eine Verbesserung bewirken, wird der Planer aufgrund der Zielfunktion die Aktivität aus dem Zeitfenster verschieben. Abbildung 5-7 veranschaulicht den Planungsvorgang.

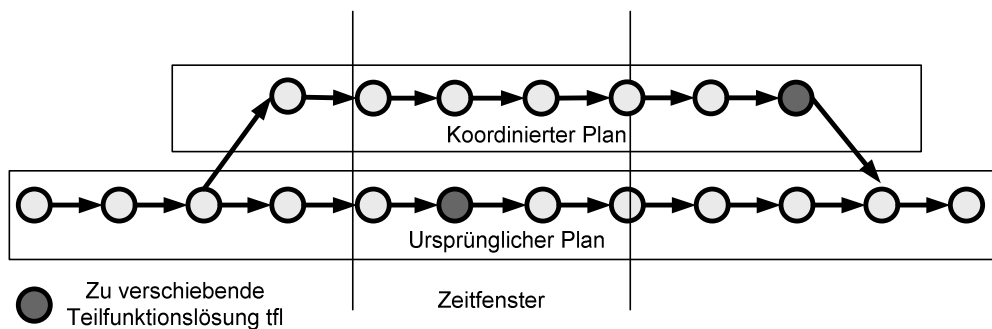


Abbildung 5-7: Planung beim Verschieben einer Teilaktion aus einem Zeitfenster

Ganz ähnlich lässt sich mit der Verschiebung in ein Zeitfenster umgehen. Die Planung muss einen Bereich vor der ursprünglichen Aktivität und einen Zielzustand im Bereich nach dem Zeitfenster umfassen. Abbildung 5-8 veranschaulicht den Planungsvorgang.

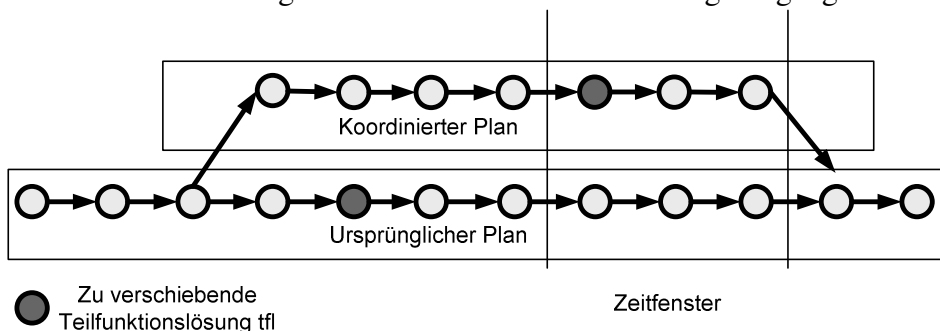


Abbildung 5-8: Planung beim Verschieben einer Teilaktion in ein Zeitfenster

Grundlage für die vorgestellte Abbildung von Koordinationsaktionen auf PDDL-Planungsprobleme ist eine Integration der Wechselwirkung in die verwendeten Planungsmetriken, um den Standardplaner in gewünschter Weise zu steuern.

## 5.2.2 Blackbox-Modellierung der Wechselwirkungen zwischen Teilsystemen

Ziel des in diesem Kapitel entwickelten Verfahrens ist die Koordination der Wechselwirkungen zwischen Teilsystemen. Es soll den Teilsystemen ermöglichen, Konflikte zu vermeiden und Synergien verstärkt zu realisieren. Dazu ist zunächst eine Modellierung dieser Wechselwirkungen notwendig. Sie stellt das Aufgabenobjekt der Koordination dar. Sie muss den Teilsystemen die notwendigen Informationen liefern, um Möglichkeiten zur Durchführung einer der zuvor vorgestellten Koordinationsaktionen zu ermitteln. Dabei erfolgt die Modellierung in zwei Schritten. Für die Koordination vor der Planung ist eine zeitunabhängige Darstellung (es liegen keine Informationen über die zeitliche Abfolge von Aktivitäten anderer Systeme vor) und für die Koordination nach der Planung eine zeitabhängige Darstellung (es liegen genaue Informationen über die zeitliche Abfolge vor) notwendig. Die Informationen, die die Modellierung umfassen muss, sind vielfältig. Zunächst muss dargestellt werden, ob und in welchem Maße Wechselwirkungen zwischen Teilfunktionslösungen unterschiedlicher Teilsysteme bestehen. Um die Relevanz dieser Wechselwirkungen zu beurteilen, ist es notwendig, die Häufigkeit der Anwendung von Teilfunktionslösungen zu kennen. Wechselwirkungen mit einer Teilfunktionslösung sind weniger relevant, wenn diese Teilfunktionslösung selten angewandt wird. Und schließlich muss ein Situationsbezug hergestellt werden, insbesondere hinsichtlich des Zielsystems, das mit der Hauptaufgabe verbunden ist. Die Beurteilung von Teilfunktionslösungen sowie von Wechselwirkungen muss das jeweilige Zielsystem berücksichtigen.

Zur Darstellung dieser Informationen wird auf Elemente zweier in Kapitel 5.1 vorgestellter Ansätze zurückgegriffen: Generalized Partial Global Planning (siehe 5.1.3.2.3) und die Satisficing Game Theory (siehe 5.1.3.3.3). GPGP definiert eine Menge von Relationen zwischen Systemaktionen, die zur Koordination von Teilsystemen genutzt werden. Die Satisficing Game Theory führt abhängige Nutzenfunktionen ein, deren Struktur zur Darstellung der Abhängigkeiten vom Zielsystem und der Anwendungshäufigkeit der Teilfunktionslösungen genutzt werden kann. Weiterhin wird in der Satisficing Game Theory die Betrachtung des Nutzens einer Handlungsalternative aufgespalten in eine Nutzen- und Kostenbetrachtung. Eine ähnliche Aufspaltung erfolgt auch hier, wobei nach lokalen und globalen Nutzen einer Handlungsalternative differenziert wird. Nutzen ist dabei immer als ein Beitrag zur Erfüllung des Zielsystems zu verstehen. Dieser zusammengesetzte Nutzen wird in der lokalen Planung berücksichtigt, wofür das Mittel der Planmetrik der PDDL eingesetzt wird.

### 5.2.2.1 Zeitunabhängige Modellierung der Wechselwirkung

Das Zielsystem wird mithilfe einer Menge  $\Omega$  von Zielsymbolen dargestellt. Ein Zielsymbol repräsentiert dabei ein verbal beschriebenes Ziel, das das Gesamtsystem verfolgen kann. Ein Zielsymbol ist nicht zu verwechseln mit einer Zielfunktion, die eine zu minimierende oder maximierende Funktion ist. Ein Zielsymbol wirkt sich auf die Ausformulierung der vom System verfolgten Zielfunktion aus, wenn das entsprechende Ziel verfolgt wird. Ziele aus dem einleitenden Beispiel (Kapitel 1) sind die Fahrgeschwindigkeit (G), der Luftspalt (L), der Komfort (K) und der Energieverbrauch (E), die sich im RailCab während der Planung ergeben.

Um die verfolgten Ziele an die jeweilige Situation, Aufgaben und Vorgaben anpassen zu können, muss es möglich sein, die Zielsymbole zu gewichten. Dazu dient eine Funktion  $weight: \Omega \rightarrow [0,1]$ . Ohne Beschränkung der Allgemeinheit soll gelten:

$$\sum_{\omega \in \Omega} weight(\omega) = 1$$



Neben der Gewichtung der Ziele ist auch der Einfluss der Ziele auf Teilfunktionslösungen, die ein Teilsystem auswählt, relevant. Dabei wird dieser Einfluss zunächst bestimmt, ohne Wechselwirkungen mit anderen Systemen zu berücksichtigen. Dazu wird ermittelt, wie oft eine Teilfunktionslösung  $tfl$  (z. B. ein bestimmtes Fahrprofil des Antriebs) unter bestimmten Umweltbedingungen  $U$  (die sich aus den Vorbedingungen der Teilfunktionslösung ergeben) angewandt wurde, wenn ein Ziel  $\omega$  (z. B. die Geschwindigkeit des RailCabs) verfolgt wurde. Bei der Ermittlung dieser bedingten Häufigkeit muss zusätzlich die jeweilige Gewichtung des Ziels  $\omega$  berücksichtigt werden:

- $$\theta(tfl|\omega, u) = \frac{\sum_{(tfl|\omega_i, u)} weight(\omega_i)}{\sum_{\omega_i} weight(\omega_i|u)}$$

Dabei bezeichnet die Laufvariable  $\omega_i$  alle Fälle, in denen das Zielsymbol  $\omega$  mit einer Gewichtung höher als 0 verfolgt wurde und  $(tfl | \omega_i)$  alle Fälle, in denen dann die Teilfunktionslösung  $tfl$  angewandt wurde. Mit  $\theta(tfl | \omega_i)$  ist ein erster Zusammenhang zwischen dem Zielsystem und dem Auftreten einer Teilfunktionslösung hergestellt. Das mögliche Auftreten einer Teilfunktion kann auf dieser Basis anhand eines gegebenen Zielsystems beurteilt werden als:

- $$\theta(tfl|\omega_1, \dots, \omega_n, u) = \sum_{\omega \in \Omega} weight(\omega) \cdot \frac{\sum_{(tfl|\omega_i, u)} weight(\omega_i)}{\sum_{\omega_i} weight(\omega_i|u)}, \text{ kurz } \theta(tfl)$$

Falls ein ausreichend großer Satz an historischen Daten vorliegt, kann die Häufigkeit  $\theta(tfl)$  auch direkt aus dem gegebenen Zielsystem abgeleitet werden. Die Funktion  $\theta(tfl)$  liefert eine Abschätzung darüber, wie häufig eine bestimmte Teilfunktionslösung bei gegebenem Zielsystem angewandt wird.

Eine weitere wichtige Information zur Koordination der lokalen Planungen ist die Darstellung und Quantifizierung der Wechselwirkung zwischen den Teilfunktionslösungen unterschiedlicher mechatronischer (Teil-)Systeme. Grundlage dieser Quantifizierung stellt eine Evaluationsfunktion ( $E$ ) für jedes der Zielsymbole dar. Diese Evaluationsfunktion ordnet jedem möglichen Zustand eine Bewertung zu, die zwischen 0 und 1 liegt:

- $$E: \Omega \rightarrow [0,1]$$

Diese Beschränkung der Evaluationsfunktion ist notwendig, damit sämtliche Zielbewertungen gleich skalieren, wodurch eine Gewichtung der Ziele überhaupt erst sinnvoll möglich wird. In der Regel sind solche Evaluationsfunktionen einfach zu konstruieren. Für das Ziel Geschwindigkeit (das maximiert werden soll) legt man zum Beispiel die durchschnittliche Geschwindigkeit  $v_\emptyset$  zugrunde und normiert diese mithilfe der maximalen Geschwindigkeit. Für den zu minimierenden Luftspalt die durchschnittliche Abweichung ( $\Delta ls$ ) des tatsächlichen Luftspalts vom technisch möglichen oder sinnvollen maximalen Luftspalt:

- $$E(G) = \frac{v_\emptyset}{v_{max}}, E(L) = \frac{\Delta ls_\emptyset}{ls_{max}}$$

Um diese Information für die Planung nutzbar zu machen, muss ein Zusammenhang zwischen den Zustandsvariablen und damit den herrschenden Umwelteinflüssen im Plan (den Vorbedingungen) und der Evaluationsfunktion hergestellt werden:

- $$E_{tfl}^{\omega, u}: \Omega \times TFL \times U \rightarrow [0,1]$$

Da diese Bewertung des Nutzens von Teilfunktionslösungen im Vorfeld der Planung erfolgt, also bevor genaue Informationen über das Verhalten der übrigen Teilsysteme vorliegen, wird

der durchschnittliche Beitrag der Teilfunktionslösung  $tfl$  zum Ziel  $\omega$   $X_u^\omega(tfl)$  zur Beurteilung herangezogen. Der Wert  $X_u^\omega(tfl)$  wird dann verwendet, um Wirkzusammenhänge zwischen unterschiedlichen Teilfunktionslösungen zu bemessen. Da  $X_u^\omega$  den sofortigen Beitrag der Teilfunktionslösung bemisst, ist es ausreichend, sich auf solche Anwendungen anderer Teilfunktionslösungen zu beschränken, die vor oder gleichzeitig mit der betrachteten Teilfunktionslösung  $tfl$  angewendet werden. Um zu bestimmen, ob ein Wirkzusammenhang zwischen zwei Teilfunktionen  $tfl$  und  $tfl'$  besteht, wird der Beitrag herangezogen, den  $tfl$  zur Erreichung eines Ziels  $\omega$  liefert, wenn zuvor oder gleichzeitig mit  $tfl$  die Teilfunktion  $tfl'$  ausgeführt wird (geschrieben als  $X_u^\omega(tfl|tfl')$ ).

Wenn also beispielsweise der Antrieb auf einem Streckenabschnitt mit ungenau verlegten Statoren (U) ein Fahrprofil ( $tfl'$ ) wählt, das sehr schnell zu einer hohen Endgeschwindigkeit führt, erreicht die Luftspaltverstellung mit der Betriebsstrategie *kontinuierliche Verstellung* ( $tfl$ ) ihr Ziel Minimierung des Luftspalts ( $\omega$ ) wesentlich schlechter als im Durchschnitt. Umgekehrt führt die Anwendung eines Fahrprofils mit niedriger Durchschnittsgeschwindigkeit dazu, dass die Luftspaltverstellung ihr Ziel besser erreicht.

Bei der Bestimmung von  $X_u^\omega(tfl|tfl')$  kann zusätzlich zur reinen Reihenfolgebetrachtung auch noch ein zeitlicher Abstand zwischen den Anwendungen der Teilfunktionslösungen herangezogen werden. Dadurch können Fälle, in denen ein zeitlich großer Abstand gegeben ist und daher eine tatsächliche Wechselwirkung als unwahrscheinlich oder vernachlässigbar schwach unterstellt werden kann, aus der Bewertung genommen werden.

Anhand des durchschnittlichen Beitrags  $X_u^\omega(tfl)$  kann abgeschätzt werden, ob  $tfl'$  eine positive oder negative Wirkung auf  $tfl$  hat. Zur Abschätzung dient die Differenz der Werte von  $X_u^\omega(tfl|tfl')$  und  $X_u^\omega(tfl)$ , bezeichnet mit  $X_\Delta^\omega(tfl|tfl')$ . Ist  $X_\Delta^\omega(tfl|tfl')$  positiv (negativ), liegt eine positive (negative) Beeinflussung vor. Falls  $X_\Delta^\omega(tfl|tfl')$  gleich 0 ist, liegt im Mittel keine Beeinflussung vor. Da die *Evaluationsfunktion* auf den Wertebereich zwischen 0 und 1 normiert ist, sind der Beitrag und auch der durchschnittliche Beitrag auf den Wertebereich von -1 und 1 normiert. Damit ist auch die Abweichung von  $X_u^\omega(tfl|tfl')$  gegenüber  $X_u^\omega(tfl)$  beschränkt. Im Generalized Partial Global Planning wird der Begriff der *Begünstigung* verwendet, wenn die abgestimmte Ausführung einer Aktion  $a'$  die Dauer der Durchführung verkürzt oder die Qualität des Ergebnisses einer Aktion  $a$  verbessert. Da auch die Dauer der Durchführung ein mögliches Zielsymbol darstellt, reicht es für den hier beschriebenen Ansatz aus, die Auswirkung der Ausführung einer Teilfunktionslösung  $tfl'$  auf die Zielerreichung einer anderen Teilfunktionslösung  $tfl$  zu betrachten. Damit stellt  $X_\Delta^\omega(tfl|tfl')$  eine Verallgemeinerung der Begünstigung aus dem GPGP dar.  $X_\Delta^\omega(tfl|tfl')$  wird im weiteren Verfahren nicht als skalarer Wert wie  $X_u^\omega(tfl)$  behandelt, sondern als eine Häufigkeitsverteilung über dem Wertebereich von +1 bis -1. Nach frequentistischer Auffassung nähert sich diese Häufigkeitsverteilung einer Wahrscheinlichkeitsverteilung über die Wechselwirkung zwischen den Teilfunktionslösungen an. Die Häufigkeitsverteilung wird bzgl. eines Ziels  $\omega$  als  $H_u^\omega(tfl|tfl')$  bezeichnet. Gemäß dieser Interpretation von  $H_u^\omega(tfl|tfl')$  und der Häufigkeitsverteilung über die Teilfunktionslösungen  $\Theta(tfl)$  kann der globale Nutzen einer Teilfunktionslösung abgeschätzt werden. Dabei entspricht der globale Nutzen der Summe der Wechselwirkungen auf andere Teilfunktionslösungen. Für ein gegebenes Ziel  $\omega$  ergibt sich damit der globale Nutzen einer Teilfunktionslösung  $tfl'$  bzgl.  $\omega$  als:

- $N_u^\omega(tfl') = \sum_{tfl \in TFL} \Theta(tfl) \cdot H_u^\omega(tfl|tfl')$

Wie bereits bei  $H_u^\omega(tfl|tfl')$  handelt es sich auch bei  $N_u^\omega(tfl')$  um eine Häufigkeits- bzw. Wahrscheinlichkeitsverteilung über den Wertebereich -1 bis 1. Diese Funktion bewertet nun beispielsweise, wie ein bestimmtes Fahrprofil hinsichtlich seiner Auswirkung auf das Ziel Luftspalt zu beurteilen ist. Dazu wird über die Teilfunktionslösungen aller anderen Teilsysteme

me die Wechselwirkung zwischen Teilfunktionslösungen betrachtet und mit der Anwendungshäufigkeit der jeweiligen Teilfunktionslösung gewichtet. Ebenso sind andere Berechnungen von  $N_u^\omega(tfl')$  vorstellbar. So könnten insbesondere *robuste* Aktionen bevorzugt werden, indem die schlechteste Wechselwirkung über alle Teilfunktionslösungen zur Berechnung herangezogen wird. Das würde einer risiko-aversen max-min Strategie entsprechen:

$$\bullet N_{max-min}^{\omega,u}(tfl') = \underset{tfl \in TFL}{argmin} H_u^\omega(tfl|tfl')$$

Ebenso können besonders unwahrscheinliche Teilfunktionslösungen herausgefiltert werden. Derartige Teilfunktionslösungen könnten sonst die Lösung verfälschen und es ist stattdessen günstiger, sie im Falle des tatsächlichen Auftretens durch die Koordination nach der Planung (siehe 5.2.4) zu behandeln. Dazu wird ein Schwellenwert  $\alpha$  eingeführt, der alle Teilfunktionslösungen mit einer relativen Anwendungshäufigkeit kleiner  $\alpha$  (z. B. 0,05) herausfiltert:

$$\bullet N_\alpha^{\omega,u}(tfl') = \sum_{tfl \in \{tfl|tfl \in TFL \wedge \theta(tfl) \geq \alpha\}} \theta(tfl) \cdot H_u^\omega(tfl|tfl')$$

Über das gesamte Zielsystem ergibt sich der globale Nutzen in jedem Fall als:

$$\bullet N_u(tfl') = \sum_{\omega \in \Omega} weigh(\omega) \cdot N_u^\omega(tfl'),$$

wobei  $N_u(tfl')$  wiederum eine Wahrscheinlichkeitsverteilung über dem Wertebereich von -1 bis 1 ist. Mit der Bestimmung von  $N_u(tfl')$  ist Folgendes geleistet: Abhängig vom aktuellen Zielsystem kann bestimmt werden, wie günstig oder ungünstig eine Teilfunktionslösung hinsichtlich seiner Wechselwirkungen mit anderen Teilsystemen ist. Dabei wird berücksichtigt, in welchem Umfang Teilfunktionen und Teilfunktionslösungen von anderen Systemen eingesetzt werden. Diese Information kann im lokalen Planungsprozess der Teilsysteme ausgenutzt werden, um gezielt Teilfunktionslösungen mit vorwiegend positiver Wechselwirkung einzusetzen und solche mit vorwiegend negativer Wechselwirkung zu vermeiden. Ein hoher Wert von  $P^0(ts)$  bedeutet, dass das Verhalten des Teilsystems  $ts$  eine starke Wirkung auf das Erreichen des Ziels  $\omega$  besitzt. Für die Teilsysteme und Ziele aus dem Beispiel beträgt der Wert immer 0 oder 1. Das Ziel Geschwindigkeit wird z. B. ausschließlich durch den Antrieb verfolgt und das Ziel Luftspalt ausschließlich durch die Luftspaltverstellung.  $P^0(ts)$  wird später genutzt, um zu steuern, in welchem Umfang ein System seine Ziele lokal verfolgt oder Rücksicht auf andere Systeme nimmt.

Abschließend ist anzumerken, dass die Modellierung der Wechselwirkungen und die Herleitung der relativen Häufigkeiten in einem gewissen Maße *naiv* sind. Wie auch bei naiven Bayes-Modellen werden mögliche Abhängigkeiten, hier zwischen Zielgewichtungen und Teilfunktionslösungen, vernachlässigt. Es werden Wechselwirkungen zwischen jeweils zwei Teilfunktionslösungen betrachtet. Einige Wechselwirkungen können unter Umständen aber erst bei Betrachtung mehrerer Teilfunktionslösungen berücksichtigt werden. Die hier gewählte Modellierung ist jedoch leicht auf eine differenziertere Modellierung umzustellen. Die Verteilung  $H^0(tfl|tfl')$  lässt sich systematisch ohne Weiteres erweitern zu  $H^0(tfl|tfl_1', \dots, tfl_n')$ . Damit würde der Nutzenbeitrag von  $tfl$  abhängig von der Anwendung von  $n$  anderen Teilfunktionslösungen bewertet. Dadurch wird zwar eine höhere Genauigkeit in der Modellierung erreicht, das Erlernen der Parameter (beschrieben in Abschnitt 5.2.2.2.1) wird jedoch erschwert und auch die Koordination im Anschluss an die Planung würde komplexer. Es sei daher an dieser Stelle auf die große Praxisrelevanz von naiven Bayes-Modellen verwiesen, die ebenfalls innerhalb des Modells Unabhängigkeiten unterstellen, die in der Realität nicht gege-

ben sind. Es ist davon auszugehen, dass Abhängigkeiten zwischen Zielen und zusätzlichen Teilfunktionslösungen sich vielfach gegenseitig aufheben und daher die Unabhängigkeitsannahme eine ausreichend gute Näherung ergibt.

### 5.2.2.2 Zeitabhängige Modellierung der Wechselwirkung

Um ein möglichst gutes Koordinationsergebnis zu erzielen, ist es erforderlich, sämtliche verfügbaren Informationen zu nutzen. Im Anschluss an die lokale Planung liegen detaillierte Informationen über die zeitliche Abfolge der Aktivitäten vor. Daher ist eine Erweiterung der Modellierung dahingehend notwendig, dass Wechselwirkungen auch bzgl. der genauen zeitlichen Abfolge von einzelnen Aktivitäten betrachtet werden.

Mit diesen Informationen besteht die Möglichkeit, die Kosten und den Nutzen einer Aktion wesentlich differenzierter zu beurteilen, da die Kosten- und Nutzenwirkung einzelner Aktionen abhängig von den gegebenen Plänen der übrigen Teilsysteme bestimmt werden kann. Dazu wird der gesamte Planungshorizont bzgl. einer Teilfunktionslösung in  $t$  Zeitabschnitte eingeteilt. Ausgehend vom Beginn des Plans bis zum Ende können die Zeitabschnitte durchnummeriert werden. Für jeden dieser Zeitabschnitte wird bestimmt, wie gut oder schlecht der Einsatz der Teilfunktionslösung  $tfl$  an dieser Stelle ist. Um möglichst genaue Kostenwerte für die Zeitintervalle zu erhalten, ist die Analyse der Pläne der  $n-1$  anderen Teilsysteme notwendig. Für jede Teilfunktionslösung  $tfl_{ts}$ , die im Plan eines anderen Teilsystems  $ts$  auftritt, lässt sich anhand der Häufigkeitsverteilung  $H_u^\omega(tfl_{ts}|tfl)$  bestimmen, welche Wirkung von  $tfl$  zu erwarten ist, wenn sie im Zeitabschnitt  $i$  ausgeführt wird. Indem diese Kosten für den Zeitabschnitt  $i$  über alle Teilfunktionslösungen der Pläne aller Teilsysteme gewichtet summiert wird, ergibt sich eine zeitabhängige Kostenabschätzung für den Einsatz einer Teilfunktionslösung. Diese Abschätzung lässt sich im Rahmen der Synchronisation der einzelnen Teilpläne nutzen, um die lokalen Anpassungen an den Plänen zu steuern. Die Berechnung des Nutzens für die Ausführung einer Aktion im Zeitabschnitt  $i$  bestimmt sich durch:

$$N(tfl_i) = \sum_{ts \in TS} \sum_{\omega \in \Omega} weigh(\omega) \cdot P^\omega(ts) \sum_{tfl \in TFL_i} \alpha \cdot H_u^\omega(tfl|tfl_i)$$

Der globale Nutzen einer Teilfunktionslösung  $tfl$  in einem Zeitabschnitt  $i$  wird also durch die Summierung der Wirkung auf die Aktivitäten der übrigen Teilsysteme bestimmt. Dabei werden Aktivitäten, die nur teilweise in das Intervall  $i$  fallen, mit dem Anteil  $\alpha$  gewichtet, zu dem sie in das Intervall fallen. Daran wird die Flexibilität dieses Vorgehens deutlich, da die Länge der Planungsschritte der einzelnen Teilmodule völlig unabhängig voneinander definiert werden kann. Damit ist es jedem Teilsystem möglich, Änderungen an seinem Plan anhand der Wirkung auf die konkreten Pläne der übrigen Teilsysteme zu bewerten. Die so ermittelten zeitabhängigen Nutzenwerte können durch das Konstrukt der bedingten Aktion leicht in einem PDDL-Planungsproblem berücksichtigt werden. Listing 5-1 zeigt ein Beispiel für eine solche Aktionsdefinition.

Der Effektteil der Aktion bewirkt, dass die Anzahl der Planungsschritte um eins erhöht wird. Im bedingten Teil der Aktionen wird für jeden im Plan vorkommenden Schritt der zu dem Zeitpunkt von den Aktivitäten der anderen Teilsysteme abhängende Nutzenwert zum Gesamtnutzenwert des Plans addiert.

```

(:action driveFast
  ...
  ...
  :effect
    ...
    ...
    (increase (plan-steps) (1))
    when( (= (plan-steps) (1)) (increase total-
      utility(utitily_driveFast_in_Step_1)
    when( (= (plan-steps) (2)) (increase total-
      utility(utitily_driveFast_in_Step_1)
      ...
    when( (= (plan-steps) (100)) (increase total-
      utility(utitily_driveFast_in_Step_1)
  )

```

**Listing 5-1: Bedingte Aktion mit zeitabhängigem Nutzenwert**

Weiterhin können die Systeme mithilfe von  $H_u^\omega(tfl_{ts}|tfl)$  die zeitliche Verschiebung einer Aktivität qualitativ beurteilen. Um den Planungsaufwand, der durch die Koordination entsteht, zu beschränken, wird nur die lokale Koordination zwischen zwei Aktionen bewertet. Aufgrund des Austauschs über die Wechselwirkungen der Aktionen können die Teilsysteme die möglichen positiven Veränderungen auch beziffern. Dazu ist teilweise Kenntnis über die Fähigkeiten der übrigen Teilsysteme notwendig, aber nur insoweit, dass einzelne Aktivitäten einer Teilfunktion zugeordnet werden. Die mögliche Wirkung einer Plananpassung auf ein einzelnes Ziel kann dann wie folgt abgeschätzt werden:

1. bei Ersetzen einer Teilfunktionslösung  $tfl'$  durch  $tfl''$ :  

$$H_u^\omega(tfl|tfl'') - H_u^\omega(tfl|tfl')$$
2. Bei Verschieben einer Aktivität in ein anderes Zeitfenster oder aus einem Zeitfenster heraus:  

$$H_u^\omega(tfl|tfl')$$

Die so ermittelten Werte können zur Bestimmung einer Rangfolge des Koordinationspotenzials einzelner lokaler Pläne genutzt werden.

#### 5.2.2.2.1 Bestimmung der Modellelemente für eine konkrete Anwendung

In diesem Abschnitt wurde bisher ein formales Modell der Wechselwirkungen zwischen Teilfunktionslösungen unterschiedlicher mechatronischer Systeme vorgestellt. Unbeantwortet ist noch die Frage, woher die notwendigen Informationen stammen. Dabei sind grundsätzlich zwei Ansätze denkbar:

1. Vorgabe aller relevanten Informationen durch einen menschlichen Experten.
2. Erlernen der relevanten Informationen im laufenden Betrieb.

Beide Ansätze haben ihre spezifischen Vor- und Nachteile. So ist die Vorgabe durch einen Experten problematisch, da sie vollständig von der Qualität des Expertenwissens abhängt. Außerdem fällt es Experten oft schwer, ihr Wissen explizit und formal auszuformulieren. Darüber hinaus sind die Informationen statisch und können sich nicht an geänderte Situationen anpassen. Auf der anderen Seite steht die Information den Teilsystemen sofort zur Verfügung und kann unmittelbar zur Verbesserung der Planung verwendet werden. Genau dies ist bei Lernverfahren nicht möglich. Diese müssen zunächst über eine ausreichende Menge an Trainingsdaten verfügen, um brauchbare Ergebnisse zu erzielen. Der Nutzen der Interaktionsplanung steht also verzögert zur Verfügung. Ideal wäre eine Integration der beiden Ansätze,

wobei ein maschinelles Lernverfahren die Expertenangaben mit den Messungen im Betrieb abgleicht und ggf. korrigiert.

Um ein entsprechendes Verfahren zu konstruieren, ist es zunächst notwendig, die zu erlernenden Daten und Informationen genauer zu betrachten. Direkt in der Planung verwandt wird der globale Nutzenwert, den eine Teilfunktionslösung erzielt. Dieser Wert wird situations- und zielabhängig aus einer Menge von anderen Parametern hergeleitet. Die folgende Aufzählung fasst die notwendigen Parameter noch einmal zusammen:

- $\theta(tfl, \omega, u)$ : Wie häufig wird eine Teilfunktionslösung angewandt, wenn ein bestimmtes Ziel unter bestimmten Umgebungseinflüssen verfolgt wird?
- $\theta(tfl | \omega_1, \omega_2, \dots, \omega_n, u)$ : Wie häufig wird eine Teilfunktionslösung in einem Zustand angewandt, wenn ein bestimmtes Zielsystem unter bestimmten Umgebungseinflüssen verfolgt wird?
- $X_u^\omega(tfl)$ : Wie gut erfüllt eine Teilfunktion ein bestimmtes Ziel?
- $H_u^\omega(tfl | tfl')$ : Wie gut erfüllt eine Teilfunktion  $tfl$  ein bestimmtes Ziel, wenn auch die Teilfunktion  $tfl'$  ausgeführt wird?
- $P^\omega(ts)$ : Wie wichtig ist ein Teilsystem zur Erreichung eines Ziels?

Die unterschiedlichen Parameter sind zunächst unterschiedlich gut zur Ermittlung durch Experten oder Lernverfahren geeignet. Ein Experte wird in der Regel mit der Gewichtung der Bedeutung der Teilsysteme  $P^\omega(ts)$  nicht überfordert sein, während die übrigen Werte schwieriger anzugeben sind. Bei allen anderen Parametern außer  $P^\omega(ts)$  handelt es sich um statistische Werte, sodass die statistischen Lernverfahren, die in Abschnitt 4.1.2.1 vorgestellt wurden, geeignet scheinen. Die dort vorgestellten Verfahren eignen sich zum Erlernen von Parametern in Bayes-Netzwerken. Daher wird an dieser Stelle nun aufgezeigt, dass alle hier verwandten Planungsparameter sich in Form von Bayes-Netzwerken darstellen lassen. Tabelle 5-1 stellt die zu erlernenden Parameter und eine entsprechende Modellierung als Bayes-Netzwerk gegenüber.

Dabei werden die zielabhängigen und die zielsystemabhängigen Parameter hinsichtlich der Teilfunktionslösungen durch das gleiche Netzwerk abgebildet. Das Netzwerk stellt grundsätzlich die relative Häufigkeit der Anwendung in Abhängigkeit vom gesamten Zielsystem dar – der Parameter für eine einzelne Zielgröße lässt sich durch Summierung und Gewichtung all derjenigen Fälle ableiten, in denen das Ziel mit einer bestimmten Gewichtung verfolgt wurde. Die Variable  $X_u^\omega$  lässt sich als Erwartungswert von  $E(tfl)$  aus dem Netzwerk von  $H_u^\omega(tfl | tfl')$  berechnen. Dieses Netzwerk beschreibt, wie gut unter gegebenen Umweltbedingungen und dem Vorliegen einer bestimmten gleichzeitigen Teilfunktionslösung  $tfl'$  die Teilfunktionslösung  $tfl$  ihr jeweiliges Ziel erreicht.

Allen Netzwerken ist gemein, dass sie in der Struktur einem naiven Bayes-Klassifikatoren entsprechen. Der Einsatz eines solchen Klassifikators zum Erlernen der Verteilungen wäre ein äußerst gradliniger Ansatz. Jedoch liefern Maximum Likelihood Ansätze wie die naiven Bayes-Klassifikatoren ungenaue Ergebnisse bei wenigen Trainingsdaten. Daher würde die Planung zunächst vermutlich ein eher schlechtes Verhalten erzielen. Ein Verfahren wie das Lernen mit Dirichlet-Verteilungen, das die Berücksichtigung von Expertenwissen ermöglicht, ist klar vorzuziehen (siehe Abschnitt 4.1.2.1).

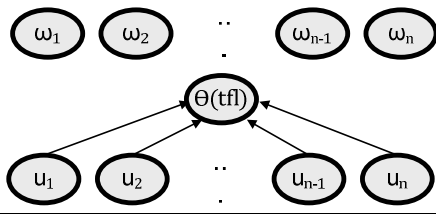
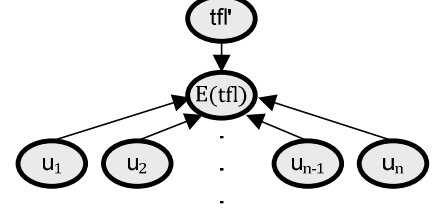
Zu lernender Parameter	Netzwerkstruktur
$\theta (tfl \omega)$	
$H_u^\omega (tfl tfl')$ :	

Tabelle 5-1: Interaktionsplanungsparameter als Bayes-Netzwerke

### 5.2.3 Koordination vor der Planung

Die im vorigen Absatz hergeleiteten und beschriebenen globalen Wirkungen von Teilfunktionslösungen müssen im lokalen Planungsprozess der Teilsysteme berücksichtigt werden, so dass möglichst wenige Konflikte und viele Synergieeffekte im Zusammenspiel der Pläne entstehen. Dazu muss zunächst zwischen den lokalen Interessen eines Teilsystems und der globalen Wirkung seines Verhaltens abgewogen werden.

Lokal verfolgen alle Teilsysteme eine ähnliche Zielfunktion, die mit einer entsprechenden Planmetrik dem lokalen Planer vorgegeben wird. Es handelt sich dabei um die gewichtete Summe der Gütefunktionen  $E(\omega)$ . Die Planmetrik hat also zunächst die Form:

- $\sum_{z \in P} \sum_{\omega \in \Omega} weight(\omega) \cdot E_{tfl}^\omega$ :

Diese rein lokale Zielfunktion muss um eine globale Komponente erweitert werden, die die Wechselwirkungen zwischen den Teilfunktionslösungen der Systeme berücksichtigt. Ansatzpunkt für diese Erweiterung ist wieder die Menge der gemeinsamen Ziele  $\Omega$  mit der entsprechenden Gewichtung.

Ein niedriger Wert von  $P^\omega(ts)$  deutet an, dass das Teilsystem  $ts$  für  $\omega$  von nachrangiger Bedeutung ist und vorzugsweise auf andere Teilsysteme Rücksicht nehmen sollte. Die Planmetrik wird daher erweitert und umgestellt zu folgender Form:

- $\sum_{z \in P} \sum_{\omega \in \Omega} (w(\omega) \cdot E_{tfl}^\omega) + (\sum_{\omega \in \Omega} w(\omega) \cdot N)$ ; wobei  $w(\omega) = P^\omega(ts) \cdot weigh(\omega)$

Der Parameter  $N$  steht für die globale Wirkung des Systemverhaltens. Diese Bildung der Planmetrik bewirkt, dass diejenigen Ziele direkt verfolgt werden, auf die das betrachtete Teilsystem eine besonders große Wirkung hat. Hat ein Teilsystem geringen oder gar keinen Einfluss auf ein Ziel, wird diesem Ziel in der lokalen Planung ein entsprechend geringes Gewicht zugewiesen. Weiterhin ergibt sich, dass ein Teilsystem mit großem Einfluss auf die Erfüllung der globalen Ziele sich tendenziell egoistisch verhält. Hat ein Teilsystem kaum direkten Einfluss auf das Zielsystem, ist das Gegenteil der Fall: Es vernachlässigt die lokale Zielverfolgung und verhält sich vor allem gemeinnützig. Dies ist im Sinne des Gesamtsystems mit seiner globalen Zielstellung im Allgemeinen ein sinnvolles Verhalten, wenn auch keine Optimalität garantiert wird. Im Wesentlichen entspricht diese Bildung der Zielfunktionen einem einfachen Greedy- oder heuristischen Verfahren.

Während die Bewertung der Zielerreichung anhand der Evaluationsfunktionen  $E_{tfl}^\omega$  bereits klar ist, muss die Bestimmung des globalen Systemverhaltens näher beleuchtet werden. Ank-

nüpfungspunkt sind wieder die einzelnen Teilfunktionslösungen mit den im vorigen Kapitel beschriebenen und hergeleiteten Wechselwirkungen. Zusätzlich wird an dieser Stelle noch betrachtet, welches Teilsystem die beeinflussten Teilfunktionslösungen ausführt. Ein direkter Ansatz bestände darin, den eben eingeführten globalen Nutzen zur Bestimmung heranzuziehen. Dabei würden aber die unterschiedlichen Gewichtungen der Teilsysteme berücksichtigt werden und es würde auf wichtige und unwichtige Teilsysteme im gleichen Maße Rücksicht genommen. Stattdessen wird an dieser Stelle der globale Nutzen so erweitert, dass er für das betrachtete Planungsproblem geeignet ist:

$$\bullet \quad N_u(tfl') = \sum_{\omega \in \Omega} \sum_{tfl \in TFL} weight(\omega) \cdot \Theta(tfl) \cdot P^\omega(ts_{tfl}) \cdot H_u^\omega(tfl|tfl')$$

In dieser Form beachtet die globale Nutzenfunktion einer Teilfunktionslösung alle wesentlichen Informationen über das Planungsproblem:

- die Gewichtungen innerhalb des aktuellen Zielsystems  $weigh(\omega)$ ,
- die Anwendungshäufigkeit von Teilfunktionslösungen anderer Teilsysteme  $\Theta(tfl)$ ,
- die Bedeutung anderer Teilsysteme  $P^\omega(ts_{tfl})$ <sup>48</sup>
- und letztlich die Richtung und Stärke der Wechselwirkungen mit den Aktivitäten anderer Teilsysteme  $H_u^\omega(tfl|tfl')$ .

Indem dieser Nutzenwert in die Planungsmetrik der lokalen Planung eingebunden wird, wird die lokale Planung so gesteuert, dass sich ein Teilsystem nur in Fällen, in denen es sinnvoll erscheint, altruistisch verhält. Aufgrund der vielfältigen Informationen ist somit sowohl ein egoistisches Verhalten zum Schaden des Gesamtsystems ausgeschlossen als auch eine überflüssige „Aufopferung“ zugunsten höher priorisierter Teilsysteme.

Durch die Formulierung der Planungsmetriken wird erreicht, dass ernsthafte Konflikte zwischen den Aktivitäten von Teilsystemen mit wesentlich geringerer Wahrscheinlichkeit auftreten. Gleichzeitig werden Situationen begünstigt, in denen Synergieeffekte realisiert werden. Da die lokalen Planungen aber isoliert voneinander erfolgen, sind diese Effekte nicht garantiert. Daher empfiehlt es sich, an die Planung eine Koordination der Pläne anzuschließen. Durch die vorgeschaltete Bestimmung einer globalen Zielfunktion wird der nachträgliche Koordinationsaufwand allerdings voraussichtlich reduziert.

#### 5.2.4 Koordination nach der lokalen Planung

Um die mit Vorliegen der lokalen Pläne bereitstehende zusätzliche Information zur weiteren Koordination der lokalen Pläne zu nutzen, bieten sich, wie bereits im Abschnitt 5.1.3.2 dargestellt, vor allem die Verfahren *Generalized Partial Global Planning* sowie die darauf aufbauende *Shared Activity Coordination* an. Beide Ansätze gehen davon aus, dass zum Zwecke der Koordination Wechselwirkungen zwischen den Aktivitäten der zu koordinierenden Teilsysteme bekannt sind. Diese Voraussetzung ist durch die Modellierung aus Abschnitt 5.2.2 erfüllt. Wie beim GPGP werden daher die lokalen Pläne der Teilsysteme ausgetauscht. Im Gegensatz zum GPGP erfolgt dabei der Austausch immer zwischen allen beteiligten Teilsystemen. Anschließend untersucht jedes Teilsystem die lokalen Pläne der übrigen Systeme auf noch vorliegende Konflikte und mögliche nicht realisierte Synergien. Konflikte und Synergien bezeichnen dabei immer eine Beziehung zwischen genau zwei lokalen Plänen.<sup>49</sup> Diese Form der Koordination der lokalen Pläne unterschiedlicher Systeme kommt nur bei Teilsys-

<sup>48</sup> dabei steht  $ts_{tfl}$  für das Teilsystem, das die Teilfunktionslösung anwendet.

<sup>49</sup> Konflikte und Synergien werden zwar nur zwischen zwei Systemen betrachtet, die Koordination erfolgt aber über die Konflikte und Synergien zwischen allen Paaren von Systemen gleichzeitig.



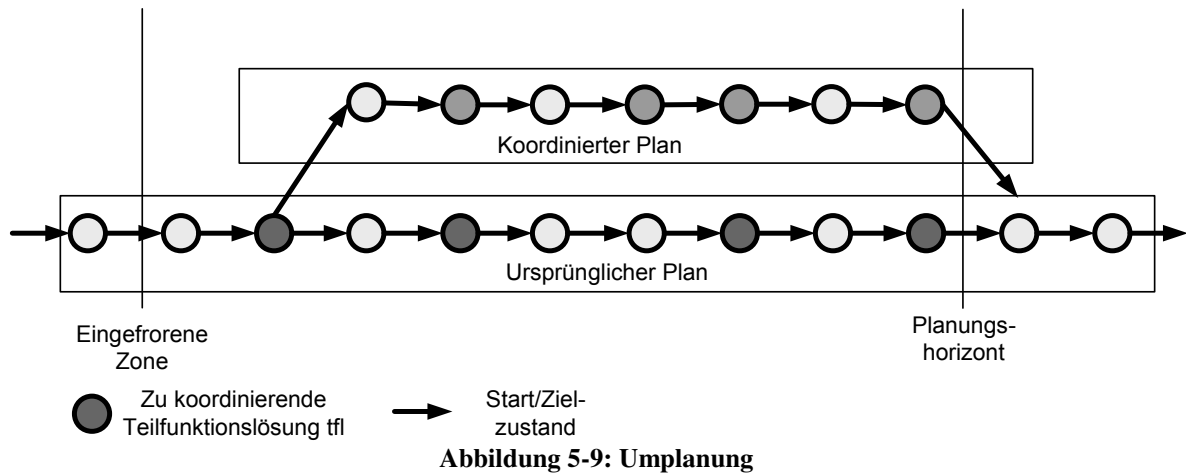
temen mit gemeinsamer Hauptaufgabe in Frage, da sie einen offenen und korrekten Austausch über die lokal angewandten Aktivitäten und die möglichen Wechselwirkungen voraussetzt. Diese *Ehrlichkeit* kann bei Systemen mit individuellen Hauptaufgaben nicht vorausgesetzt werden.

Grundlage der Koordination nach der lokalen Planung sind die in Abschnitt 5.2.2.2 hergeleiteten zeitabhängigen Nutzenwerte einer Teilfunktionslösung  $N(\text{tfl}_i)$ . Damit kann nun auch die Planungsmetrik auf die zeitabhängigen Nutzenwerte umgestellt werden (dabei steht  $z \in P$  für einen Schritt im Plan):

$$\sum_{z \in P} \sum_{\omega \in \Omega} (w(\omega) \cdot E_u^\omega(\text{tfl}_z)) + \sum_{\omega \in \Omega} (1 - w(\omega)) \cdot N_u^\omega(\text{tfl}_z)$$

Mithilfe dieser Funktion kann beispielsweise der Antrieb seine verfügbaren Fahrprofile hinsichtlich seiner lokalen Ziele (Geschwindigkeit) und der Auswirkungen auf die Ziele der Luftspaltverstellung (Größe des einstellbaren Luftspalts) beurteilen. Je nach Gewichtung der Ziele wird ein Fahrprofil als günstig oder ungünstig beurteilt. In Abschnitt 5.2.1 wurde beschrieben, wie die drei Koordinationsaktionen auf PDDL-Planungsprobleme zurückgeführt werden können, sodass sie von einem Standardplaner durchgeführt werden können. Dabei wurde davon ausgegangen, dass bestimmte Zustände aus dem existierenden Plan als Start- und Zielzustand ausgewählt werden. Eine genauere Bestimmung dieser Auswahl wurde nicht durchgeführt, da dafür die Einbettung der Koordination in einen Gesamtablauf erfolgen muss. Im Rahmen dieses Gesamtablaufs muss auch berücksichtigt werden, dass zur Koordination der Pläne nicht unbegrenzt Zeit zur Verfügung steht. Die Shared Activity Coordination (SHAC) liefert mit dem Commit-Window ein Konzept zur Behandlung dieses Problems (vergleiche Abschnitt 5.1.3.2.4). Dieses Konzept entspricht dem einer eingefrorenen Zone, die sich mit der Ausführung immer weiter verschiebt. Dadurch werden immer weitere Teile des Plans fixiert und können durch die Koordination nicht mehr verändert werden. Zusätzlich kann der Planungsaufwand, der durch die Koordination entsteht, weiter eingeschränkt werden, indem man den Planungshorizont beschränkt. Das bedeutet, es wird eine Grenze definiert, die die Neuplanung nicht überschreiten soll. Dadurch wird der Umfang des Planungsproblems reduziert, wodurch schneller angepasste Teilpläne erzeugt werden können. Aufgrund des Fortschreitens der eingefrorenen Zone müssen aber vor allem solche Koordinationsaktionen durchgeführt werden, die sich auf Aktivitäten nahe der eingefrorenen Zone beziehen.

Beide Grenzen der Umplanung eignen sich, um den Ausgangs- und Zielzustand für die Umplanung zu bestimmen. Der Ausgangszustand ist immer der Zustand, der auf die letzte Aktion innerhalb der eingefrorenen Zone folgt, während der Zielzustand derjenige ist, der auf die letzte Aktion innerhalb des Planungshorizonts folgt. Dadurch ist gewährleistet, dass der übrige Plan, der nach dem Planungshorizont folgt, auch nach der Umplanung noch ausführbar ist. Werden Ausgangs- und Zielzustand auf diese Art gewählt, fallen alle in Abschnitt 5.2.1 vorgestellten Planungsprobleme zur Koordination zusammen. Die neue Planungsaufgabe wird dahingehend formuliert, dass eine Sequenz von Teilfunktionslösungen gesucht wird, die vom Startzustand nach der eingefrorenen Zone den Zustand nach dem Planungshorizont herstellt. Da die oben vorgestellte Planungsmetrik die aktuellen Informationen über die Pläne der übrigen Teilsysteme verwendet, wägt diese Planung zwischen der lokalen Zielerreichung und der Koordination mit den übrigen Teilsystemen automatisch ab. Daher werden alle Konflikte und Synergien, die innerhalb des Intervalls zwischen eingefrorener Zone und Planungshorizont vorliegen, simultan betrachtet. Eine wiederholte Umplanung zur Berücksichtigung mehrerer Wechselwirkungen ist nicht notwendig. Abbildung 5-9 illustriert das Umplanungsproblem mit seinem Start- und Zielzustand.



Der Umstand, dass alle möglichen Konflikte und Synergien im Intervall zwischen der eingefrorenen Zone und dem Planungshorizont simultan betrachtet werden, erleichtert auch die Durchführung und Koordination über alle Teilsysteme hinweg. In einem ersten Schritt bewertet jedes Teilsystem das Koordinationspotenzial seines lokalen Plans. Das Koordinationspotenzial wird ermittelt, indem die theoretisch maximal realisierbare Koordination ermittelt wird. Dazu wird für jede Teilfunktionslösung im eigenen Plan ermittelt, wie weit es von der optimalen Abstimmung mit den Teilfunktionslösungen der anderen Teilsysteme entfernt ist. Dabei wird die Untersuchung auf solche Teilfunktionslösungen beschränkt, die im aktuellen Planungshorizont liegen. Das maximal realisierbare Koordinationspotenzial berechnet sich also wie folgt:

$$\sum_{tfl_i \in P_{ts}} \sum_{ts_j \in TS/ts} \sum_{tfl_k \in P_{ts_j}} \sum_{\omega \in \Omega} w(\omega) H_{max}(tfl_k|tfl_i) - H_{akt}(tfl_k|tfl_i)$$

Dabei ist  $H_{max}(tfl_k|tfl_i)$  die bestmögliche Abstimmung zwischen den Teilfunktionslösungen und  $H_{akt}(tfl_k|tfl_i)$  die aktuelle erreichte Abstimmung, jeweils gemessen in der Beeinflussung der beiden Teilfunktionslösungen. Der Wert  $N_{max}(tfl_k|tfl_i)$  ist für jedes Paar  $tfl_k$  und  $tfl_i$  statisch und muss daher nicht für jede konkrete Aktivität neu berechnet werden:

$$N_{max}(tfl_k|tfl_i) = argmax(argmax(H_u^\omega(tfl_k| -tfl_i); argmax(H_u^\omega(tfl_k|tfl_{i*})).$$

Er bestimmt sich als das Maximum aus der bestmöglichen zeitlichen Verschiebung der Teilfunktionslösung  $tfl_i$  ( $tfl_i$  wird in einem anderen Planungsschritt verschoben)<sup>50</sup> und dem besten Wert, der durch den Austausch der Teilfunktionslösung  $tfl_i$  (Fahrprofil B statt Fahrprofil A) mit einer alternativen Teilfunktionslösung  $tfl_{i*}$  erzielt werden kann.

Nach der Ermittlung ihres individuellen Koordinationspotenzials tauschen sich die Teilsysteme über das jeweilige Koordinationspotenzial aus. Das Teilsystem mit dem höchsten Koordinationspotenzial führt eine Umplanung durch. Anschließend ermitteln die Systeme das Koordinationspotenzial neu und der Vorgang wird wiederholt. Dabei wird das Koordinationspotenzial immer für das aktuell gültige Intervall zwischen eingefrorener Zone und Planungshorizont ermittelt.

<sup>50</sup> Ein Beispiel für eine Teilfunktion, die zeitlich verschoben werden kann, ist der Druckaufbau im Hydrauliksystem der aktiven Federung. Dieser kann beispielsweise vorgezogen werden oder durch geringe Federungsaktivität verzögert werden.

### 5.2.5 Berücksichtigung gemeinsamer Ressourcen

Eine besondere Form der Koordination ist erforderlich, wenn die Funktionsmodule eine gemeinsame Ressource wie zum Beispiel einen Energiespeicher teilen. Hier muss sichergestellt sein, dass die individuellen Pläne hinsichtlich des Ressourcenverbrauchs konsistent sind. In diesem Fall bietet es sich an, diese Bedingung durch eine zentrale Instanz sichern zu lassen, die die Konsistenz des Ressourcenverbrauchs vor der Koordination nach der Planung herstellt. Diese zentrale Instanz soll im Weiteren als Ressourcenverwalter bezeichnet werden. Auf Basis der Nutzenfunktionen kann die Herstellung der Konsistenz so geschehen, dass das Zielsystem weiterhin so gut wie möglich erfüllt ist.

Im Anschluss an die lokale Planung informieren sämtliche Ressourcen verbrauchenden Teilsysteme den Verwalter über den Verbrauch, der mit der Durchführung ihrer jeweiligen Aktivitäten verbunden sein wird. Mit dieser Information und den gegebenen Ressourcen kann der Verwalter überprüfen, ob der Plan lösbar ist. Ist dies nicht der Fall, können direkt die Zeitabschnitte bestimmt werden, in denen eine Einsparung erforderlich ist. Die Realisierung der Einsparung kann mithilfe einer Ausschreibung des Ressourcenverwalters an die Teilsysteme realisiert werden. Der Ressourcenverwalter fordert also die Teilsysteme auf, eine bestimmte Menge an Ressourcen in einem bestimmten Zeitabschnitt einzusparen. Mithilfe der lokalen Planungsmetrik kann dabei jedes Teilsystem die Auswirkung der Wahl einer alternativen Teilfunktionslösung ermitteln und diese Auswirkung zusammen mit dem geänderten Verbrauch an den Ressourcenverwalter übermitteln. Anhand der übermittelten Werte kann der Ressourcenverwalter dann entscheiden, welches Teilsystem wann auf eine alternative Teilfunktionslösung wechseln soll. Da der zentrale Verwalter keine Planung im eigentlichen Sinne durchführt und kein zentrales Planungsmodell mit hohem Verzweigungsgrad vorliegt, gelten die Kritikpunkte (vergl. Kapitel 2.4) an einer zentralen Planungsinstanz in diesem Fall nicht.

### 5.2.6 Ablauf der Koordination

Abbildung 5-10 stellt den Gesamtprozess der Interaktionsplanung bei gemeinsamer Hauptaufgabe dar. Dabei sind zwei nebenläufige Prozesse zu erkennen: die Planung und die Ausführung. Der erste Prozess beginnt mit der lokalen Planung, die die aktuell bekannten Wechselwirkungen zwischen den Aktivitäten einzelner Teilsysteme berücksichtigt. Als Ergebnis liegt eine Menge lokaler Pläne vor. Für die vorliegenden lokalen Pläne werden die Nutzenwerte anhand der bekannten Wechselwirkungen und den nun vorliegenden zeitlichen Beziehungen aktualisiert. Dadurch ergeben sich planbezogene Nutzenwerte. Im darauf folgenden Schritt bewertet jedes Teilsystem seinen lokalen Plan bezüglich des enthaltenen Koordinationspotenzials. Der Plan mit dem maximalen Koordinationspotenzial wird ausgewählt und anschließend einer lokalen Umplanung zugeführt.

Parallel dazu wird der jeweils fixierte Teil der lokalen Pläne ausgeführt. Diese Ausführung bewirkt zweierlei: Der Planungshorizont verschiebt sich mit der Ausführung und neue Trainingsdaten für das Lernverfahren liegen vor. Die Verschiebung des Planungshorizonts wird im Planungsprozess bei der Bestimmung der Nutzenwerte und der Auswahl des umzuplanenden lokalen Plans berücksichtigt. Das Lernverfahren aktualisiert die bekannten Wechselwirkungen auf Grundlage der neu vorliegenden Trainingsdaten. Dieser Vorgang kann bedarfsweise angestoßen werden, wenn zum Beispiel eine bestimmte Menge Trainingsdaten vorliegt oder freie Rechenzeit bereitsteht.

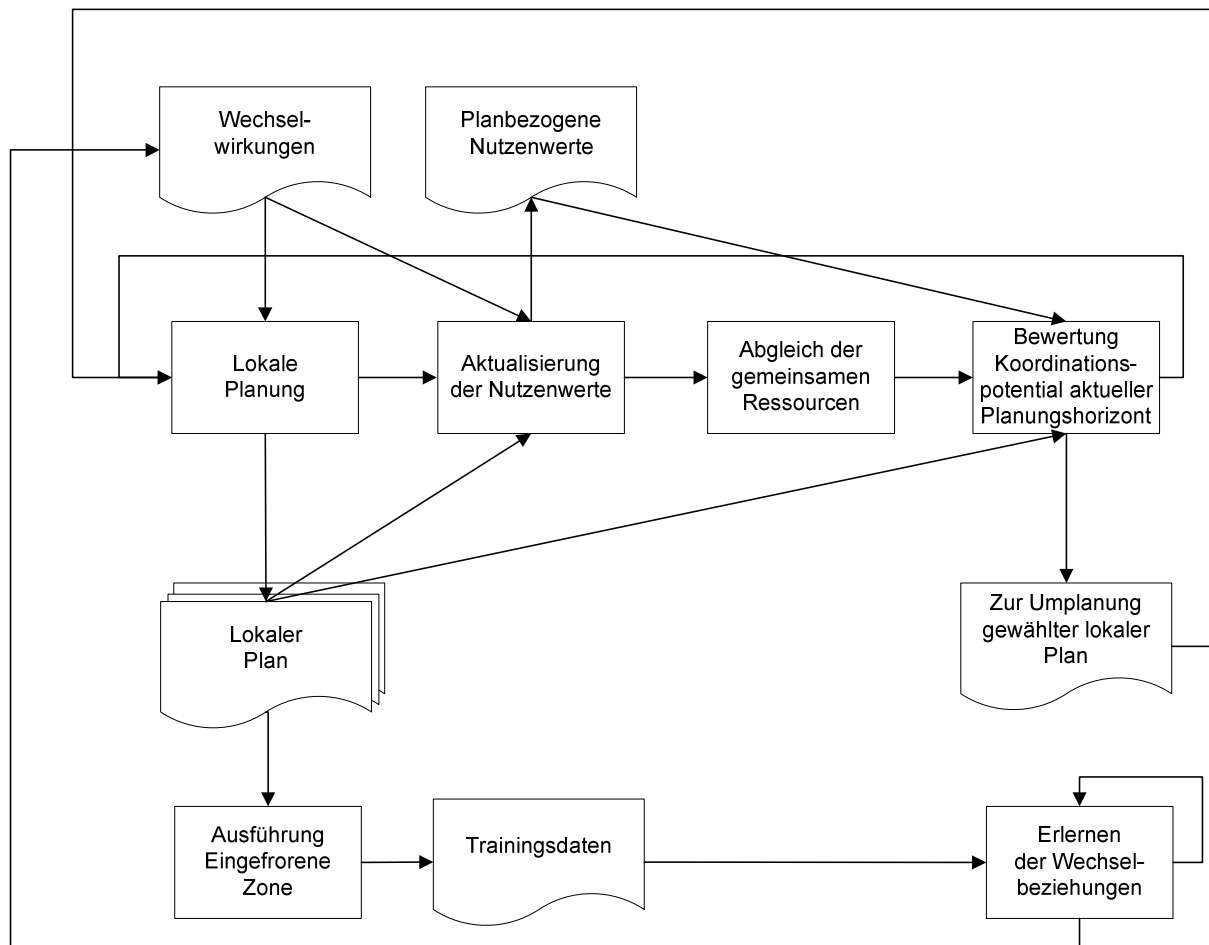


Abbildung 5-10: Ablauf der Interaktionsplanung bei gemeinsamer Hauptaufgabe

Zur Durchführung des hier vorgeschlagenen Verfahrens zur Planung von Systeminteraktionen müssen sich die Teilsysteme über ihre lokalen Pläne und deren Koordinationspotential austauschen. Zur Umsetzung dieses Austauschs bietet sich eine Blackboard-Architektur<sup>51</sup> an, wie sie häufiger im Bereich der Multiagenten-Planung eingesetzt wird (z. B. im Verfahren zur Coordination before Planning, siehe 5.1.3.2.5.). Bei einem Blackboard handelt es sich um einen gemeinsamen Bereich, auf den alle Teilsysteme zugreifen können. Dazu gehört lesender und schreibender Zugriff. Auf dem Blackboard veröffentlichen die Teilsysteme zunächst ihre lokalen Pläne und anschließend das jeweilige Koordinationspotential der Pläne. Dabei müssen die Pläne nur als eine Liste von Teilfunktionslösungsbezeichnungen mit Start- und Endzeit beschrieben werden. Ein genauerer Austausch über die Vor- und Nachbindungen der Teilfunktionslösungen oder die Angabe einer Zustandstrajektorie ist nicht notwendig. Genau so wird das Koordinationspotential einfach als reelle Zahl ausgetauscht, die sich aus der Aufsummierung der Koordinationsmöglichkeit der einzelnen Teilfunktionslösungen ergibt. Der Vorzug des Blackboards liegt dabei in der Reduzierung der Kommunikationsbeziehungen im Vergleich zu einer direkten Kommunikation zwischen den Teilsystemen. Weiterhin werden blockierende Nachrichten vermieden, die auftreten würden, wenn Teilsystem A den lokalen Plan von Teilsystem B anfragt, während B den Plan noch erzeugt oder andere Pläne analysiert.

<sup>51</sup> Dieses Kommunikationsprinzip ist nicht zu verwechseln mit den Blackboards im Matchmaking aus Abschnitt 5.1.3.1.3.

Mit diesem Vorgehen überwindet das Planungsverfahren die wesentlichen Kritikpunkte am GPGP und SHAC, indem ganze Sequenzen von Aktionen synchronisiert werden und auf die Voraussetzung der weitgehenden Unabhängigkeit von Aktivitäten verzichtet werden kann.

### **5.3 Planung der Interaktion bei individueller Hauptaufgabe**

In Anlehnung an Wooldridge und Jennings (siehe Abschnitt 5.1.2.2) muss die Interaktionsplanung bei individueller Hauptaufgabe die folgenden Schritte unterstützen:

1. Erkennen von Kooperationspotenzial,
2. Identifikation von Kooperationspartnern,
3. Bestimmung gemeinsamer Planungsziele und
4. Gemeinsame Planfindung.

Anhand dieser Phasen lässt sich bereits erkennen, dass der Ausgangspunkt einer Interaktionsplanung bei individueller Hauptaufgabe immer ein einzelnes System ist, das in seinem lokalen Plan die Möglichkeit zur Verbesserung der Planqualität entdeckt. Treibende Kraft ist also das Ziel, die Erfüllung der individuellen Hauptaufgabe zu verbessern. Bei Systemen mit individueller Hauptaufgabe kann eine bauliche und räumliche Integration der Systeme ausgeschlossen werden. Allerdings handeln die Systeme in einer gemeinsamen Umgebung, sonst wäre Interaktion ausgeschlossen. Daher ist eine Interaktion immer verbunden mit einem Umgebungsobjekt oder genauer: mit Zustandsveränderungen an einem Umgebungsobjekt.

#### **5.3.1 Mögliche Kooperationsaktionen**

Um ihr Ziel zu erreichen, die Qualität des lokalen Plans mithilfe anderer Systeme zu verbessern, stehen den Systemen mehrere grundsätzliche Kooperationsaktionen zur Verfügung. Diese Aktionen besitzen sehr unterschiedlichen Charakter hinsichtlich der notwendigen Abstimmung mit anderen Systemen.

Die möglichen Kooperationsaktionen, die innerhalb dieser Arbeit betrachtet werden, sind im Einzelnen:

- die Abgabe von Aufträgen an andere Systeme,
- die Abgabe von Teilfunktionen an andere Systeme und
- Abstimmung von Aktivitäten.

Bei der Abgabe von Aufträgen an andere Systeme gibt ein System eine Aufgabe, die ihm ursprünglich zugeordnet war, vollständig an ein anderes System ab. Ein solches Vorgehen ist immer dann sinnvoll, wenn die Ausführung der Aufgabe für das andere System mit geringem Aufwand verbunden ist. Dies wird in der Regel sehr häufig der Fall sein bei Aufträgen, die in der eigenen Planung einen verhältnismäßig großen Aufwand bedeuten, indem sie beispielsweise zu Zuständen führen, die für das Erfüllen anderer Aufträge ungünstig sind. Eine Abstimmung der Systeme ist hier nicht notwendig, solange nur die Aufgabe erfolgreich erfüllt wird.

Ähnlich gelagert ist der Fall der Abgabe einzelner Teilfunktionen. Nur wird in diesem Fall eben nicht der gesamte Auftrag abgegeben, sondern nur eine einzelne Teilfunktion, während die Ausführung des übrigen Auftrags in der Verantwortung des abgebenden Systems verbleibt. Daraus ergibt sich auch ein höherer Koordinationsaufwand zwischen den beteiligten Systemen, da die Ausführung der betrachteten Teilfunktion hinsichtlich ihres Ergebnisses und der zeitlichen Ausführung in den Plan des abgebenden Systems eingepasst werden muss.

Beiden Kooperationsaktionen ist weiterhin gemeinsam, dass ein fremdes System Tätigkeiten übernimmt. Aus einer solchen Übernahme von Tätigkeiten wird sich für das übernehmende System in der Regel keine Verbesserung der lokalen Planqualität ergeben. Entsprechend muss in irgendeiner Form eine Kompensation erfolgen. Für die Umsetzung dieser Kooperationsplanung wird hier eine vereinfachte Darstellung von Kompensationen gewählt, um die Umsetzung der Interaktionsplanung exemplarisch demonstrieren zu können. Eine Kompensation, die für eine Kooperationsaktion notwendig ist, wird daher als eine reelle Zahl aus dem Intervall  $[0,1]$  dargestellt. Auf der anderen Seite wurde zur Beurteilung der Planqualität jeweils eine Planungsmetrik in Form einer gewichteten Summe der möglichen Ziele eines mechatronischen Systems herangezogen. Diese Planungsmetrik soll – wie bereits im vorherigen Kapitel 5.2 - ebenfalls auf den Wertebereich  $[0,1]$  normiert werden. Dadurch werden die Kompensation, die im Rahmen einer Kooperation anfällt, und die sich einstellende Planverbesserung vergleichbar. Eine anschauliche Interpretation wäre eine monetäre Bewertung der einzelnen Glieder der Planmetrik und eine Kompensation in Form einer Ausgleichszahlung.

Im Falle der Abstimmung von Aktivitäten können sich durchaus Situationen einstellen, in denen alle beteiligten Systeme von einer Kooperation profitieren. Damit ist diese Koordinationsmöglichkeit die einzige, die auch anwendbar ist, wenn das Anwendungsgebiet eine Darstellung von Kompensationen nicht ermöglicht. Jedoch wird auch für das gemeinsame Durchführen von Aktivitäten der Spielraum durch Kompensationen vergrößert, wodurch sich mehr Kooperationen einstellen können. Diese Form der Kooperationsaktion erfordert auch den größten Aufwand an Abstimmung zwischen den beteiligten Systemen. So müssen die zeitliche Abstimmung und die durchzuführenden Aktionen genau festgelegt werden.

### **5.3.2 Modellierung von Kooperationsaktionen**

Um die Möglichkeit zur Durchführung von Kooperationsaktionen zu identifizieren und diese anschließend umzusetzen, bedarf es einer Erweiterung der Modellierung des Planungsproblems aus Unterkapitel 2.1.2. Ohne ein geeignetes Modell ist der Planer nicht in der Lage, mit den Möglichkeiten der Kooperation umzugehen.

#### **5.3.2.1 Modellierung der Abgabe von Aufträgen**

Die Modellierung der Abgabe von Aufträgen lässt sich von allen Kooperationsaktionen am leichtesten bewerkstelligen, da bereits in Unterkapitel 2.1.2 alle notwendigen Modellelemente vorgegeben sind. Dort ist ein Auftrag verbunden mit einer Menge von Zielzuständen, die zur Erfüllung des Auftrags erreicht werden müssen. Wird ein Auftrag abgegeben, entfallen die mit ihm verbundenen Zielzustände. Dadurch ergibt sich ein neues Planungsproblem, für das ein neuer Plan generiert wird. Notwendige Kompensationszahlungen und die Differenz der Planqualitäten mit und ohne den entsprechenden Auftrag können daraufhin verglichen werden.

Am Beispiel des RailCabs ist ein Zielzustand, der mit einem Auftrag verbunden ist, der Aufenthalt des Fahrzeugs in der Startstation und Endstation des Auftrags, jeweils innerhalb eines bestimmten Zeitfensters. Gibt ein RailCab einen Auftrag ab, kann es seine Route anpassen, ohne die beiden Stationen in den jeweiligen Zeitfenstern anfahren zu müssen. Dadurch kann sich z. B. eine erhebliche Verkürzung des Fahrweges ergeben.

#### **5.3.2.2 Modellierung der Abgabe von Teilfunktionen**

Die Darstellung der Abgabe einer Teilfunktion erfordert dagegen die Erweiterung des Modells aus Unterkapitel 2.1.2. Die bisherige Modellierung sah noch keine Teilfunktionen vor, die an andere Systeme abgegeben werden, sondern nur die eigenen Teilfunktionslösungen. Eine Teilfunktion, die an ein anderes System abgegeben wird, ist aber auch nur eine spezielle

Form der Teilfunktionslösung. Ihre besonderen Eigenschaften liegen darin, dass eine solche Teilfunktionslösung nur zwei unterschiedliche Arten von Effekten haben kann. Dies sind zunächst die Effekte auf Umfeldobjekte des Systems und andererseits eben die Kompensationen, die fällig werden. Mithilfe derartiger Aktionen in der Modellierung des Planungsproblems lassen sich die Chancen und Möglichkeiten der Kooperation direkt in der Planerstellung berücksichtigen. Dabei wird die Abgabe einer Teilfunktion (dargestellt durch ein Aktionschema) nur berücksichtigt, wenn sie den Plan hinsichtlich der Zielfunktion verbessert. Da bei der Abgabe von Teilfunktionen nicht davon ausgegangen werden kann, dass sie immer in völlig identischer Art und Weise erfüllt werden, können eigentlich keine skalaren Werte für die Effekte an Umfeldobjekten und die Kompensation angegeben werden. Wie bereits im Rahmen der Interaktionsplanung für Systeme mit gemeinsamer Hauptaufgabe sollen auch hier Häufigkeitsverteilungen herangezogen werden. Zur Darstellung in Form einer PDDL-Aktion kann wiederum ein repräsentativer Wert einer historischen Häufigkeitsverteilung (also etwa der Mittelwert oder ein P-Quantil) herangezogen werden.

Ein Beispiel für ein mechatronisches System, das Teilfunktionen abgeben kann, ist eine Bohr-Fräsmaschine, die die Arbeitsgänge Bohren und Fräsen durchführen kann. Grundsätzlich besteht aber auch die Möglichkeit, einen der Arbeitsgänge an eine andere Maschine abzugeben (die beispielsweise eine höhere Fertigungsgenauigkeit erreicht oder den Arbeitsgang schneller durchführen kann). So kann für die Abgabe eines Bohrarbeitsganges im Planungsmodell der Bohr-Fräsmaschine eine spezielle Teilfunktionslösung vorgesehen werden. Diese Teilfunktionslösung besteht darin, das Werkstück durch eine andere Maschine bearbeiten zu lassen. Wie diese Bearbeitung und der Transport durchgeführt werden, ist für die Bohr-Fräsmaschine unerheblich, solange das Werkstück rechtzeitig und korrekt bearbeitet wieder zur Verfügung steht.

### 5.3.2.3 Modellierung zur Abstimmung von Aktivitäten

Auch für die Darstellung der Wirkung von gemeinsam ausgeführten Aktivitäten ist eine Erweiterung der bisherigen Modellierung notwendig. Anders als bei der Modellierung der Abgabe von Teilfunktionen werden hier allerdings keine neuen Aktionen für die Darstellung eingeführt. Stattdessen baut die Darstellung von gemeinsamen Aktivitäten auf den bereits bekannten Teilfunktionslösungen auf. Um ein Verbesserungspotenzial in der Abstimmung auszuführender Teilfunktionslösungen erkennen zu können, ist es notwendig, die möglichen entsprechenden Teilfunktionslösungen und ihre Wechselwirkungen zu kennen. Ohne diese Kenntnis ist eine Abstimmung nicht möglich. Bereits in Kapitel 5.3 wurde eine Modellierungsmethode eingeführt, die diese Informationen darstellen kann. Die dort eingeführte Modellierung kann für den Fall der Interaktionsplanung mit individueller Hauptaufgabe übernommen werden. Dabei ist allerdings zu berücksichtigen, dass die folgenden Voraussetzungen für den Einsatz dieser Kooperationsaktion auf Basis der gewählten Modellierung erfüllt sein müssen:

- Die Systeme müssen die Teilfunktionslösungen kennen, die andere Systeme in ihrer Umgebung ausführen können.
- Die Systeme müssen die Wechselwirkungen kennen oder erlernen.
- Die Systeme müssen feststellen können, welche Systeme an welchen Umfeldobjekten agieren.

Die Voraussetzungen sind nicht in jedem Fall erfüllt und bedingen, dass zumindest einige Systeme in der Umgebung grundsätzlich bereit sind, Informationen auszutauschen, um die Verbesserungsmöglichkeiten auszunutzen, die sich durch gemeinsame Aktivitäten ergeben.

Ein Beispiel für die Abstimmung einer Aktivität ist die Konvoifahrt der RailCab-Fahrzeuge. Die Abstimmung besteht darin, dass die Fahrzeuge gleichzeitig den gleichen Streckenabschnitt (Umgebungsobjekt) befahren (Teilfunktionslösung). Durch die Konvoifahrt reduziert sich der Energieverbrauch während der Fahrt (Wechselwirkung).

### 5.3.3 Durchführung der Abgabe von Aufträgen

Die Durchführung der Abgabe von Aufträgen kann grob in drei Phasen unterteilt werden:

1. Identifikation abzugebender Aufträge,
2. Ausschreibung abzugebender Aufträge und
3. Verhandlung über abzugebende Aufträge.

Die erste Phase der Abgabe von Teilaufträgen läuft vollkommen lokal in den Teilsystemen ab und analysiert nur den eigenen Plan. Bei der Ausschreibung der abzugebenden Aufträge ist zu unterscheiden, inwieweit für diesen Zweck auf eine Infrastruktur zurückgegriffen werden kann. Davon ist abhängig, welche Methoden des Matchmakings verwendet werden können. Die Verhandlung über abzugebende Aufträge ist relativ einfach, da nur eine Einigung über die Kompensation erreicht werden muss.

#### 5.3.3.1 Identifikation abzugebender Aufträge

Unabhängig von der gegebenen Problemstruktur kann ein abzugebender Auftrag dadurch identifiziert werden, dass die Verhaltensplanung jeweils einmal mit und ohne den Auftrag durchgeführt wird. Stellt sich eine relevante Verbesserung der Planungsqualität ein, ist es lohnend, den Auftrag abzugeben. Wird dieser Vorgang aber für alle Aufträge eines Systems durchgeführt, entsteht ein erheblicher Mehraufwand in der Planung. Die in Abschnitt 5.3.2.1 eingeführte Darstellung von Aufträgen kann dazu genutzt werden, Aufträge, die sich zur Abgabe eignen, anhand ihrer Eigenschaften zu identifizieren. Aufträge, die sich sehr stark von allen anderen Aufträgen unterscheiden, sind voraussichtlich nur mit einem relativ hohen Aufwand zu erfüllen und können besser von solchen Systemen übernommen werden, die bereits ähnliche Aufträge zugeordnet haben.

Mit dem Zielvektor, der mit jedem Auftrag verbunden ist, liegt die Information vor, die zur Beurteilung der Kompatibilität zweier Aufträge notwendig ist. Dabei kann davon ausgegangen werden, dass zwei Aufträge gut zueinanderpassen, wenn die im Vektor enthaltenen Elemente ähnliche Zustände zu ähnlichen Zeiten beschreiben. Umgekehrt sind zwei Aufträge nur schwer oder mit hohem Aufwand miteinander vereinbar, wenn sie zu ähnlichen Zeiten unterschiedliche Zustände erzielen wollen. Ein solcher Zusammenhang lässt sich in Form eines Distanzmaßes (siehe Abschnitt 3.2.1.1) beschreiben.

Für das hier untersuchte Problem ist die Menge  $M$  die Menge der Aufträge eines mechatronischen Systems. Das Distanzmaß der Aufträge ist aus einem Distanzmaß der Zustände abzuleiten. Ein solches Distanzmaß ist aufbauend auf die PDDL relativ einfach zu konstruieren, da Zustandsvariablen sich lediglich durch den Wahrheitswert logischer Prädikate und die Belegung der numerischen Zustandsvariablen unterscheiden. Die Konstruktion des Distanzmaßes soll am Beispiel einer euklidischen Distanz erläutert werden, obwohl je nach Anwendungsgebiet durchaus andere Distanzen wie die Minkowski-Distanz oder Mahalanobis-Distanz geeigneter sein können. Auf Basis der im Zielvektor enthaltenen Zustandsvariablen kann die Distanz zwischen zwei Aufträgen  $a_1$  und  $a_2$  ermittelt werden als:

$$d(a_1, a_2) = \sqrt{\sum_{k=1}^p (x_{1,k} - x_{2,k})^2}$$



Dabei bezeichnet  $x_{i,k}$  die in Auftrag  $i$  beschriebene Ausprägung der  $k$ -ten Zustandsvariable. Damit ist der zeitliche Abstand, zu dem die einzelnen Zustandsvariablen realisiert werden sollen, aber noch völlig unberücksichtigt. Da ein großer zeitlicher Abstand weniger ins Gewicht fallen soll als ein geringer, bietet es sich an, die Distanz der Zustandsvariablen mit dem zeitlichen Abstand zu gewichten:

$$d_t(a_1, a_2) = \sqrt{\sum_{k=1}^p \frac{1}{(t(x_{1,k}) - t(x_{2,k}))^2} \cdot (x_{1,k} - x_{2,k})^2}$$

Diese Formulierung des Distanzmaßes kann auch dann bestehen bleiben, wenn eine Zustandsvariable mehrfach zu unterschiedlichen Zeitpunkten in der Zielformulierung eines Auftrags auftritt. In diesem Fall wird jede Zustandsvariable mit der korrespondierenden Zustandsvariablen des anderen Auftrags verglichen, die ihr zeitlich am nächsten ist.

Mit der Definition eines Distanzmaßes ist es möglich, die Aufträge durch ein Clustering [JMF99] in Gruppen einzuteilen, die sich besonders ähnlich sind. Fallen nur wenige oder sogar nur ein einzelner Auftrag in eine Klasse, scheint dieser für die Abgabe besonders geeignet.

Führt ein RailCab beispielsweise ein Clustering über seine Transportaufträge aus, dienen die Koordinaten der Start- und Endstationen sowie die entsprechenden Zeitfenster als Teile des Distanzmaßes. Interessant sind Cluster mit nur wenigen oder sogar einem einzigen Transportauftrag, der dazu auch noch eine große Distanz zu den übrigen Aufträgen aufweist. Kann ein solcher Cluster vollständig abgegeben werden (daher sind Cluster mit nur wenigen Elementen zu bevorzugen), wird ein Umweg in der aktuellen Route eingespart und die übrigen Aufträge werden günstiger bedient.

### 5.3.3.2 Ausschreibung abzugebender Aufträge

Das Ziel der Ausschreibung ist es, ein System zu finden, das den abzugebenden Auftrag übernimmt. In diesem Sinne handelt es sich um ein Matchmaking-Problem, wie in 5.1.3.1 beschrieben. Dementsprechend können auch die dort beschriebenen Verfahren eingesetzt werden. Um den anfallenden Kommunikationsaufwand zu beschränken, sind Middle-Agents (vergleiche 3.4.3.1.3) die geeignetste Variante. Der Vorteil liegt darin, dass Systeme, die in der Lage sind, einen Auftrag zu übernehmen, aktiv nach einem geeigneten Auftrag suchen können. Im Gegensatz zum reinen Contract Net Protocol, bei dem mit allen Systemen kommuniziert wird, erfolgt die Kommunikation also wesentlich zielgerichteter. Das abgebende System stellt dabei die den Auftrag betreffenden Informationen den Middle-Agents zur Verfügung. Dadurch können Systeme, die an einer Übernahme interessiert sind, vor der Kontaktaufnahme überprüfen, ob sie den Auftrag überhaupt und mit welchem Aufwand realisieren können.

Ist es nicht möglich oder gewollt, eine Infrastruktur in Form von Middle-Agents aufzubauen, bleiben nur die Möglichkeiten des verteilten Matchmakings oder eine Kommunikation nach dem Contract Net Protocol mit den Systemen, die sich in Kommunikationsreichweite befinden, übrig. Das verteilte Matchmaking setzt dabei voraus, dass die übrigen Systeme bereit sind, eine Ausschreibung weiterzuleiten.

### 5.3.3.3 Verhandlung über abzugebende Aufträge

Kontaktiert ein an einem ausgeschriebenen Auftrag interessiertes System das ausschreibende System, müssen die beiden Systeme sich über die entsprechende Kompensation einigen. Dazu muss das ausschreibende System zunächst das durch die Abgabe realisierbare Verbesserungs-

potenzial bestimmen. Dies ist – wie bereits oben erläutert – durch eine einfache Neuplanung ohne die mit dem Auftrag verbundenen Teilziele möglich. Indem die Planungsmetrik für beide Pläne ausgewertet wird und die Differenz gebildet wird, bestimmt das abgebende System die Obergrenze der Kompensation. Übersteigt die Kompensation diese Differenz, führt die Abgabe zu keiner Verbesserung mehr.

Auf der anderen Seite muss das interessierte System ebenfalls die Pläne mit und ohne Berücksichtigung des Auftrags bewerten. Es ermittelt so seine Untergrenze für die Kompensation, unterhalb derer eine Übernahme nicht sinnvoll ist. Damit ist der Verhandlungsraum bereits abgesteckt, und die tatsächlich realisierte Kompensation hängt von der Verhandlung zwischen den Systemen ab. Für die Situation, dass es nur ein einziges interessiertes System gibt, bietet sich zum Beispiel das Protokoll für das wechselseitige Bieten an (beschrieben in 5.1.3.3.1). Falls mehrere Systeme an dem Auftrag interessiert sind, kann das abgebende System auch eine umgekehrte Auktion durchführen, um die geringstmögliche Kompensation zu erhalten. Einigen sich die Systeme auf eine Abgabe des Auftrags, stellen die beiden Systeme ihre lokalen Pläne entsprechend um. Eine darüber hinausgehende Abstimmung ist nicht notwendig.

### 5.3.4 Durchführung der Abgabe von Teilfunktionen

Der allgemeine Ablauf der Abgabe von Teilfunktionen ähnelt dem Vorgehen bei der Abgabe von Aufträgen. Bei grober Betrachtung zerfällt er in die gleichen Phasen:

1. Identifikation abzugebender Teilfunktionen,
2. Ausschreibung abzugebender Teilfunktionen und
3. Verhandlung über abzugebende Teilfunktionen.

Die Ausgestaltung der Phasen und ihre Gewichtung unterscheiden sich aber erheblich. Da die abgegebene Teilfunktion in den ursprünglichen Plan des abgebenden Systems passen muss, ist in der Verhandlung deutlich mehr abzustimmen als nur die Kompensation. Das Ergebnis der Teilfunktion muss sowohl hinsichtlich des Ergebnisses als auch der zeitlichen Bereitstellung passen.

#### 5.3.4.1 Identifikation abzugebender Teilfunktionen

Durch die Modellierung der Abgabe von Teilfunktionen als eine spezielle Form einer Teilfunktionslösung kann die Identifikation der abzugebenden Teilfunktionen direkt in der Planung geschehen. Indem die Kompensationen direkt in die Planungsmetrik aufgenommen werden, kann der Planer zwischen der Abgabe der Teilfunktion und der Ausführung der Teilfunktion mithilfe eigener Teilfunktionslösungen abwägen. Dabei wird der gewählte charakteristische Wert für die einzelnen Effekte und die Kompensation angesetzt. Alle Teilfunktionen, die anschließend in dem Plan enthalten sind, eignen sich für die Abgabe.

Mit dem so konstruierten Plan hängt die erfolgreiche Durchführung aber davon ab, dass alle ausgeschriebenen Teilaufgaben abgegeben werden können. Davon ist aber nicht zwingend auszugehen. Daher schließt sich an diese erste Planungsphase, die die abzugebenden Teilfunktionen ermittelt hat, eine zweite Phase an. In dieser zweiten Phase wird der ursprüngliche Plan in mehrere Segmente unterteilt. Als Nahtstellen der Segmente dienen die abzugebenden Teilfunktionen, die nun aus dem Plan entfernt werden. Auf diese Art werden für  $n$  abzugebende Teilfunktionen  $n$  neue Planungsprobleme erzeugt, die die Fremdvergabe ersetzen sollen. Die Startsituation für jedes dieser Planungsprobleme ist gegeben durch den Vorgängerzustand der Teilfunktionslösungen. Ebenso könnte als Zielzustand direkt der Nachfolgezustand bei Abzug der Kompensationszahlung verwendet werden. Damit definiert man den Zielzustand unter Umständen aber zu restriktiv und nimmt so dem Planer die Möglichkeit, einen guten Plan zu bestimmen. Statt also den Nachfolgezustand der abzugebenden Teilfunktion zu verwenden, sollte der Zielzustand so bestimmt werden, dass der restliche Plan weiterhin aus-

fürbar bleibt. Ein Beispiel für diesen Unterschied stellen Fertigungstoleranzen dar. Statt also als Zielzustand die genaue Fertigungsgenauigkeit aus dem ursprünglichen Plan vorzugeben, ist es sinnvoller, die Toleranzen anzugeben, mit denen die nachfolgenden Schritte ausgeführt werden können.

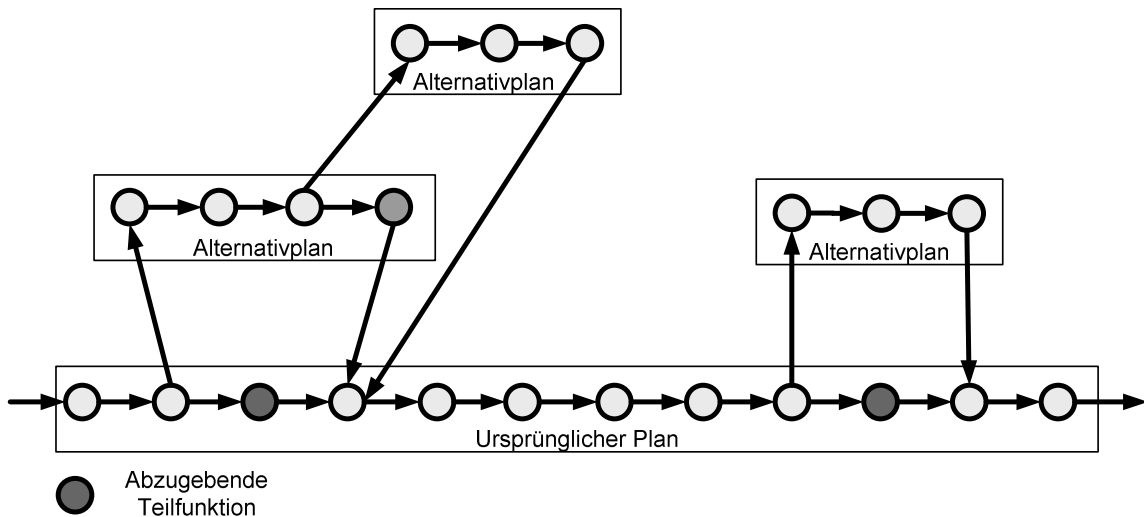
Um einen solchen Zustand zu finden, wird die Funktion  $\text{req}(s_i): s_i \rightarrow P(s_i)$  eingeführt, die angibt, welche Zustandsvariablen eines Zustands  $s_i$  erfüllt sein müssen, damit der  $s_i$  nachfolgende Plan weiterhin ausgeführt werden kann. Zur Berechnung dieser Funktion kann der in Listing 5-2 dargestellte Algorithmus verwendet werden. Die Zeit, um alle Zustände im Plan zu aktualisieren, beträgt  $O(n)$  ( $n = \text{Anzahl der Zustände}$ ). Indem bei einer Neuplanung eines Teilplans mit den Aktionen  $a_j \dots a_n$  die Menge der Zustandsvariablen-Belegungen  $\text{req}(s_{i+1}) \cup g(s_{i+1})$  als zu erreichendes Ziel vorgegeben wird, kann gewährleistet werden, dass der neue Teilplan konsistent mit dem restlichen Plan ist. Logischerweise stellt  $s_{ij-1}$  die Vorbedingungen des neuen Teilplans dar.

```
temp = pre(j)
req(zj-1) = temp
Läufe rückwärts über alle Aktionen und Zustände solange temp • ∅
  Wenn
    die aktuelle Aktion die Zustandsbelegung s als Nachbedingung erzeugt
  Dann temp = temp / s
    req(aktueller Zustand) = req(aktueller Zustand) ∪ temp
```

#### Listing 5-2: Berechnung der Funktion $\text{req}(s_i)$

Anhand der Start- und Zielzustände, die die Anknüpfungspunkte für den bisherigen Plan darstellen, lassen sich nun alternative Pläne ermitteln, die verwendet werden können, falls eine Teilfunktion nicht erfolgreich abgegeben wird. Im Rahmen der Planung zur Ermittlung dieser Alternativpläne muss natürlich die Teilfunktion, die ersetzt werden soll, aus der Menge der verfügbaren Aktionen gestrichen werden, also gewissermaßen *tabu* gesetzt werden. Nur so kann sich ein alternativer Plan ergeben. Lässt man andere Teilfunktionen allerdings zu, ergibt sich ein rekursives Problem, da für diese ggf. eingesetzten Teilfunktionen wiederum alternativ Pläne einzusetzen sind. Die Menge der tabu gesetzten Teilfunktionen wächst mit jeder Rekursionsstufe an. Dies ist notwendig, damit die Rekursion sicher terminiert. Steht das schnelle Auffinden eines Plans, den das System eigenständig ausführen kann, im Vordergrund, können auch alle Teilfunktionen gleichzeitig tabu gesetzt werden. Abbildung 5-11 veranschaulicht das Vorgehen.

Sind beispielsweise an einem Werkstück einer Bohrfräse mehrere Bohrvorgänge auszuschreiben, können diese einzeln ausgeschrieben werden. Damit muss eine Vielzahl von alternativen Plänen für den Fall konstruiert werden, dass einzelne Vorgänge nicht vollständig abgegeben werden können. Stattdessen können die Bohrvorgänge aber gemeinsam betrachtet werden. Es werden also entweder alle Teilfunktionen abgegeben oder die Bohrfräse führt alle Bohrvorgänge selber durch.



**Abbildung 5-11: Einbettung alternativer Pläne als Ersatz für abzugebende Teilfunktionen**

Natürlich muss ein mechatronisches System nicht jede Teilfunktion, die es an ein anderes System abgeben kann, zwingend durch eigene Teilfunktionslösungen durchführen können. In einem solchen Fall hängt die erfolgreiche Ausführung des Plans tatsächlich von der Vergabe der Teilfunktion ab und die eben beschriebene Bildung von alternativen Plänen kann entfallen.

#### 5.3.4.2 Ausschreibung abzugebender Teilfunktionen

Wie bereits bei der Ausschreibung von Aufträgen erfolgt auch die Ausschreibung von Teilfunktionslösungen idealerweise über Middle-Agents. Stehen keine Middle-Agents zur Verfügung, können als Alternativen ebenfalls das verteilte Matchmaking und das Contract Net Protocol verwendet werden.

Die Informationen, die über die Middle-Agents oder andere Formen des Matchmakings ausgetauscht werden umfassen dabei:

- die Zeit, zu der die Teilfunktion ausgeführt werden muss,
- die Parameter (also die Objekte, auf denen die Teilfunktion arbeitet),
- die Menge der zu erzeugenden Zustandsvariablen ( $req(s_i)$ ) und schließlich
- den Namen der ausführenden Teilfunktion.

Anhand dieser Informationen kann ein interessiertes Teilsystem schnell überprüfen, ob die Übernahme einer Teilfunktion möglich ist oder nicht.

#### 5.3.4.3 Verhandlung über abzugebende Teilfunktionen

Im Gegensatz zur Verhandlung über abzugebende Aufträge handelt es sich bei der Verhandlung über abzugebende Teilfunktionen grundsätzlich um eine Verhandlung mit mehreren Attributen. Dementsprechend müssen Verhandlungsprotokolle aus 5.1.3.3 gewählt werden. Die einzelnen Attribute umfassen dabei die Kompensation und die genaue Bestimmung der Effekte der Ausführung der Teilfunktion. Es besteht ein gewisser Spielraum, da diese Effekte nicht nur einen Einfluss auf die Durchführbarkeit des restlichen Teilplans besitzen, sondern auch auf die Planqualität. Im Falle mehrerer interessierter Systeme kann auch hier wieder eine Auktion stattfinden.

Ist eine Einigung über diese Verhandlungsattribute gefunden, ist der Prozess der Abgabe der Teilfunktion damit abgeschlossen.

### 5.3.5 Abstimmung von Aktivitäten

Der Ablauf der Durchführung der Planung von gemeinsamen Aktivitäten unterscheidet sich von den beiden vorangegangenen. Dabei wird nicht mit einer Analyse in der lokalen Planung begonnen, um Aktivitäten zu identifizieren, die durch Abstimmung mit anderen Teilsystemen verbessert werden können. Stattdessen lässt sich der Suchraum für mögliche Kooperationen in diesem Fall in der Regel schneller einschränken, indem in einem ersten Schritt untersucht wird, welche anderen Systeme im gleichen Umfeld agieren. Sind diese Systeme identifiziert, kann versucht werden, Aktivitäten mit ihnen abzustimmen.

#### 5.3.5.1 Identifikation möglicher Kooperationspartner

Wie bereits eingangs von 5.3 dargelegt, hängen gemeinsame Aktivitäten immer direkt mit Umgebungsobjekten zusammen. Zustände und Eigenschaften von Objekten sind bestimmt durch die Zustandsvariablen der PDDL-Problemdefinition. Über die Zustandsvariablenbelegungen lassen sich damit auch Beziehungen zwischen den Plänen verschiedener Systeme herstellen. Da Zustandsvariablen in engem Zusammenhang mit Objekten stehen, können solche Zusammenhänge auch über die Variablen festgestellt werden.

Um einem System zu ermöglichen, potenzielle Kooperationspartner zu identifizieren, stellen die Objekte Informationen darüber bereit, welche Agenten sie zu welcher Zeit wie manipulieren:  $\text{man}: O \times T \times T \rightarrow \text{System} \times TFL$ . Die zweite und dritte Stelle des Tupels stellen die Grenzen des Zeitintervalls dar, in dem die Manipulation am Objekt erfolgt. Unter folgenden Umständen kann ein Agent als Kooperationspartner in Frage kommen:

- der Agent manipuliert das gleiche Objekt oder
- der Agent manipuliert ein Objekt desselben Typs.

Diese Erfassung möglicher Kooperationspartner ist aber evtl. zu einschränkend. So würde man im Beispiel der Konvoiplanung nur Fahrzeuge betrachten, die bereits die gleichen Streckenabschnitte befahren wie das Fahrzeug, das auf der Suche nach Konvoipartnern ist. Als Kooperationspartner kommen auch solche Agenten in Frage, die zwar nicht direkt am gleichen Objekt, aber in der näheren Umgebung eines solchen Objekts agieren (also in der Nähe des suchenden Fahrzeugs verkehren).

Den ersten Sachverhalt kann eine Funktion  $\text{distance}: O \times O \rightarrow \mathbb{R}$  darstellen, die den (z. B. euklidischen) Abstand zwischen zwei Objekten als reelle Zahl liefert. Für das Beispiel der Konvoiplanung ist die Definition sehr intuitiv: Die Distanz entspricht dem kürzesten Weg zwischen zwei Streckenpunkten. Eine andere Möglichkeit, die nicht voraussetzt, dass eine zentrale Instanz die Abstände zwischen allen Objekten kennt, wäre eine Funktion  $\text{neighbour}: O \rightarrow \mathcal{P}(O)$ , die für ein Objekt die benachbarten Objekte zurückliefert (zu einem Streckenpunkt beispielsweise die direkt verbundenen Streckenpunkte). Damit erweitert sich der Kreis der möglichen Kooperationspartner um alle Systeme, die in der eigenen Nachbarschaft handeln.

Wiederum erscheinen Middle-Agents als ein geeignetes Mittel, den Kommunikationsaufwand zu beschränken. Für diese Koordinationsaktionen bietet es sich insbesondere an, Blackboards (vergleiche Abschnitt 5.1.3.1.3) einzusetzen, die einzelnen Umgebungsobjekten zugeordnet sind. So kann direkt überprüft werden, welche anderen Systeme in einem relevanten Zeitfenster das gleiche oder ähnliche Objekt manipulieren.

#### 5.3.5.2 Abstimmung der Aktivitäten

Im Gegensatz zur Interaktionsplanung bei gemeinsamer Hauptaufgabe kann bei individueller Hauptaufgabe nicht davon ausgegangen werden, dass die Systeme bereitwillig ihre lokalen

Pläne austauschen. Das schränkt die Möglichkeiten zur Abstimmung natürlich sehr stark ein. Eine Abstimmung bedingt aber zumindest einen teilweisen Austausch über die Pläne.

Somit steht am Beginn der Abstimmung die Anfrage eines Systems, ob eine Abstimmung der Pläne gewünscht ist. Sind die angefragten Systeme an einer Abstimmung interessiert, müssen die relevanten Teilpläne ausgetauscht werden. Die Begrenzungen dieser Teilpläne können aus den Objekten abgeleitet werden, an denen die Systeme zu gleichen Zeiten handeln. Um den nachfolgenden Abstimmungsprozess möglichst zielgerichtet zu gestalten, besteht außerdem die Möglichkeit, gemeinsame Ziele zu definieren, die eingehalten werden müssen, um die Gültigkeit der lokalen Pläne sicherzustellen. Dazu kann wiederum die Funktion  $req(s_i)$  herangezogen werden, deren Berechnung in Listing 5-2 dargestellt ist.

### 5.3.6 Gesamtablauf

Die vorgestellten Kooperationsaktionen lassen sich jeweils einzeln, also ohne Verbindung zu den anderen Aktionen realisieren. Der Gesamtablauf der Koordination zerfällt dann in die bereits eingangs des Kapitels eingeführten Phasen:

1. Erkennen von Kooperationspotenzial,
2. Identifikation von Kooperationspartnern,
3. Bestimmung gemeinsamer Planungsziele und
4. Gemeinsame Planfindung.

Dabei ist zu beachten, dass im Falle der Abstimmung von Aktivitäten die Reihenfolge der Phasen 1 und 2 vertauscht wird. Die Abläufe innerhalb der einzelnen Phasen sind in den vorangegangenen Abschnitten ausführlich beschrieben.

Einer genaueren Betrachtung bedarf der Gesamtablauf aber, wenn die Kooperationsaktionen gleichzeitig eingesetzt werden sollen. Die Ursache hierfür besteht darin, dass Kooperationsaktionen in Konkurrenz stehen, sich also teilweise gegenseitig ausschließen. So fallen bei erfolgreicher Abgabe eines Auftrags ganze Teilpläne weg, die selbst Gegenstand von Kooperationsaktionen wie der Abgabe von Teilfunktionen oder der Abstimmung von Aktivitäten sein können. Die gleiche Beziehung liegt bei der Abgabe von Teilfunktionen hinsichtlich der Abstimmung von Aktivitäten vor. Aufgrund dieser Konkurrenz muss der gleichzeitige Einsatz der Kooperationsaktionen koordiniert werden.

Der erste Schritt in Richtung einer Koordination der Kooperationsaktionen besteht darin, die Beziehung zwischen den Kooperationsaktionen bzw. den mit ihnen verbundenen Planelementen festzuhalten. Abbildung 5-12 stellt diese hierarchische Beziehung in Form eines Baumes dar. Die Wurzel des Baumes ist der eigentliche Plan, wobei die enthaltenen Aktivitäten in Teilmengen zerfallen, die sich den Aufträgen zuordnen lassen. Daher stellen die Aufträge die nächste Ebene des Baumes dar. Zur Realisierung eines Auftrages, der nicht abgegeben wird, dienen eine Menge atomarer Aktivitäten oder Teilfunktionen, die wiederum an andere Systeme abgegeben werden können. Teilfunktionen wiederum sind in ihrer Struktur identisch mit den Aufträgen. Sie zerfallen also wiederum in Teilfunktionen oder Aktivitäten. Anhand der Baumstruktur ist direkt erkennbar, dass die erfolgreiche Abgabe eines Auftrags oder einer Teilfunktion alle möglichen Kooperationen hinsichtlich der Kindsnoten obsolet macht.

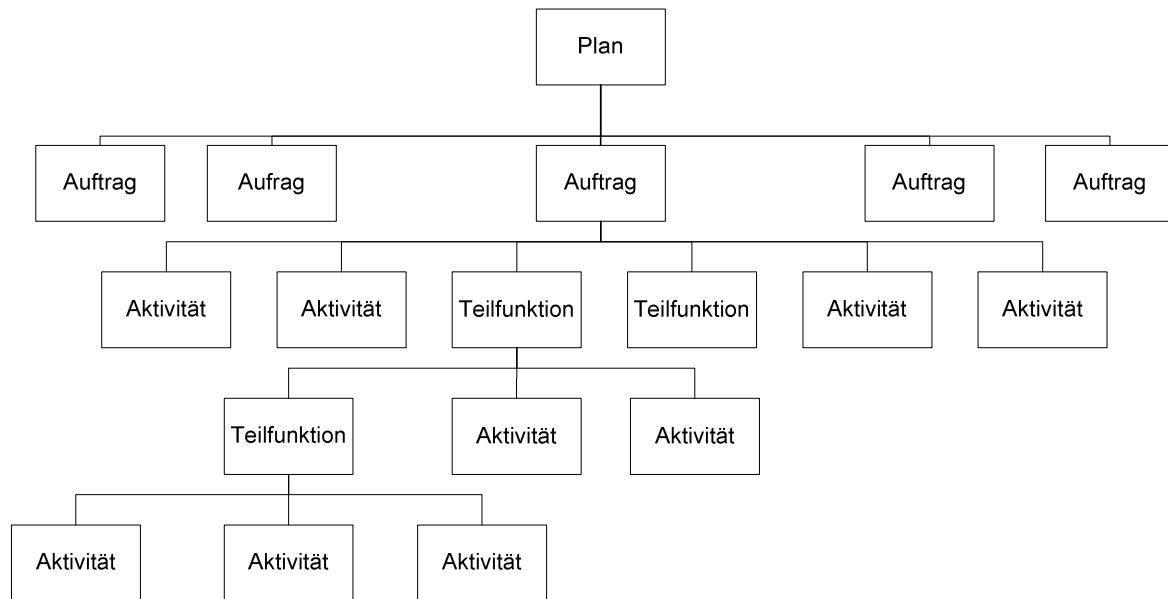


Abbildung 5-12: Hierarchische Beziehung zwischen Kooperationsgegenständen

Im allgemeinen Fall der Interaktionsplanung bei individueller Hauptaufgabe wird das mechatronische System versuchen, alle drei Formen der Kooperation gleichzeitig zu verfolgen. Dabei bietet sich ein Vorgehen von der umfassendsten Kooperationsaktion hin zur feinsten an, zumal die (positive) Wirkung der Abgabe von Aufträgen in der Regel größer sein wird als die Abgabe einzelner Teilfunktionen und die Abstimmung einzelner Aktivitäten. Abbildung 5-13 stellt den entsprechenden Ablauf dar. Ausgangspunkt der Interaktionsplanung ist die Auswahl abzugebender Aufträge, die noch vor dem Start der eigentlichen Planung ausgeschrieben werden. Als zweiter Schritt schließt sich die lokale Planung der Aufträge an, wobei die in Abschnitt 5.3.2.2 eingeführten Teilfunktionsaktivitäten genutzt werden, um solche Teilfunktionen zu identifizieren, deren Abgabe eine Verbesserung der Planqualität verspricht. Auch diese Teilfunktionen werden direkt ausgeschrieben. Die nächsten beiden Schritte können grundsätzlich parallel ausgeführt werden. Es sind einerseits die Alternativpläne für die ausgeschrieben Teilfunktionen zu erstellen. Andererseits wird für die im Plan vorhandenen *atomaren* Aktivitäten nach Kooperationspartnern gesucht. Abhängig von den identifizierten Kooperationspartnern ergeben sich dann Abstimmungsmöglichkeiten. Ebenfalls grundsätzlich parallel zum übrigen Planungsprozess können Verhandlungen mit anderen Systemen durchgeführt werden. Die Ergebnisse einer solchen Verhandlung werden zwischenzeitlich gepuffert. Nachdem der lokale Plan vollständig konstruiert ist, werden unter den möglicherweise konkurrierenden Einigungen eine Auswahl durchgeführt und die ausgewählten Kooperationen fixiert. Am Ende des Prozesses steht ein abgestimmter Plan.

Der Auswahlprozess zwischen den Einigungen bedarf noch einer genaueren Erläuterung. Zunächst bedingt er, dass die Einigungen, die Ergebnis der vorherigen Verhandlungen sind, nur vorläufig sind. Weiterhin ist zu klären, wie das mechatronische System zwischen den möglicherweise konkurrierenden Einigungen auswählt. Als Datenstruktur zur Verwaltung der Einigung bietet sich die in Abbildung 5-12 eingeführte Baumstruktur an. Wird eine beliebige Kooperationsaktion fixiert, sind alle weiteren Kooperationsaktionen, die in der Baumstruktur oberhalb oder unterhalb des entsprechenden Knotens in der Baumstruktur liegen, zu entfernen und den entsprechenden Systemen eine Absage zu erteilen. Daran ist ersichtlich, dass vor einer Fixierung die positiven Effekte einer Kooperation abgewogen werden müssen gegen diejenigen, die dadurch verworfen werden. Stehen bereits alle Einigungen über mögliche Kooperationsaktionen endgültig fest, so ist dies auch ohne Weiteres möglich. Es liegen dann für jede

mögliche Kooperationsaktion die Verhandlungsergebnisse vor, sodass die resultierende Planqualität errechnet werden kann.

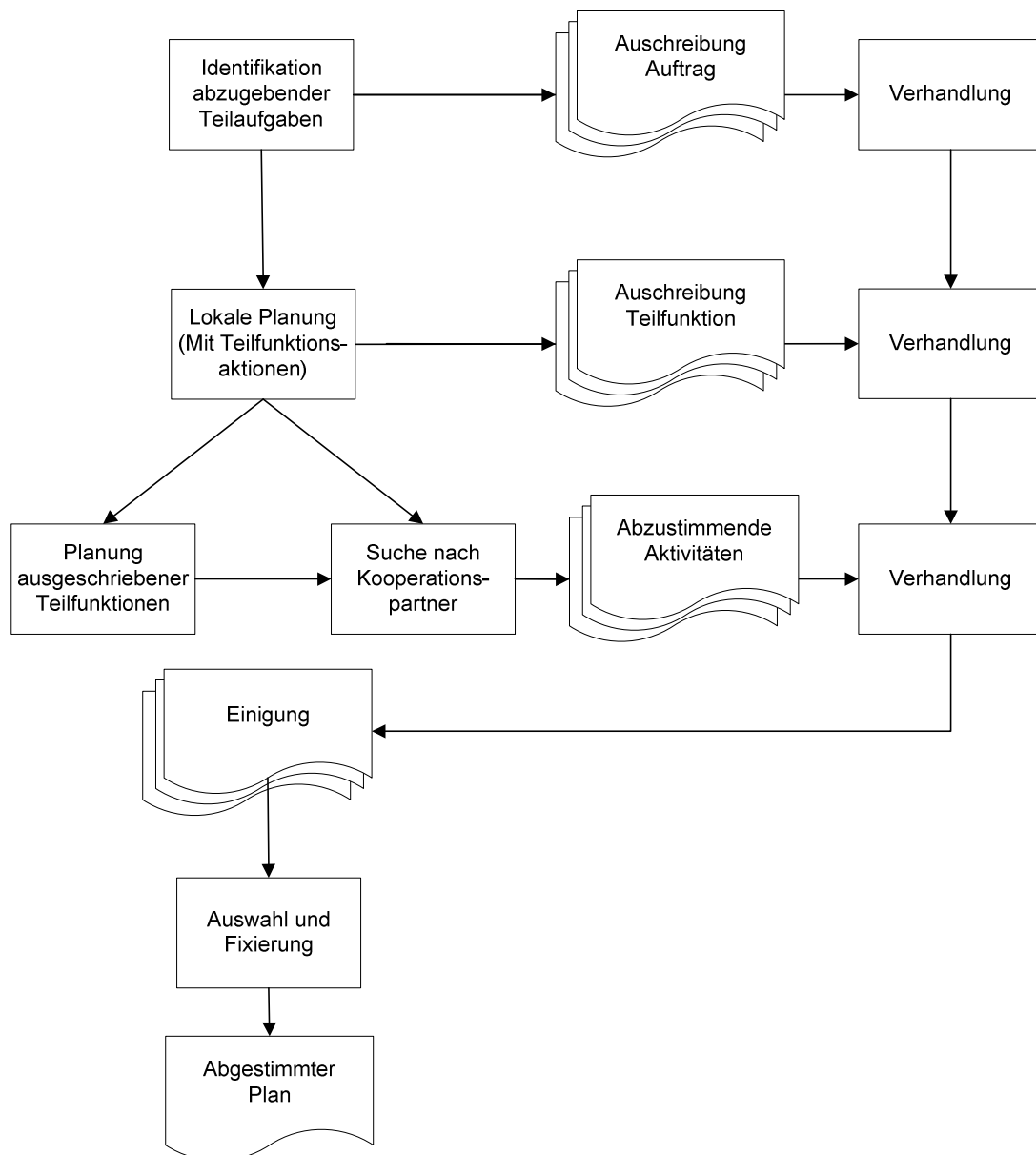


Abbildung 5-13: Ablauf der Interaktionsplanung bei individueller Hauptaufgabe

In der Regel wird aber nicht unbegrenzte Zeit zur Planung und Koordination zur Verfügung stehen, sondern die Kooperationen müssen fixiert werden, bevor die entsprechenden Aktionen ausgeführt werden. Diesem Zweck dient ein Commit-Window, das angibt, bis wann die Koordinationen hinsichtlich einer Aktion abgeschlossen sein müssen. Gleichzeitig ist es aber nicht realistisch zu unterstellen, dass Systeme ihre Kooperationsangebote immer aufrecht erhalten, bis das Commit-Window abgelaufen ist. Daher wird für jede Einigung ebenfalls ein Commit-Window eingeführt, das Gegenstand der Verhandlungen zwischen den kooperierenden Systemen ist. Damit besteht grundsätzlich die Möglichkeit, dass die Entscheidung über die Fixierung einer Kooperation getroffen werden muss, bevor die Kooperationsmöglichkeiten der Eltern- und Kinds-knoten bekannt sind. In diesem Falle bleibt nichts anderes übrig, als wie auch schon bei der Definition der Kompensationskosten für abgegebene Aufträge und Teilfunktionen (siehe 5.3.4.1) auf historische Daten zurückzugreifen. Diese werden sinnvollerweise in Abhängigkeit von der Länge des Commit-Window der betreffenden Aktivitäten



definiert, sodass die verbleibende Zeit, alternative Kooperationen zu realisieren, berücksichtigt wird.



## 6 Integrationszenario für die Planungsprozesse

Die in den Kapiteln 3, 4 und 5 beschriebenen Konzepte stellen die Bausteine einer Verhaltensplanung für mechatronische Systeme dar. Diese Bausteine sind so zusammenzufügen, dass sie ein Gesamtkonzept ergeben. Durch die Entscheidung, alle Problemfelder auf Basis des Formalismus der PDDL anzugehen, ist eine Schnittstelle zwischen den einzelnen Konzepten gegeben: Die einzelnen Verfahren können durch den Austausch von Plänen miteinander verbunden werden. So entsteht eine Art von Workflow, in dem die Verfahren Plandokumente miteinander austauschen und schrittweise weiterverarbeiten. Daher bietet sich eine zentrale Instanz an, in der die Verfahren die bearbeiteten Pläne ablegen und zur Weiterverarbeitung bereitstellen. Diese zentrale Instanz ist die Planbibliothek.

In der Planbibliothek kann eine Menge von Plänen abgelegt werden, die Beziehungen zueinander besitzen. Dabei wird ein Plan zerlegt in seine Aktionen und Zustände. Die Planbibliothek ist also:

- eine Menge  $PP$  von Planungsproblemen,
- eine Menge  $P$  von vollständig geordneten Plänen, wobei
- $p_i = S_i \cup A_i$ ,  $p_i \in P$  und
- $S_i$  die Zustände und  $A_i$  die Aktionen sind, die im Plan vorkommen.

Das ursprüngliche Planungsproblem des mechatronischen Systems wird als  $pp^*$  bezeichnet. Im weiteren Verlauf werden neue Planungsprobleme bestimmt, die zur Generierung von Alternativplänen oder zur Überarbeitung eines Plans erzeugt werden. Zwischen den Elementen der Menge  $S_i$  und  $A_i$  besteht eine Vorgänger-Nachfolger-Relation.

- $s_j \leftarrow a_k$ , wenn Aktion  $a_k$  im Zustand  $s_j$  ausgeführt wird
- $a_k \mapsto s_j$ , wenn Aktion  $a_k$  den Zustand  $s_j$  erzeugt<sup>52</sup>

Den Start des Workflows stellt eine gewöhnliche PDDL-Planung dar. Die Formulierung des initialen Planungsproblems berücksichtigt Blackbox-Modellierung von Systeminteraktionen (Kapitel 5.2), berücksichtigt also bereits die Interaktion mit Systemen, die dieselbe Hauptaufgabe erfüllen. Ebenso können die abzugebenden Teilfunktionen in das initiale Planungsproblem integriert werden, wodurch entsprechend des Ablaufs in Unterkapitel 5.3.4 mehrere alternative Pläne Ergebnis des Schritts *initiale Planung* sind. Jeder der alternativen Pläne steht dabei für eine andere Menge an Teilfunktionen, die abgegeben werden sollen.

Im nächsten Verarbeitungsschritt können die alternativen Pläne einer kontinuierlichen Analyse und Optimierung im Sinne des Unterkapitels 3.2.2 unterzogen werden. Können alle Aktionen der Planungsdomäne mithilfe vorgelagerter Optimierung dargestellt werden, entfällt dieser Schritt des Workflows. Anschließend wird derjenige Plan ausgewählt, der mit der Menge der erfolgreich an andere Systeme abgegebenen Teilfunktionen korrespondiert. Dieser Plan stellt die Lösung eines deterministischen Planungsproblems dar und wird in eine *Planbibliothek* eingestellt. Auf der *Planbibliothek* arbeiten drei weitere Planungsprozesse.

Die *probabilistische Analyse* (vergleiche Kapitel 4.2) analysiert die in der Bibliothek vorliegenden Pläne hinsichtlich möglicher Abweichung in der Planausführung. Die Analyse erzeugt

---

<sup>52</sup> In beiden Fällen muss natürlich gelten:  $z_j \in S_i$  und  $a_k \in A_i$  und  $p_i = S_i \cup A_i$ .

wie beschrieben eine Menge relevanter Verzweigungen, für die Alternativpläne zu erstellen sind. Zwischenergebnis der probabilistischen Analyse ist also:

- eine Menge  $B$  von Verzweigungen,
- zu jeder Verzweigung  $b \in B$  eine Bedingung, unter der sie angewandt wird, und
- ein neues deterministisches Planungsproblem  $pp \in PP$ , das durch den deterministischen Planungsablauf verarbeitet werden kann

Entsprechend ihrer *Selectability* (siehe 4.2.1.6) werden die neuen Planungsprobleme im deterministischen Planungsablauf gelöst. Ist die Planung abgeschlossen, wird der neue Alternativplan in die Planbibliothek integriert. Dabei besteht eine Relation zur entsprechenden Verzweigungsstelle:

- $p \mapsto b$ , wenn  $p$  der Plan ist, der auszuführen ist, wenn die Bedingung von  $b$  eintritt.

Die Abstimmung von Aktivitäten (vergleiche Kapitel 5.3.5) sucht anhand der in den Plänen enthaltenen Informationen nach möglichen Kooperationspartnern und versucht mit diesen einen gemeinsamen Plan zu generieren. Um den neuen Plan zu generieren, wird wieder ein neues Planungsproblem  $pp'$  erzeugt. Die Kooperation und das Planungsproblem  $pp'$  beziehen sich dabei auf genau einen alternativen Plan  $p$  zur Lösung des ursprünglichen Planungsproblems. Ist die Abstimmung in mehreren Verzweigungen möglich, ist jeweils ein eigenes Planungsproblem  $pp$  notwendig. In Abschnitt 5.3.4.1 wurde die Funktion  $req(s_i)$  eingeführt, die jedem Zustand  $s_i$  aus dem folgenden Plan die Zustandsvariablen bestimmt, die gegeben sein müssen, damit der auf  $s_i$  folgende Plan ausführbar bleibt. So kann für das Planungsproblem  $pp'$  ein Zielzustand  $s_{ziel}$  bestimmt werden, der eine Einbettung in den Ursprungsplan ermöglicht. Ist das Planungsproblem  $pp'$  gelöst, werden die Zustände und Aktionen zwischen den Zuständen  $s_{start}$  und  $s_{ziel}$  des ursprünglichen Plans  $p$  ersetzt durch die Aktionen und Zustände des neuen Teilplans  $p'$ , der die abgestimmten Aktionen enthält.

Der Prozess *Abgabe von Aufträgen* bearbeitet im Gegensatz zu den anderen beiden Planungsprozessen nicht nur einen einzelnen Plan in der Planbibliothek. Stattdessen sind alle Pläne betroffen, wenn ein Auftrag erfolgreich abgegeben wurde. Jeder in der Bibliothek enthaltene Plan muss durch einen neuen Plan ersetzt werden, der den oder die abgegebenen Aufträge nicht mehr enthält. Auch hier kann die Funktion  $req(s_i)$  eingesetzt werden, um den Planungsaufwand zu beschränken.

Im Rahmen des Workflows sind also drei unterschiedliche Typen von Plänen zu bearbeiten und müssen entsprechend in der Bibliothek abgelegt werden:

1. Die ursprüngliche deterministische Lösung eines Planungsproblems;
2. Alternativpläne, die in bestimmten Planschritten von einem anderen Plan abzweigen;
3. Teilpläne, die durch Kooperation mit anderen mechatronischen Systemen eine Verbesserung des Plans bewirken.

Ein Plan in der Bibliothek ist der jeweils aktuell auszuführende Plan. Auf diesem Plan kann die Koordination nach der Planung für Systeme mit gemeinsamer Hauptaufgabe durchgeführt werden (vergleiche Unterkapitel 5.2.4). Da dieser Prozess jeweils einen Teil des Plans fixiert und zur Ausführung *freigibt*, ist das Ergebnis eine Sequenz von durchzuführenden Aktivitäten. Im Rahmen der Planausführung wird der tatsächliche Planungsverlauf mit den in der Planbibliothek enthaltenen Alternativplänen abgeglichen, um ggf. den aktuell auszuführenden Plan wechseln zu können.

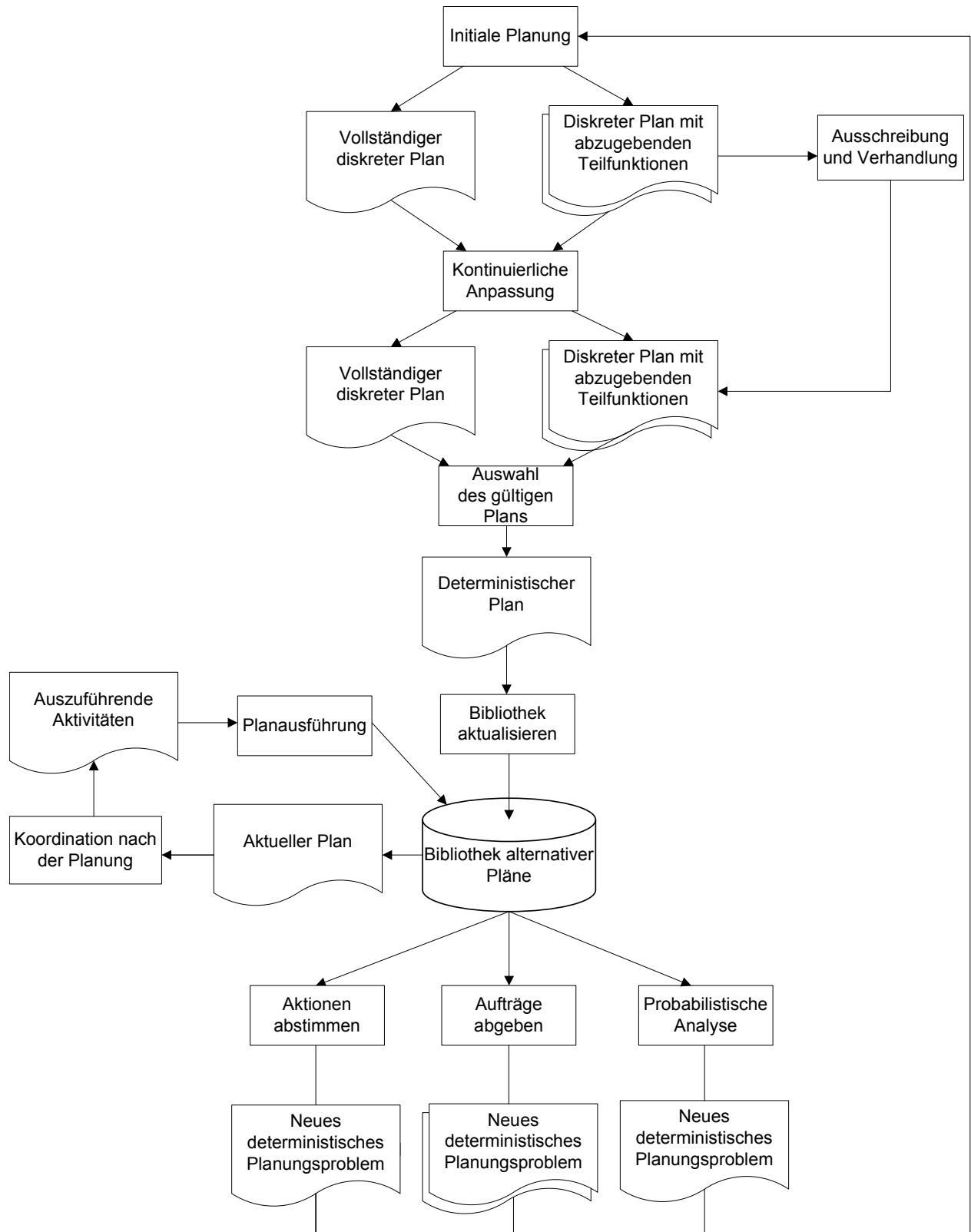


Abbildung 6-1: Integrationszenario für die Planungsverfahren

Der Auswahlprozess zwischen den Einigungen bedarf noch einer genaueren Erläuterung. Zunächst bedingt er, dass die Einigungen, die Ergebnis der vorherigen Verhandlungen sind, nur vorläufig sind. Weiterhin ist zu klären, wie das mechatronische System zwischen den möglicherweise konkurrierenden Einigungen auswählt. Als Datenstruktur zur Verwaltung der Einigung bietet sich die in Abbildung 5-12 eingeführte Baumstruktur an. Wird eine beliebige

Kooperationsaktion fixiert, sind alle weiteren Kooperationsaktionen, die in der Baumstruktur oberhalb oder unterhalb des entsprechenden Knotens in der Baumstruktur liegen, zu entfernen und den entsprechenden Systemen eine Absage zu erteilen. Daran ist ersichtlich, dass vor einer Fixierung die positiven Effekte einer Kooperation abgewogen werden müssen gegen diejenigen, die dadurch verworfen werden. Stehen bereits alle Einigungen über mögliche Kooperationsaktionen endgültig fest, so ist dies auch ohne Weiteres möglich. Es liegen dann für jede mögliche Kooperationsaktion die Verhandlungsergebnisse vor, sodass die resultierende Planqualität errechnet werden kann.

Treten unvorhergesehene Abweichungen auf, ist eine Umplanung im Sinne des Unterkapitels 4.2.2 aufzurufen (nicht in der Abbildung). Da hier besondere Anforderungen an die Planungsgeschwindigkeit zu stellen sind, werden in der Regel die kontinuierliche Analyse und die kooperativen Verbesserungsaktionen wegfallen. Daher muss es Ziel der Umplanung sein, möglichst schnell auf einen Zustand in einem der in der Planbibliothek enthaltenen Pläne zurückzukehren. Abbildung 6-1 stellt den Gesamtablauf dar.

Sind im betrachteten Anwendungsfall die einzigen kooperierenden Systeme mit gemeinsamer Hauptaufgabe die Funktionsmodule innerhalb der autonomen mechatronischen Systeme (AMS), bietet sich eine Alternative zum hier beschriebenen Ablauf an. In diesem Fall können die Planungsprozesse hierarchisch organisiert werden. Auf einer überlagerten Ebene planen die AMS mithilfe eines vereinfachten Planungsmodells. Kontinuierliche Aspekte treten hier nur in Form von Kostenwerten auf, die aus den detaillierten Planungsmodellen der Funktionsmodule abgeleitet werden (z. B. Komfortwerte abhängig vom Streckentyp). Auf dieser Ebene führen die AMS zunächst alle Kooperationsaktionen für interagierende mechatronische Systeme mit individueller Hauptaufgabe durch. Ergebnis dieses Planungsprozesses ist ein grober Ablaufplan für die Aufgabenerfüllung der mechatronischen Systeme. Dieser Planungsprozess wird auf der Ebene der Funktionsmodule verfeinert. Hier finden eine detaillierte Abstimmung der Aktivitäten innerhalb des AMS (mittels der Interaktionsplanung für Systeme mit gemeinsamer Hauptaufgabe) und eine genauere Analyse des kontinuierlichen Verhaltens statt. Treten auf dieser Ebene unlösbare Planungsprobleme auf oder ergibt sich eine zu schlechte Planqualität (im Verhältnis zur erwarteten Planqualität im groben Ablaufplan), ist ggf. eine Umplanung auf der Ebene der AMS notwendig.

## 7 Anwendungsbeispiele

Die folgenden Anwendungsbeispiele beziehen sich alle auf das RailCab-System der *Neuen Bahntechnik Paderborn*. Innerhalb des RailCab-Systems bietet sich die Möglichkeit, die Planungsprozesse in den unterschiedlichen mechatronischen Ebenen hierarchisch zu verknüpfen. Dabei planen die Fahrzeuge auf der obersten Hierarchieebene ihre Routen, um eine Menge von Transportaufträgen durchzuführen. Natürlich können die Fahrzeuge dabei miteinander interagieren und so Konvois bilden oder Transportaufträge austauschen. Die zweite Planungsebene, die auf Ebene der MFM angesiedelt ist, erhält eine Abfolge von Streckenabschnitten als Eingabe. Diese Abfolge ist Ergebnis der Planungsprozesse auf der oberen Planungsebene. Hier wählt der Antrieb beispielsweise seine Fahrprofile so aus, dass die Route in der entsprechenden Zeit abgefahren wird. Abbildung 7-1 veranschaulicht die Planungsebenen und ihre Interaktion.

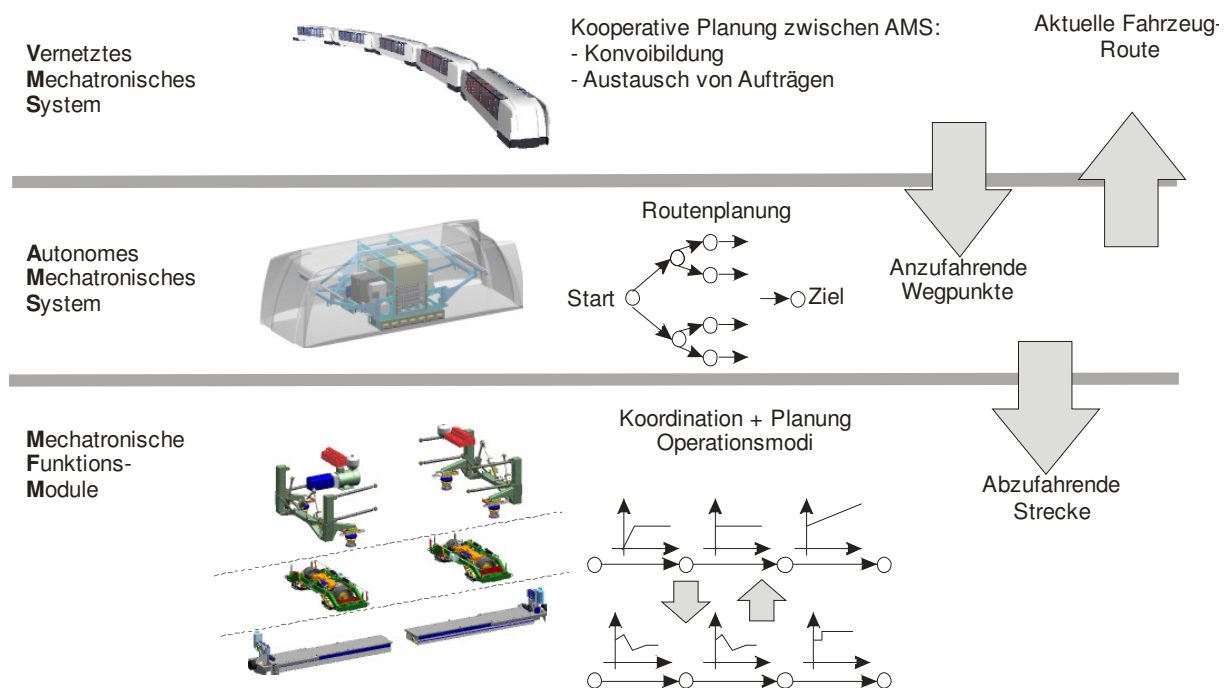


Abbildung 7-1: Planungshierarchie im RailCab-Beispiel

Zur Planung der Abläufe im RailCab-Verkehrssystem (VMS-Ebene) wird ein Multiagenten-System eingesetzt. Ein Grund für die Wahl dieses dezentralen Planungsparadigmas ist zunächst die hohe Zahl von zu berücksichtigenden Systemen: Kunden, Fahrzeuge, Streckenabschnitte und Stationen müssen hinsichtlich ihrer Anforderungen und Kapazitäten abgestimmt werden. Gleichzeitig muss das System laufend auf neue Eingaben reagieren können. So können jederzeit neue Kunden und damit Transportaufträge in das System eintreten oder Streckenausfälle müssen berücksichtigt werden.

Das Multiagenten-System ist in seiner ursprünglichen Fassung ausführlich in [Frank04] beschrieben. Das System umfasst vier Arten von Agenten: Streckenagenten, Stationsagenten, Fahrzeugagenten und Kundenagenten. Abbildung 7-2 zeigt die Abläufe im RailCab-MAS als stark vereinfachtes UML-Sequenzdiagramm. Ausgangspunkt der Planungsaktivitäten ist die Bekanntgabe eines Transportauftrags an die Fahrzeugagenten. Jeder Fahrzeugagent, der grundsätzlich an dem Transportauftrag interessiert ist, führt zunächst eine lokale Routenplanung durch. Diese Routenplanung dient der Überprüfung, ob der Transportauftrag zur aktuel-

len Fahrzeugroute passt. Ist dies der Fall, werden in einem iterativen Prozess (Details siehe [Frank04]) die Überfahrten über Streckenabschnitt und Aufenthaltszeiten in Stationen bei den entsprechenden Agenten gebucht.

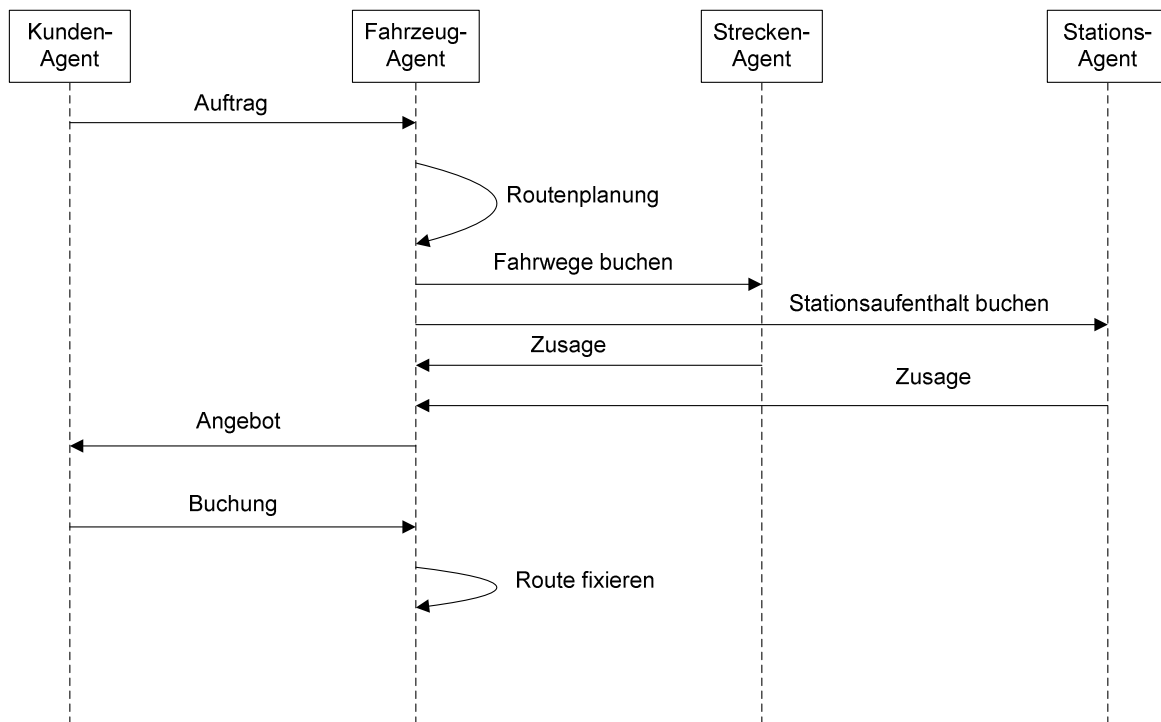


Abbildung 7-2: Vereinfachte Darstellung der Abläufe im RailCab-MAS

Formal betrachtet löst das RailCab-MAS eine Onlinevariante des *Pick Up and Delivery Problem with Time Windows (PDPWTW)*. Dabei handelt es sich um eine Onlinevariante, da die Transportaufträge nur sukzessiv ins System eintreten, aber unverzüglich eingeplant werden müssen (da die Zeitfenster der Kunden eingehalten werden müssen). Dadurch können Situationen eintreten, in denen eine ursprünglich getroffene Entscheidung als nicht optimal oder sogar als schlecht erkannt wird. Bis zur Ausführung des Plans (oder genauer: bis zur Ausführung einer Aktivität im Plan) kann sich die Situation im System so verändert haben, dass eine Anpassung des ursprünglichen Plans sinnvoll ist.

Dabei wird das Systemverhalten anhand von Kostenfunktionen bewertet. Die individuelle Kostenfunktion eines Fahrzeugs ermittelt sich anhand der Kosten, die für Benutzung von Strecken und von Stationen anfallen:

$$cost_i = \sum_{t=1}^n c_t + e_t + \sum_s^m c_s (d_s - a_s)$$

Dabei bezeichnet  $c_t$  die Maut für das Befahren eines Streckenabschnitts,  $e_t$  die Kosten für den Energieverbrauch auf dem Streckenabschnitt,  $c_s$  einen Kostensatz für das Halten in einer Station sowie  $a_s$  die Ankunftszeit in einer Station und  $d_s$  die Abfahrtszeit. Zur Beurteilung der Leistung des Gesamtsystems und der Planungsmechanismen werden die lokalen Kostenfunktionen aufsummiert:

$$cost_g = \sum_{i=1}^o cost_i$$



## 7.1 Integration kontinuierlicher Verläufe für die aktive Federung

Die Integration der kontinuierlichen Verläufe der Aktivitäten eines mechatronischen Systems wird am Beispiel der aktiven Federung des RailCabs erläutert. Wie bereits in Kapitel 5 beschrieben, ist das für die Federung entscheidende Merkmal des Komforts nur vor dem Hintergrund eines kontinuierlichen Verlaufs zu bestimmen. Dies geschieht durch die Integration der Beschleunigungen, die am Aufbau des RailCabs wirken.

Im Falle der aktiven Federung wurde ein Vorgehen auf Basis einer vorgeschalteten Optimierung gewählt. Die Optimierung ist nicht Bestandteil dieser Arbeit, sondern in [VT08] beschrieben. Die vorgelagerte Optimierung bestimmte diejenigen Reglerparameter und Einstellungen, die bei einem vorgegebenen maximalen mittleren Leistungsbedarf (Nebenbedingung) den optimalen Komfort liefern. Die Parameter, die im Rahmen der Optimierung angepasst wurden, waren im Einzelnen:

- jeweils in horizontaler, vertikaler und Drehrichtung (um die Längsachse), Einstellungen für:
  - die virtuelle Relativfedersteifigkeit ( $c_{rel}$ ),
  - den virtuellen Relativ-Dämpfer ( $d_{rel}$ ),
  - den Skyhook-Dämpfer ( $d_{sky}$ ),
- der Versorgungsdruck ( $p_{supp}$ ) und
- Aktivierung einer Störgrößenkompensation.

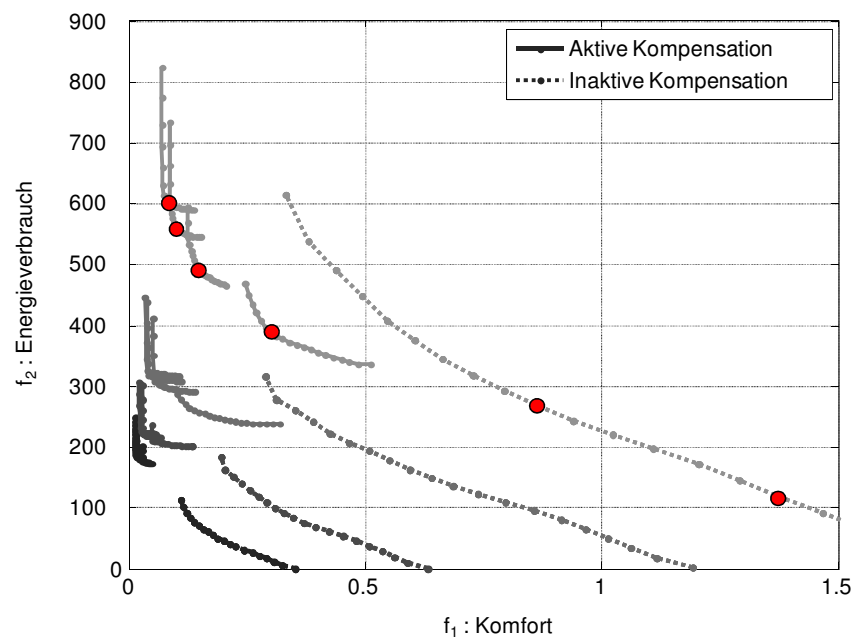


Abbildung 7-3: Pareto-Mengen im Bildraum für die Federung [VT08]

Das Optimierungsproblem wurde für bestimmte Gewichtungen der Ziele gelöst. Ergebnis sind also Kombinationen aus Komfort und mittlerer Leistung. Diese entsprechen einer sinnvollen Zielkombination. Als relevante Umfeldvariablen wurden die Amplitude (Stärke) und Frequenz (Häufigkeit) der Streckenanregungen verwendet. Das Optimierungsproblem wurde für eine Menge von typischen Werten dieser beiden Eigenschaften gelöst. Ergebnis der Optimierung ist eine Pareto-Menge, die die effizienten Kompromisse der beiden Zielfunktionen „maximiere Komfort“ ( $f_1$ ) und „minimiere mittlere Leistung“ ( $f_2$ ) enthält. Abbildung 7-3 zeigt die Ergebnisse der Optimierung für unterschiedliche Streckentypen und mit aktiver oder inaktiver Störgrößenkompensation. Die fett hervorgehobenen Punkte auf den sich ergebenden

Pareto-Mengen sind die ausgewählten Teilfunktionslösungen, die im Rahmen der Planung betrachtet werden sollen. Aus den Werten für Komfort und mittlere Leistung ergeben sich die Nachbedingungen der Anwendung einer solchen Teilfunktionslösung, nämlich die Güte der Aktion (Komfort) und ihr Ressourcenverbrauch (mittlere Leistung über die Zeit). Damit kann für einen Streckenabschnitt bekannter Länge grundsätzlich eine entsprechende Aktion oder Teilfunktionslösung formuliert werden. Damit war zuletzt noch zu klären, zu welchem Streckentyp ein konkreter Streckenabschnitt in der Planung gehört. Wie bereits in Abschnitt 3.2.1.1 beschrieben, ist für diese Aufgabe grundsätzlich ein Klassifikationsverfahren geeignet.

```
(:action suspensionComfort
  :parameters
    (?r railcab ?t tracksection)
  :precondition
    (and on ?r ?l1)
    (>=(energy_storage ?r)(*distance ?l1 ?l2)(suspension-active ?r)
  :effect
    (
      when( (is ?t trackTypA) (increase used_energy (* (distance
        ?t)(avg_power ?trackTypA) and increase comfort (* comfortLevel-
        SuspensionComfort(?trackTypA) (distance ?t))
      when( (is ?t trackTypB) (increase used_energy (* (distance
        ?t)(avg_power ?trackTypB) and increase comfort (* comfortLevel-
        SuspensionComfort(?trackTypB) (distance ?t))
      ...
      when( (is ?t trackTypN) (increase used_energy (* (distance
        ?t)(avg_power ?trackTypN) and increase comfort (* comfortLevel-
        SuspensionComfort(?trackTypN) (distance ?t))
    )
```

**Listing 7-1: Aus der Optimierung abgeleitete Aktionsdefinition**

Der Eigenschaftsvektor, der dazu verwendet wurde, enthält die Amplitude und die Frequenz der Anregung. Je stärker diese Werte voneinander abweichen, desto unähnlicher sind sich zwei Streckenabschnitte. Die einfache Aufgabe, einen konkreten Streckenabschnitt einem der charakteristischen zuzuordnen, kann z. B. mithilfe eines Naive Bayes Classifiers gelöst werden.

Die Aktivität des *Federns* liess sich nun für unterschiedliche Zielkonfigurationen und Streckentypen formulieren. Listing 7-1 zeigt die Formulierung einer bedingten Aktion. Für jeden in der vorgelagerten Optimierung behandelten Streckentyp ist ein bedingter Aktionsteil enthalten. Wenn der aktuell zu planende Streckenabschnitt einem Typen entspricht, wird die entsprechende mittlere Leistung mit der Länge des Streckenabschnitts multipliziert, um auf die tatsächlich verbrauchte Energie zu kommen. Das Gleiche geschieht mit dem Komfortwert, der innerhalb einer Planmetrik als Optimierungsziel genutzt werden kann. Der Streckentyp wird wie beschrieben vor einem Planungslauf durch ein Klassifikationsverfahren bestimmt.

Durch die Definition dieser Aktion sind die kontinuierlichen Einstellungen und Reglerparameter sowie das kontinuierliche Gütekriterium Komfort in die diskrete Planung integriert.

## 7.2 Planung unter Unsicherheit

### 7.2.1 Probabilistische Routenplanung für bedarfsorientierten Schienentransport

Auf der Ebene des AMS muss ein RailCab-Fahrzeug wiederholt seinen Weg planen, um zugeordnete Transportaufträge zu erfüllen. Klassischerweise wird dies durch ein deterministisches Kürzeste-Wege-Verfahren gelöst, das die Fahrzeiten zwischen zwei Streckenpunkten als sicher bekannt voraussetzt.

In diesem Kapitel wird ein Verfahren vorgestellt, das mithilfe der in Kapitel 4.2 beschriebenen Expertenagenten eine genauere Prognose von Fahrzeiten realisiert als Verfahren auf Basis unbedingter Durchschnittswerte und dadurch zu einer besseren Routenplanung für die RailCab-Fahrzeuge führt. Das Verfahren wurde umgesetzt innerhalb des RailCab-MAS, das im vorherigen Abschnitt beschrieben wurde. Dazu wurden sowohl der Track-Agent als auch der Fahrzeugagent erweitert.

### 7.2.2 Die Agentenklasse Track-Experte

Die Prognose für die Fahrtdauer auf einem einzelnen Streckenabschnitt wird wie in Abschnitt 4.2.1.3 beschrieben mittels Bayes-Netzwerken modelliert. In dieses Netzwerk geht die erwartete Ankunftszeit des Fahrzeugs als Eingabe ein. Das Netz selbst beschreibt die Wirkung von Umfeldeinflüssen auf die Fahrtdauer. Von Faktoren wie Verkehrsdichte oder Witterungsbedingungen hängt die Fahrzeit aber nur mittelbar über die durchschnittliche Geschwindigkeit ab. Sie beeinflussen die auf dem Streckenabschnitt realisierbare Durchschnittsgeschwindigkeit  $sv_{avg}$ . Aus ihr ergibt sich die Fahrtdauer rechnerisch als  $rd = \frac{Länge}{|v_{avg}|}$  mit  $v_{avg} = \min\{fv_{max}, sv_{avg}\}$ , wobei *Länge* die Länge des Streckenabschnitts und  $fv_{max}$  die Höchstgeschwindigkeit des Fahrzeugs bezeichnen.<sup>53</sup> Der Track-Experte hat also folgende Wirkungskette zu modellieren:

*Ankunftszeit* → *Umfeldeinflüsse* → *realisierbare Durchschnittsgeschwindigkeit*  
→ *Fahrzeit*

Zur Modellierung dieser Wirkungskette wird der Track-Experte mit einem Bayes-Netzwerk ausgestattet, das die Zusammenhänge seiner Umgebung und die Einflüsse der Umgebung auf die Fahrzeit darstellt. Es wird ein hybrides Bayes-Netzwerk eingesetzt, das sowohl kontinuierliche als auch diskrete Zufallsvariablen enthält. Die verwendeten Netzwerke umfassen mindestens die folgenden Knoten:

- Der Knoten Arrival wird als kontinuierliche Zufallsvariable betrachtet und von Fahrzeugagenten mit einer Verteilung der geplanten Ankunftszeit initialisiert.
- Der Knoten Ride Duration (Fahrzeit) ist eine normalverteilte Zufallsvariable. Sie wird direkt aus den Verteilungen der Umfeldvariablen abgeleitet, die prognostizierte Fahrtdauer hat also die Form  $P(\text{Ride Duration} | e)$ .

Der rein funktionale Zusammenhang zwischen realisierbarer Durchschnittsgeschwindigkeit und Fahrzeit wird also nicht explizit modelliert oder berechnet, sondern im Netzwerk implizit in der Verteilung  $P(\text{Ride Duration} | e)$  dargestellt. Abbildung 7-4 zeigt die Struktur der verwendeten Netzwerke und veranschaulicht die Interaktion zwischen Fahrzeug- und Expertenagent.

<sup>53</sup> Dieser Zusammenhang basiert auf der Annahme, dass die Fahrzeuge jeweils mit der höchstmöglichen Geschwindigkeit fahren.

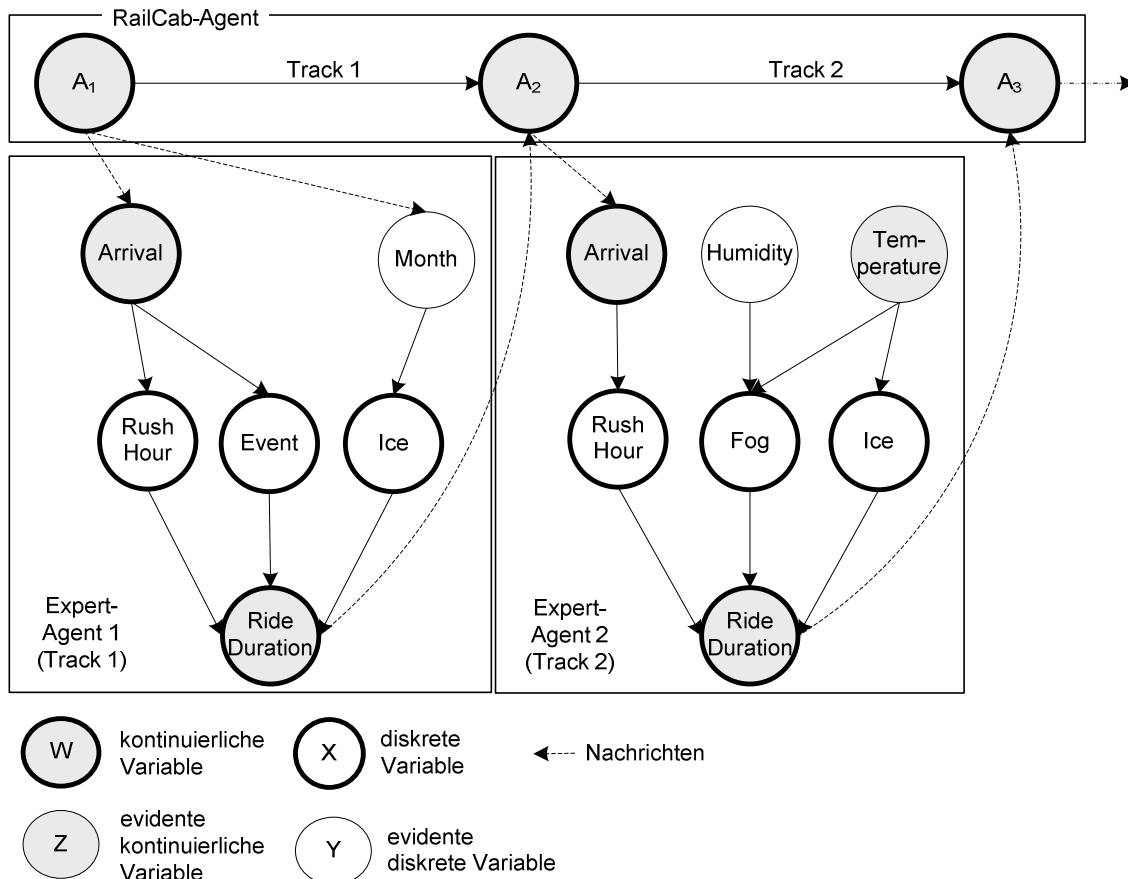


Abbildung 7-4: Zwei mit einem Fahrzeugagenten kommunizierende Track-Experten

Zur Durchführung der Inferenz wurde eine eigene Bibliothek mit dem Namen pns (probability network support) implementiert. Diese Bibliothek oder Package enthält alle Klassen, die für den Aufbau und die Verarbeitung der Bayes-Netzwerke in den Expertenagenten notwendig sind. Abbildung 7-5 zeigt ein Klassendiagramm des Packages.<sup>54</sup>

Um die Experten einfach mit individuellen Netzwerken ausstatten zu können, wurde auf Basis des XML-Formats XMLBIF<sup>55</sup> eine Erweiterung namens XBIF (eXtended Bayesian Network Interchange Format) entwickelt, die auch hybride Netzwerke im XML-Format ablegen kann.<sup>56</sup> Der Konstruktor der Klasse BeliefNet erwartet als Eingabe den Namen einer solchen Datei und erzeugt anschließend das entsprechende Netzwerk.

Die Klasse BeliefExpert ist eine Spezialisierung der Klasse BeliefNet und stellt die Methoden zur Inferenz zur Verfügung. Mithilfe dieser Methoden ist es möglich, harte oder weiche Evidenzen zu setzen, Evidenzen zu entfernen und a-posteriori-Wahrscheinlichkeiten zu berechnen.

Im Rahmen der Umsetzung wurde der Track-Agent aus dem ursprünglichen RailCab-MAS so erweitert, dass er in der Lage ist, Anfragen bzgl. einer Prognose durch die Fahrzeugagenten anzunehmen und mittels der im Package pns implementierten Klassen zu beantworten.

<sup>54</sup> Um eine kompakte Darstellung zu erreichen, wird im Diagramm der kompaktere Begriff Belief Net anstelle von Bayes-Netzwerk gewählt.

<sup>55</sup> <http://www.cs.cmu.edu/~javabayes/Home/index.html>, 2001.

<sup>56</sup> Die DTD sowie Beispiele für XML Dateien finden sich im Anhang.

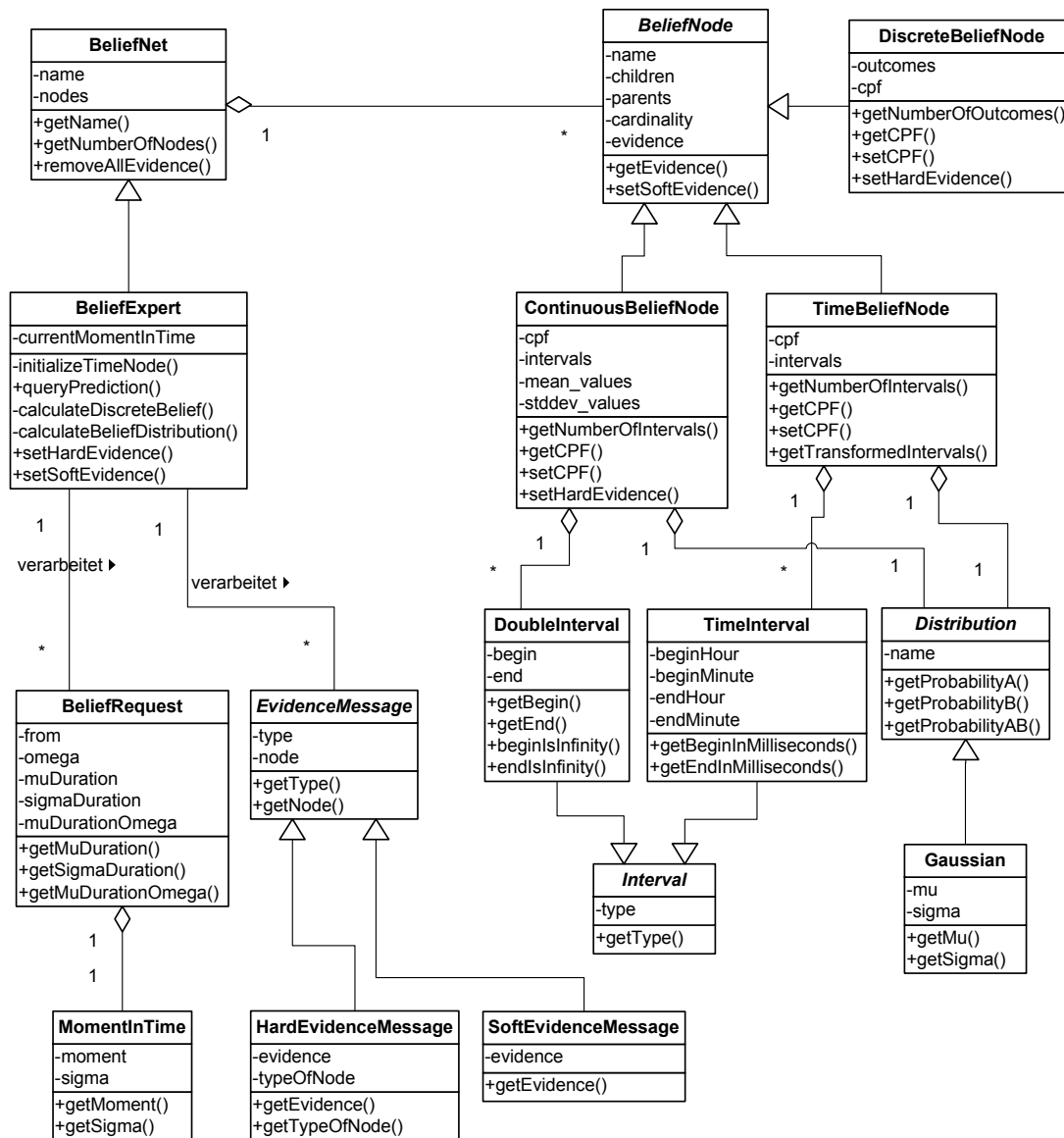


Abbildung 7-5: Klassendiagramm des Packages PNS

### 7.2.3 Erweiterung des Fahrzeugagenten

Um die verbesserte Routenplanung zu realisieren, musste auch der Fahrzeugagent des ursprünglichen Systems erweitert werden. Als der eigentlich planende Agent und Repräsentant eines autonomen mechatronischen Systems war der Fahrzeugagent mit einem Planungsalgorithmus auszustatten, der die Informationen der Expertenagenten nutzt. Dazu wurde eine spezielle Version von Dijkstras Kürzeste-Wege-Algorithmus [CLRS01] entwickelt.

Eine weitere Möglichkeit, die sich dem Fahrzeugagenten durch die Behandlung der Fahrzeiten als Zufallsvariablen bietet, ist die Umsetzung eines differenzierten Verhaltens bzgl. Verspätungen. Es kann berücksichtigt werden, ob eine Route mit zuverlässiger Ankunftszeit und möglicherweise längerer Fahrzeit bevorzugt wird oder eine Route, die die Chance einer möglichst schnellen Ankunft bei geringerer Zuverlässigkeit bietet. Um diese Möglichkeit zu realisieren, wird der Erwartungswert der Fahrzeit mit einer Gewichtung für die Möglichkeit einer Verspätung belegt. Dieses Gewicht ergibt sich aus der Wahrscheinlichkeit, dass sich eine Verspätung maximal in Höhe eines Anteils der erwarteten Fahrzeit ergibt:

$$rd_\omega = \mu_{\text{Fahrzeit}}[1.5 - P(\mu_{\text{Fahrzeit}} \leq \text{Fahrzeit} \leq (1 + \omega)\mu_{\text{Fahrzeit}})]$$

Der gewichtete Erwartungswert  $\mu_{\text{Fahrzeit}}$  wird vom Dijkstra-Algorithmus als Kostenwert angesetzt. Der Parameter  $0 \leq \omega \geq$  definiert den Anteil der erwarteten Fahrzeit, in dessen Höhe mögliche Verspätungen nicht beachtet werden. Liegen annähernd alle Verspätungen innerhalb des Intervalls, gilt  $rd_\omega \approx \mu_{\text{Fahrzeit}}$ . Liegen dagegen viele Verspätungen außerhalb dieses Intervalls, gilt  $rd_\omega > \mu_{\text{Fahrzeit}}$ . Die Möglichkeit der Verspätung wird also bestraft. Listing 7-2 zeigt den Pseudo-Code des (BeliefDijkstra genannten) erweiterten Kürzeste-Wege-Algorithmus.

#### BeliefDijkstra

```

s // Startstation
z // Zielstation
adj[] // Feld mit den Adjazenzlisten aller Knoten
length[][] //zwei dimensionales Feld mit den Entfernungen
vmay // technische Höchstgeschwindigkeit

//Initialisierung wie im Standard-Dijkstra
For i := 1 to n
    S[i] = true
    Fahrzeit[i] := ∞
    wFahrzeit[i] := ∞
    pred[i] = null
tue

//Der Startort ist von sich selbst 0 LE entfernt
Fahrzeit[s] := 0.0
wFahrzeit[s] := 0.0
//Erwartungswert und Standardabweichung der Abfahrzeit
μA[s] := Abfahrtszeit
σA[s] := 0.0.
Solange S nicht leer ist
    Wähle u mit minimalen wFahrzeit[u] und S[u] = true
    S[u] := false;
    Wenn u = Z Dann STOP
    Über alle v ∈ adj[u] with S[v]= true
        Wenn (wFahrzeit[v] > wFahrzeit[u]+length[u,v]/velocity) Dann
            Frage Fahrzeitverteilung für (u,v) für Ankunft (μA[u],σA[u])
ab
    Warte auf Prognose von μFahrzeit und σFahrzeit, rdω und Pω ab
    Wenn (wFahrzeit[v] > wFahrzeit + rdω) Dann
        Fahrzeit[v] := wFahrzeit[u] + rdω
        Vorgänger[v] := u
        con(v) := (0.5 + Pω)
        μA[v] := add(μA[v], μFahrzeit)
        σA[s] := √(σA[s]2 + σFahrzeit[s]2)

```

Ende

Listing 7-2: Pseudo-Code des BeliefDijkstra

## 7.2.4 Experimente und Ergebnisse

Um die Funktionsweise des in den vorherigen Kapiteln vorgestellten Verfahrens zu überprüfen und zu validieren, ist die Beantwortung der folgenden Fragen relevant:

- Wie gut sind die Fahrzeitprognosen?
- Wie wirken sich ungenaue a-priori-Verteilungen in den Bayes-Netzwerken der Experten-Agenten auf die Prognose aus?
- Falls sich die durch den BeliefDijkstra gewählte Route von der des Standard-Dijkstras unterscheidet, ergibt sich im Durchschnitt eine Fahrzeiterparnis?

Um die Experimente durchzuführen, wurden drei unterschiedliche Transportnetzwerke mit 100, 150 und 300 Stationen verwendet. Die Track-Agenten wurden mit Bayes-Netzwerken mit zwei unterschiedlichen Topologien ausgestattet (Typ A und Typ B), aus denen durch Anpassung der a-priori-Verteilungen der Umfeldeinflüsse insgesamt 12 Instanzen gebildet wurden.<sup>57</sup> Jedem Track-Agenten wurde zufällig eines der 12 Netzwerke zugewiesen.

Die einzelnen Testabläufe wurden nach folgendem Muster durchgeführt:

1. Routen planen
  - a. Planen der fahrtzeitoptimalen Route  $(s, \dots, z)_D$  mittels des normalen Dijkstra-Algorithmus. (Die Basis für die Fahrtzeitbewertung bildet die aus der Länge der Tracks und der Durchschnittsgeschwindigkeit des Shuttles ermittelte Fahrtdauer.)
  - b. Planen der fahrtzeitoptimalen Route  $(s, \dots, z)_{BD}$  mittels des BeliefDijkstra-Algorithmus. (Die Basis für die Fahrtzeitbewertung bilden die Prognosen der Track-Agenten.)
2. Für jedes Testschema n-Mal: Simulation der tatsächlichen Fahrtdauer der Routen
  - a. Für jeden Abschnitt  $t$  aus  $(s, \dots, z)_D$  und  $(s, \dots, z)_{BD}$ : Simulation der tatsächlichen Fahrtdauer.
    - i. Setzen der erwarteten Ankunftszeit, der Evidenz für den Knoten Monat, sowie weicher Evidenzen für die Knoten Feuchtigkeit und Temperatur.
    - ii. Für alle diskreten Knoten, für die keine Evidenz vorliegt: A-priori-Wahrscheinlichkeiten entsprechend dem aktuellen Testschema um einen festen Wert modifizieren.
    - iii. Verteilungen der Elternknoten der Fahrzeit entsprechend dem aktuellen Testschema, um einen festen Anteil modifizieren.
    - iv. Für alle Knoten oberhalb der Fahrzeit:
      1. A-posteriori-Wahrscheinlichkeiten berechnen
      2. Monte-Carlo-Ziehung einer evidenten Ausprägung aus der a-posteriori-Verteilung. (Das Ergebnis sind simulierte harte Evidenzen für alle Knoten oberhalb von Fahrzeit).
      3. Ziehen der simulierten Fahrtdauer  $\text{Fahrzeit}^{\text{sim}}_t$  aus der, durch die simulierten harten Evidenzen gegebenen, simulierten Verteilung für Ride Duration.
    - v. Bestimmen der simulierten Fahrzeit  $\text{Fahrzeit}^{\text{sim}}_D$  und  $\text{Fahrzeit}^{\text{sim}}_{BD}$  für  $(s, \dots, z)_D$  und  $(s, \dots, z)_{BD}$  als Summe der entsprechenden  $\text{Fahrzeit}^{\text{sim}}_t$ .

Das erste Testergebnis zeigt den Vergleich beider Verfahren bzgl. der Abweichung der im Planungsprozess vorhergesagten Fahrzeit von der simulierten Fahrzeit. Dabei wurden im Rahmen der Simulation die gleichen Verteilungen wie in der Prognose verwendet. Abbildung 7-6 zeigt die Ergebnisse für 23 Transportaufträge. Dargestellt ist die mittlere Abweichung von der durchschnittlichen simulierten Fahrzeit. Die Prognose des Beliefs-Dijkstras war in jedem Fall genauer als die des Standard-Dijkstras. In den allermeisten Fällen war das Verhältnis der Prognose sehr deutlich. Beim BeliefDijkstra lag die Abweichung zwischen 0,36% und 6,04%. Die durchschnittliche Abweichung betrug 2,68%. Dagegen lag die Abweichung für den Standard-Dijkstra zwischen 0,45% und 30,97%. Im Mittel lag die Abweichung bei 14,81%, also mehr als doppelt so hoch als die größten Abweichungen des BeliefDijkstras.

<sup>57</sup> Details zum Versuchsaufbau im Anhang

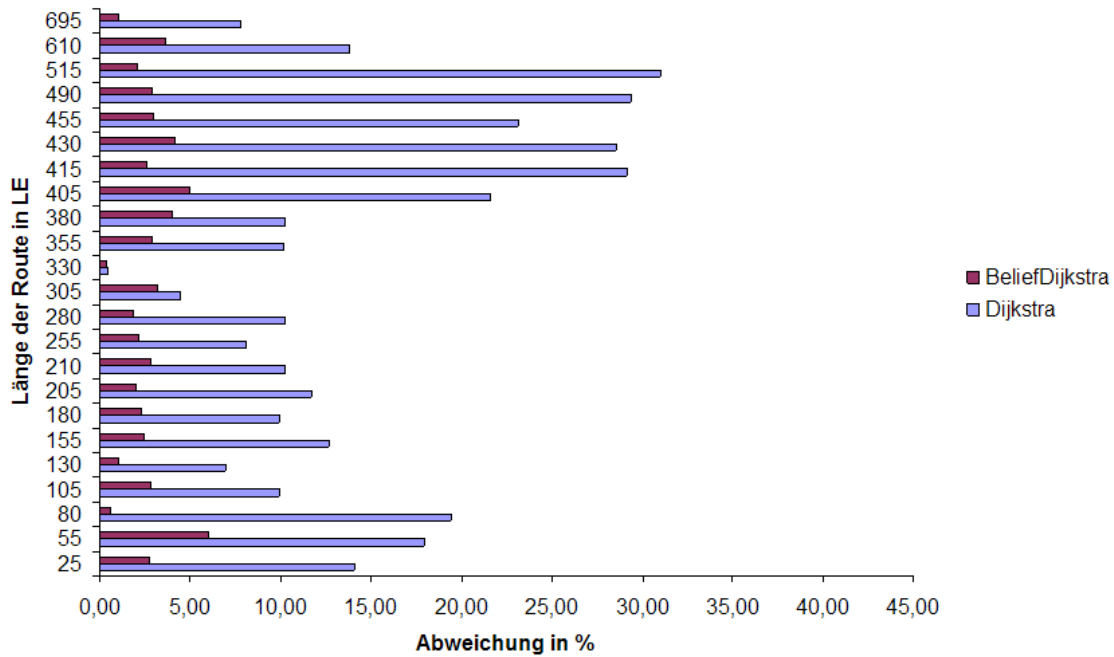


Abbildung 7-6: Ergebnisse bei exakten Prognoseverteilungen

Variiert man die in der Simulation eingesetzten Verteilungen (im Vergleich zu denen in der Prognose), so verändert sich das Bild natürlich etwas. Verschiebt man beispielsweise die  $\mu$ -Werte in den bedingten Normalverteilungen der Fahrzeit, verschlechterte sich der Belief-Dijkstra erheblich. Ab einer Abweichung der  $\mu$ -Werte von -8% ergaben sich in den Experimenten erstmals Ergebnisse, in denen der BeliefDijkstra schlechter abschnitt als der Standard-Dijkstra (Abbildung 7-7).

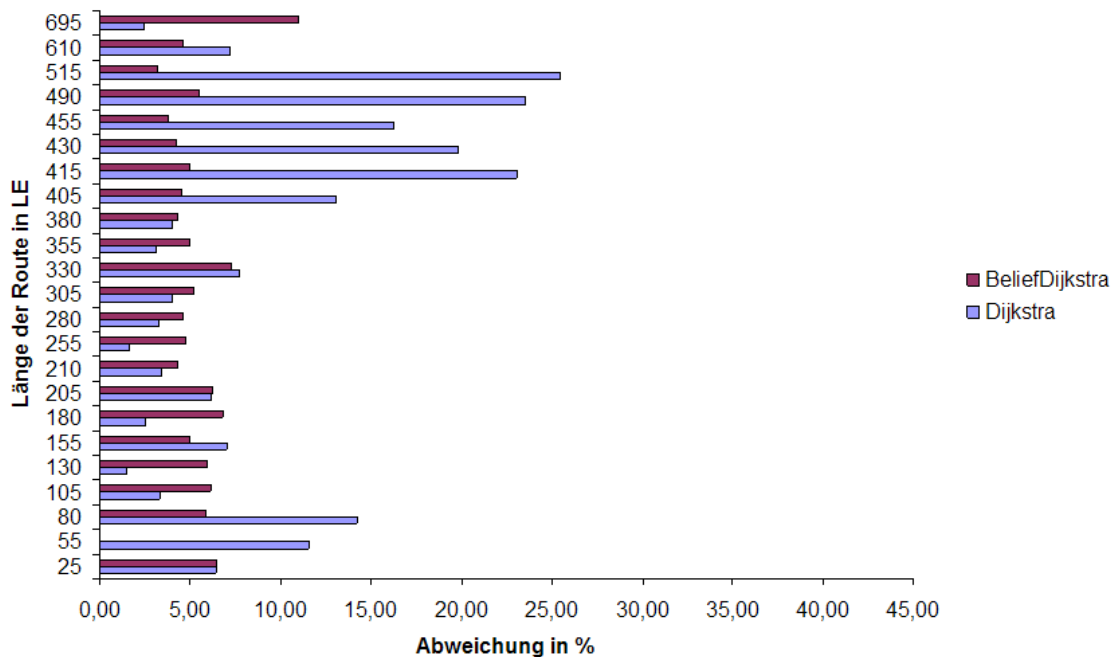


Abbildung 7-7: Ergebnisse bei einer Variation  $\mu$  von -8%

Robuster zeigt sich der BeliefDijkstra dagegen bei Abweichungen der  $\mu$ -Werte nach oben. Bei einer Erhöhung der  $\mu$ -Werte um 5% (Abbildung 7-8) ergab sich eine mittlere Abweichung der Prognosen von 7,31% gegenüber 19,04% für den Standard-Dijkstra. Bei einer Erhöhung um



15% (Abbildung 7-9) betrug die mittlere Abweichung des BeliefDijkstras 15,45%, die des Dijkstras 26,34%.

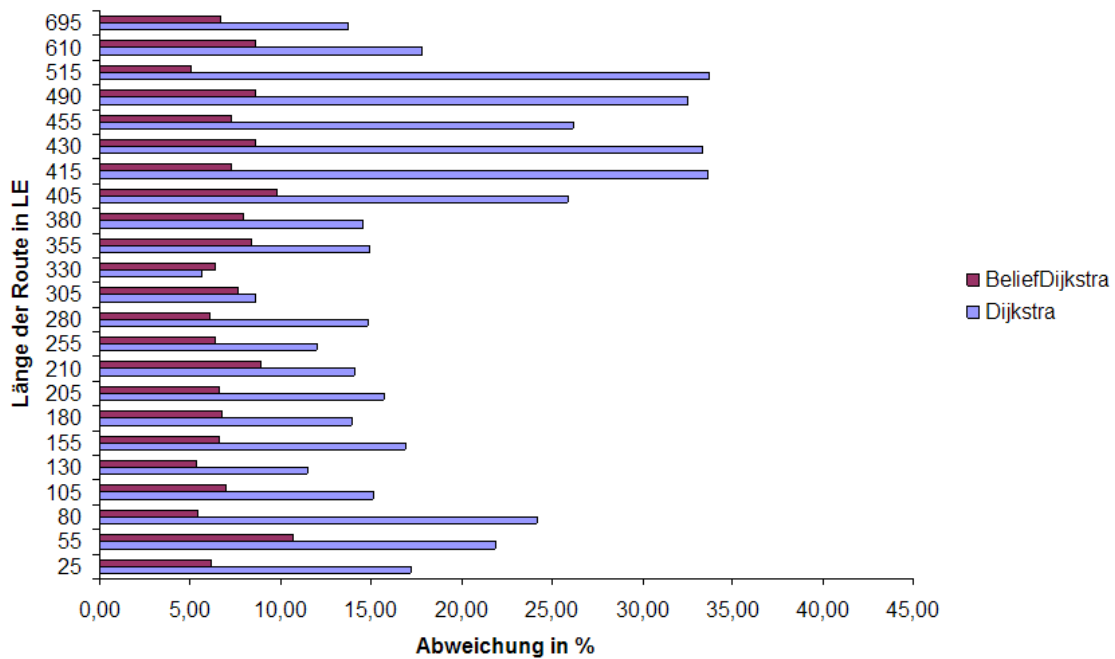


Abbildung 7-8: Ergebnisse bei einer Variation  $\mu$  von +5%

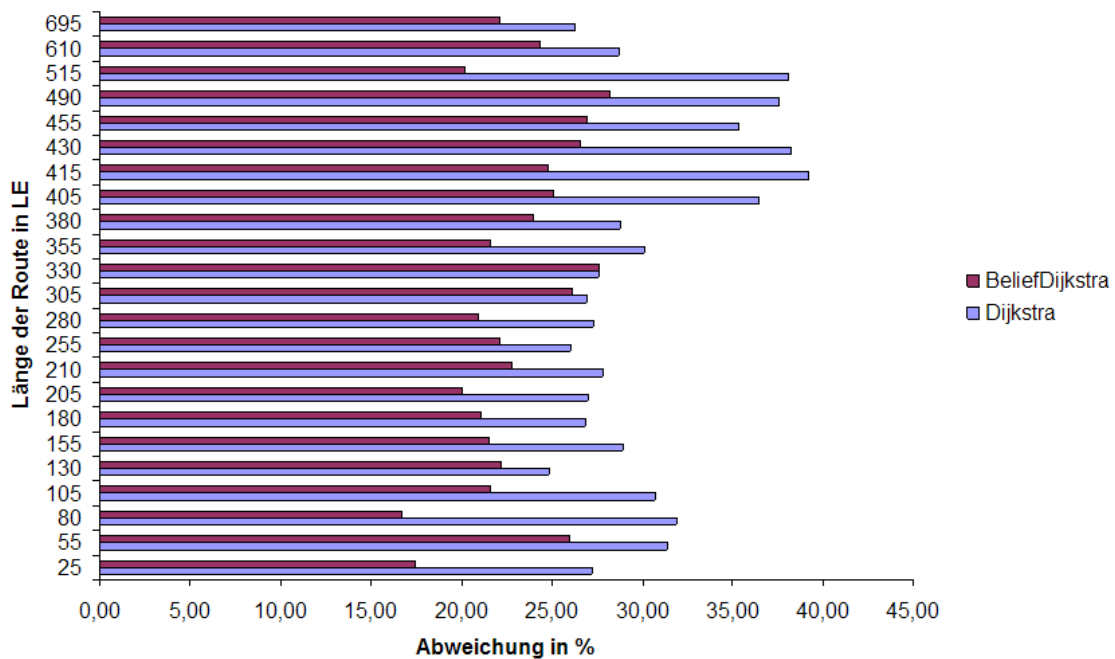


Abbildung 7-9: Ergebnisse bei einer Variation  $\mu$  von +15%

Abbildung 7-10 und Abbildung 7-11 zeigen die Auswirkungen, wenn alle Verteilungsparameter manipuliert werden. Dabei wurde in diskreten Verteilungen jeweils der unwahrscheinlichste Wert manipuliert, also seine Eintrittswahrscheinlichkeit erhöht oder weiter vermindert (wodurch auch alle anderen Werte angepasst werden müssen, um sicherzustellen, dass die Gesamtwahrscheinlichkeit eins beträgt). Es ergab sich bei einer Anpassung aller Verteilungswerte um 5% (Abbildung 7-10) für den BeliefDijkstra eine mittlere Abweichung von 15,61%, für den Dijkstra eine mittlere Abweichung von 25,05%. Bei einer Modifikation aller Verteilungswerte um 15% (Abbildung 7-11) betrug die mittlere Abweichung des BeliefDijkstras 15,45%, die des Dijkstras 26,34%.

lungparameter um -10 % (Abbildung 7-11), war für den BeliefDijkstra eine durchschnittliche Abweichung von 8,18%, für den Dijkstra eine durchschnittliche Abweichung von 18,30% zu beobachten.

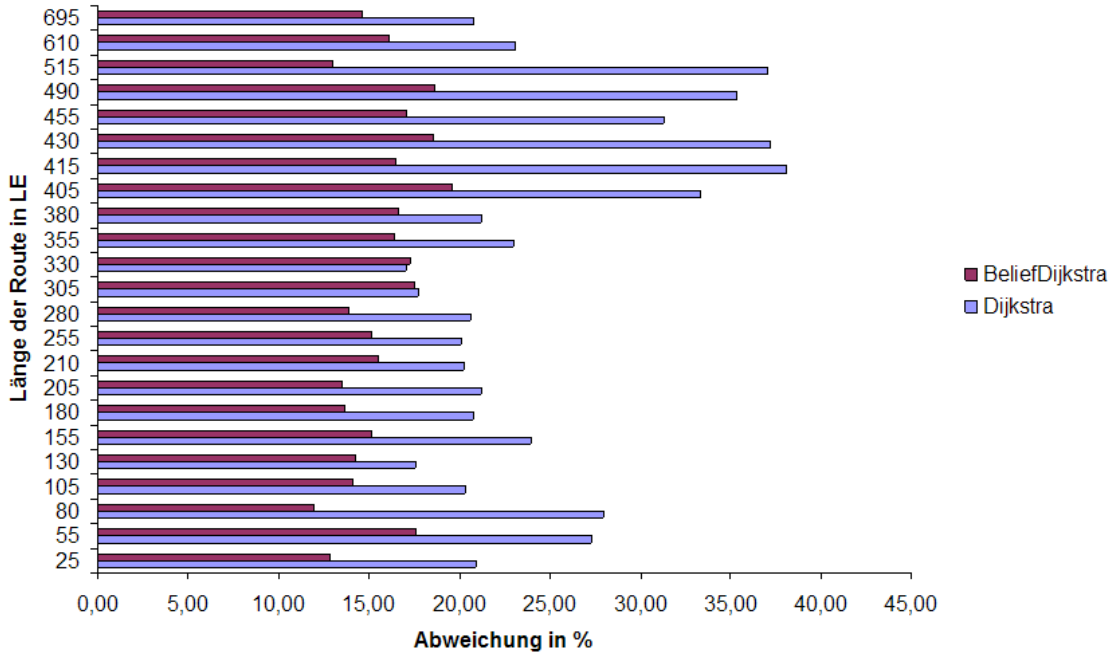


Abbildung 7-10: Ergebnisse bei einer Variation +5% aller Parameter

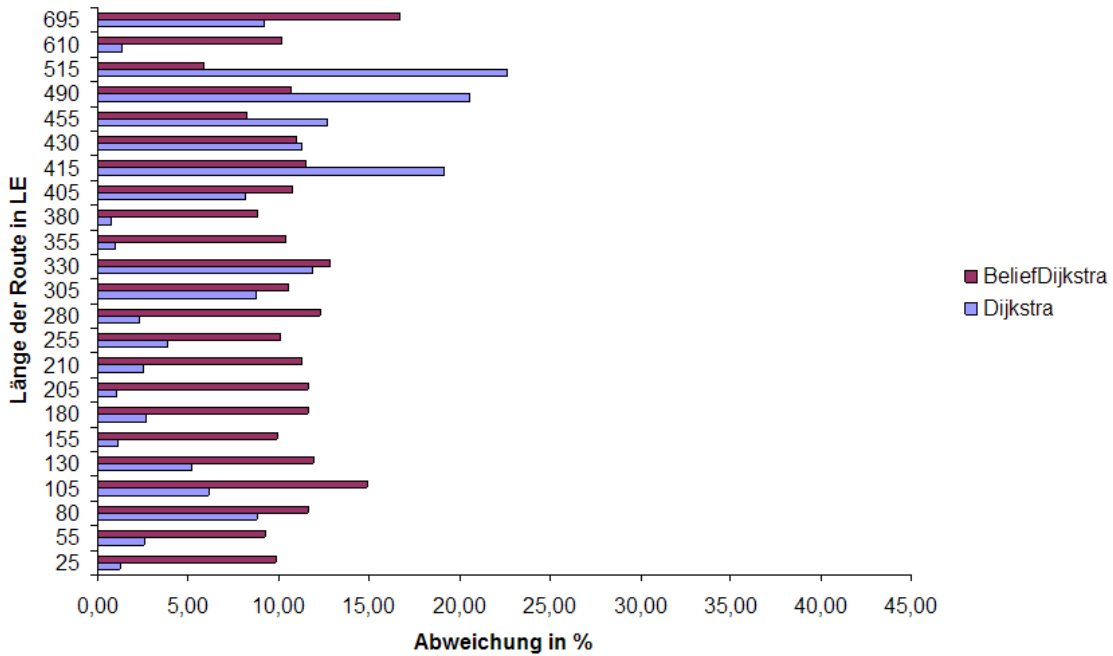


Abbildung 7-11 Ergebnisse bei einer Variation von -10% aller Parameter

Die besseren Prognosen schlugen sich auch in kürzeren Fahrzeiten nieder, vorausgesetzt die beiden Algorithmen fanden aufgrund der Streckentopologie unterschiedliche Routen. Abbildung 7-12 zeigt die Fahrzeiterparnis, die sich einstellte, als die Verteilungen der Prognose und der Simulation übereinstimmen. Die mittlere Einsparung betrug 13,5%.

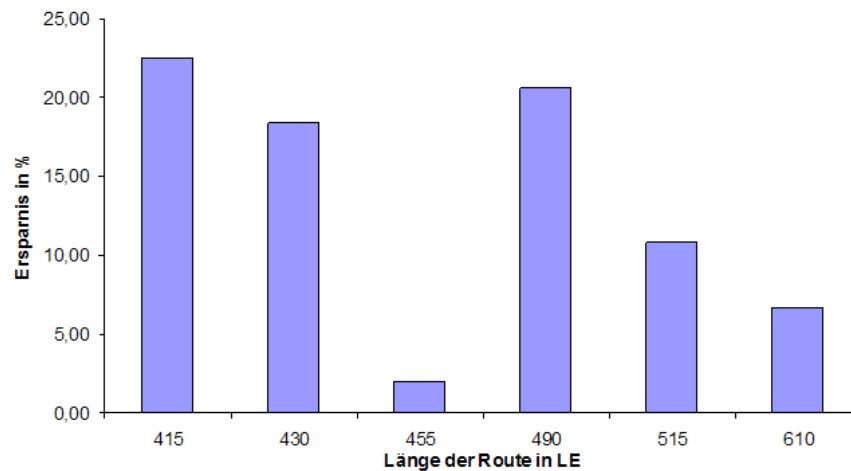


Abbildung 7-12: Fahrzeiterparnis durch den BeliefDijkstra

Auch bei abweichenden Verteilungen konnte in den meisten Fällen noch eine Ersparnis erzielt werden. Abbildung 7-13 zeigt die Fahrzeiterparnis, die sich einstellte, als alle Verteilungsparameter um 10% vermindert wurden. Die durchschnittliche Einsparung betrug in diesem Fall noch 11,87%. Bei zufällig um 15% veränderten Verteilungen konnte immerhin noch eine mittlere Einsparung von 8,03% erzielt werden (vgl. Abbildung 7-13). Auffällig ist, dass für die Testroute der Länge 455 keine Ersparnis mehr erzielt werden konnte. Gleiches konnte auch bei anderen Testszenarien mit veränderten Verteilungen beobachtet werden. Der Grund dafür muss darin gesucht werden, dass der Längenunterschied zwischen dem vom BeliefDijkstra und dem vom Dijkstra geplanten Pfad 90 LE beträgt. Dies ist die höchste Differenz unter allen Testrouten. Da der Pfad des BeliefDijkstra 90 LE länger ist, war die Ersparnis trotz der schlechten Verkehrsverhältnisse auf dem Pfad des Dijkstra mit 1,98% relativ klein. Ist das Zutrauen in die Wahrscheinlichkeitsverteilung gering, sollte also bei großen Abweichungen bzgl. der Fahrstrecke eher auf die Lösung des Standard-Algorithmus zurückgegriffen werden.

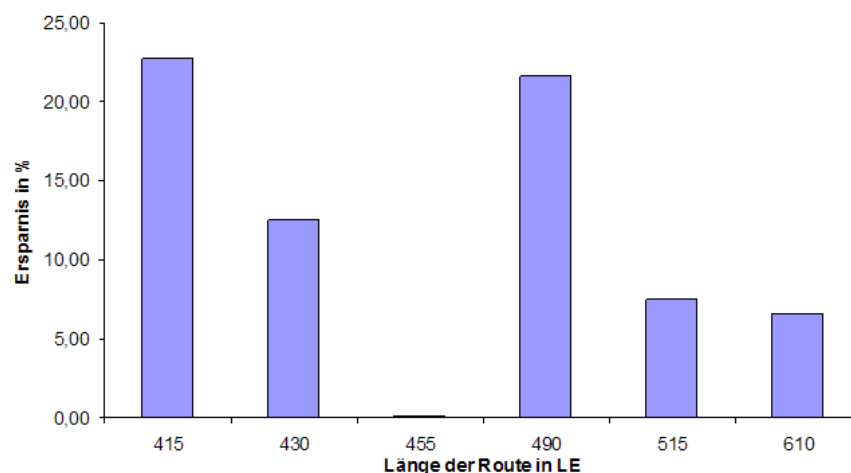


Abbildung 7-13: Fahrzeiterparnis bei Variation aller Verteilungsparameter um +10%

Um zu prüfen, ob auch die Berücksichtigung des Verspätungsrisikos in gewünschter Weise funktioniert, wurde ein ergänzender Test durchgeführt. In diesem Test wurden wiederum zwei Vergleichsplanungen durchgeführt, einmal mit einem Parameter von  $\omega=0,02$  und ein zweites

Mal mit  $\omega=2$ . Bei  $\omega=2$  wird die risikodämmende Wirkung des gewichteten Mittelwerts der Fahrzeit fast völlig ausgehebelt, diese Einstellung entspricht also einer risikoneutralen Planung. Während der Simulationen wurde für jeden geplanten Pfad auch die mittlere Verspätung berechnet. Im Transportnetzwerk mit 100 Stationen führten die Planungsläufe bei 31 Kombinationen von Start und Ziel zu unterschiedlichen Routen. Für diese 31 Fälle wurde die mittlere Verspätung verglichen. Bei allen Pfaden, die mit der Einstellung  $\omega=0,02$  geplant wurden, ergab sich eine Verspätung um 8,39%, für die mit  $\omega=2$  erstellten Pläne eine Verspätung von 9,31%. Auf den ersten Blick erscheint dies als ein schwaches Ergebnis, jedoch bedeutet es eine Reduzierung der Verspätung um fast 10%. Außerdem setzte bereits der Vergleichsalgorithmus die fortgeschrittene Prognosetechnik ein, sodass Verspätungen ohnehin schon unwahrscheinlicher sind.

## 7.3 Planung von Systeminteraktionen

Dieses Kapitel beschreibt ausgewählte Anwendungsbeispiele aus dem Gebiet der Planung von Systeminteraktionen, wie sie in Kapitel 5 beschrieben ist. Dabei teilen sich die Anwendungsbeispiele wiederum in die beiden Gebiete „Planung bei gemeinsamer Hauptaufgabe“ und „Planung bei individueller Hauptaufgabe“ auf.

### 7.3.1 Planung bei gemeinsamer Hauptaufgabe

#### 7.3.1.1 Koordination von Funktionsmodulen in einem Schienenfahrzeug

Als Anwendungsbeispiel für die Planung bei gemeinsamer Hauptaufgabe dient die Abstimmung der beiden Module Antrieb (Index P für Propulsion) und Luftspaltverstellung (AGAS, Index A) aus dem RailCab (vergleiche auch Kapitel 1). Der erste Schritt in der Beschreibung des Anwendungsbeispiels ist die Darstellung des konkreten Problems in der in Unterkapitel 5.2 beschriebenen Modellierung.

Der Antrieb betrachtet unterschiedliche Geschwindigkeitsstufen (Höchstgeschwindigkeit  $M_{ax}$  (180 km/h) und  $\frac{v_{max}}{2}$  (also 90 km/h)), zwischen denen er auf einzelnen Streckenabschnitten mit gleichmäßiger Beschleunigung wechseln kann. Nimmt man noch das Anfahren aus dem Stand hinzu, ergeben sich somit insgesamt 6 Teilfunktionslösungen. Das AGAS betrachtet im Beispiel zwei unterschiedliche Teilfunktionslösungen: die einmalige Einstellung des Luftspalts für einen Streckenabschnitt und die wiederholte Einstellung während der Überfahrt.

Im Beispiel verfolgen die beiden Systeme zwei Ziele, die sich jeweils genau einem der beiden Module zuordnen lassen: die Maximierung der durchschnittlichen Geschwindigkeit (Antrieb) und die Minimierung des Luftspalts (AGAS).

Eine Querwirkung geht im Beispiel ausschließlich vom Antrieb aus: Da der Antrieb durch hohe Geschwindigkeiten die Einstellungsmöglichkeiten des AGAS behindert, erreicht dieser das Ziel der Laufspaltminimierung bei Fahrprofilen mit hohen Geschwindigkeiten schlechter. Dies gilt insbesondere auf Strecken mit ungenauer Statorlage, da hier die größeren Höhendifferenzen zwischen den unterschiedlichen Statoren wesentlich schlechter abgefahren werden können.

#### 7.3.1.2 Modellierung des Interaktionsplanungsproblems

Die Definition des Interaktionsplanungsproblems bei gemeinsamer Hauptaufgabe beginnt mit der Angabe der Zielmenge  $\Omega$ . Im Beispiel umfasst  $\Omega$  zwei Elemente:

- $\Omega = \{\emptyset \text{Geschwindigkeit (G)}, \emptyset \text{Luftspalt (L)}\}$

Als Umweltbedingungen wird lediglich die Verlegungsgenauigkeit der Statoren betrachtet:

- $U = \{\text{Exakt (E)}, \text{Ungenau (U)}\}$

Im nächsten Schritt gilt es, die Anwendungshäufigkeiten der Teilfunktionslösungen zu bestimmen. Da im Beispiel eine Querwirkung vom Antrieb ausgeht, muss nur dieser die Anwendungshäufigkeit der Teilfunktionslösungen des AGAS berücksichtigen. Da die Teilfunktionslösung *einmalige Verstellung (abgekürzt eV)* die Ziele des AGAS in allen Situationen schlechter erreicht als *kontinuierliche Verstellung (abgekürzt kV)*, versucht das AGAS unab-

hängig vom Zielsystem immer die *kontinuierliche Verstellung* auszuführen. Lediglich wenn Umwelteinflüsse wie Witterung (z. B. veränderte Statorlage durch Eis auf den Statoren) die sichere Anwendung der *kontinuierlichen Verstellung* verhindern, wird die *einmalige Verstellung* verwendet. Dabei fallen die Umwelteinflüsse bei ungenauer Statorlage etwas stärker ins Gewicht, da die Differenzen zwischen den Positionen der einzelnen Statoren größer sind. Somit ergibt sich z. B. eine Häufigkeitsverteilung von:

- $\theta(kV|E) = 0,9$  und  $\theta(eV|E) = 0,1$
- $\theta(kV|U) = 0,8$  und  $\theta(eV|U) = 0,2$

Die Evaluationsfunktionen für die Ziele *durchschnittliche Geschwindigkeit* und *durchschnittlicher Luftspalt* werden genau wie im Beispiel aus Abschnitt 5.2.2 definiert:

$$\bullet \quad E(G) = \frac{v_{\emptyset}}{v_{max}}, \quad E(L) = \frac{\Delta l s_{\emptyset}}{l s_{max}}$$

Der nächste zu bestimmende Modellbestandteil ist der durchschnittliche Zielereichungsgrad (E), den ein bestimmter Modus unter gegebenen Umweltbedingungen erreicht:  $X_u^{\omega}(tfl)$ . Tabelle 7-1 zeigt die entsprechenden Werte für den Antrieb und sein ausschließliches Ziel durchschnittliche Geschwindigkeit t (G). Da die Statorlage keine Auswirkungen auf die Durchschnittsgeschwindigkeit besitzt, die ein bestimmtes Fahrprofil erreicht, sind die Werte für beide Umweltbedingungen identisch. Die Werte errechnen sich direkt aus der sich aus einem Fahrprofil ergebenden Durchschnittsgeschwindigkeit und der maximalen Geschwindigkeit von 180 km/h.

Teilfunktionslösung	$X_E^G(tfl)$ und $X_U^G(tfl)$
A	0,25
B	0,5
C	0,5
D	0,75
E	0,75
F	1

Tabelle 7-1: Funktionswerte von  $X_E^G(tfl)$  und  $X_U^G(tfl)$

Anders liegt der Fall natürlich beim AGAS. Hier haben die Umwelteinflüsse Auswirkungen auf den einstellbaren Luftspalt.

Teilfunktionslösung	$X_E^L(tfl)$	$X_U^L(tfl)$
kV	0,9375	0,4875
eV	0,7	0,2

Tabelle 7-2: Funktionswerte von  $X_E^L(tfl)$  und  $X_U^L(tfl)$

Um die Wechselwirkungen darstellen zu können, ist es schließlich erforderlich,  $H_u^{\omega}(tfl|tfl')$  (Tabelle 7-3) zu bestimmen. Wie zu erkennen ist, führen hohe Durchschnittsgeschwindigkeiten tendenziell zu einem schlechteren Luftspalt. Dieser Effekt ist bei ungenau verlegten Streckenabschnitten stärker als bei exakt verlegten Streckenabschnitten.

Damit sind fast alle notwendigen Variablen definiert, damit das Antriebsmodul im Rahmen seiner Planung bei Vorliegen konkreter Zielgewichtungen die Querwirkungen auf das AGAS

berücksichtigen kann. Lediglich die Rangfunktion  $P$  fehlt. Da die beiden Ziele aber jeweils exklusiv durch ein Modul verfolgt werden, ergibt sich diese einfach durch:

- $P^G(P) = 1$  und  $P^G(A) = 0$
- $P^L(P) = 0$  und  $P^L(A) = 1$

Tabelle 7-3: Funktionswerte von  $H_E^{\omega}(tfl|tfl')$

Teilfunktionslösung	Bedingende Teilfunktionslösung	$H_E^L(tfl tfl')$	$H_E^L(tfl tfl')$
<b>eV</b>	A	0,3125	0,7
	B	0,0625	0,7
	C	0,0625	0,7
	D	-0,1875	0,7
	E	-0,1875	0,7
	F	-0,4375	0,7
<b>kV</b>	A	0,0625	0,2
	B	0,0625	0,2
	C	0,0625	0,2
	D	-0,0625	0,2
	E	-0,0625	0,2
	F	-0,4375	0,2

### 7.3.1.3 Ein Algorithmus für die initiale Planung des Antriebs

Aufgabe der Planung für den Antrieb ist nun, eine Sequenz von Fahrprofilen für die unterschiedlichen Streckenabschnitte auszuwählen, sodass die gewichtete Summe aus durchschnittlichem Luftspalt und Geschwindigkeit möglichst optimiert wird.

Ein sehr einfacher Ansatz wäre die Verwendung einer Breitensuche, die den Planbaum Ebene für Ebene nach der besten Abfolge durchsucht. Die Breitensuche durchläuft dabei alle Knoten und Kanten im Baum, hat also eine Laufzeit von  $O(|V|+|E|)$  (mit  $V$  Menge der Knoten und  $E$  Menge der Kanten). Im Beispiel handelt es sich beim Planbaum um einen Ternär-Baum<sup>58</sup>. Ein Ternär-Baum der Höhe  $h$  enthält  $3^h - 1$  Knoten. Da sich die Höhe des Planbaums im Beispiel aber direkt aus der Anzahl der abzufahrenden Streckenabschnitte ergibt, ergäbe sich für die Breitensuche als Planungsalgorithmus eine Komplexität von  $(2^{\text{Streckenabschnitt}})$ , also eine exponentielle Laufzeit. Benötigt die Untersuchung eines Knoten 1 Millisekunde, ergibt sich bei einer Problemgröße von 10 Streckenabschnitten eine Laufzeit von etwa 1 Sekunde und bei 50 Streckenabschnitten bereits eine Laufzeit von etwa 300 Stunden. Offensichtlich ist dieses Vorgehen daher nicht sinnvoll.

Durch die Aufteilung des Planungsproblems auf einzelne Module besteht aber die Möglichkeit, sich die jeweilige Problemstruktur zunutze zu machen, um zu einem schnelleren Lösungsverfahren zu kommen. Im beschriebenen Planungsproblem hängt die Ausführbarkeit eines Modus immer nur von seinem direkten Vorgänger ab. Ein Teilbaum in Ebene  $n$  mit der Wurzel  $tfl$  enthält nachfolgend genau die gleichen Teilbäume wie alle anderen Teilbäume, die ebenfalls die Wurzel  $tfl$  besitzen.

<sup>58</sup> Jeder Modus des Antriebs und damit jeder Knoten im Planbaum besitzt maximal drei gültige Nachfolger: die nächste höhere und tiefere Geschwindigkeitsstufe und die eigene Geschwindigkeitsstufe.

**Planer**

```

Knoten start //der Startknoten des Planungsproblems
Liste aktuelle_Ebene //Knotenliste der aktuell untersuchten Ebene
Liste nächste_Ebene //Knotenliste der nächsten Ebene

aktuelleEbene.fügeHinzu(start)//die erste Ebene enthält nur den Start

Solange die aktuelle_Ebene noch Knoten enthält
    Knoten k := erster Knoten aus der aktuellen Ebene
    entferne den ersten Knoten aus der aktuellen Ebene

    Wenn die Zielebene noch nicht erreicht ist
        //expandiere die Nachfolger von k
        Liste Nachfolger := mögliche Nachfolger von k

        Über alle Knoten n in Nachfolger
            //der Zielwert entspricht den Summierten
            //globalen Nutzen über
            //alle Knoten in einem Pfad
            Zielwert von n := Zielwert von k +  $H_u^o(tfl) + X_u^G(tfl)$ 

        //Suche Element e aus nächste_Ebene mit der gleichen TFL wie n

        Wenn es ein solches Element nicht gibt
            Dann Füge den Knoten zu nächste_Ebene hinzu
            Sonst Wenn der Zielwert von n > Zielwert von e
                Dann entferne e aus nächste_Ebene und
                    füge n ein
            Sonst -

        Sonst
            tue

        Wenn aktuelle Ebene keinen Knoten mehr enthält
            Dann Aktuelle_Ebene := nächste_Ebene
                Nächste_Ebene leeren
            Sonst -

    Tue

//Am Ende der Schleife enthält aktuelle_Ebene mehrere Knoten aus der
//letzten Ebene des Planbaums, wenn es eine gültige Lösung gibt

besterKnoten = erstes Element aus aktuelle_Ebene
//Diese Schleife ermittelt den Endknoten der besten Lösung
Wenn aktuelle_Ebene nicht leer ist
    Dann
        Über alle Elemente k aus der aktuellen Ebene
            Wenn die Zielfunktion k > Zielfunktion besterKnoten
                Dann besterKnoten := k
            Sonst -

    Tue
Sonst -

return besterKnoten
Ende

```

**Listing 7-3: Planungsalgorithmus für den Antrieb in Pseudo-Code**

Da nun von jedem Knoten der Ebene mit der Teilfunktionslösung *tfl* die gleichen Pfade zu erreichen sind, muss nur der jeweils beste Knoten  $k_{\text{best}}$  für jede Teilfunktionslösung weiter untersucht werden. Die Wege über alle anderen entsprechenden Knoten können im weiteren Verlauf nicht mehr zur Lösung beitragen als die Wege über  $k_{\text{best}}$ . Dadurch wird auf jeder Ebene nur noch maximal ein Knoten für jede Teilfunktionslösung weiter untersucht. Daher bleibt die Anzahl der Knoten in jeder Ebene konstant und der Plan kann in linearer Zeit  $O(|\text{Teilfunktionslösungen}| \cdot |\text{Streckenabschnitte}|)$  gefunden werden. Damit ergebe sich im konkreten Beispiel bei 50 Streckenabschnitten eine Lösungszeit von 200 Millisekunden.



Listing 7-3 zeigt einen entsprechenden Algorithmus in Pseudo-Code. Er durchsucht den Planungsbaum Ebene für Ebene und behält jeweils nur die aktuell beste Lösung zu jeder Teilfunktionslösung in der betrachteten Ebene. Dabei ergibt sich die Lösungsqualität eines Knoten durch die kumulierten  $H_u^\omega(\text{tfl}|\text{tfl}')$  Werte aller seiner Vorgänger. Dieser Wert wird den Knoten jeweils im Schritt der Expansion von ihrem Elternknoten mitgegeben. Am Ende des Pseudo-Codes ist der letzte Knoten der besten Lösung bestimmt. Indem jeder Knoten einen Verweis auf seinen Elternknoten beinhaltet, kann die Lösung des Planungsproblems nun problemlos rückwärts aufgebaut werden.

### 7.3.1.4 Koordination nach der Planung

Aufgrund von zusätzlichen Umwelteinflüssen, die dem Antrieb während der initialen Planung nicht bekannt sind, wählt das AGAS gelegentlich nicht die vermutete Teilfunktionslösung. Wechselt das AGAS von der *kontinuierlichen Verstellung* zur *einmaligen Verstellung*, kann der Antrieb seinen Plan anpassen, um die aktuellen Ziele optimal zu erreichen.

#### Plananpassung

```

Knoten start := Ende der eingefrorenen Zone
Knoten ziel := Ende des Planungshorizonts
Tabelle aktuelleKostenWerte //diese Tabelle enthält zu jedem Modus
                                // die aktuellen Zielwerte auf jedem
                                // Strecken abschnitt abhängig vom Plan
                                // des AGAS
Liste aktuelle_Ebene //Knotenliste der aktuell untersuchten Ebene
Liste nächste_Ebene //Knotenliste der nächsten Ebene

aktuelleEbene.fügeHinzu(start)//die erste Ebene enthält nur den Start

solange die aktuelle_Ebene noch Knoten enthält
    ... // Entspricht Listing 7.2

        //bei der Anpassung wird die Wechselwirkungen auf den
        //tatsächlichen Modus des AGAS zur Berechnung des Zielwertes
        //verwendet
        Zielwert von n := Zielwert von k +  $H_u^\omega(\text{tfl}|\text{tfl}_{\text{agas}}^n) + X_u^G(\text{tfl})$ 
    ... // Entspricht Listing 18
Tue

//Am Ende der Schleife enthält aktuelle_Ebene mehrere Knoten aus der
//letzten Ebene des Planbaums, wenn es eine gültige Lösung gibt
//Aus diesem muss nur derjenige Knoten ausgewählt werden, der den
//gleiche Teilfunktionslösung wie der Knoten ziel besitzt
//Dadurch ist die Konsistenz zum restlichen Plan gewährleistet

aktuellerKnoten := erstes Element aus aktuelle_Ebene
//Diese Schleife ermittelt den Endknoten der besten Lösung
Solange aktuelle_Ebene nicht leer ist
    Wenn Modus aktuellerKnoten := Modus ziel
        return aktueller Knoten
Tue
Ende

```

Listing 7-4: Anpassungsalgorithmus in Pseudo-Code

Wie in Abschnitt 5.2.4 kann die Planung mit einem rollierenden Planungshorizont und einer eingefrorenen Zone durchgeführt werden, um die Berechnungszeit zu beschränken. Obwohl die Lösungszeiten des im vorigen Abschnitt beschriebenen Algorithmus sehr gut sind, wurde eine rollierende Planung implementiert, um die Auswirkung der eingefrorenen Zone analysieren zu können.

Listing 7-4 zeigt den Algorithmus für die Plananpassung in Pseudo-Code. Ihm werden das Ende der eingefrorenen Zone und das Ende des Planungshorizonts als Nachfolger übergeben.

Zwischen diesen beiden Punkten konstruiert der Algorithmus auf ähnliche Weise wie die initiale Planung einen neuen Teilplan. Dabei verwendet die Plananpassung im Unterschied zur initialen Planung zur Berechnung des Zielwerts die vom AGAS im jeweiligen Planungsschritt gewählte Aktivität. Dadurch operiert die Anpassungsplanung auf genaueren Informationen. Da sowohl der Startknoten als auch der Zielknoten nicht verändert werden (zumindest hinsichtlich der gewählten Teilfunktionslösung), kann der restliche Plan ohne Beeinflussung durch die Plananpassung weiter ausgeführt werden.

### 7.3.1.5 Experimentelle Ergebnisse für die Planung des Antriebs

Zur Evaluierung der oben beschriebenen Planungsmechanismen wurde eine Variante des Planungsalgorithmus implementiert, die eine optimale Abstimmung mit dem AGAS für eine maximale Gesamtfahrzeit liefert.

Im Rahmen der im Folgenden beschriebenen Experimente wurde eine Gesamtstreckenlänge von 300 km, unterteilt in Streckenabschnitte von 1 km, verwendet. Dabei wurden unterschiedliche Testszenarien hinsichtlich der Häufigkeit von Streckenabschnitten mit ungenau verlegten Statoren betrachtet. Es wurden Szenarien mit einem Anteil von 30% bis 70% (in Schritten von 10%-Punkten) ungenauer Streckenabschnitte erzeugt. Für jedes Szenario wurden 100 Gesamtstrecken mittels Zufallsgenerator erzeugt. Für die 300 km ergibt sich eine Mindestfahrzeit von etwa 6000 Sekunden (100 Minuten). Die Planung wurde jeweils für maximale Fahrzeiten zwischen 6500 und 9000 Sekunden durchgeführt. Dabei war zu erwarten, dass das Planungsverfahren den jeweils größeren Spielraum ausnutzt und eine bessere Abstimmung mit dem AGAS erreicht, wodurch sich ein kleinerer durchschnittlicher Luftspalt ergibt.

Da kein Simulationsmodell zur Bestimmung des einstellbaren Luftspalts zur Verfügung stand, wurden in der Evaluation die in Tabelle 7-4 dargestellten Schätzungen verwendet.

Modus AGAS	Streckentyp	Modus Antrieb	Luftspalt
Kontinuierlich	Ungenau	A	10.0 mm
		B	12.5 mm
		C	12.5 mm
		D	15.0 mm
		E	15.0 mm
		F	17.1 mm
	Exakt	A	8.0 mm
		B	8.0 mm
		C	8.0 mm
		D	9.25 mm
		E	9.25 mm
		F	10.5 mm
Einmalig	Ungenau	Alle	17.1 mm
	Exakt	Alle	11.0 mm

Tabelle 7-4: Situationsabhängiger mittlerer Luftspalt auf einem Streckenabschnitt

Abbildung 7-14 zeigt die Ergebnisse für die Versuchseinstellung mit 70% ungenauen Streckenabschnitten. Es ist klar erkennbar, dass das Planungsverfahren die Informationen über die Wechselwirkungen mit dem AGAS bei den längeren Fahrzeiten ausnutzte. Bei der längsten zur Verfügung stehenden Fahrzeit wurde der durchschnittliche Luftspalt über die 100 Testläufe um 13,36% reduziert. Die durchschnittliche Planungsdauer lag bei ca. 2,6 Sekunden.<sup>59</sup>

<sup>59</sup> Gemessen auf einem AMD Athlon 64 Prozessor 3000+ mit 1,8 GHz und 1,00 GB RAM mit Windows XP SP2.

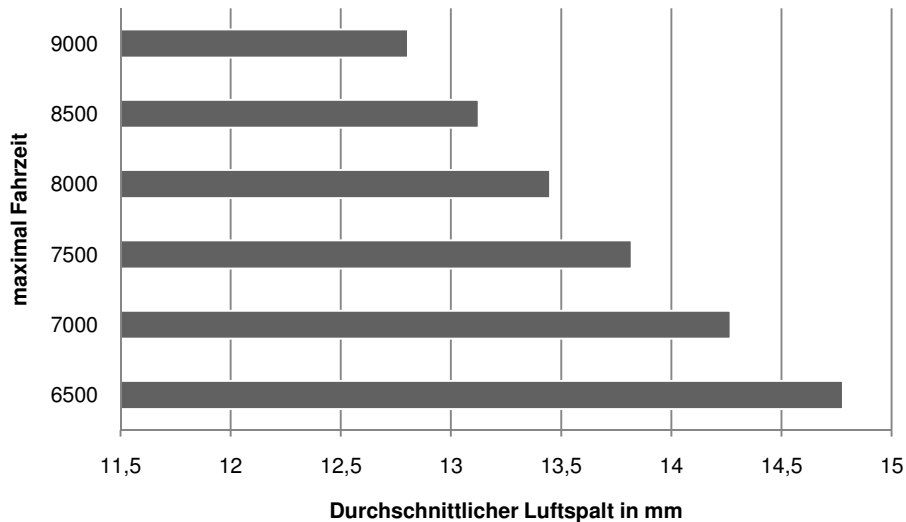


Abbildung 7-14: Ergebnisse bei 70% ungenauen Streckenabschnitte

Die Tendenz ist in allen Testszenarien die gleiche. In jedem Fall wird der Spielraum der längeren Fahrzeiten ausgenutzt, um die Koordination mit dem AGAS zu verbessern. Abbildung 7-15 zeigt die Ergebnisse in den einzelnen Szenarien. Es ist deutlich erkennbar, dass der durchschnittliche Luftspalt in jedem Fall mit steigender maximaler Fahrzeit reduziert wurde. Die durchschnittliche Reduzierung des Luftspalts bei einer Fahrzeit von 9000 Sekunden gegenüber der Fahrzeit von 6500 Sekunden liegt in jedem Szenario zwischen 13% und 14%.

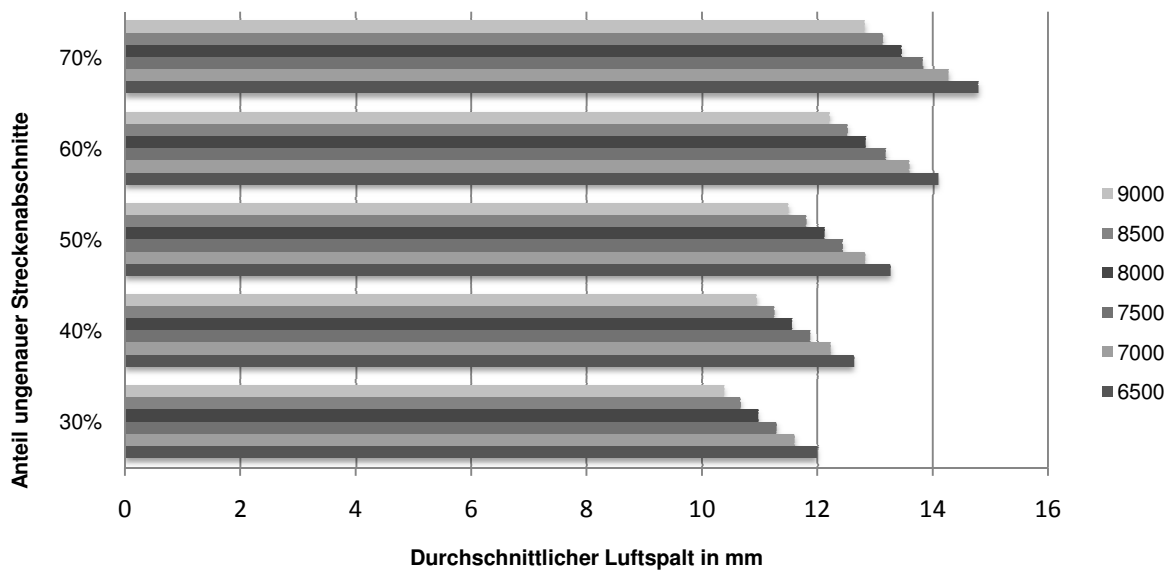


Abbildung 7-15: Ergebnisse in allen Testszenarien

Auch die rollierende Planung wurde untersucht. Hier wurde angenommen, dass das AGAS aufgrund aktueller Informationen seine Entscheidung ändert und auf einzelnen Streckenabschnitten von der Einstellung *kontinuierlich* auf *einmalig* wechselt. Da bei einmaliger Verstellung das Fahrprofil des Antriebs keinen Einfluss auf die Größe des einstellbaren Luftspalts besitzt, bieten sich solche Streckenabschnitte an, um schnell befahren zu werden. Im Ergebnis kann der Antrieb daher seinen Plan anpassen und auf den Abschnitten langsam fahren, die eine Verbesserung der Abstimmung mit dem AGAS bewirken und dennoch die maximale Fahrzeit einhalten. Dabei wurde die rollierende Planungsvariante mit unterschiedlichen Pla-

nungshorizonten durchgeführt. Abbildung 7-16 zeigt die Ergebnisse der Versuche<sup>60</sup>. Ein Zusammenhang zwischen der Länge des Planungshorizonts und der Planqualität ist klar erkennbar. Das liegt vor allem an dem größeren Spielraum, der bei einem langen Planungshorizont vorlag. Auffällig ist, dass selbst bei einem langen Planungshorizont von 90 Planungsschritten noch eine erhebliche Lücke zur optimalen Lösung (ganzer Plan) des Umplanungsproblems bestand.

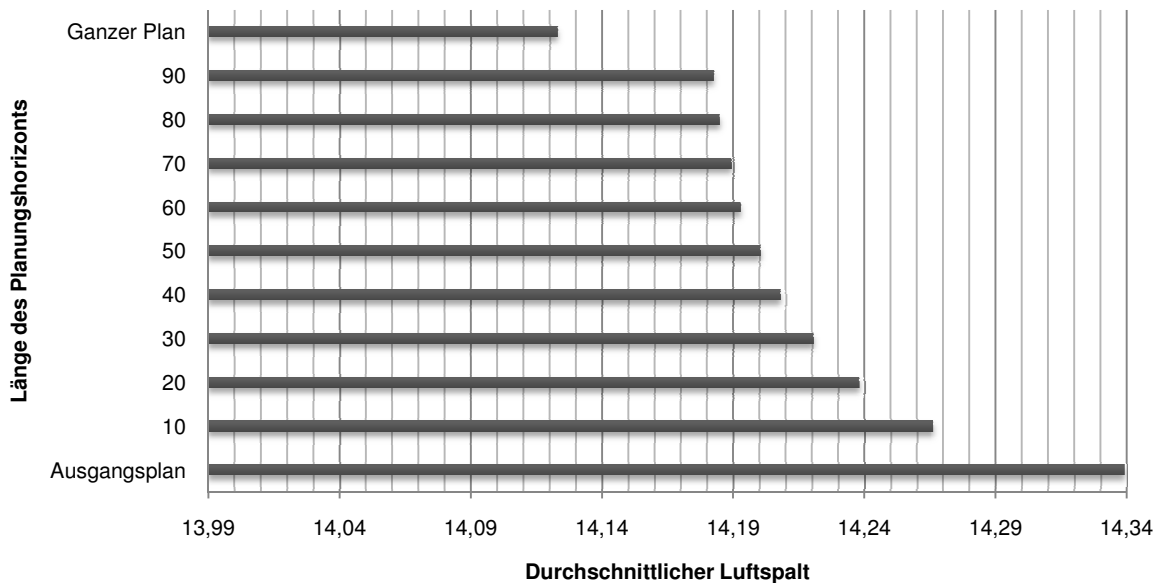


Abbildung 7-16: Ergebnisse der Umplanung

Es bestand aber ein ebenso deutlicher Zusammenhang zwischen der Dauer einer Iteration und der Länge des verwendeten Planungshorizonts. Die Dauer einer Iteration (also die Umplanung auf dem aktuellen Planungshorizont) ist aber eine wesentliche Kennzahl: Es ist die Zeit, die vergeht, bis das System den nächsten ausführenden Schritt erhält. Abbildung 7-17 zeigt die durchschnittliche Dauer einer Iteration in Abhängigkeit zur Länge des Planungshorizonts. Es lässt sich unmittelbar ein quadratischer Zusammenhang feststellen. Auch wenn dieser Zusammenhang für das betrachtete Anwendungsszenario unbedeutend ist (die Überfahrt eines Streckenabschnitts dauert mindestens 20 Sekunden), kann er für andere Anwendungen, in denen weniger Zeit zwischen dem Beginn und Ende der Ausführung eines Planungsschritts vergeht oder die Planungszeiten insgesamt höher sind, von großer Bedeutung sein. Es besteht also ein Trade-Off zwischen der Dauer einer Iteration der Umplanung und der erreichbaren Planqualität, die für jede Anwendung abgewogen werden muss.

<sup>60</sup> Die Ergebnisse basieren auf einem Szenario mit 70% Anteil an ungenauen Streckenabschnitten, einer maximalen Fahrzeit von 8000 Sekunden und einer Wechselwahrscheinlichkeit für die Modi des AGAS von 90%.

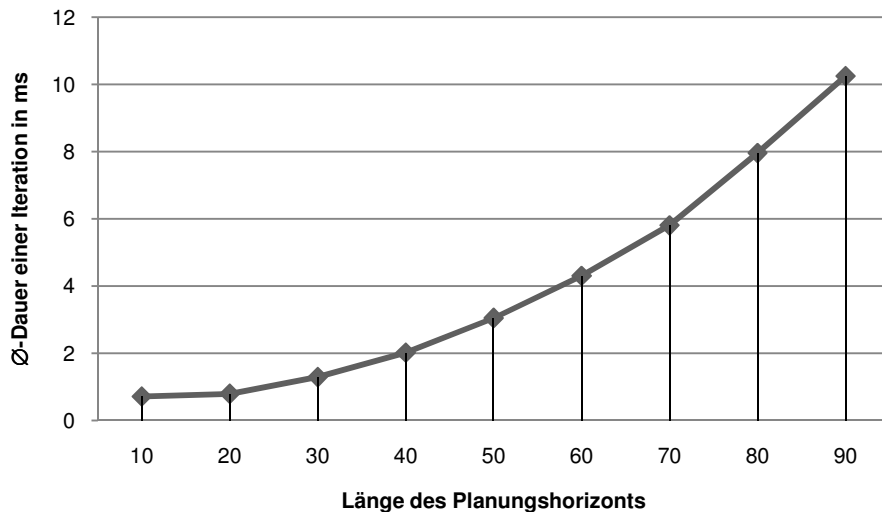


Abbildung 7-17: Dauer der Iterationen der Umplanung

### 7.3.2 Planung bei individueller Hauptaufgabe

Wie bereits in Abschnitt 5 erläutert, stellen kooperierende RailCabs ein gutes Anwendungsbeispiel für die kooperative Planung zwischen mechatronischen Systemen mit individueller Hauptaufgabe dar.

Das logistische Konzept des RailCab-Systems sieht den Verzicht auf starre Fahrpläne vor und propagiert einen bedarfsorientierten Schienenverkehr. Dabei ist vorstellbar, dass innerhalb eines Schienennetzes mehrere konkurrierende Unternehmen ihre Transportdienstleistungen anbieten. Damit ist es nicht mehr möglich, über alle Fahrzeuge hinweg ein eindeutiges Zielsystem zu definieren (Maximierung des Gewinns über alle Fahrzeuge, optimale Versorgung der Kunden, etc.), sondern die Fahrzeuge verfolgen jeweils die individuellen Ziele ihres Betreibers. Weiterhin besitzen zwar alle Fahrzeuge eine identische Aufgabe (Transport von Gütern und/oder Menschen), aber die konkreten Ausprägungen der Aufgaben unterscheiden sich im Zeitverlauf von Fahrzeug zu Fahrzeug.

Dennoch sind grundsätzlich Aktionen vorstellbar, in denen zwei oder mehr Fahrzeuge kooperieren, um ihre jeweiligen Pläne zu verbessern. Dazu gehören, entsprechend der Kategorisierung aus Abschnitt 5.3.1, die Abgabe von (Transport-)Aufträgen und die Abstimmung von Aktionen durch die Bildung von Konvois.

In Unterkapitel 5.3 wurde die Planung der Interaktion bei individueller Hauptaufgabe in vier Schritte unterteilt: Erkennen von Kooperationspotenzial, Identifikation von Kooperationspartnern, Bestimmung gemeinsamer Planungsziele und gemeinsame Planfindung. Die beiden nun folgenden Beispielprozesse einer Interaktionsplanung orientieren sich an dieser Unterteilung. In den folgenden Absätzen ist eine dezentrale Umsetzung der Kooperationen beschrieben. Um zu aussagekräftigen und vergleichbaren Ergebnissen zu gelangen, wurde auf die Umsetzung eines egoistischen Verhaltens der Fahrzeuge verzichtet. Daher kommt eine Kooperation in Form einer Auftragsvergabe oder eines Konvois immer zustande, wenn über alle beteiligten Fahrzeuge eine Verbesserung erreicht wird, auch Fahrzeuge, die sich durch eine Kooperation verschlechtern, nehmen ohne Kompensation teil.

#### 7.3.2.1 Austausch von Transportaufträgen zwischen RailCabs

Bei Onlineproblemen wie der Transportdisposition im RailCab-System besteht die Möglichkeit, dass Entscheidungen getroffen werden, die bei Vorliegen neuer Informationen als ungünstig erkannt werden. Ein Beispiel für eine solche Entscheidung ist die Übernahme eines

Transportauftrags, der im Anschluss zu einer langen Leerfahrt führt. Um diese Art von ungünstigen Entscheidungen rückgängig machen zu können, wurde das RailCab-MAS um die Möglichkeit des Austauschs von Transportaufträgen erweitert. Die Idee dabei ist, den ungünstigen Transportauftrag an ein RailCab abzugeben, in dessen aktuelle Route er sich besser einfügt (sodass z. B. eine kürzere Leerfahrt resultiert). In den folgenden Absätzen wird erläutert, wie dieser Kooperationsmechanismus realisiert wurde. Wie bereits in 5.3.2.1 beschrieben, ist eine gesonderte Modellierung der Abgabe von Transportaufträgen nicht notwendig. Lediglich eine geeignete Beschreibung der Aufträge, anhand der die Ähnlichkeit von Aufträgen beurteilt werden kann, ist erforderlich. Die hierfür notwendigen Merkmale sind die Koordinaten der Stationen eines Transportauftrags sowie die Zeitfenster zum Abholen und Absetzen eines Fahrgastes.

### 7.3.2.1.1 Erkennen des Kooperationspotenzials – Auswahl von abzugebenden Aufträgen

Um Aufträge zu identifizieren, die zur Abgabe geeignet sind, durchsuchen die Fahrzeugagenten ihre aktuellen Routen. Dieser Prozess wird immer durchgeführt, wenn eine Veränderung an der Route vorgenommen wird, also ein neuer Auftrag eingefügt oder ein alter abgegeben wird. Abbildung 7-18 zeigt, wie die Fahrzeugroute in einem solchen Fall nach einem abzugebenden Transportauftrag durchsucht wird. Ausgehend von einem neu eingefügten Transportauftrag oder dem Vorgänger eines abgegebenen Transportauftrags  $T$  wird die Strecke untersucht. Zunächst wird überprüft, mit welchen anderen Transportaufträgen sich  $T$  überlappt. Von einer Überlappung wird immer dann gesprochen, wenn ein Passagier auf dem Weg von der Startstation von  $T$  zur Endstation von  $T$  zu- oder aussteigt. Für den Fall, dass genau ein Transportauftrag  $T^*$  überlappt, wird geprüft, ob die Abgabe dieses Auftrags eine Verbesserung bringen kann. Dies ist insbesondere der Fall, wenn für  $T^*$  ein *Umweg* gefahren werden muss. Wenn der Auftrag  $T$  sich mit keinem anderen Auftrag überschneidet, wird der Auftrag  $T^*$  betrachtet, der als letzter vor  $T$  endet.

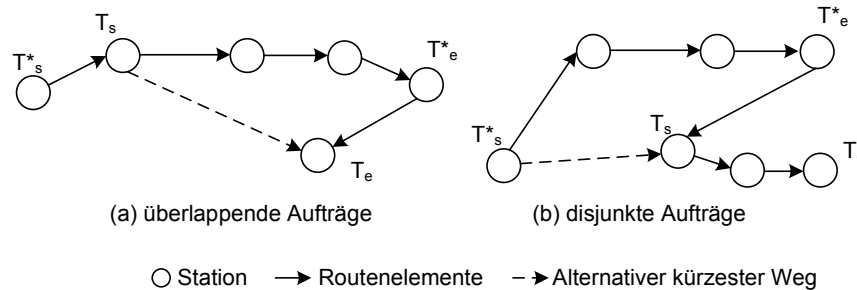


Abbildung 7-18: Beispiele für Wegeinsparungen durch die Abgabe von Aufträgen

In jedem Fall wird anhand des kürzesten Weges zwischen den Startpunkten des ausgewählten Transportauftrages geprüft, ob eine Abgabe des Auftrags eine Einsparung verspricht. Dabei ist zu beachten, dass diese Wegeinsparung nicht in jedem Fall realisierbar ist, da der Reservierungsprozess entlang der Stationen und Streckenabschnitte dazu führen kann, dass der kürzeste Fahrweg nicht verfügbar ist.

**Auftragsauswahl**

```

Route R
// Der neu eingefügte Auftrag oder der direkte Nachfolger eines abge-
//gebenen Auftrags
Auftrag T
// Ermitteln des Teilstücks aus R zwischen Start und Ende von T
Route r* = Teilroute(R, T)
// Ermitteln der Transportaufträge, die in diesem Teilstück starten
// oder enden
Liste überlappendeAufträge = lieferÜberlappendeAufträge(R, teil)

Wenn überlappendeAufträge genau einen Aufträge enthält
  Dann
    Auftrag T*
    //Kosten ermitteln für die Fahrt von der Startstation von
    //T* zur Endstation von T* und von dort zur Endstation von T
    alteKosten = KostenVon(R, T*_s, T*_e) + KostenVon(R, T*_e, T_e)
    Wegpunkt P = T*_s
  Sonst Wenn keine Überlappenden Aufträge existieren
    Dann
      A = voriger Wegpunkt mit ein- oder aussteigendem Kunden
      Wenn A genau einen aussteigenden Kunden hat
        Dann
          T* = A
          P = voriger Wegpunkt mit ein- oder aussteigendem
            Kunden
          alteKosten = KostenVon(R, P, T_e) + KostenVon(R, T*_e, T_e)
        Sonst -
      Sonst -
    // wie würden die Kosten im günstigsten Fall aussehen, wenn der vorige
    // Auftrag abgegeben würde
    minimaleNeueKosten = KostenFürDenKürzestenWeg(P, T_s) +
      KostenFürDenKürzestenWeg(T_s, T_e)
    Wenn alteKosten > minimaleNeueKosten
      Dann
        Versuche T* abzugeben
    Sonst -
Ende

```

**Listing 7-5: Suche nach abzugebenden Transportaufträgen****7.3.2.1.2 Identifikation von Kooperationspartnern**

Die Identifikation von Kooperationspartnern ist im Fall des Austauschs von Aufträgen mithilfe von Blackboards realisiert (vergleiche 5.1.3.1.3). Als Blackboards agieren in diesem Fall die Stationsagenten. Indem die Transportaufträge an den Stationen ausgeschrieben werden, in denen sie beginnen, können Fahrzeugagenten gezielt nach Aufträgen suchen, die sich bereits (zumindest teilweise) auf ihrer Route befinden. Dabei wird auch das Einsparungspotenzial veröffentlicht, das in Listing 7-5 ermittelt wurde. Kann ein Fahrzeug den Transportauftrag zu geringeren Kosten realisieren, bietet es die Übernahme beim abgebenden Fahrzeug an. Da die Ausschreibung auf Grundlage der größtmöglichen Einsparung erfolgte, muss nun überprüft werden, ob sich die Abgabe tatsächlich lohnt. Das abgebende Fahrzeug plant dazu seine Route ohne den abzugebenden Auftrag und ermittelt so die tatsächliche Einsparung. Wenn tatsächlich eine Einsparung entsteht, wird diese an das übernehmende Fahrzeug übermittelt. Sind die Mehrkosten durch die Übernahme geringer als die Einsparung, wird der Transportauftrag getauscht.

### 7.3.2.2 Experimente und Ergebnisse zum Austausch von Transportaufträgen

Der Austausch von Transportaufträgen wurde in einem Vergleichsszenario evaluiert. Dabei wurden mit variierenden Parametern jeweils 100 unabhängige Testläufe in einem Transportnetzwerk mit 50 Stationen und 10 Fahrzeugen durchgeführt.

Als Vergleichsszenario diente ein Experiment, in dem 200 zufällige Transportanfragen innerhalb eines Zeitraums von 18 Stunden in das System gestellt wurden. Die einzelnen Transportaufträge besaßen ein Zeitfenster (zwischen frühester Abfahrtszeit und spätester Ankunftszeit) zwischen 3 und 6 Stunden. In diesem Experiment wurde eine durchschnittliche Einsparung an Streckennutzungsgebühren von 15% erreicht. Dabei schwankten die Einsparungen in den Testläufen zwischen 3% und 35%. Die durchschnittliche Bedienrate (Anteil der erfolgreich bearbeiteten Transportaufträge) lag bei durchschnittlich 49% (zwischen 34% und 67%).

In einem abweichenden Szenario wurde untersucht, wie sich längere Zeitfenster auf das System auswirken. Hier konnte natürlich ein größerer Anteil von Aufträgen bearbeitet werden (zwischen 59% und 91%, im Durchschnitt 78%). Dazu wurden Transportaufträge mit einer Dauer von 6 bis 12 Stunden generiert. Es ergab sich eine durchschnittliche Einsparung von 18%. Die Einsparung in den einzelnen Läufen schwankte zwischen 3% und 56%. Das Verfahren ist also in der Lage, die größeren Zeitfenster in der Planung auszunutzen.

In einem weiteren Vergleichsszenario wurde untersucht, welche Auswirkungen eine geringe Auslastung hat. Dazu wurden nur 100 Transportanfragen innerhalb der 18 Stunden generiert, die aber auch nur von 5 Fahrzeugen bedient werden konnten. Im Ergebnis konnten nur wenige Transportaufträge bedient werden. Das liegt daran, dass die Fahrzeuge sich stärker im Netzwerk verteilen und längere Leerfahrten zwischen den Transportaufträgen absolvieren müssen. Da sich auch nur wenige Transportaufträge überhaupt überlappten, fanden sich weniger Kooperationspartner, die einen ausgeschriebenen Auftrag übernehmen konnten. Daher lag die durchschnittliche Einsparung nur bei 3% und in einigen Testläufen konnte gar keine Einsparung realisiert werden.

Zusammenfassend kann festgestellt werden, dass der Mechanismus des Austauschs von Transportaufträgen durchaus geeignet ist, Fehlentscheidungen im Online-Planungsprozess zu revidieren. Dabei sind zwei entscheidende Einflussfaktoren auszumachen: Zunächst profitiert das Verfahren davon, wenn die Transportaufträge wenig restriktiv sind, also großzügige Zeitfenster besitzen. Das liegt daran, dass sich eher Fahrzeuge finden lassen, die einen Transportauftrag übernehmen können. Für ein reales System wären hier finanzielle Anreize für eine hohe Flexibilität der Kunden in Erwägung zu ziehen. Außerdem zeigt sich, dass der Mechanismus nur bei einer gewissen Auslastung des Systems effektiv ist. Ist diese nicht gegeben, sind ggf. andere Optimierungsstrategien zu nutzen.

### 7.3.2.3 Konvoibildung im RailCab-System

Durch die Bildung eines Konvois sind die Fahrzeuge des RailCab-Systems in der Lage, „günstiger“ im Schienennetzwerk zu verkehren. Aufgrund des Windschatten-Effekts reduziert sich der Luftwiderstand der Fahrzeuge, die nicht an der Spitze des Konvois fahren, um etwa 80%. Auf ebenen oder fast ebenen Strecken hängt der Leistungsbedarf des Fahrzeugs fast ausschließlich vom Luftwiderstand ab. Dementsprechend besteht hier ein großes Einsparungspotenzial. Es handelt sich bei dieser Kooperation um eine Abstimmung von Aktivitäten. Abzustimmen ist hier das Befahren von Streckenabschnitten, das zeitlich und räumlich so zusammengefasst wird, dass einzelne Fahrzeuge gleichzeitig auf denselben Streckenabschnitten verkehren. Die Anpassung der ursprünglichen Routenpläne hat dabei so zu erfolgen, dass die sich anschließenden Teile der Routenpläne der beteiligten Fahrzeuge eingehalten werden können.



Betrachtet man das Konvoiplanungsproblem zunächst aus einer zentralen Perspektive und unterstellt ein statisches Problem (im Gegensatz zum tatsächlichen Online-Problem), fallen Ähnlichkeiten zum Vehicle Routing Problem (VRP) auf. Im Vehicle Routing Problem ist eine Menge von Bedarfsorten in mehreren Touren anzufahren, sodass der insgesamt zurückgelegte Weg minimiert wird. Dieses sehr komplexe Problem<sup>61</sup> zerfällt in zwei Teilprobleme: eine Gruppierung von Bedarfsorten, die durch ein Fahrzeug innerhalb einer Tour angefahren werden, und die Bestimmung der kürzesten Route, die diese Orte verbindet. Daraus resultiert die erhebliche Komplexität des Problems: Zunächst muss aus der Menge aller möglichen Touren die beste Kombination von Touren gewählt und anschließend das Rundreise-Problem für die einzelnen Touren gelöst werden. Die Lösung des ersten Teilproblems beeinflusst offensichtlich die Lösung des zweiten Teilproblems, sodass beide Schritte für alle gültigen Kombinationen durchgeführt werden müssen, um zu einer optimalen Lösung zu gelangen.

Die kombinatorischen Eigenschaften der Konvoiplanung sind sehr ähnlich. In der zentralen Offline-Variante sind aus allen Fahrzeugen diejenigen zusammenzufassen, die die besten Konvois bilden. Anschließend ist die beste Route für jeden Konvoi zu finden, wobei die Stationen, an denen die beteiligten Fahrzeuge Güter oder Passagiere aufnehmen oder absetzen, innerhalb bestimmter Zeitfenster zu durchfahren sind. Bei der Bildung der Konvois ist zu berücksichtigen, dass einige Fahrzeuge von ihrer ursprünglichen (für die Alleinfahrt) optimalen Route abweichen werden. Die Verbesserung des Konvois muss diese Abweichung mindestens wieder aufwiegen. Aufgrund dieser kombinatorischen Eigenschaften ist es notwendig, den untersuchten Lösungsraum möglichst schnell einzuschränken.

#### 7.3.2.3.1 Modellieren der abgestimmten Aktionen

Die Modellierung der abgestimmten Aktionen im Fall der Konvoifahrt ist problemlos möglich. Ein Fahrzeug, das an der Spitze eines Konvois oder alleine fährt, verbraucht abhängig von der Länge eines Streckenstücks und seiner Eigenschaften (z. B. Steigung) Energie. Durch die Fahrt als nachfolgendes Fahrzeug reduziert sich dieser Energieverbrauch um einen bestimmten Anteil. Entsprechend kann die Fahraktion aus Unterkapitel 2.1.3 angepasst werden. Listing 7-6 zeigt, wie die Konvoifahrt durch eine zusätzliche Fahraktion dargestellt werden kann. Anhand des von der Alleinfahrt abweichenden Energieverbrauchs können die Fahrzeuge die Möglichkeit erkennen, den Plan zu verbessern.

```
(:action drive_convoy
:parameters
  (?r railcab ?s railcab ?l1 ?l2 location

:precondition
  (and at ?r ?l1 and at ?s ?l2)
  (>=(energy_storage ?l)(*distance ?l1 ?l2)(convoyConsumption ?r)))

:effect
  (and (not (at ?r ?l1))
  (at r? ?l2)
  (increase (total-energy-used) (*distance ?l1
  ?l2)(convoyConsumption ?r)))
  (decrease (energy_storage ?r) (*distance ?l1
  ?l2)(convoyConsumption ?r)))
)
```

**Listing 7-6: Fahraktionen mit Berücksichtigung der Kooperationsform "Konvoi"**

#### 7.3.2.3.2 Erkennen des Kooperationspotenzials

<sup>61</sup> Das VRP ist np-hart.

Die Fahrzeugagenten entdecken das Einsparungspotenzial durch Analyse ihrer aktuellen Routenpläne. Finden sich mehrere zusammenhängende Streckenabschnitte, auf denen das Fahrzeug noch nicht im Konvoi fahren soll, besteht die Möglichkeit, eben durch die Bildung eines Konvois die Fahrtkosten zu reduzieren.

### 7.3.2.3.3 Identifikation von Kooperationspartner und Bestimmung der Ziele

Die Identifikation der Kooperationspartner ist im Fall der Konvoiplanung vergleichsweise kompliziert. Die Eignung eines anderen Fahrzeugs als Konvoipartner hängt ganz entscheidend von den ursprünglichen Routen der Fahrzeuge ab. Je besser sie bereits vor der Konvoiplanung zusammenpassen, desto leichter lässt sich der Konvoi anschließend bilden. Der Prozess der Identifikation von Kooperationspartnern macht von mehreren Techniken Gebrauch, die in Abschnitt 5.1.3.1 beschrieben sind.

Die erste Technik ist die Kombination von Middle-Agents und verteiltem Matchmaking. Der erste Schritt der Konvoiplanung besteht in einer verteilten Tiefensuche, die durch die Stationen durchgeführt und durch einen Fahrzeugagenten ( $F_1$ , I für Initiator) ausgelöst wird. Die Suche läuft entlang des Abschnitts der ursprünglichen Route des Fahrzeugagenten  $F_1$ , auf der ein Konvoi gebildet werden soll. Gesucht wird nach Fahrzeugen, die die gleichen oder benachbarte Stationen durchfahren. Dabei leiten die Stationen die Anfrage an benachbarte Stationen weiter, wodurch eine verteilte Tiefensuche im Streckennetzwerk entsteht. Um die Suchzeit zu beschränken und außerdem nur solche Fahrzeuge zu finden, die auch tatsächlich in der gleichen Region wie das auslösende Fahrzeug agieren, lässt sich die Suchtiefe beschränken. Die Stationen leiten die Anfrage nur weiter, wenn ihre eigene *Tiefe* einen Schrankenwert nicht überschreitet. Die Tiefe  $t$  einer Station berechnet sich durch:

$$t(s) = \begin{cases} 0, & \text{wenn die Station vom Fahrzeugagenten befragt wird} \\ d(s') + 1, & \text{wenn die Station von Stationsagenten } s' \text{ befragt wird} \end{cases}$$

Dadurch enthält bereits das Suchergebnis nur solche Agenten, deren Routen zumindest durch eine Station in der Nähe der Route von  $F_1$  führen. Das Suchergebnis und damit der Lösungsraum kann durch Anwendung von *Filtering* weiter verkleinert werden. Die Konvoiplanung setzt zwei unterschiedliche Filter ein. Der erste Filter entfernt solche Fahrzeuge, die außerhalb eines definierten Zeitfensters in der Nähe des auslösenden Fahrzeugs verkehren:

$$t_{first} \leq t_i \leq t_{end}$$

Für jedes Fahrzeug wird jede Station innerhalb des durchsuchten Teils des Transportnetzwerks betrachtet. Liegt die Ankunftszeit in der Station  $i$  ( $t_i$ ) nicht innerhalb des Zeitfensters zwischen der Ankunft von  $F_1$  in der ersten Station der betrachteten Teilroute ( $t_{first}$ ) und dem Verlassen der letzten Station ( $t_{end}$ ), kommt es für eine Konvoibildung nicht in Betracht.

Der zweite Filter nutzt Informationen über die allgemeine Reiserichtung der Fahrzeuge aus, um weitere Fahrzeuge aus der Kandidatenmenge zu eliminieren. Dazu wird ein Richtungsvektor für jedes Fahrzeug berechnet. Der Vektor der Koordinaten der letzten Station im Suchbereich  $s_i$  wird von den Koordinaten der ersten Station im Suchbereich subtrahiert:  $r_i = e_i - s_i$ . Dieser Richtungsvektor wird verglichen mit dem Richtungsvektor des Fahrzeugs  $F_1$  durch Berechnung des Schnittwinkels  $\alpha_i$ :

$$\sin \alpha = \frac{\langle r_o, r_i \rangle}{|r_o||r_i|} \Leftrightarrow \alpha = \arcsin \frac{\langle r_o, r_i \rangle}{|r_o||r_i|}$$

Die eigentliche Filterregel ist definiert durch einen maximalen Winkel  $\gamma$ , um den die Richtung eines Fahrzeugs maximal von  $F_1$  abweichen darf. Die Filterung kann dabei jeweils direkt durch die jeweilige Station erfolgen, da sie über alle notwendigen Informationen verfügt.

Als Zwischenschritt muss der auslösende Fahrzeugagent  $F_1$  bei den Agenten aus der Ergebnismenge anfragen, ob sie an einer Kooperation interessiert sind. Diese Anfrage ist notwendig, da die weiteren Schritte einen Austausch der Agenten über ihre lokalen Pläne voraussetzen. Im Rahmen dieses Austausches erfolgt auch die Bestimmung der gemeinsamen Ziele. Mit einer Zusage übermitteln die angefragten Agenten alle Stationen, die sie im Suchraum der verteilten Tiefensuchen anfahren müssen, um Passagiere oder Transportgüter aufzunehmen oder abzusetzen. Weiterhin wird die Ankunftszeit in der ersten Station außerhalb des Suchraums übermittelt. Diese Station muss rechtzeitig erreicht werden (also wie im Ursprungsplan vorgesehen), um die restliche Route des Fahrzeugs beibehalten zu können.

Als letzter Schritt der Identifikation von Kooperationspartnern sind im Fall der Konvoiplanung Gruppen von Fahrzeugen zu bilden, die vermutlich einen guten Konvoi bilden können. Dabei kann ausgenutzt werden, dass die einzelnen Fahrzeuge bereits ein Optimierungsproblem gelöst haben, das sehr ähnlich zur Routenplanung für den Konvoi ist. Alle Fahrzeuge haben bereits eine Route gebildet. Offensichtlich ist das Optimierungsproblem „*finde Konvoi-route*“ genau dann gut zu lösen, wenn die Randbedingungen der einzelnen Fahrzeuge sich ähneln (sie also zu ähnlichen Zeiten in Stationen sein müssen, die nahe aneinander liegen). Über die Menge der Kandidaten wird ein Clustering [JMF99] durchgeführt, wobei die Restriktionen der lokalen Routenplanungsprobleme der einzelnen Agenten in das Distanzmaß eingehen. Damit befinden sich in einem Cluster solche Agenten, deren lokale Optimierungsprobleme ähnliche Nebenbedingungen haben. So sind die Chancen besonders hoch, dass das gemeinsame Optimierungsproblem eine Lösung besitzt. Beim verwendeten Clustering handelt es sich um ein hierarchisches Clustering, dessen Ergebnis eine Baumstruktur ist. Knoten des Baumes sind Cluster, wobei jeder Cluster die Vereinigung seiner Kindsknoten ist. Die Wurzel enthält sämtliche Fahrzeugagenten, während die Blätter aus einzelnen Fahrzeugagenten bestehen. Jeder Level definiert also eine Lösung für das Gruppierungsproblem in der Konvoiplanung. Dadurch wird das Problem der Einteilung der Fahrzeuge in Gruppen auf die Auswahl der günstigsten Ebene reduziert.

#### **7.3.2.3.4 Gemeinsame Routenplanung**

Die gemeinsame Planung zum Abschluss der Konvoiplanung kann zurückgeführt werden auf ein einfaches Routenplanungsproblem mit Zeitfenstern. Gesucht ist eine Route, die die Pflichthaltepunkte verbindet und die späteste Austrittszeit aller Fahrzeuge einhält. Zur Lösung dieses Routenplanungsproblems wurde eine Insertion-Heuristik angewandt, wie sie aus dem Operations Research als Teillösung des VRP bekannt ist [CS04]. Dieses einfache und schnelle Verfahren wird eingesetzt, um die Lösungsqualität einer Ebene aus dem Clustering zu ermitteln und anschließend mit den Ergebnissen der übrigen Ebenen zu vergleichen. Dabei kann die Baumstruktur ausgenutzt werden, um die Berechnungsergebnisse wieder zu verwenden. Da alle Fahrzeuge (mit ihren lokalen Randbedingungen) innerhalb eines Clusters auf der Ebene  $i+1$  auch auf der Ebene  $i$  im selben Cluster sind, besitzt der Cluster auf der Ebene  $i$  mindestens die gleichen Randbedingungen wie sein Kindsknoten. Daher ergibt sich aus der Unlösbarkeit des Routenplanungsproblems auf der Ebene  $i+1$  die Unlösbarkeit des Problems auf der Ebene  $i$ . Es muss also auf der Ebene des Elternknotens keine erneute Routenplanung durchgeführt werden. Stattdessen können im Falle einer Unlösbarkeit die Ergebnisse von der untersten Ebene mit gültiger Lösung (im schlimmsten Fall die Ebene mit einzelnen Fahrzeugagenten) nach oben propagiert werden.

#### **7.3.2.4 Experimente und Ergebnisse zur Konvoiplanung**

Experimente mit einer prototypischen Implementierung haben gezeigt, dass in 5% aller ausgelösten Planungsläufe ein Konvoi erfolgreich gebildet wurde. Dieses Ergebnis ist vor dem Hintergrund der Vielzahl von Beschränkungen (Zeitfenster der Kunden und Belegung der Tracks und der Stationen) akzeptabel. Wurde ein Konvoi erfolgreich gebildet, konnte unter Berücksichtigung der Abweichungen von den (ggf. kürzeren) Ursprungsrouten 15%-20% der Energiekosten eingespart werden. Dabei wurde eine durchschnittliche Ersparnis von 50% je befahrenem Streckenabschnitt angenommen (auf ebener Strecke liegt die Einsparung bei fast 80%).

## 8 Zusammenfassung und Ausblick

Zunehmend *intelligente* Maschinen sollen den Aufgabenbereich und den Nutzwert technischer Systeme auf eine neue Stufe anheben. Die Entwicklung der jungen Disziplin der Mechatronik stellt einen wichtigen Schritt in diese Richtung dar. Durch das Zusammenwirken von Maschinenbau, Elektrotechnik und Informatik werden die technischen Systeme in die Lage versetzt, relevante Aspekte ihrer Umgebung zu erkennen und ihr Verhalten auf diese Aspekte anzupassen.

### 8.1 Zusammenfassung

Ein wesentlicher Aspekt von Intelligenz, der in heutigen mechatronischen Systemen unberücksichtigt bleibt, ist die Fähigkeit zur Planung. Planung versetzt intelligente Systeme in die Lage, zukünftige Handlungsweisen und Situationen zu antizipieren und bereits im Vorfeld das eigene Verhalten anzupassen. Proaktive Handlungsweisen anstelle einer reinen Reaktion auf Ereignisse werden damit möglich. Ziel der Arbeit war also die Konzeption einer Verhaltensplanung für mechatronische Systeme.

Vergleicht man bestehende Modelle, die im Rahmen einer Planung eingesetzt werden können, und Modelle zur Beschreibung des Verhaltens mechatronischer Systeme, fällt eine zu schließende Lücke auf. Die meisten Planungsformalismen betrachten Handlungsweisen und Aktivitäten als eine Art diskrete Blackbox, also begrenzt durch einen klar definierten Zustand vor der Ausführung und einen klar definierten Zustand nach der Ausführung. Für mechatronische Systeme ist dagegen der kontinuierliche Verlauf einer Handlungsweise von sehr großer Bedeutung. Diese Lücke musste geschlossen werden.

Darüber hinaus waren zwei weitere Aspekte unbedingt zu beachten: Unsicherheit (also unbekannte oder stochastische Einflüsse) und mögliche Interaktion mit anderen Systemen. Konventionelle Planungsverfahren lassen diese Aspekte außer Acht. Spezielle Planungsverfahren und Modelle, die diese Aspekte berücksichtigen, passen nicht vollständig auf die Anforderungen einer Verhaltensplanung für mechatronische Systeme. Für beide Aspekte sind in dieser Arbeit daher angepasste Planungsverfahren entwickelt worden.

Die Lücke zwischen der diskreten Formulierung der Planungsprobleme und dem kontinuierlichen Verhalten der mechatronischen Systeme wurde geschlossen. Dazu wurden zwei mögliche Ansätze beschrieben. Der erste Ansatz verfolgt eine Bestimmung kontinuierlicher Verläufe durch eine dem Betrieb vorgelagerte Optimierung. Die so bestimmten Verläufe werden in den verwendeten Planungsformalismus integriert, indem die Optimierungsergebnisse als diskrete Kennzahlen verwendet werden. Die zweite Möglichkeit besteht in der Approximation kontinuierlicher Charakteristika von Verhaltensweisen im Betrieb oder im Rahmen von Experimenten vor dem Betrieb. Das jeweilige Vorgehen hängt vom Anwendungsfall und beispielsweise den Eigenschaften der einsetzbaren Optimierungsprobleme ab.

Um einen effizienten und intelligenten Umgang mit unsicheren oder stochastischen Einflüssen zu ermöglichen, wurde ein Konzept zur Vereinigung der beiden Planungsparadigmen *bedingte Planung* und *Planüberwachung und Neuplanung* erarbeitet. Das Ziel des Konzepts ist die proaktive Erzeugung von optimalen Plänen für wahrscheinliche alternative Ausführungsverläufe. *Planüberwachung und Neuplanung* dient als eine Rückfallebene, falls unvorhergesehene Ereignisse eintreten. Da *Planüberwachung und Neuplanung* keine optimalen Planungsergebnisse gewährleisten kann, ist das Konzept so ausgelegt, dass das Auftreten eines unvorhergesehenen Zustands besonders unwahrscheinlich ist.

Um die Wahrscheinlichkeiten der möglichen Ausführungsverläufe beurteilen zu können, wurde eine ausdrucksstarke Modellierung auf Basis von Bayes-Netzwerken entwickelt und in den Planungsprozess integriert. Durch die Aufnahmen von Expertensystemen oder –agenten gelingt es außerdem, aktuelle und spezifische Umweltinformationen in die Planung zu integrieren.

In der Behandlung der Interaktion zwischen mechatronischen Systemen oder eben der Interaktionsplanung sind zunächst zwei unterschiedliche verschiedene Situationen identifiziert worden. In der ersten Situation wirken die interagierenden mechatronischen Systeme zusammen, um gemeinsam eine Aufgabe zu erfüllen. Daraus lässt sich ableiten, dass die Systeme sich an einem gemeinsamen Zielsystem ausrichten und sich über ihre jeweiligen lokalen Pläne austauschen. Im zweiten Fall haben die Systeme jeweils eigene Aufgaben zu erfüllen, weswegen diese starken Annahmen nicht mehr aufrechtzuerhalten sind.

Für die erste Situation wurde ein Planungsmodell entwickelt, das es den mechatronischen Systemen ermöglicht, die Auswirkungen ihrer eigenen Handlungsweise auf andere Systeme zu ermitteln und im Rahmen der Planung zu berücksichtigen. Anhand des Planungsmodells konnte gezeigt werden, wie mittels Planungsverfahren, die vollständig geordnete Pläne erzeugen, eine Koordination der Systeme im Sinne ihres gemeinsamen Zielsystems erfolgen kann. Dabei wurden zwei aufeinander folgende Phasen beschrieben: die Koordination vor der Planung und die Koordination nach der Planung, die zur Behandlung unerwarteter Verhaltensweisen eingesetzt werden kann. Um den Rechenaufwand der Umplanung im Rahmen der Koordination nach der Planung zu begrenzen und möglichst mit der Planausführung zu verzahnen, wurde ein rollierendes Planungskonzept mit beschränktem Planungshorizont vorgeschlagen. Somit kann eine Koordination auch bei komplexen Planungsproblemen erfolgen.

Für die zweite Situation wurden zunächst Kooperationsformen herausgearbeitet (Abgabe von Aufträgen, Abgabe von Teilfunktionen und Abstimmung von Aktivitäten), die zwischen mechatronischen Systemen mit individuellen Aufgaben möglich sind. Anschließend wurde für jede der Kooperationsformen erläutert, wie sie mittels Techniken aus dem Bereich der Multiagenten-Systeme und durch Einbettung in die lokale Planung der Systeme durchgeführt werden. Es wurde auch ein Vorgehen zur Integration der unterschiedlichen Kooperationsformen vorgeschlagen.

Im Rahmen eines ausführlichen Anwendungsbeispiels aus der *Neuen Bahntechnik Paderborn* konnte gezeigt werden, wie die Planungsverfahren und –modelle auf den unterschiedlichen mechatronischen Funktionsebenen eingesetzt werden können. Dabei wurde auch erarbeitet, wie die unterschiedlichen Planungsebenen ineinandergreifen können.

Im Rahmen einer stochastischen Routenplanung konnte gezeigt werden, wie zuverlässigere Aussagen über Fahrzeiten und bessere Fahrpläne mithilfe der vorgeschlagenen Modellierung auf Basis von Bayes-Netzwerken erreicht werden können. Dabei konnte in Experimenten gezeigt werden, dass der Ansatz auch robust gegen eine ungenaue Spezifikation der Parameter des Modells ist.

Die RailCab-Module AGAS und Antrieb dienen als Beispiel für die Interaktionsplanung innerhalb eines mechatronischen Systems. Hier gelang es, auf Basis der vorgeschlagenen Modellierung die Querverwirkung des Antriebs auf das AGAS in die lokale Planung des Antriebs aufzunehmen. An diesem Beispiel bestätigte sich außerdem, dass bei Zerteilung des Planungsproblems auf mehrere Modelle eher gut lösbare Problemformulierungen gefunden werden können als in integrierten zentralen Planungsmodellen.

Die Planungsverfahren für die Interaktion zwischen Systemen wurden am Beispiel der Rail-Cabs demonstriert. In enger Anlehnung an die zuvor definierten Abläufe wurde gezeigt, wie durch die gezielte Abgabe von Transportaufträgen und die bewusste Bildung von Konvois Umwege und Energiekosten eingespart werden können.

Die Ableitung eines diskreten Planungsformalismus aus einem vorgelagerten Optimierungsproblem wurde am Beispiel der aktiven Federung illustriert. Durch die Integration ist es möglich, das kontinuierliche Gütemaß *Komfort* auch in der diskreten Planung einzusetzen. Weiterhin können kontinuierliche Entscheidungsgrößen für die Ausführung einer Verhaltensweise aus dem diskreten Plan abgeleitet werden.

## **8.2 Ausblick**

Nicht alle der in Kapitel 5 beschriebenen Ansätze konnten an den Anwendungsbeispielen demonstriert werden. So wäre es wünschenswert, die noch nicht demonstrierte Kooperationsform *Abgabe von Teilfunktionen*, die im RailCab-Beispiel nicht sinnvoll anzuwenden ist, auch in einem konkreten System durchzuführen. Ein geeignetes Anwendungsbeispiel scheinen flexible Fertigungssysteme zu sein, in denen die Fertigungsmaschinen beispielsweise die Schritte eines Arbeitsplans (Teilfunktionen) optimal unter sich aufteilen.

Ein zweiter wesentlicher Aspekt, der aufgrund nicht zur Verfügung stehender Daten aus den Demonstratoren aktive Federung, Antrieb und AGAS nicht durchgeführt werden konnte, ist das Erlernen der Parameter der unterschiedlichen Planungsmodelle.

Es wurde vorgeschlagen, diese automatisierte Modellbildung mithilfe von existierenden Lernverfahren durchzuführen. Damit handelt es sich hierbei um ein lohnendes Arbeitsgebiet, um diesen Beitrag zu einer Verhaltensplanung für intelligente interagierende mechatronische Systeme in nicht-deterministischen Umgebungen abzurunden.

Natürlich können die allgemein formulierten Konzepte auch auf neue Anwendungen übertragen werden und dabei verbessert und validiert werden.





## Literaturverzeichnis

- [ADG+08] Adelt, P.; Donoth, J.; Gausemeier, J.; Geisler, J.; Henkler, S.; Kahl, S.; Klöpfer, B.; Krupp, A.; Münch, E.; Oberthür, S.; Paiz, C.; Porrmann, M.; Radkowski, R.; Romaus, C.; Schmidt, A.; Schulz, B.; Vöcking, H.; Wittkowski, U.; Witting, K., Znamenshchikov, A.: Selbstoptimierende Systeme des Maschinenbaus - Definitionen, Anwendungen und Konzepte, HNI-Verlagsschriftenreihe, 2008
- [APM01] Amgoud, L., Parson, S., Maudet, N.: Arguments, dialogue, and negotiation. In: 14th European Conference on Artificial Intelligence, IOS Press, Amsterdam, 2001, S. 338-342
- [BB01] Boutilier, C., Brafman, R.I.: Partial-Order Planning with Concurrent Interacting Actions. In: Journal of Artificial Intelligence Research, Vol. 14, 2001, S. 105-136
- [BCLP03] Bertoli, P.; Cimatti, A.; Lago, U. D.; Pistore, M.: Extending PDDL to Nondeterminism, Limited Sensing and Iterative Conditional Plans. In: Workshop on PDDL, Proceedings of ICAPS'03, 2003.
- [BCM00] Beynon, M.; Curry, B.; Morgan, P.: The Dempster-Shafer theory of evidence: an alternative approach to multicriteria decision modelling. In: The International Journal of Management Science, 28, 2000, S. 37-50
- [BCR01] Bertoli, R., Cimatti, A., Roveri, M.: Conditional Planning under Partial Observability as Heuristic-Symbolic Search in Belief Space. Technical Report, IRST, Trento, Italy, 2001
- [BDM+02] Bresina, J.L., Dearden, R., Meuleau, N., Smith, D.E., Washington, R.: Planning Under Continuous Time and Resource Uncertainty: A Challenge for AI. In: AIPS Workshop on Planning for Temporal Domains, 2002, S. 91-97
- [BF97] Blum, A.L. und Furst, M.: Fast planning through planning graph analysis. In: Artificial Intelligence, 90, S. 279-298., 1997
- [BKR+97] Binder, J.; Koller, D.; Russel, S.; Kanazawa, H.: Adaptive Probabilistic Networks with Hidden Variables. In: Machine Learning, 29, 1997, S. 213-244.
- [BL00] Barbuceanu, M., Lo, W.-K.: A multi-attribute utility theoretic negotiation architecture for electronic commerce. In: AGENTS '00, Proceedings of the fourth international conference on Autonomous agents, ACM Press, New York 2000 S. 239-246
- [Blyt99] Blythe, J.: An Overview of Planning Under Uncertainty. In: Artificial Intelligence Today, Lecture Notes of Artificial Intelligence, 1600, 1999, Springer Verlag, Berlin, S. 85-110
- [BMVW06] Buzing, P., ter Mors, A., Valk, J., Witteveen, C.: Coordinating Self-interested Planning Agents. In: Autonomous Agents and Multi-Agent Systems, 12, 2006, S. 199-218
- [Bren03] Brenner, M.: Multiagent Planning with Partially Ordered Temporal Plans. In: Proceedings of the 9th International Conference on Artificial Intelligence (IJ-CAI-03), Morgan Kaufmann, San Francisco, 2003
- [BTG04] Burmester, S.; Tichy, M.; Giese, H.: Modeling Reconfigurable Mechatronic Systems with Mechatronic UML. In: Proceedings of Model Driven Architecture: Foundations and Application (MDAFA 2004), 2004
- [CB03] Clement, B.J., Barret, A.C.: Continual coordination through shared activities. In: AAMAS'03: Proceedings of the Second International Conference on Autonomous agents and Multiagent Systems, ACM Press, New York, 2003, S. 57-64

- [CCI89] N.N.: CCIT Z.100 (1989) „Specification and Description Language SDL“. Recommendation Z.100, Geneva: ITU 1989
- [CCI93] N.N.: CCIT Z.100 (1993) „Specification and Description Language SDL“. Recommendation Z.100, Geneva: ITU 1993
- [CCO02] Calabuig, V., Cunyat, A., Olcina, G.: Commitment and choice of partner in negotiation with a deadline. In: Spanish Economic Review, 4(1), 2002, S. 61-78
- [CDL+03] Cowell, Robert G.; Dawid, A. Philip; Lauritzen, Steffen L.; Spiegelhalter, David J.: Probabilistic Networks and Expert Systems. Springer Verlag, Berlin, 2003
- [CGGT97] Cimatti, A.; Giunchiglia, E.; Giunchiglia, F.; Traverso, P.: Planning Via Model Checking: A Decision Procedure for AR. In: Proceeding of the 4th European Conference on Planning (ECP'97), Springer Verlag, 1997, S. 130–142
- [CGH97] Castillo, Enrique; Gutiérrez, José Manuel; Hadi, Ali S.: Expert Systems and Probabilistic Network Models. Springer Verlag, Berlin, 1997.
- [CH71] Clifford, P.; Hammersley, J.M.: Markov fields on finite graphs and lattices. Technical Report, unpublished, 1971.
- [CJ04] Coehoorn, R.M., Jennings, N.R.: Learning an Opponents Preferences to Make Effective Multi-Issue Negotiation Trade-Offs. In: Proceedings of the 6th International Conference on Electronic Commerce, 2004
- [CLRS01] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. Second Edition, MacGraw-Hill, New York, 2001
- [Coni08] Conitzer, V.: Comparing Multiagent Systems Research in Combinatorial Auctions and Voting. In: 10th International Symposium on Artificial Intelligence and Mathematics, 2008
- [CR99] Cimatti, A.; Roveri, M.: Conformant Planning via Model Checking. In: European Conference on Planning. In: Lecture Notes in Artificial Intelligence, 1809, Springer Verlag, Berlin, 1999, S. 21–34
- [CRS07] Cobb, B.R.; Rumi, R., Salmerón, A.: Bayesian Network Models with Discrete and Continuous Variables. In: Advances in Probabilistic Graphical Models, Series: Studies in Fuzziness and Soft Computing, 213. Springer Verlag, Berlin S. 81-102.
- [CS04] Campbell, A. M. ; Savelsbergh, M.: Efficient insertion heuristics for vehicle routing and scheduling problems. In: Transportation Science, 38, 2004, S. 369–378
- [DDOW99] DesJardins, M.E., Durfee, E.H., Ortiz, C.L., Wolverton, M.J.: A Survey of Research in Distributed Planning. In: AI Magazine, 20, 4, 1999, S. 13-22
- [Dear05] Dearden, R.: Planning and Learning in Hybrid Discrete-Continuous Models, International Joint Conference on Artificial Intelligence, 2005
- [DGLH92] Decker, K.S., Garvey, A.J., Lesser, V.R., Humphrey, M.A.: An approach to modeling environment and task characteristics for coordination. In: Proceedings of the First International Conference on Enterprise Integration Modeling, MIT Press, Cambridge, 1992.
- [DHW94] Draper, D., Hanks, S., Weld, D.: Probabilistic planning with information gathering and contingent execution. In: Proceedings Second International Conference on Artificial Intelligence Planning Systems, 1994 S. 31-37
- [DL00] Decker, K.S., Li, J.: Coordinating mutually exclusive resources using gppg. In: Autonomous Agents and Multi-Agent Systems, 3, 2000. S. 113-157
- [DL87] Durfee, E.H., Lesser, V.R.: Planning coordinated action in dynamic domains. In: Proceedings of the DARPA Knowledge-Based Planning Workshop, 1987. S. 18.1-18.10

- [DL92] Decker, K.S., Lesser, V.R.: Generalizing the partial global planning algorithm. In: International Journal of Intelligent and Cooperative Information Systems, 1992, S. 319-346
- [DL94] Decker, K.S., Lesser, V.R.: Designing a family of coordination algorithm. In: Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence (DAI-94), 1994, S. 65-84
- [DLR77] Dempster, A. P.; Laird, N. M.; Rubin, D. B.: Maximum likelihood from incomplete data via the EM algorithm. In: Journal of the Royal Statistical Society, 39, 1977
- [Durf01] Durfee, E.H.: Distributed problem solving and planning. In: Selected Tutorial Papers from the 9th ECCAI Advanced Course ACAI 2001 and Agent Link's, 3rd European Agent Systems Summer School on Multi-Agent Systems and Applications, Springer Verlag, Berlin, 2001, S. 118-149,
- [DW99] DeJardins, M.E., Wolverton, M.J.: Coordinating a Distributed Planning Systems. In: AI Magazine, 20, 1999, S. 45-53
- [DWS96] Decker, K., Williamson, M., Sycara, K.: Matchmaking and brokering. In: Proceedings of the Second International Conference in Multi-Agent Systems (ICMAS'96), 1996
- [Ferb01] Ferber, J.: Multiagentensysteme . Eine Einführung, Addison Wesley, Reading, 2001
- [FGK+04] Frank, U. Giese, H., Oberschelp, O., Schmidt, A., Vöcking, H., Witting, K.: Selbstoptimierende Systeme des Maschinenbaus – Definitionen und Konzepte. HNI-Verlagsschriftenreihe, Paderborn, Band 155, Heinz Nixdorf Institut, Universität Paderborn, 2004
- [Fike71] Fikes, R.E.: Monitored execution of robot plans produced by STRIPS. In: Proceedings of the IFIP Congress 71, 1971, S. 189-194,
- [FL03] Fox, M., Long, D.: PDDL 2.1: An Extension to PDDL for Expressing Temporal Planning Domains. In: Journal of Artificial Intelligence Research (JAIR), 20, AAAI Press, Menlo Park, 2003, S. 61-124
- [FMB+98] Füssel, D.; Ballè, P.; Moseler, O.; Willimoswski, M.; Höfling, T.: Residuenbasierte Fehlererkennung und Diagnose an komplexen Prozessen. at-Automatisierungstechnik 46, 1998, S. 435-443
- [FN71] Fikes, R.E., Nilsson, N.J.: STRIPS: A New Approach to Application of Theorem Proving to Problem Solving. In: Artificial Intelligence, 2, Elsevier Science Publisher, Essex, 1992, S. 189-208
- [Föl94] Föllinger, O.: Regelungstechnik – Eine Einführung in die Methoden und ihre Anwendung, Hüthig GmbH, Heidelberg, 1994
- [Frank04] Franke, H.: Eine Methode zur unternehmensübergreifenden Transportdisposition durch synchron und asynchron kommunizierende Agenten. Dissertation, Universität Paderborn, Heinz Nixdorf Institut, Wirtschaftsinformatik, insbesondere CIM, HNI-Verlagsschriftenreihe, Paderborn, Band 152, 2004
- [Fres92] Frese, E. (Hrsg): Handwörterbuch der Organisation, 3. Auflage, Poeschel-Verlag, Stuttgart, 1992
- [Frie98] Friedman, N. : The Bayesian Structural EM Algorithm, In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, 1998 , S. 129-138.
- [FS93] Fershtman, C., Seidmann, D.J.: Deadline and Inefficient Delay in Bargaining with Endogenous Commitment. In: Journal of Economic Theory, 60, 1993, S 306-321
- [FS06] Ferstl, O.K., Sinz, E.: Grundlagen der Wirtschaftsinformatik, Band 1, 5. Auflage, Oldenbourg, München, 2006

- [FSJ98] Faratin, P., Sierra, C., Jennings, N.R.: Negotiation Decision Function for Autonomous Agents. In: *International Journal of Robotics and Autonomous Systems*, 24, 1998, S.159-182
- [FWJ04] Fatima, S., Wooldridge, M., Jennings, N.R.: An agenda-based framework for multi-issue negotiation. In: *Artificial Intelligence*, 152, 2004, S. 1-45
- [FWJ05] Fatima, S., Wooldridge, M., Jennings, N.R.: Bargaining with Incomplete Information. In: *The Annals of Mathematics and Artificial Intelligence*, 44, 2005, S. 207-232
- [GB96] Goldman, R. P.; Boddy, M. S.: Expressive Planning and Explicit Knowledge. In: *Planning with Incomplete Information for Robot Problems: Papers from the 1996 AAAI Spring Symposium*, AAAI Press, Menlo Park, 1996, S. 17 –24
- [Gier98] Gierend, C.: Fuzzy-Control in Kraftwerken – Potentiale und Anwendungsmöglichkeiten. *VGB Kraftwerkstechnik*, 78, 1998, S. 116-125
- [GL05] Gerevini, A.; Long, D.: Plan Constraints and Preferences in PDDL3, Technical Report, Department of Electronics for Automation, University of Brescia, Italy, August 2005
- [GPP+98] Georgeff, M.; Pell B., Pollack, M. Tambe, M.; Wooldridge, M.: The Belief-Desire-Intention Model of Agency. In: *Proceedings of Intelligent Agents V. Agent Theories, Architectures, and Languages: 5th International Workshop, ATAL'98*, 1998, S. 1-10
- [GSS03] Gerevini, A.; Saetti, A.; Serina, I.: On Managing Temporal Information for Handling Durative Actions in LPG. In: *AI\*IA 2003: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, 2829, Springer Verlag, Berlin, 2003
- [GTK+08] Geisler, J.; Trächtler, A.; Witting, K.; Dellnitz, M.: Self-Optimization of the Guidance Module of a Rail-bound Vehicle. In: *Self-Optimizing Mechatronic Systems: Design the Future*, HNI-Verlagschriften Reihe, Paderborn, 2008
- [Hahn96] Hahn, D.: Strategische Fertigungsplanung. In: Kern, W., Schröder, H.-H., Weber, J. (Hrsg.): *Handwörterbuch der Fertigungswirtschaft*. 2. Aufl., Schäffer-Poeschel, Stuttgart, 1996
- [Heck95] Heckermann, D.: A Tutorial on Learning with Bayesian Networks – Microsoft Technical Report, MSR-TR-95-06, Redmond, 1995
- [HJA+05] Hill, J.C., Johnson, F.R., Archibald, J.K., Frost, R.L., Stirling, W.C: A Cooperative Multi-Agent Approach to Free Flight. In: *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM Press, New York, 2005, S. 1083-1080
- [HKW95] Hanks, S.; Kushmerick, N.; Weld, D. S.: An algorithm for probabilistic planning. In: *Artificial Intelligence*, 76, 1995, S.239-286
- [HN01] Hoffmann, J. ; Nebel, B.: The FF Planning System: Fast Plan Generation Through Heuristic Search. In: *Journal of Artificial Intelligence Research*, 14, 2001, S. 253–302
- [HS94] Haddawy, P., Suwandi, M.: Decision-theoretic planning using inheritance abstraction. In: *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems*, AAAI Press, Menlo Park, 1994, S. 266-271
- [Huan01] Huang, M.: Funktionsmodellierung und Lösungsfindung mechatronischer Produkte, Dissertationen, Fakultät für Maschinenbau, Universität Karlsruhe, 2001
- [HV05] Holleczeck, P., Vogel-Heuser, B.: UML for Systems Engineering (SysML) - Eine Notation zur Beschreibung von Systemen. Springer Verlag, Berlin, 2005
- [HWHC06] Hsu, C.W.; Wah, B. W.; Huang, R.; Chen, Y. X.: New Features in SGPlan for Handling Soft Constraints and Goal Preferences in PDDL3.0. In: *Proceedings of the Fifth International Planning Competition*, 2006

- [Iser99] Isermann, R.: Mechatronische Systeme – Grundlagen. Springer Verlag, Berlin, 1999
- [Jens01] Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer Verlag, Berlin, 2001
- [Jenn95] Jennings, N.R.: Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions. In: Artificial Intelligence, 75, 1995, S. 195-240
- [JMA04] Jang, M.-Y., Momen, A.A., Agha, G.: ATSpace: A Middle Agent to Support Application Oriented Matchmaking and Brokering and Services. In: Proceedings. IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT), 2004, S. 393-396
- [JMF99] Jain, A. K.; Murty, M.N.; Flynn, P. J.: Data clustering: a review. ACM Computing Surveys. 31, 1999, S. 264–323
- [Kahl95] Kahlert, J.: Fuzzy Control für Ingenieure. Analyse, Synthese und Optimierung von Fuzzy-Regelungssystemen. vieweg, Braunschweig/Wiesbaden, 1995
- [Kirk70] Kirk, D. E. : Optimal control theory . - Prentice-Hall , Englewood Cliffs, 1970
- [Kaut06] Kautz, H.: SATPLAN04: Planning as Satisfiability. In: Working Notes on the Fifth International Planning Competition, 2006,S. 45-46
- [KN80] Kindermann, R.; Snell, J. L.: Markov Random Fields and their Applications. In: Contemporary Mathematics,1. American Mathematical Society, Rhode Island, 1980.
- [Koen01] Koenig, S.: Minimax Real-Time Heuristic Search. In: Artificial Intelligence, 129, Elsevier Science Publishers, Essex, 2001, S. 165-197
- [Kolg50] Kolmogorov, A. N.: Foundations of the Theory of Probability. Chelsea Press, New York, 1950
- [Korf90] Korf, R.: Real-Time heuristic search. In: Artificial Intelligence, 42, 1990, S. 189-211,
- [KS95] Koenig, S., Simmons, R.: Real-Time Search in non-deterministic domains. In: Proceedings of the International Joint Conference on Artificial Intelligence, 1995, S. 1660-1667
- [Kuhn00] Kuhn, C.: Merkmalsgenerierung und Klassifikation. In: Berichtsband 10, Workshop „Fuzzy Control“ des GMA-FA 5.2.2, Dortmund, 2000, S. 177-191
- [LCN90] Levesque, H.J.; Cohen, P.R.; Nunes, J.H.T.: On Acting Together. In Proceedings of AAAI 90, Boston,1990, S. 94-99
- [Lin04] Lin, R.J.: Bilateral Multi-Issue Contract Negotiation for Task Redistribution Using a Mediation Service. In: Agent Mediated Electronic Commerce, Lecture Notes in Computer Science, 3937, 2001
- [Lipp06] Lippe, W.M.: Soft-Computing mit Neuronalen Netzen, Fuzzy-Logic und Evolutionären Algorithmen, Springer Verlag, 2006
- [Lobo97] Lobo, J.: Adding Knowledge to the Action Description Language A. In: Proceedings of the AAAI National Conference on Artificial Intelligence, MIT Press, Cambridge, 1997, S. 454-459
- [LS88] Lauritzen, S.L.; Spiegelhalter, D.J.; Local computation with probabilities on graphical structures and their application to expert systems. In: Journal of the Royal Statistical Society, B 50(2), 1988, S, 157-224
- [LT06] Little, I.; Thiebaux, S.: Concurrent Probabilistic Planning in the Graphplan Framework. In: Proceedings of the International Conference on Automated Planning & Scheduling (ICAPS), 2006, S. 263-273
- [Math02a] Mathworks: Optimization Toolbox User’s Guide, 2002
- [Math02b] Mathworks: System Identification Toolbox User’s Guide, 2002
- [Mite97] Mitchell, T.: Machine Learning, McGraw Hill, New York 1997.

- [Möll92] Möller, D.: Modellbildung, Simulation und Identifikation dynamischer Systeme, Springer Verlag, Berlin, 1992
- [NAI+03] Nau, D.; Au, T.-C.; Ilghamim, O.; Kuter, U.; Murdock, J.W.; Wu, D.; Yaman, F.: SHOP2: An HTN Planning System. In: Journal of Artificial Intelligence Research, 20, 2003, S. 379-404
- [Nea04] Neapolitan, R. E.: Learning Bayesian Networks. Pearson Education, Inc., New Jersey, 2004.
- [OG90] Olawsky, D.; Gini, M.: Deferred Planning and Sensor Use. In: Innovative Approaches to Planning, Scheduling, and Control, Morgan Kaufmann Publisher, San Diego, 1990, S. 166–174
- [OV01] Ogston, E., Vassiliadis, S.: Matchmaking among minimal agents without facilitator. In: Agents '01: Proceedings of the Fifth International Conference on Autonomous Agents, 2001, ACM Press, New York, S. 608-615
- [OV02] Ogston, E., Vassiliadis, S.: Unstructured Agent Matchmaking: experiments in timing and fuzzy matching. In: Proceedings of the 2002 ACM Symposium on Applied Computing, ACM Press, New York, 2002, S. 300-305
- [PB93] Pahl, G., Beitz, W.: Konstruktionslehre. Methoden und Anwendung. 3. Auflage, Springer Verlag, Berlin,, 1993
- [Pear88] Pearl, Judea: PROBABILISTIC REASONING IN INTELLIGENT SYSTEMS: Networks of Plausible Inference. Morgan Kaufmann Publishers, San Mateo, 1988.
- [Pearl84] Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading, 1984
- [Pedn86] Pednault, E.: Formulation Multiagent Dynamic-World Problems in the Classical Planning Framework. In: Reasoning about Actions and Plans: Proceedings of 1986 Workshop, Morgan Kaufmann, San Mateo, 1986
- [Pfeif93] Pfeiffer, B.-M.: Identifikation von Fuzzy-Regeln aus Lerndaten. In: 3. Workshop des GMA-FA 5.2.2, Dortmund, 1993, S. 238-250
- [PJ96] Parsons, S.D., Jennings, N.R.: Negotiation Through Argumentation – A Preliminary Report. In: Proceedings of the Second International Conference Multi-Agent Systems (ICMAS'96), 1996, S. 67-274
- [PJK+02a] Pfeiffer, B.M.; Jäkel, J.; Kroll, A.; Kuhn, C.; Kuntze, H.B.; Lehmann, U.; Slawinski, T.; Tews, V.: Erfolgreiche Anwendungen von Fuzzy Logik und Fuzzy Control (Teil 1). In: at Automatisierungstechnik, 50, 2002, S.461-471
- [PJK+02b] Pfeiffer, B.M.; Jäkel, J.; Kroll, A.; Kuhn, C.; Kuntze, H.B.; Lehmann, U.; Slawinski, T.; Tews, V.: Erfolgreiche Anwendungen von Fuzzy Logik und Fuzzy Control (Teil 2). In: at Automatisierungstechnik, 50, 2002, S. 511-521
- [PKSA06] Pitt, J.; Kamara, L.; Sergot, M.; Artikis, A.: Voting in Multi-Agent Systems. In: The Computer Journal, 49, 2006, S.156-170
- [PL96] Nagendra Prasad, M. V.; Lesser, V. R.: Learning Situation-Specific Coordination in Generalized Partial Global Planning. In: 1996 AAAI Spring Symposium on Adaptation, Co-evolution and Learning in Multi-agent Systems, AAAI Press, Menlo Park, 1996
- [PS92] Peot, M.A., Smith, D.E: Conditional Nonlinear Planning, In: Proceedings of the First International Conference on Artificial Intelligence Planning Systems, 1992
- [PW92] Penberthy, J. S. and Weld, D., UCPOP: A Sound, Complete, Partial-Order Planner for ADL. In: Third International Conference on Knowledge Representation and Reasoning (KR-92), 1992.
- [Rau03] Rau, M.: Nicht-Lineare modellbasierte prädiktive Regelung auf Basis lernfähiger Zustandsraummodelle, Dissertation, TU München, 2003

- [RN03] Russell, S. Norvig, Peter: Artificial Intelligence - A Modern Approach, 2. Edition, Prentice Hall, Upper Saddle River, 2003.
- [Rode91] Rodenacker, W.G.: Methodisches Konstruieren. Konstruktionsbücher Bd. 27, 4. Auflage., Springer Verlag, Berlin, 1991
- [Roth00] Roth, K.H.: Konstruieren mit Konstruktionskatalogen, Band 1 Konstruktionslehre, 3. Auflage, Springer Verlag, Berlin, 2000
- [RZ94] Rosenschein, J.S., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers, MIT Press, Cambridge, 1994
- [SB98] Sutton, R. S.; Barto, A. G: Reinforcement Learning: An Introduction. MIT Press, Cambridge, 1998.
- [SC96] Smith, I.A.; Cohen, P.R.: Toward a Semantics for an Agent Communication Language Based on Speech Acts. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eight Innovative Applications of Artificial Intelligence Conference, AAAI Press, Menlo Park, 1996, S. 24-31.
- [Schw01] Schwarz, J.: Grundlagen der Statistik, Band 2: Wahrscheinlichkeitsrechnung und induktive Statistik, 7. Aufl., NWB-Studienbücher Wirtschaftswissenschaften, Herne, 2001
- [SCN99] Schädel, H.M., Ctitis, C., Nikolai, D.: Fuzzy Adaption von PI-Reglern im geschlossenen Regelkreis ohne Prozesskenntnis. In: 9. Workshop „Fuzzy Control“ des GMA-FA 5.2.2, Dortmund, 1999, S. 270-283
- [SDW97] Sycara, K., Decker, K., Williamson, M.: Middle Agents for the Internet. In: Proceedings of International Joint Conferences on Artificial Intelligence 97, 1997, S. 578-583
- [Smi80] Smith, R.G.: The Contract Net Protocol. In: IEEE Transaction on Computers, C-29, 1980, S. 1104-1113
- [SO02] Sapena, O., Onaindía, E.: Domain-Independent Online Planning for STRIPS Domains. In: Lecture Notes of Artificial Intelligence, 2527, Springer-Verlag, Berlin, 2002
- [SS01] Ströbel, M., Stolze, M.: A Matchmaking Component for the Discovery of Agreement and Negotiation Spaces in Electronic Markets. In: Proceedings of the Group Decision and Negotiation Conference, 2001, S. 61-75
- [Stir03] Stirling, W.C.: Satisficing Games and Decision Making: With Applications to Engineering and Computer Science, Cambridge University Press, Cambridge, 2003.
- [SW98] Smith, D. E.; Weld, D. S.: Conformant Graphplan. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98), 1998
- [SWMV06] Steenhuisen, J.R., Witteveen, C., ter Mors, A.W., Valk, M.: Framework and Complexity Results for Coordinating Non-Cooperative Planning Agents. In: Multiagent System Technologies, 4th German Conference, MATES 2006, Springer Verlag, Berlin, 2006
- [TAB06] Tsamardinos, I.; Aliferis, F.A.; Brown, L.E.: The max-min hill-climbing Bayesian network structure learning algorithm. In: Machine Learning, 65 Springer Netherlands, 2006, S. 31-78.
- [Tam97] Tambe, M.: Agent architectures for flexible, practical teamwork. In: National Conference on Artificial Intelligence (AAAI), AAAI Press, Menlo Park 1997
- [Tho90] Thoma, J.U.: Simulation by Bondsgraphs Introduction to Graphical Method. Springer-Verlag, Berlin 1990
- [TR95] Tews, V., Reine, F.: Ein Reglerbaukasten aus konventionellen und Fuzzy-Komponenten für Echtzeitanwendungen. In: Kongreß Echtzeit 95, Karlsruhe, 1995

- [VT08] Vöcking, H.; Trächtler, A.: Self-optimization of an Active Suspension System Regarding Energy Requirements. In: International Conference on Control, Automation and Systems 2008, 2008
- [VV05] Vlahavas, I., Vrakas, D.: Intelligent Techniques for Planning, Idea Group, Publishing, Hershey, 2005
- [Warr76] Warren, D. H.D.: Generating Conditional Plans and Programs. In: Proceedings of the Summer Conference on AI and Simulation of Behavior, 1976
- [WAS98] Weld, D.S., Anderson, C.R., Smith, D.E.: Extending graphplan to handle uncertainty and sensing actions. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Application of Artificial Intelligence (IAAI-98), AAAI Press, Menlo Park, 1998, S. 897-904
- [Weld94] Weld, D.: An Introduction to Least Commitment Planning, AI Magazine, 15, 1994, S. 27-61
- [Well90] Wellmann, M.P.: Formulation of Tradeoffs in Planning Under Uncertainty. Pitman, London, 1990
- [WMW05] de Weerdt, M., ter Mors, A., Witteveen C.: Multi-agent Planning: An introduction to planning and coordination. In: Handouts of the European Agent Summer School, 2005, S. 1-32
- [XCZ06] Xing, Z.; Chen, Y.; Zhang, W.: MaxPlan: Optimal Planning by Decomposed Satisfiability and Backward Reduction. In: Proceedings of the 5th International Planning Competition, 2006, S. 53-55
- [YL04] Younes, H.L.S., Littman, M.L.: PPDDL1.0: An Extension to PDDL for Expressing Planning Domains with Probabilistic Effects, Technical Report CMU-CS-04-167, School of Computer Science, Carnegie Mellon University, Pittsburgh, 2004
- [YS03] Younes, H.; Simmons, R.: VHPOP: Versatile heuristic partial order planner, Journal of Artificial Intelligence, 20, 2003, S. 405-430
- [ZHWZ04] Zhou, P.-C.; Hong, B.-R.; Wang, Y.-H., Zhou, T.: Multi-Agent Cooperative Pursuit Based on Extended Contract Net Protocol. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics, 2004, S. 169-173
- [Zilb96] Zilberstein, S.: Using Anytime Algorithms in Intelligent Systems. In: AI Magazine, 17, 1996, S. 73-83



## A. Modellierungsmethoden im Entwurf mechatronischer Systeme

Es existiert eine Vielzahl von Modellierungsmethoden, mechatronische Systeme oder Teilaspekte mechatronischer Systeme im Entwurfprozess zu beschreiben. An dieser Stelle wird ausschließlich auf Modellierungsmethoden zur Beschreibung des Verhaltens eingegangen. Methoden, die auf Struktur oder Gestalt des Produkts abzielen, sind für diese Arbeit nicht relevant.

Eine der einfachsten Methoden zur Beschreibung des Systemverhaltens auf einem allgemeinen Niveau sind Blockdiagramme oder Blockschaltbilder [Föl94]. Ein einzelnes Teilsystem oder eine einzelne Teilfunktion wird dabei als Block mit spezifizierten Eingangs- und Ausgangsgrößen dargestellt. Die Transformation der Ein- und Ausgangsgrößen kann durch mathematische Modelle sehr konkret beschrieben werden. Wegen ihrer starren Verknüpfung und der geringen Formalisierung sind Blockdiagramme aber nicht für eine Verhaltensplanung geeignet.

Viele Modellierungsansätze stellen das Verhalten mechatronischer Systeme mithilfe endlicher Zustandsautomaten oder Erweiterungen dieses Konzepts dar. Die Grundidee dabei ist jeweils, die möglichen Zustände des Systems und erlaubte Übergänge zwischen ihnen darzustellen. Zu den Vertretern dieses Modellierungsansatzes gehören die SDL, Abstrakte Datenmodellierung [CCI89, CCI93], die Mechatronic UML [BTG04] und die SysML [HV05]. SDL wurde zur Spezifikation kommunizierender technischer Systeme entwickelt. Damit eignet es sich grundsätzlich zur Modellierung interagierender Systeme. Die Kommunikation wird über Signale zwischen Prozessen beschrieben, die ihrerseits in Form erweiterter endlicher Zustandsautomaten beschrieben werden. Auf diese Weise wird eine getrennte Modellierung der Zustände einzelner Systeme möglich. Jedoch geben die erweiterten Zustandsautomaten einen relativ starren Rahmen für das Systemverhalten vor. Auf Einflüsse, Ziele und Situationen, die zur Entwurfszeit nicht bekannt sind oder erkannt wurden, kann das System nicht mehr reagieren.

Die RealTime State Charts der Mechatronic UML dienen in erster Linie der Modellierung und Verifikation der Kommunikation und Koordination mechatronischer Systeme. Eine Erweiterung stellen hybride Statecharts dar, die jedem Zustand einen (implementierten) Regler mit kontinuierlichem Verhalten zuordnen. Gegenüber der SDL ist diese Berücksichtigung kontinuierlichen Verhaltens ein wesentlicher Vorteil der Mechatronic UML. Jedoch ist diese Modellierungsmethode zur Verifikation des sicherheitskritischen Verhaltens eines Systems konzipiert und zur algorithmischen Verarbeitung in Planungsalgorithmen kaum geeignet. Insbesondere die Darstellung kontinuierlichen Verhaltens in Form von Reglern läuft dieser Verwendung zuwider, da Regler nur ein reaktives Verhalten implementieren. Da die Verhaltensbeschreibung in der SysML genau wie in der Mechatronic UML auf der UML 2.0 Spezifikation aufsetzen, gelten auch hier die gleichen Einschränkungen. Außerdem ist die SysML nicht in der Lage, kontinuierliches Verhalten darzustellen.

Sogenannte Bondgraphen [Tho90] liefern eine Beschreibungsform der Energie- und Leistungsflüsse zwischen Komponenten (Subsysteme). Grundlage für die Spezifikation von Bondgraphen sind die allgemeinen Gleichungen für Leistung, Impuls und Verschiebung. Energie- und Leistungsflüsse zwischen Systemen werden durch gerichtete Kanten dargestellt. Mithilfe von Softwarewerkzeugen können automatisch Differenzialgleichungen abgeleitet werden, um das System zu modellieren. Da Bondgraphen keine Informations- und Stoffflüsse darstellen können, stellen sie nur einen Teilaspekt mechatronischen Verhaltens dar. Schön [Sch00] verbindet Statecharts und Bondgraphen zur Modellierung zeitkontinuierlicher und ereignisorientierter Aspekte mechatronischer Systeme. Dabei werden die Gleichungssysteme

und Ereignisverarbeitung jeweils in Zeitabschnitten (taktweise) abgearbeitet. Damit liefert Schön einen Ansatz zur Integration von kontinuierlichen Verhaltensweisen und diskreter Funktionsausführung. Allerdings gelten bzgl. der Statecharts und der aus ihnen ausgeleiteten Programme die gleichen Einschränkungen wie bei der SDL und UML.

Allen hier beschriebenen Modellierungsansätzen ist gemeinsam, dass sie dazu dienen, vom menschlichen Entwickler zur Spezifikation des Verhaltens eines zu entwickelnden Produkts genutzt zu werden. Obwohl einige der Modellierungsmethoden durchaus formale und sogar operable Modelle liefern, sind sie als Aufgabenobjekt für eine Verhaltensplanung nicht geeignet. Ihre Anwendung scheitert, da keines der erzeugten Modelle Informationen darüber enthält, wie das Verhalten des Systems im Hinblick auf ein konkretes Ziel gestaltet werden muss. Während des Entwurfs leistet diese Gestaltung der Entwickler. Für einen algorithmischen Ansatz käme beispielsweise auf Basis von endlichen Zustandsautomaten nur das Durchspielen aller möglichen Zustandssequenzen und die anschließende Auswahl der besten Sequenz in Frage. Dieses Vorgehen wäre allerdings nicht sonderlich effizient.

## B. Planungsverfahren der Künstlichen Intelligenz

Zur Lösung des Planungsproblems stehen mehrere unterschiedliche Typen von Lösungsverfahren zur Verfügung. Eine ausführliche Übersicht über die Lösungsverfahren kann beispielsweise in [RN03] nachgelesen werden. An dieser Stelle soll nur ein kurzer Überblick über die unterschiedlichen Lösungsansätze gegeben werden, die anwendungsspezifische Vor- und Nachteile besitzen.

Ein intuitiver Ansatz ist die Lösung des Planungsproblems mithilfe eines Suchverfahrens [RN03]. Dabei wird das Planungsproblem als gerichteter Graph aufgefasst, in dem die Knoten Zustände repräsentieren, die durch Aktionen miteinander verbunden sind. Eine Aktion ist eine ausgehende Kante eines Zustands, wenn dieser ihre Vorbedingungen erfüllt. Der sich ergebende Nachfolgerknoten lässt sich anhand der Effekte der Aktionen (unter den Annahmen der klassischen Planung) genau bestimmen. Die Zustandsraumsuche liefert tatsächlich eine Sequenz von Aktionen, der die Ausführungsreihenfolge der Aktionen eindeutig zu entnehmen ist. Man spricht daher von total geordneten Plänen. Im praktischen Einsatz kommt der Verwendung von Heuristiken entscheidende Bedeutung zu, um den Suchprozess zu beschleunigen (z.B. durch den Algorithmus A\* [Pearl84]). Dabei ist es in der Regel besser möglich, für konkrete Problemstellungen heuristische Schätzfunktionen zu formulieren. Ein Beispiel für einen heuristischen Suchplaner ist FastForward [HN01].

Planbasiertes Planen oder partiell ordnendes Planen liefert im Gegensatz zur Zustandsraumsuche keine vollständig geordneten Pläne [Weld94]. Das bedeutet insbesondere, dass das Paradigma der Problemzerlegung genutzt werden kann. Der Verzicht auf eine vollständige Ordnung des Plans führt zu einer größeren Flexibilität des Planungsalgorithmus, durch die der Lösungsraum schneller einzuschränken ist. Der Planungsalgorithmus kann zunächst *offensichtliche* oder *wichtige* Aktionen einplanen, anstatt der chronologischen Reihenfolge folgen zu müssen. Im Anschluss an den Planungsprozess muss eine vollständig geordnete Variante des Plans (Linearisierung) erzeugt werden, um den Plan ausführen zu können. Es existieren diverse Umsetzungen von partiell ordnenden Planern (siehe z. B. [PW92], [YS03]).

Blum und Furst stellten mit GRAPHPLAN [BF97] ein System vor, das wesentlich schneller arbeitete als die damaligen partiell ordnenden Planer. Es gehört zu den graphbasierten Planern. Es handelt sich um eine Klasse von Planungsalgorithmen, die für Planungsaufgaben genutzt werden können, die in reiner Aussagenlogik formuliert werden.

Das System verwendete dabei einen Planungsgraphen. Ein Planungsgraph besteht aus zwei Knotenarten: Propositionsknoten und Aktionsknoten, wobei Propositionsknoten nur mit Aktionsknoten verbunden sind und umgekehrt. Der Graph wird in Ebenen unterteilt, die jeweils einen Zeitschritt im Plan (also einen Schritt, in dem eine Aktion ausgeführt wird) enthalten. Die erste Ebene enthält dabei die Literale des Initialzustands, die Aktionen, die darauf ausgeführt werden können, und die Literale, die durch die Ausführung dieser Aktionen erzeugt werden können. Damit enthält der Plangraph *in etwa* alle Literale, die in einem Zeitschritt wahr sein könnten, und *in etwa* alle Aktionen, die in einem Zeitschritt ausgeführt werden könnten. Die Zuordnung der Aktionen und Literale ist aber nur „*in etwa*“ korrekt, da der Planungsgraph zunächst nur einen Teil der Konflikte zwischen Aktionen berücksichtigt. Ein Algorithmus zur Erzeugung eines Plans aus einem Plangraphen iteriert über zwei Lösungsschritte. Im ersten Schritt, der Expansion des Graphen, werden die Aktionen der aktuellen Ebene und die Literale der nächsten Ebene eingefügt. Die folgende Lösungsextraktion überprüft, ob alle Literale des Zielzustands in der aktuellen Ebene enthalten sind und ob zwischen den enthaltenen Literalen Mutex-Relationen bestehen. Falls alle Literale vorhanden sind und keine Mutex-Relationen bestehen, enthält der Plangraph möglicherweise einen gültigen Plan. Dies wird überprüft, indem der Planungsgraph rückwärts nach einer Sequenz von Aktionen durchsucht wird, wobei die in den Mutex-Relationen enthaltene Information genutzt wird.

## C. XML-Format der Bayes-Netzwerke

Im Rahmen der Validierung des Agentensystems zur Vorhersage probabilistischer Fahrzeiten wurde zur Initialisierung der Agenten eine XML-Beschreibung der Bayes-Netzwerke verwendet. Das folgende Listing zeigt die DTD des XBIF-Formats, das zur Beschreibung der Expertennetzwerke eingesetzt wurde.

```
<! -- DTD of eXtended Bayesian Network Interchange Format v0.1 ( XBIF ) -->
<! ELEMENT XBIF ( NETWORK ) * >
  <! ATTLIST BIF VERSION CDATA # REQUIRED >
<! ELEMENT NETWORK ( ( PROPERTY | VARIABLE | DEFINITION ) * ) >
<! ATTLIST NETWORK NAME ID # REQUIRED TEMPLATE CDATA # IMPLIED
  TOPOLOGY CDATA # IMPLIED >
<! ELEMENT VARIABLE ( ( ( OUTCOME | INTERVAL ) | PROPERTY ) * ) >
  <! ATTLIST VARIABLE NAME ID # REQUIRED
  TYPE ( discrete | continuous | continuous_time ) " discrete ">
<! ELEMENT OUTCOME ( # PCDATA ) >
<! ELEMENT INTERVAL ( # PCDATA ) >
<! ELEMENT DEFINITION ( GIVEN | ( TABLE | FUNCTION ) | PROPERTY ) * >
  <! ATTLIST DEFINITION FOR CDATA # REQUIRED >
<! ELEMENT FOR ( # PCDATA ) >
<! ELEMENT GIVEN ( # PCDATA ) >
<! ATTLIST GIVEN MEAN_VALUES CDATA # IMPLIED STD_DEVS CDATA # IMPLIED >
<! ELEMENT TABLE ( # PCDATA ) >
<! ELEMENT FUNCTION ( PARAM * ) >
  <! ATTLIST FUNCTION TYPE ( gaussian ) " gaussian ">
<! ELEMENT PARAM ( # PCDATA ) >
  <! ATTLIST PARAM NAME CDATA # REQUIRED >
<! ELEMENT PROPERTY ( # PCDATA ) >
```

**Listing C-1: DTD des eXtended Bayesian Network Interchange Format**

Die in Tabelle 6.1 dargestellte CPF (Bedingte Wahrscheinlichkeitsverteilung) einer Variable C mit zwei Elternknoten A und B und jeweils drei Ausprägungen pro Variable würde als Konkatenation der Zeilen der Tabelle im XML-Format angegeben als:

```
<TABLE>0.30 0.20 0.50 0.60 0.50 0.80 0.80 0.70 0.90 0.30 0.40 0.30 0.30 0.40 0.15 0.15
0.25 0.10 0.40 0.40 0.20 0.10 0.10 0.05 0.05 0.05 0.00</TABLE>
```

	a <sub>1</sub>			a <sub>2</sub>			a <sub>3</sub>		
	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>
c <sub>1</sub>	0,3	0,2	0,5	0,6	0,5	0,8	0,8	0,7	0,9
c <sub>1</sub>	0,3	0,4	0,3	0,3	0,4	0,15	0,15	0,25	0,1
c <sub>3</sub>	0,4	0,4	0,2	0,1	0,1	0,05	0,05	0,05	0,0

**Tabelle C-1: Wahrscheinlichkeitsverteilung einer Variablen C mit Elternknoten A und B**

Das folgende Listing zeigt ein Beispiel für eine XML-Datei, die den Aufbau und die Parameter eines Expertennetzwerks beschreibt.

```
<? xml version =" 1.0 " encoding ="UTF -8"?>
<! DOCTYPE XBIF SYSTEM " xbif . dtd ">
<! -- Bayesian network in XBIF v0 .1 -->
<XBIF VERSION =" 0.1 ">
<NETWORK NAME =" trackexpert0 " TEMPLATE =" expert1 " TOPOLOGY ="B">

  <DEFINITION FOR =" arrival ">
    <FUNCTION TYPE =" gaussian ">
      <PARAM NAME ="mu">8.0 </ PARAM >
      <PARAM NAME =" sigma ">1.2 </ PARAM >
    </ FUNCTION >
  </ DEFINITION >

  <DEFINITION FOR =" temperature ">
    <FUNCTION TYPE =" gaussian ">
      <PARAM NAME ="mu">12.0 </ PARAM >
      <PARAM NAME =" sigma ">1.2 </ PARAM >
    </ FUNCTION >
  </ DEFINITION >

  <DEFINITION FOR =" rush_hour ">
    <GIVEN >arrival </ GIVEN >
    <TABLE >0.00 0.02 0.00 0.00 0.00 0.02 0.00 0.00 0.80 0.30 26 0.40
              0.30 0.80 0.00 1.00  0.18 0.70 0.60 0.70 0.18 27 1.00
    </ TABLE >
  </ DEFINITION >

  <DEFINITION FOR =" ride_duration ">
    <GIVEN MEAN_VALUES =" 37.5 21.0 18.75 " STD_DEVS =" 3.0 1.7 0.9 ">
      fog
    </ GIVEN >
    <GIVEN MEAN_VALUES =" 37.5 23.0 18.75 "34 STD_DEVS =" 2.6 3.6 0.9 ">
      ice
    </ GIVEN >
  <GIVEN MEAN_VALUES =" 33.3 25.0 18.75 "36 STD_DEVS =" 6.1 2.5 0.9 ">
    rush_hour
  </ GIVEN >
  <FUNCTION TYPE =" gaussian ">
    <PARAM NAME ="mu">20.0 </ PARAM >
    <PARAM NAME =" sigma ">2.0 </ PARAM >
  </ FUNCTION >
</ DEFINITION >
</ NETWORK >
</ XBIF >
```

**Listing C-2: Beispiel einer XBIF Beschreibung eines Experten-Netzwerks**

## D. Versuchsaufbau Belief-Dijkstra

Es wäre zu aufwendig gewesen, für alle Streckenabschnitte der Transportnetzwerke von Hand ein Bayes-Netz mit trackspezifischen a-priori-Verteilungen der Fahrdauer zu erstellen. Daher wurden zwei Bayes-Netze mit unterschiedlicher Topologie (Typ A und B, dargestellt in Abbildung 7-4) erstellt, anhand derer durch Anpassung der a-priori-Verteilungen der Einflussfaktoren 12 Schablonen erstellt wurden, die verschiedenen Szenarien entsprechen. Die Netze vom Typ B enthalten zwei Knoten: *Humidity* und *Temperature*. Für sie sollen vor jeder Anfrage weiche Evidenzen eines Wetter-Agenten gesetzt werden. Da ein solcher Wetter-Agent nicht zur Verfügung steht, wurden für die Tests für alle Tracks eines Transportnetzwerkes die gleichen (zuvor festgelegten) weichen Evidenzen für die Knoten *Humidity* und *Temperature* angenommen. In der Testvorbereitung wurde jedem Track zufällig eines der Szenarien zugeordnet und für jede Ausprägung der Elternknoten von Ride Duration die a-priori-Verteilungen der Fahrdauer entsprechend der Länge des jeweiligen Tracks generiert.

Schablone	Beschreibung	Typ
1	starker Berufsverkehr	B
2	moderater Berufsverkehr	B
3	geringer Berufsverkehr	B
4	kein Berufsverkehr	B
5	kein Berufsverkehr, Großveranstaltung (groß) von 14-18 h	A
6	geringer Berufsverkehr, Großveranstaltung (groß) von 9-14 h	A
7	moderater Berufsverkehr, Großveranstaltung (klein) von 9-14 h	A
8	starker Berufsverkehr, keine Großveranstaltung	A
9	starker Berufsverkehr, keine Großveranstaltung	A
10	geringer Berufsverkehr, Großveranstaltung (klein) von 14-18 h, verringerte Eisgefahr	A
11	moderater Berufsverkehr, keine Großveranstaltung, verringerte Eisgefahr	A
12	moderater Berufsverkehr, keine Großveranstaltung, keine Eisgefahr	A

Tabelle D-1: Schablonen für die Bayes-Netzwerke der Track-Agenten

Der Knoten *Humidity* des Netztyps B repräsentiert die relative Luftfeuchtigkeit und wurde entgegen seiner kontinuierlichen Natur als diskrete Variable modelliert. Sein Wertebereich ist auf das Intervall  $[0, 100]$  beschränkt, eine Normalverteilung weist jedoch grundsätzlich einen unendlichen Wertebereich auf. Für den praktischen Einsatz der Fahrdauerprognose soll ein spezieller Wetter-Agent vor einer Prognose weiche Evidenzen für die Knoten *Humidity* und *Temperature* des Netztyps B liefern. Da ein solcher Wetter-Agent bislang nicht zur Verfügung steht, wurden in den Tests für *Humidity* die in Tabelle D-2 angegebenen weichen Evidenzen manuell gesetzt. Für den kontinuierlichen Knoten *Temperature* wurde eine  $N(2^{\circ}\text{C}, 3^{\circ}\text{C})$  Verteilung als evident angenommen.

Tabelle D-3 gibt einen Überblick über die verwendeten Zufallsvariablen und ihre Ausprägungen, bzw. im Falle kontinuierlicher Variablen die verwendeten Intervalle. Gleichnamige Variablen haben in beiden Netzen die gleichen Ausprägungen/Intervalle.

Ausprägung	Wahrscheinlichkeit
0-25%	0,00
25-50%	0,05
50-75%	0,70
75-90%	0,24
90-95%	0,01
95-100%	0,00

Tabelle D-2: Weiche Evidenzen für den Knoten Humidity

Knoten	Ausprägungen	Typ	in Netztyp
Arrival	0:00-7:00 h 7:00-9:00 h 9:00-11:45 h 11:45-14:00 h 14:00-16:00 h 16:00-18:00 h 18:00-0:00 h	continuous_time	A,B
Event	big small none	discrete	A
Fog	thick some none	discrete	B
Humidity	0-25% 25-50% 50-75% 75-90% 90-95% 95-100%	discrete	B
Ice	area-wide partly none	discrete	A,B
Month	January,...,December	discrete	A
Ride Duration	-	continuous	A,B
Rush Hour	heavy moderate none	discrete	A,B
Temperature	-infinity - -5 -5 - -1 -1 - 0 0 - 1 1 - 5 5 - 10 10 - 15 15 - 20 20 - 25 25 - infinity	continuous	B

Tabelle D-3: Zufallsvariablen der beiden verwendeten Netztypen

Die beiden folgenden Tabellen zeigen die zur Generierung der Verteilungen verwendeten Durchschnittsgeschwindigkeiten und Anteile. Für den Test der Mechanismen zur Eindäm-

mung des Verspätungsrisikos wurden bei der Generierung der Standardabweichungen andere Anteile verwendet als für die restlichen Tests: Bei den Tests zur Abweichung von der simulierten Fahrtdauer und zur Fahrtdauerersparnis bei unterschiedlichen Pfaden wurden die in der Spalte „Anteil Test 1“ von Tabelle D-5 angegebenen Anteile benutzt. Für den Test der Mechanismen zur Eindämmung des Verspätungsrisikos wurde dagegen zufällig einer der beiden Anteile aus den Spalten „Anteil I Test 2“ und „Anteil II Test 2“ verwendet.

Variable	Ausprägung	Durchschnittsgeschwindigkeit
Event	big	30,0
	small	60,0
	none	80,0
Fog	thick	40,0
	some	70,0
	none	80,0
Ice	area-wide	40,0
	partly	65,0
	none	80,0
Rush Hour	heavy	45,0
	moderate	60,0
	none	80,0

**Tabelle D-4: Mittlere Geschwindigkeiten zur Generierung der Erwartungswerte**

Variable	Ausprägung	Anteil Test 1	Anteil I Test 2	Anteil II Test 2
Event	big	0,05	0,01	0,15
	small	0,1	0,04	0,25
	none	0,05	0,01	0,12
Fog	thick	0,08	0,02	0,16
	some	0,08	0,02	0,16
	none	0,05	0,01	0,12
Ice	area-wide	0,07	0,02	0,20
	partly	0,16	0,05	0,32
	none	0,05	0,01	0,12
Rush Hour	heavy	0,185	0,07	0,40
	moderate	0,1	0,04	0,25
	none	0,05	0,01	0,12

**Tabelle D-5: Anteile zur Bestimmung der Standardabweichungen**