

Operative strukturfokussierte Gestaltung kognitiver Systeme am Beispiel hochautomatisierter Fahrzeuge

Von der Fakultät für Informatik der Carl von Ossietzky Universität Oldenburg zur Erlangung des Grades und Titels eines

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

angenommene Dissertation

von Herrn Johann Kelsch

Geboren am 14.01.1978 in Nowokusnezsk / Russland



Gutachter

Prof. Dr. Frank Köster

Weitere Gutachter

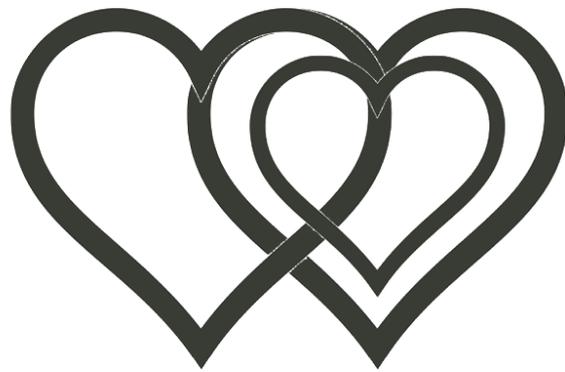
Prof. Dr. Martin Baumann

Tag der Disputation: 06.06.2019

Erklärung

Hiermit versichere ich, Johann Kelsch, dass die von mir vorgelegte Dissertation mit dem Titel „Operative strukturfokussierte Gestaltung kognitiver Systeme am Beispiel hochautomatisierter Fahrzeuge“ von mir selbständig verfasst wurde, und ich dafür keine weiteren außer in der Arbeit genannten Hilfsmittel verwendet habe. Des Weiteren versichere ich, dass ich alle von anderen Autoren wörtlich übernommenen Stellen, sowie sich an die Gedankengänge anderer Autoren eng anliegende Ausführungen meiner Arbeit besonders gekennzeichnet und die Quellen zitiert habe.

Braunschweig, den 2. November 2018



*Für meine lieben
Elena und Fabian.
Danke!*

Inhaltsverzeichnis

Kurzfassung	ix
Abstract	xi
1 Einleitung.....	1
1.1 Motivation: Komplexität der kognitiven Systeme.....	1
1.2 Problemstellung: Komplexität wirkt gegen die Gestaltung.....	3
1.3 Ziel: Reduktion der Komplexität.....	4
1.4 Lösungsansatz: Gestaltungsrahmen reduziert Komplexität	5
1.5 Forschungsfragen.....	6
1.6 Leistungen und Mehrwert der Arbeit	7
1.7 Struktur und Ergebnisse der Arbeit	8
2 Systemgestaltungsrahmen	9
2.1 Einordnung relevanter Systemgestaltungsrahmen	9
2.2 Techniknahe Systemgestaltung.....	11
2.2.1 Systems Engineering	12
2.2.2 Systemmodellierung	13
2.2.3 Objektorientierte Systeme.....	14
2.2.4 Multiagentensysteme.....	15
2.2.5 Techniknahe Gestaltungsleitlinien	17
2.2.6 Techniknahe Gestaltungsmuster	19
2.2.7 Werkzeugketten	19
2.3 Menschnahe Systemgestaltung.....	20
2.3.1 Usability Engineering	20
2.3.2 Nutzerzentrierte Methoden.....	22
2.3.3 Menschzentrierte Methoden.....	24
2.3.4 Kognitive Architekturen	25
2.3.5 Menschnahe Gestaltungsleitlinien	25

2.3.6	Menschnahe Gestaltungsmuster	26
2.4	Konsistente Interaktionsgestaltung	27
2.5	Zusammenfassung	29
3	Verstehen und Beschreiben kognitiver Systeme	31
3.1	Kognitive Multiagentensysteme	32
3.1.1	Verknüpfung zwischen Menschen, Maschinen und MAS.....	32
3.1.2	Reduktion und Homogenisierung der Definitionsvielfalt	33
3.1.3	Kognitive Systeme als Multiagentensysteme	34
3.2	Kommunikationsstrukturen kognitiver Systeme.....	36
3.3	Strukturen und Verhalten der Agenten	38
3.4	Strukturen und Verhalten kognitiver Systeme	41
3.5	Semi-Formale Repräsentation Kognitiver Systeme	44
3.6	Verwandte Konzepte und Diskussion.....	49
3.7	Zusammenfassung	50
4	Gestalten kognitiver Systeme	51
4.1	Grundlegendes zum Gestalten kognitiver Systeme	51
4.1.1	Beeinflussung des Agentenverhaltens	52
4.1.2	Zeithorizont der Beeinflussung des Agentenverhaltens	54
4.1.3	Zusammenfassung.....	56
4.2	Aktionstension	56
4.2.1	Problemstellung: Zeitlich fehlabgestimmte Agentenhandlungen	56
4.2.2	Lösungsansatz: Steuerstrukturen in kognitiven Systemen	57
4.2.3	Semi-Formale Repräsentation der Aktionstensionen	61
4.2.4	Verwandte Konzepte und Diskussion	62
4.2.5	Aktionstensionen in hochautomatisierten Fahrzeugen	64
4.2.6	Zusammenfassung.....	65
4.3	Arbitrierung	66
4.3.1	Problemstellung: Qualitativ fehlabgestimmte Agentenhandlungen	66

4.3.2	Lösungsansatz: Allgemeine Methode der Handlungsabstimmung	67
4.3.3	Zu Problemen und Lösungen in kognitiven Systemen	70
4.3.4	Verwandte Konzepte und Diskussion	73
4.3.5	Semi-Formale Repräsentation der Arbitrierung	74
4.3.6	Arbitrierung in hochautomatisierten Fahrzeugen	75
4.3.7	Zusammenfassung.....	79
4.4	Gestaltungskompatible Architektur kognitiver Systeme	79
4.4.1	Anforderungen an Architektur und Gestaltungsschnittstelle	80
4.4.2	Zur Architektur	80
4.4.3	Zur Gestaltungsschnittstelle	82
4.4.4	Zusammenfassung.....	85
4.5	Zusammenfassung der Gestaltungskonzepte	86
5	Prozess, Regeln und Werkzeuge der Gestaltung.....	87
5.1	Konfliktanalysewerkzeug	87
5.2	Lösungsexploration	88
5.3	Semantisches Mapping.....	89
5.4	Arbitrierungsmatrix	91
5.5	Syntaktisches Storyboard	92
5.6	Arbitrierungswerkzeug	93
5.7	Interaktionskontroller	95
5.8	Zusammenfassung	97
6	Erläuterung und Evaluation durch Anwendung.....	99
6.1	Beispielsystem 1: Annähern, Bremsen und Ausweichen.....	99
6.1.1	Anforderungen.....	99
6.1.2	Szenario und Systemdekomposition	99
6.1.3	Konfliktanalyse und Semantisches Mapping	100
6.1.4	Lösungsexploration.....	100
6.1.5	Operationalisierung der Aktionstension	103

6.1.6	Arbitrierungsstrategie	105
6.1.7	Umsetzung als Prototyp	106
6.1.8	Usability Studie	107
6.1.9	Zusammenfassung	108
6.2	Beispielsystem 2: Kooperation bei Spurwechseln	108
6.2.1	Anforderungen	108
6.2.2	Szenario und Systemdekomposition	109
6.2.3	Konfliktanalyse und Semantisches Mapping	109
6.2.4	Lösungsexploration	110
6.2.5	Operationalisierung der Aktionstension	112
6.2.6	Arbitrierungsstrategie	114
6.2.7	Umsetzung als Prototyp	116
6.2.8	Usability Studie	117
6.2.9	Zusammenfassung	118
6.3	Zusammenfassung der Evaluation und Erläuterung	118
7	Zusammenfassung und Ausblick	121
8	Anhang	123
8.1	Anhang A: Wissensrepräsentation mit Concept-Maps	123
8.2	Anhang B: Beschreibungsmittel für kognitive Systeme	125
8.3	Anhang C: Zur grafischen Nomenklatur in Abbildungen	127
8.4	Anhang D: Evaluation des Syntaktischen Storyboards	129
	Abbildungsverzeichnis	131
	Tabellenverzeichnis	135
	Abkürzungsverzeichnis	137
	Literaturverzeichnis	139

Kurzfassung

Die Gestaltung eines modernen Fahrerassistenzsystems konfrontiert den Entwickler im Allgemeinen mit der Komplexität der technischen Entwicklung, dem Fahrerverhalten, dem Verkehr sowie der Heterogenität auf dem Markt bereits verfügbarer Assistenzsysteme. Um mit einer solchen Komplexität effektiv umgehen und stabile, gut nutzbare Systeme entwickeln zu können, benötigt der operative Gestaltungsprozess einen Rahmen. Solcher kann einen Zusatznutzen und Ordnung in eine Gestaltung bringen und somit den Systementwicklungsprozess bezogen auf den Zeit- und Kosteneinsatz beschleunigen und optimieren.

In der vorliegenden Arbeit ist ein solcher Gestaltungsrahmen präsentiert und evaluiert. Er basiert auf einem holistischen, deskriptiven, skalierbaren und formalisierten Beschreibungsmittel für kognitive Systeme, denen auch die Fahrerassistenzsysteme zuzuordnen sind. Dieser Rahmen besteht aus Gestaltungskonzepten, einem Regelwerk und einer Werkzeugkette.

Zunächst ist der Inhalt der Arbeit relativ zu bereits existierenden Systemgestaltungsrahmen eingeordnet. Danach sind relevante Grundlagen der Systemgestaltung, der Informationstechnik und der Psychologie zum Konzept der Manipulations-Potenzial-Systeme (MPS) transformiert. Der Fokus ist dabei auf die Strukturen bzw. Organisation kognitiver Systeme gelegt. Innerhalb des MPS-Konzeptes sind zwei weitere Konzepte erarbeitet: Aktionstension und Arbitrierung. Es ist argumentiert, dass mithilfe dieser Konzepte und des MPS kognitive Systeme holistisch verstanden, beschrieben und operativ gestaltet werden können.

Aus allen Konzepten sind eine gestaltungskompatible Architektur der Fahrerassistenzsysteme und die dazugehörige Gestaltungsschnittstelle abgeleitet. Es ist gezeigt, wie die vorgeschlagene Architektur sowie die Gestaltungsschnittstelle als Software implementiert werden können. Über die Konzepte hinaus sind für die operative Anwendung des Gestaltungsrahmens dessen Regelwerk sowie die dafür notwendigen Werkzeuge eingeführt.

Der Gestaltungsrahmen ist anhand von zwei umfassenden Beispielen für innovative Fahrerassistenzsysteme erläutert. Das erste Beispiel behandelt ein System mit der zwischen einem Fahrer und einer Automation geteilten Quer- und Längsführungskontrolle eines hochautomatisierten Fahrzeuges. Bei dem zweiten Beispiel handelt es sich um ein System zur Förderung der Kooperation zwischen zwei Verkehrsteilnehmern bei Fahrstreifenwechseln. Die Beispiele sind so gewählt, dass die Räume der möglichen Wechselwirkungen, wie Konkurrenz und Kooperation, sowie der möglichen Handlungen, wie Fahrzeugkontrolle und Koordination, exemplarisch abgedeckt sind. Ergebnisse der Nutzerbewertungen dieser beiden Beispielsysteme sind zur Evaluation des Gestaltungsrahmens verwendet.

Abstract

The design of a modern driver assistance system generally challenges the developer with the complexity of the technical development, the driver behaviour, the traffic as well as the heterogeneity of assistance system already available on the market. In order to deal effectively with such complexity and to develop stable, well-usable systems, a framework is essential for the operative design process. Such a framework can bring additional benefits and order into a design and consequently accelerate and optimize the development in terms of time and cost.

This thesis presents and evaluates such a design framework. It is based on a holistic, descriptive, scalable and formalized means for describing cognitive systems, to which the driver assistance and automation systems are also assigned. This framework consists of design concepts, a set of rules and a tool chain.

First, the content of this work is set relative to already existing system design frameworks. After that, the relevant basics of system design, information technology and psychology are transformed into the concept of Manipulation Potential Systems (MPS). The focus is on the structures and organization of cognitive systems. Within the MPS concept, two further concepts are developed: action tension and arbitration. It is argued that with these concepts and the MPS, cognitive systems can be holistically understood, described, and operationalized.

A design-compatible architecture of driver assistance systems and the corresponding design interface are derived from these concepts. It is shown how the proposed architecture as well as the design interface can be implemented as software. Beyond the concepts for the operative application of the framework, the design rules and necessary tools are introduced.

The design framework is explained using two comprehensive examples of innovative driver assistance systems. The first example deals with a system for lateral and longitudinal guidance control of a highly automated vehicle shared between a driver and an automation. The second example is a system for supporting cooperation between two road users when changing lanes. The examples are chosen to exemplarily cover the spaces of possible interferences in cognitive systems, such as competition and cooperation, as well as possible actions, such as vehicle control and coordination. Results of the usability studies for these two example systems are used to evaluate the design framework.

1 Einleitung

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.

(The Zen of Python, by Tim Peters)

1.1 Motivation: Komplexität der kognitiven Systeme

Unter dem immer weiter steigenden Druck auf dem „Handlungsfeld Mobilität“^[Ler11] ist es abzusehen, dass die Entwicklung von Fahrerassistenzsystemen und der automatisierten Fahrzeuge^[GaA12] in den Mittelpunkt der Forschung und der Entwicklung der Automotivdomäne gerückt werden. Doch damit wächst auch die Wichtigkeit der Frage: Können wir die damit ebenfalls steigende Komplexität der entwickelten und verkauften Systeme noch beherrschen?

Unter einem *System* versteht man eine „Menge von geordneten Elementen mit Eigenschaften, die durch Relationen verknüpft sind. Die Menge der Relationen zwischen den Elementen eines Systems ist seine *Struktur*. Unter einem *Element* versteht man einen Bestandteil eines Systems, der innerhalb dieser Gesamtheit nicht weiter zerlegt werden kann“^[SP13a]. „Der Begriff System [...] bezeichnet allgemein eine Gesamtheit von Elementen, die so aufeinander bezogen bzw. miteinander verbunden sind und in einer Weise wechselwirken, dass sie als eine aufgaben-, sinn- oder zweckgebundene Einheit angesehen werden können“^[W13g].

Eine *Kognition* kann als „Fähigkeit zur Wahrnehmung und zielgerichteten Interpretation der Lebenswelt“^[SP13c] und als „von einem verhaltenssteuernden System ausgeführte Informationsumgestaltung“^[W13b] definiert werden. *Kognitive Systeme* sind demnach solche Systeme, die kognitive Eigenschaften und integrierte Strukturen, Verhalten und Prozesse aufweisen^[Lin07]. Im Gegensatz zum meist deterministischen Verhalten technischer dynamischer Systeme, kann das Verhalten der kognitiven Systeme oft nichtdeterministisch, d.h. von außen „nicht bestimmbar“, „nicht vorhersehbar“ sein. Solche Systeme können ein willkürliches Verhalten zeigen, das sich bei einer Systemgestaltung schwer erfassen und behandeln lässt.

Des Weiteren wird in dieser Arbeit der Begriff „kognitives System“ (CS) als eine Aggregation für alle Mensch/Maschine- bzw. Multiagentensysteme^[Kör16] verwendet. Alle Arten von kognitiven Elementen innerhalb eines Systems, wie Menschen, Fahrer, intelligente Maschinen, Automation, Fahrerassistenzsysteme etc., werden als „kognitive Agenten“ (CA, CAs für Mehrzahl) bezeichnet. Aus Gründen der effektiven Informationsvermittlung wird in der gesamten Arbeit die Figur „der Systemingenieur“ mitgeführt und als fiktiver Nutzer der Ergebnisse dieser Arbeit an wichtigen Schlüsselstellen und in zentralen Argumentationsketten platziert.

Im Kontext dieser Definitionen betrachte man eine für diese Arbeit repräsentative Beispielsituation: Ein Systemingenieur muss ein neues Fahrerassistenzsystem gestalten. Dieses sollte den Fahrer bei der Geschwindigkeitshaltung und bei der Abstandshaltung zum vorausfahrenden Fahrzeug unterstützen. Ein solches Assistenzsystem existiert bereits auf dem Automobilmarkt mit der Bezeichnung „Adaptive Cruise Control“ (ACC)^[WiH09]. Unter der Zuhilfenahme einer

kurzen Analyse^[KeT13] dieses bekannten Fahrerassistenzsystems kann die wesentliche Motivation für diese Arbeit formuliert werden: Es ist die hohe Komplexität des Gesamtsystems, die von dem Systemingenieur bei der Gestaltung von Fahrer/ACC/Fahrzeug/Verkehr-Systemen und ähnlicher kognitiver Systeme gemeistert werden muss (Abbildung 1-1).

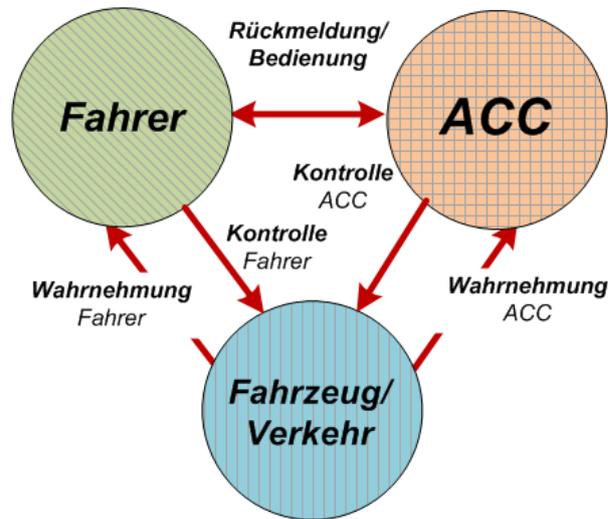


Abbildung 1-1: Komplexe Beziehung innerhalb eines (kognitiven) Fahrer/ACC/Fahrzeug/Verkehr-Systems

Moderne Fahrerassistenz- bzw. Automationssysteme können ein komplexes und für den Fahrer manchmal intransparentes Verhalten aufzuweisen. Ein ACC nimmt einen relevanten Ausschnitt der aktuellen Verkehrssituation wahr. Es entscheidet dann autonom über die Ausführung aktueller Fahrmanöver in der Längsführung, wie sich einem anderen Fahrzeug anzunähern, ihm zu folgen oder die Geschwindigkeit beizubehalten^[NaE95], und über die Rückmeldung an den Fahrer. Diese Wahrnehmung, Entscheidung und Rückmeldung können sich von denen des Fahrers unterscheiden bzw. von einem unerfahrenen Fahrer missinterpretiert werden. Des Weiteren können die bei der Entwicklung unbekanntes Umwelt- bzw. Situationseinflüsse das System im Nachhinein destabilisieren. Dies kann zu schwerwiegenden Folgen führen, wie Fehlbedienung des ACC, Unfällen, aber auch zur Unzufriedenheit und Kaufunlust der potenziellen Systemnutzer. Im Folgenden soll etwas genauer darauf eingegangen werden.

Bei der Gestaltung eines Fahrer/ACC/Fahrzeug/Verkehr-Systems, wie bei der Gestaltung eines ähnlich komplexen kognitiven Systems, muss eine Fülle an technikhnen Aufgaben gelöst werden: Welche Führungsgrößen sind für die Fahrzeugsteuerung und -regelung relevant, welche Reglerarchitektur ist geeignet, um mit Nichtlinearitäten und Störungen in der Regelstrecke umzugehen etc.? Als Ergebnis einer solchen technikhnen Gestaltung kommen eine bestimmte Struktur und ein Verhalten der *technischen Komponente* des Gesamtsystems heraus. Ein ACC bietet beispielsweise Funktionen, wie das Verzögern zur Kollisionsvermeidung nach Erscheinen des vorausfahrenden Fahrzeugs, das Abstandhalten oder das Halten der Geschwindigkeit beim Fehlen des vorausfahrenden Fahrzeugs. Technisch bedingt, kann die Verfügbarkeit dieser Funktionen nach oben und nach unten auf der Geschwindigkeitsskala eingeschränkt sein, z.B. durch eine nicht ausreichende oder nicht immer zuverlässige Sensorik.

Dazu kommt, dass im Fahrzeug mit dem eingerüsteten ACC eine *menschliche Komponente*, der Fahrer, vorhanden ist. Der Fahrer steuert das Fahrzeug neben der Automation und kann aufgrund seiner kognitiven Eigenschaften, ein sehr komplexes Verhalten zeigen. Innerhalb des insgesamt kognitiven Fahrer/ACC/Fahrzeug/Verkehr-Systems könnte der Fahrer jederzeit ab-

gelenkt, müde, überlastet oder gestresst sein. Er kann den Aktionen der Automation (zu sehr) vertrauen („overtrust“)^[MoS98] oder nachlässig sein („complacency“)^[PaS00]. Dabei kann das Gesamtsystem in Fehlerzustände geraten. Ein weiterer Grund dafür kann das „mode confusion“-Problem^[Sa95a] sein, wenn der Fahrer nicht mehr weiß, in welchem Modus sich die Automation befindet, ob sie z.B. noch an oder schon aus ist. Andere Gründe können „pilot-out-of-the-loop“^[Sa95b] oder „skill degradation“-Probleme^[PaS00] sein. Diese treten auf, wenn der Fahrer durch die Automation soweit aus der Systemkontrolle entzogen wird, dass er bei Bedarf die Fahrzeugsteuerung nicht mehr übernehmen kann. Der Fahrer kann die ihm angebotene Automation im falschen bzw. nicht vorgesehenen Kontext nutzen („misuse“, „abuse“)^[PaR97] etc.

Eine weitere Komplexitätsstufe bildet die Anzahl an möglichen Automationssystemen in einem modernen Fahrzeug. Denn der Fahrer muss nicht nur mit dem ACC zurechtkommen, sondern auch mit einer Notbremsassistentz (AEBS), einer „Lane Departure Warning“ (LDW), einer „Lane Keeping Automation“ (LKA), einer „Rear End Collision Avoidance“ (RECA) etc. Viele dieser, für sich allein genommen, guten Automationssysteme wurden oft historisch und betrieblich bedingt einzeln entwickelt. Die Mensch/Maschine-Schnittstelle kann dadurch für einige Automationssysteme und Fahrzeugmarken eine heterogene Komposition darstellen^[Blh12]. Der oft unerfahrene Fahrer kann gezwungen sein, mehrere Automationssysteme über verschiedene besondere Stellteile, wie Knöpfe, Hebel, multidimensionale Hebel mit Knöpfen darauf etc. zu bedienen. Die Rückmeldung von den Automationssystemen zu dem Fahrer hin kann ebenfalls heterogen gestaltet sein^[Blh12]. Automationssysteme können dabei verschiedene akustische Signale oder visuelle Informationen nur in Teilen, auf unterschiedlichen Displays und unter Nutzung unterschiedlicher Konzepte darstellen.

Die Komplexität des Verkehrs trägt ebenfalls zu der Gesamtkomplexität des zu gestaltenden Systems bei. Denn die verschiedenen Fahrer und ihre Automationssysteme stellen eine Menge an vorwiegend selbständig agierenden und nicht immer fahrzeugübergreifend aufeinander abgestimmten kognitiven bzw. nichtdeterministischen Elementen dar. Dies kann zu einer gefährlichen Destabilisierung des Verkehrsflusses führen, wenn z.B. viele ACC-Systeme sich gleichzeitig im Randbereich ihrer Funktionalität befinden und reihum entscheiden würden, die Kontrolle an ihre unaufmerksamen Fahrer zurückzugeben. Dies gilt gleichermaßen für technische Überlegungen zur Systemstabilität mit ACC auch bei normalen Kolonnenfahrten^[WiH09].

Die eben eingeführte Komplexität des Fahrer/ACC/Fahrzeug/Verkehr-Systems gilt für die gesamte Vielfalt kognitiver Systeme, wie andere Mensch/Maschine/Umwelt- bzw. sozio-technische Systeme. Es ist wichtig zu betonen, dass ein kognitives System mit einem oder mehreren kognitiven Elementen (Menschen, intelligente Maschinen) insgesamt ein *komplexes System* ist. Dies wird auch gestützt durch den theoretisch angenommenen emergenten Ursprung der Kognition selbst^[Mcd10]. Der Systemingenieur muss sich also im Bereich der Gestaltung solcher Systeme mit emergenten Effekten, mit hochgradig nichtlinearer Dynamik, mit sich selbst organisierenden und erhaltenden Prozessen etc. befassen^[Mar99].

1.2 Problemstellung: Komplexität wirkt gegen die Gestaltung

Die gestalterische Abstimmung von einerseits Selbständigkeit, andererseits technisch bedingten Einschränkungen des Maschinenverhaltens mit dem natürlichen und komplexen Verhalten

des Menschen und der Umwelt kann äußerst problematisch und schwierig sein. Dennoch muss bei einer Systemgestaltung dafür gesorgt werden, dass der nicht immer deterministisch agierende Mensch mit dem Verhalten der technischen Komponenten des kognitiven Systems jederzeit zurechtkommt. Denn die Bedienfehler können in kritischen Nutzungsszenarien schnell fatale Folgen haben. Das Verhalten unterschiedlicher Automationssysteme muss untereinander ebenfalls homogenisiert sein. Denn verschiedene Automationssysteme mit den heterogenen Eingabe- und Rückmeldekonzepthen können in manchen Fällen von den Nutzern nicht richtig verstanden und bedient werden. Das Verhalten und die Struktur der Umwelt müssen beim Gestalten kognitiver Systeme explizit berücksichtigt werden.

Die wesentliche Problemstellung für diese Arbeit lautet: Der Systemingenieur muss bei der operativen Gestaltung effektiv gebrauchstaugliche kognitive Systeme entwickeln und unter einem holistischen Paradigma in ein Gesamtsystem integrieren können, was bei hoher Komplexität kognitiver Systeme eine große Herausforderung darstellt.

1.3 Ziel: Reduktion der Komplexität

Bei einer Systemgestaltung, dies gilt ebenso für die Gestaltung kognitiver Systeme, geht der Systemingenieur folgendermaßen vor: Er nutzt einen bestimmten Gestaltungsrahmen, um die Systemkomplexität zu reduzieren. Gestaltungsrahmen bestehen grundsätzlich aus einer Gestaltungstheorie, einem Regelwerk und den dazugehörigen Werkzeugen. Konzepte einer Gestaltungstheorie können einschränkende und anregende Eigenschaften besitzen. Die einschränkenden Konzepte geben dem Zielsystem die äußere Form, z.B. durch physikalische Gesetzmäßigkeiten oder Vorschriften. Die anregenden Konzepte versorgen die Gestaltung mit Motivation und neuen Ideen. Mithilfe der Gestaltungstheorie des jeweiligen Rahmens, seines Regelwerkes und der Werkzeuge formt der Systemingenieur das angestrebte Zielsystem.

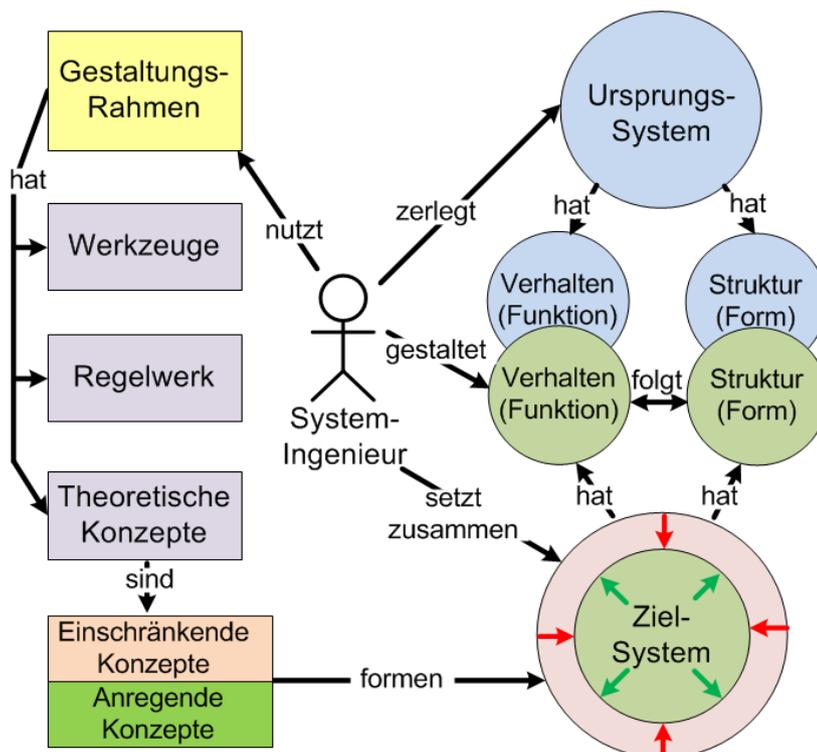


Abbildung 1-2: Concept-Map mit grundsätzlichen Komponenten einer operativen Systemgestaltung

Für die weitere Komplexitätsreduktion nutzt der Systemingenieur die Methode der Dekomposition^[Deo78], indem er das vorhandene (oder noch erdachte) Ursprungssystem in seine Verhaltens- und Strukturkomponenten zerlegt. Dabei bestimmen der Zweck und die Funktionen eines Systems sein Verhalten. Die Strukturen eines Systems bestimmen die Systemorganisation, wie die Anordnung der Systemelemente und ihre Beziehungen untereinander, die möglichen Zustände und Eigenschaften. Die Abbildung 1-2 zeigt eine Concept-Map (vgl. Anhang A) mit dem eben beschriebenen grundsätzlichen Vorgehen während einer Systemgestaltung.

Bei der Dekomposition analysiert der Systemingenieur mögliche Probleme, Konflikte und überlegt sich Lösungen dafür. Er bestimmt, meist in einem iterativen Ablauf^[MiD09], die funktionalen und die nichtfunktionalen Anforderungen an das Zielsystem und gestaltet die möglichen Prozesse, sowie die Funktionen und die Strukturen selbst. Genau an dieser Stelle kann für die konkrete Gestaltung die grundlegende Frage beantwortet werden, ob die Funktion der Form oder die Form der Funktion des Systems folgt^[Pre06]. Am Ende der Gestaltung wird das Zielsystem zusammengesetzt, getestet, optimiert, evaluiert und für die Nutzung freigegeben.

Das wesentliche Ziel dieser Arbeit ist, einen effektiven Rahmen für die operative Gestaltung kognitiver Systeme, vor allem für den Bereich automatisierter Fahrzeuge, zu beschreiben und zu evaluieren. Darüber hinaus zielt diese Arbeit darauf ab, die bekannte Gestaltungstheorie zu diesem Zweck zu transformieren und zu homogenisieren. Zusätzlich benötigte theoretische Konzepte sollen ergänzend der bekannten Gestaltungstheorie hinzugefügt werden.

1.4 Lösungsansatz: Gestaltungsrahmen reduziert Komplexität

Der adäquate Lösungsansatz kann seinen Ursprung in dem Forschungsfeld des Cognitive Systems Engineering (CSE)^[HoW82] haben. CSE ist ein wichtiger Teil des Systems Engineering^[EIG08], da die modernen Systeme immer komplexer werden und die Kognition immer noch ein entscheidender Bestandteil dieser Systeme ist. CSE liefert vor allem für die Gestaltungsphase aus dem Systems Engineering wichtige Ansätze, welche die Eigenschaften der sich innerhalb des Systems befindenden kognitiven Elemente berücksichtigen, sich zunutze machen und später ausführlich dargestellt werden.

In den letzten drei Jahrzehnten wurde auf dem Forschungsfeld des CSE mit dem Ergebnis einiger sehr nützlicher Systemgestaltungskonzepte viel unternommen. Einer der Erfolge (und vielleicht sogar der wichtigste) war die Definition der Maschine als kognitiver Agent^[HoW82] und somit als Standardelement eines kognitiven Multiagentensystems, das noch ausführlich eingeführt wird. Die modernen Maschinen können bereits komplexe Situationen wahrnehmen und verstehen, z.B. durch eine Sensordatenfusion. Die Maschinen sind in der Lage, dynamische Situationen vorherzusagen, z.B. probabilistisch, um dann autonom Entscheidungen zu treffen und Aktionen auszuführen. In anderen Worten, die heutigen Maschinen können als kognitiv bezeichnet werden, weil sie die Komponenten und Platzhalter für alle Ebenen des Situationsbewusstseins^[Eny95] sowie zahlreiche interne Wissens- und Verhaltensmodelle besitzen.

Die nächsten wichtigen Errungenschaften im Bereich des CSE waren der „User Centered Design“-Ansatz und das „Usability Engineering“^{[NoD86][Nin93]}. Diese rückten die Beachtung der menschlichen Kognition in den Mittelpunkt des CSE und schafften damit zunächst einen Ge-

genpol und danach ein Gleichgewicht zum bis dahin vorherrschenden „Technology Centered Design“-Ansatz^[Gan03]. Nach der ausführlichen Untersuchung einiger Design-Metaphern für die hochautomatisierte Fahrzeugführung, wie H-Metapher^[Flh00] oder virtueller Co-Pilot^[Hon07], wurde es immer deutlicher, dass die Gestaltung der kognitiven Maschine nicht zwingend mensch-ähnlich, sondern mensch-kompatibel sein sollte^{[FIK08][KeH12]}.

Ferner ist die Methodik der kognitiven Funktions-, Aufgabenanalyse^{[RaP94][WoH87][Boy98][HoM08]} und der kognitiven Arbeitsanalyse^[Vie99] als bewährte Ansätze zu berücksichtigen und innerhalb der CS-Gestaltung zu nutzen. Diese Ansätze ermöglichen eine Analyse von CAs durch systematische Berücksichtigung der kognitiven Prozesse und Funktionen innerhalb der CAs. Die genannten Gestaltungsansätze werden im Verlauf der Arbeit eingeführt und bewertet. Während der bei den genannten Ansätzen vorgeschlagenen *funktionalen CS-Dekomposition* können die funktionalen CA- und CS-Anforderungen gesammelt und geklärt werden.

Als ein Lösungsansatz für die genannte Problemstellung und als ein Forschungsbeitrag zum CSE wurde der **operative strukturfokussierte** Rahmen für die **Gestaltung** holistischer dynamischer **kognitiver Systeme (osGekoS)** entwickelt (Abbildung 1-3 vgl. auch Abbildung 1-2). Wie jeder Gestaltungsrahmen besteht er aus einer Gestaltungstheorie, einem Regelwerk und einer Werkzeugkette. In der vorliegenden Arbeit wird er systematisch aufgezeigt und in zwei exemplarischen Einsätzen auf seine grundsätzliche Anwendbarkeit für den Bereich hochautomatisierter Fahrzeuge evaluiert.

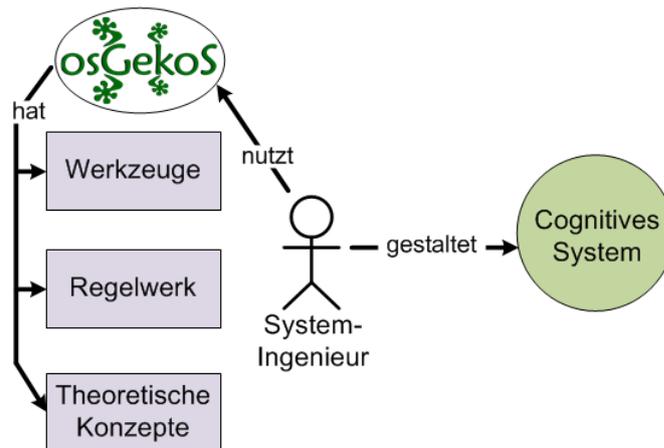


Abbildung 1-3: osGekoS ist ein Gestaltungsrahmen für kognitive Systeme

1.5 Forschungsfragen

Neben den funktionalen Anforderungen bei der Gestaltung eines CS gibt es auch eine starke Notwendigkeit für die Definition der strukturellen Anforderungen und somit die Notwendigkeit der *strukturellen CS-Dekomposition*. Bei einer operativen Gestaltung kognitiver Systeme ist es nicht ausreichend, nur die funktionalen Fragen zu beantworten, z.B. wann, wie und welche Aufgaben welchen CAs zugewiesen werden sollten, was müssen die CAs tun und können, welche Informationen sie dafür benötigen und welche Ziele sie mithilfe welcher Ressourcen zu erreichen haben? Um sich zielgerichtet und effizient in Richtung einer konkreten und erfolgreichen Gestaltung und Implementierung eines CS zu bewegen, ist es von entscheidender Bedeutung und Notwendigkeit folgende zwei Forschungsfragen zu beantworten.

Erstens: *Wie kann die Struktur eines kognitiven Systems verstanden und beschrieben werden?*

Das bedeutet im Detail: Wie können die CS mittels kognitiver Elemente zerlegt, modelliert und in der Soft- und Hardware des Entwicklers abgebildet werden? Welche Strukturelemente sollten abgebildet werden? Welche ist die allgemeine Datenrepräsentation dafür?

Es gibt viele theoretische Überlegungen zu den CA-Architekturen^{[RaG91][AnM97][Sun03]} und einige zu den CS-Architekturen^{[Pu003][ShL09]}, die später eingeführt und bewertet werden. Leider entdeckt man in der einschlägigen Literatur einen Mangel an einfach implementierbaren Konzepten auf der Ebene der operativen CS-Gestaltung. Vermutlich ist dieser Umstand der Knappheit der generischen Struktur- und Verhaltensähnlichkeiten zwischen den unterschiedlichen Arten von CS geschuldet. Welche Strukturen und Verhalten gibt es z.B., die gemeinsam für solche CS, wie ein Büroarbeitsplatz, ein Fahrer-Automationssystem oder ein UAV-Schwarm sind?

Solche Ähnlichkeitshinweise können wesentliche Informationen liefern, um ein allgemeines und einfach handhabbares Beschreibungsmittel für CS-Strukturen zu entwickeln. Sie können durch eine *Methode der Transformation und Homogenisierung* des vorhandenen Wissens über die CS aufgedeckt werden. Ferner können Konzepte für die Gestaltung der CS in einer dafür geeigneten gestaltungskompatiblen Architektur und Schnittstelle zusammengeführt werden. Damit wäre man in der Lage, die Struktur, aber auch das Verhalten eines CS neu zu gestalten oder zu verändern. Diese Gedanken führen direkt zu der zweiten Forschungsfrage.

Zweitens: *Wie kann eine operative Gestaltung an die Struktur eines kognitiven Systems angebunden werden?*

Das bedeutet im Detail: Wie sehen eine gestaltungskompatible CS-Architektur und eine Gestaltungsschnittstelle dazu aus? Wie kann diese Schnittstelle in Soft- und Hardware des Entwicklers abgebildet werden? Wie kann der kompatible Gestaltungsprozess umgesetzt sein?

Im Verlauf der Arbeit werden diese beiden Forschungsfragen ausführlich behandelt.

1.6 Leistungen und Mehrwert der Arbeit

Diese Arbeit liefert einen Mehrwert, indem hier komprimiert und daher leicht zugänglich bekanntes und teilweise sehr weit zerstreutes Wissen aus dem Umfeld des CSE miteinander verknüpft und bewertet wird. Durch die anschließende Transformation und Homogenisierung dieses Wissens als holistisches Beschreibungsmittel für kognitive Systeme (MPS) und die Erarbeitung der integrierten Gestaltungskonzepte, wie Aktionstension und Arbitrierung, wird ein Forschungsbeitrag dem CSE zurückgegeben. Gleichzeitig wird die Gestaltungstheorie soweit vereinfacht, dass holistisches Verständnis und Beschreibung beliebig komplexer CS möglich werden. Damit lassen sich z.B. alle Arten des CS-Verhaltens, wie Konkurrenz und Kooperation, adressieren, so dass neben dem Verständnis und Beschreibung eine ganzheitliche Gestaltung eines CS innerhalb des erarbeiteten Gestaltungsrahmens möglich ist.

Ein solcher Rahmen wird umso wichtiger, je näher der Systemingenieur sich an der operativen Umsetzung seiner Gestaltung befindet. Daher systematisiert und formalisiert diese Arbeit die transformierte und homogenisierte Theorie und macht sie nutzbar für eine operative Gestal-

tung beliebig komplexer kognitiver Systeme. Dabei liefert sie einen Beitrag in Form einiger neuen Ideen und Konzepte für die strukturfokussierte Gestaltung von CS. Diese werden mittels der ausführlichen Beispiele der damit entwickelten Fahrerassistenzsysteme gezeigt und erläutert. Erarbeitete Regeln, Methoden, Werkzeuge sowie die zwei damit entwickelten exemplarischen Fahrerassistenzsysteme runden den Mehrwert dieser Arbeit ab.

1.7 Struktur und Ergebnisse der Arbeit

Als erstes Ergebnis (Kapitel 2) werden die wichtigsten aus dem Umfeld des CSE bekannten Gestaltungsrahmen zusammengefasst, sortiert, miteinander verknüpft und bewertet. Dort wird auch der Beitrag dieser Arbeit relativ zu den vorhandenen Ansätzen eingeordnet.

Als zweites Ergebnis (Kapitel 3 und Abschnitt 4.1) wird das für die Ziele dieser Arbeit notwendige Wissen präsentiert, mittels der fortlaufend entwickelten Annahmen, Schlussfolgerungen, Abbildungen und Formeln homogenisiert und zum allgemeinen CS-Beschreibungsmittel (MPS) transformiert. Zum homogenisierten und transformierten Wissen gehören die gestaltungsrelevanten Grundlagen der Systemtheorie, der Informationstechnik, der Kognitions- und der Sprachwissenschaft.

Als drittes Ergebnis werden die neuen bzw. noch fehlenden theoretischen Konzepte der Aktionsdimension (Abschnitt 4.2) und der Arbitrierung (Abschnitt 4.3) erarbeitet. Mithilfe dieser Konzepte und des MPS können kognitive Systeme holistisch verstanden, beschrieben und gestaltet werden. Auf dieser Basis werden die gestaltungskompatible CS-Architektur (Abschnitt 4.4) und die dazugehörige Gestaltungsschnittstelle (Abschnitt 4.5) erarbeitet.

Als viertes Ergebnis (Kapitel 5) werden die Regeln und der Prozess der Gestaltung innerhalb des erarbeiteten Rahmens samt aller dazugehörigen Methoden und Werkzeuge dargestellt.

Als fünftes Ergebnis (Kapitel 6) wird der bis dahin erarbeitete Gestaltungsrahmen anhand umfassender Beispiele der Gestaltung zweier neuartiger Fahrerassistenzsysteme erläutert. Hier wird Wert darauf gelegt zu zeigen, wie die erarbeitete Theorie über die Methoden, Regeln, und Werkzeuge der Gestaltung hin zu einem konkreten CS- und CA-Wechselwirkungsdesign für den Bereich hochautomatisierter Fahrzeuge übersetzt werden kann. Die Ergebnisse der Nutzerbewertung der beiden Fahrerassistenzsysteme in den dazugehörigen und ebenfalls präsentierten Usability Studien dienen der Evaluation des erarbeiteten Gestaltungsrahmens.

Als abschließendes Ergebnis (Kapitel 7) folgt der Ausblick auf den weiteren Forschungs- und Entwicklungspotenzial der Inhalte der vorliegenden Arbeit.

Die erläuterten Ergebnisse der Entwicklung von osGekoS innerhalb der vorliegenden Arbeit sind zur Veranschaulichung in der Abbildung 1-4 zusammengefasst. Diese Abbildung wird sich mit Markierungen an entsprechenden Stellen zwecks Orientierung und Navigation am Anfang eines jeden Kapitels wiederfinden.

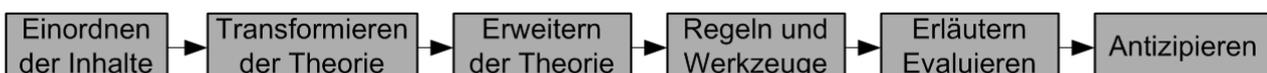
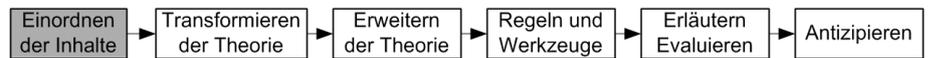


Abbildung 1-4: Ergebnisse der vorliegenden Arbeit zeigen den Weg von der Theorie zur Praxis von osGekoS



2 Systemgestaltungsrahmen

Zweck dieses Kapitels ist, die relevanten Systemgestaltungsrahmen im Umfeld des CSE zusammenzufassen, in Bezug auf die Ziele der vorliegenden Arbeit zu bewerten und zu verknüpfen. Der im Kern dieser Arbeit stehende operative strukturfokussierte Gestaltungsrahmen für kognitive Systeme (osGekoS) ist in die Landschaft anderer Rahmen einzuordnen.

2.1 Einordnung relevanter Systemgestaltungsrahmen

Vielfalt und Komplexität der heute zu gestaltenden Systeme hat sich erhöhend auf die Anzahl der verfügbaren Rahmen ausgewirkt. Dies hat zu einer gewissen Verwirrenheit von heutigen Gestaltungsansätzen geführt, die von Experten des CSE ebenfalls erkannt wird^{[MiD09][Hoc07]}. Viele Konzepte und Rahmen existieren parallel. Die verbreitetsten und nützlichsten werden je nach Anwendungsfall und je nach Priorität des jeweiligen Systemingenieurs eingesetzt. Diese Vielfalt kann einerseits vorteilhaft sein, um dem Systemingenieur die kreative Freiheit in der Wahl der Gestaltungsmethode zu gewähren. Damit eine Wahl andererseits getroffen werden kann, müssen die Gestaltungsrahmen nach Anwendungsschwerpunkten geordnet werden.

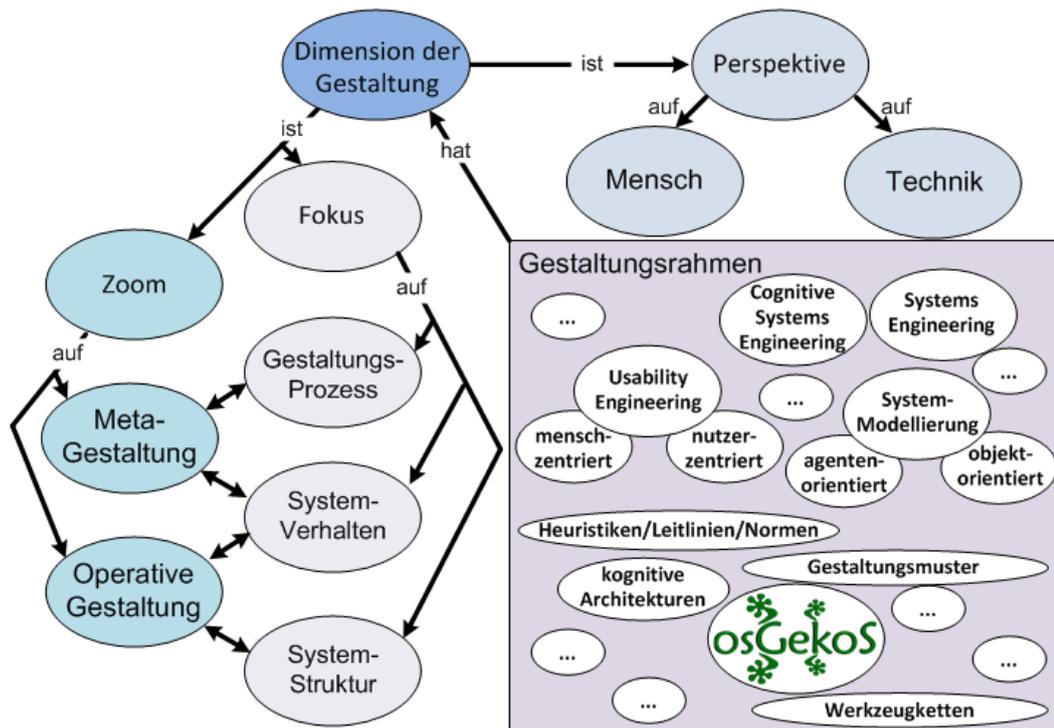


Abbildung 2-1: Einordnung und relative Positionierung des operativen strukturfokussierten Gestaltungsrahmens für kognitive Systeme (osGekoS) aus dieser Arbeit zu den existierenden Gestaltungsrahmen

Abbildung 2-1 zeigt eine Concept-Map mit der Einordnung der relevanten¹ Gestaltungsrahmen aus dem Umfeld des CSE in ein gemeinsames Schema. OsGekoS ist dort ebenfalls eingeordnet und durch seine Position relativ zu den übrigen Rahmen gesetzt. Es gibt drei grundsätzliche Dimensionen (Perspektive, Zoom und Fokus), die in Analogie zu einer fotografischen Aufnahme als Hauptparameter für die Klassifikation der Rahmen verwendet werden können.

¹ Die Relevanz wurde vom Autor selbst und ohne einen expliziten Anspruch auf Vollständigkeit eingeschätzt.

Die Dimension Perspektive zeigt, inwieweit innerhalb des jeweiligen Gestaltungsrahmens vorwiegend² vom Standpunkt des Menschen bzw. des Nutzers oder der verfügbaren Technik gestaltet wird. Damit ist auch gemeint, inwieweit eher die menschnahen oder die techniknahen Methoden und Werkzeuge innerhalb des jeweiligen Rahmens empfohlen werden. Techniknah sind die klassischen Methoden und Werkzeuge aus dem Systems Engineering, wie Projektmanagement^[DOD01], FMEA^[TiM03], semiformale Systemmodellierung mittels z.B. UML^[OMG11], Prototyping, technikleiche Normen, Werkzeugketten und Gestaltungsmuster, die später ausführlicher beschrieben werden.

Menschnah sind Gestaltungsrahmen aus den Bereichen Human Factors und menschliche Kognition. Es sind das Usability Engineering mit mensch- und nutzerzentrierten Methoden, wie Erwartungsabfragen, Usability Studien, Arbeits-, Aufgaben-, Ziele-, und Entscheidungsanalysen. Es sind kognitive Architekturen für eine mögliche Modellierung und Formalisierung der menschnahen Gestaltung. Des Weiteren sind es Leitlinien und Heuristiken bezüglich des menschlichen Situationsbewusstseins, der Belastung und Beanspruchung, des Vertrauens, der Müdigkeit, der Aufmerksamkeit etc. Eine gleichzeitige Perspektive sowohl auf die menschliche als auch auf die technische Komponente eines CS ist ebenfalls möglich. Solche Rahmen sind in der Abbildung zwischen den menschnahen und techniknahen Perspektiven eingeordnet.

Die Dimension Fokus zeigt, auf welchen Teilaspekt der Systemgestaltung der jeweilige Rahmen hauptsächlich fokussiert. Der Fokus kann auf den Gestaltungsprozess selbst gelegt sein: Womit fängt man an, wie geht man vor, wie beendet man die Gestaltung? Er kann mehr auf die Funktion des Systems gelegt sein: Was ist der Zweck, die Funktionen des Systems und seiner Teilkomponenten? Was müssen diese Komponenten leisten? Wie soll sich das System verhalten und welche Prozesse liegen diesem Verhalten zugrunde?

Ein Gestaltungsrahmen kann auch auf die Strukturen bzw. die Form des Systems fokussiert sein: Aus welchen Teilkomponenten besteht das System, welche sind die Schnittstellen und Kommunikationsstrukturen dazwischen, welche Hierarchie und Architektur liegen dem System zugrunde? Welche Daten müssen generiert und ausgetauscht werden? Wie sieht die gemeinsame Repräsentation und die Kommunikation dieser Daten aus? Was muss getan werden, damit die Teilkomponenten des Systems untereinander kompatibel sind? Obwohl der Gestaltungsprozess und die Systemfunktionen sehr wichtig sind, ist die Strukturfokussierung essenziell. Denn die gestaltete Funktionalität eines Systems kann nur auf den tatsächlichen, zur Funktionalität, zum Gestaltungsprozess und untereinander kompatiblen Systemstrukturen erfolgreich und effizient implementiert werden.

Die Dimension Zoom zeigt, wie nah sich die Regeln, Methoden, Techniken und Werkzeuge des jeweiligen Gestaltungsrahmens an der tatsächlichen Implementierung des Systems befinden. Implementierungsferne Rahmen behandeln abstraktere Probleme einer Gestaltung. Solche Rahmenart ist oft den übergeordneten Zielen einer Gestaltung gewidmet, wie der Qualitätssicherung, der Planungssicherheit und der Effizienz großer verteilter Vorhaben. Je mehr es aber um die tatsächliche Implementierung der Systeme geht, desto wichtiger werden die operativen Gestaltungsrahmen. Innerhalb solcher Rahmen wird es dem Systemingenieur erst

² Hier werden manchmal die Begriffe *vorwiegend*, *hauptsächlich* etc. verwendet, weil kein Gestaltungsrahmen *ausschließlich* einer Perspektive, einem Fokus und einem Zoom zugeordnet werden kann.

möglich, seine Gestaltung als Soft-, Hardwarelösungen für gebrauchstaugliche Systeme zu realisieren. Denn das gestaltete CS muss schließlich in seiner Implementierung und Anwendung münden. Für den Bereich hochautomatisierter Fahrzeuge heißt das konkret, dass das gestaltete CS in einem Simulator, einem Versuchsfahrzeug und einem Realfahrzeug implementiert werden muss, um seinen am Anfang der Gestaltung definierten Zweck zu erfüllen.

Neben den prozessbezogenen und funktionalen Anforderungen der Gestaltung von dynamischen kognitiven Systemen ist es von großer Bedeutung, Überlegungen zu strukturellen Anforderungen solcher Systeme anzustellen und diese operativ umzusetzen. Dafür wird ein anwendungsnaher bzw. operativer Gestaltungsrahmen benötigt, der sich zwischen der mensch- und der techniknahen Perspektive befindet und sich auf Strukturen der kognitiven Systeme fokussiert. Genau diese Perspektive, Fokus, und Zoom nimmt der in dieser Arbeit entwickelte osGekoS ein. Durch den Zweck der operativen Gestaltung ist dieser Rahmen als eine sinnvolle Erweiterung der bereits vorhandenen Gestaltungsrahmen zu verstehen und einzuordnen.

Tabelle 2-1: Bewertung der um osGekoS liegenden Gestaltungsrahmen

Rahmen Kriterium	Heuristiken, Leitlinien, Normen	Gestaltungsmuster	Kognitive Architekturen	Werkzeugketten	osGekoS
Ganzheitlichkeit, Durchgängigkeit	++	+	++	+	+++
Skalierbarkeit, Übertragbarkeit	+	++	++	+	+++
Effizienz, Effektivität	++	++	+	+	+
Benutzbarkeit, Verständlichkeit	++	++	+	+++	++

Die um osGekoS liegenden Gestaltungsrahmen (vgl. Abbildung 2-1) lassen sich bezüglich weiterer gestaltungsrelevanter Kriterien, wie Ganzheitlichkeit, Skalierbarkeit, Effizienz und Benutzbarkeit, noch enger miteinander verknüpfen und vergleichen. Dabei fällt auf (vgl. Tabelle 2-1), dass osGekoS in seiner Effizienz und Benutzbarkeit zwar etwas schlechter sein dürfte als z.B. der unmittelbare operative Einsatz von Heuristiken oder Werkzeugketten. Er ist aber deutlich ganzheitlicher und skalierbarer angelegt als übrige Gestaltungsrahmen. Nach Meinung des Autors sollten genau solche Kriterien, wie Ganzheitlichkeit und Skalierbarkeit, für einen Mittel- bis Langfristigen Erfolg eines Gestaltungsrahmens verantwortlich sein.

In nun folgenden Abschnitten werden die in der Abbildung 2-1 verknüpften Gestaltungsrahmen kurz dargestellt und zu osGekoS in eine Beziehung gesetzt.

2.2 Techniknahe Systemgestaltung

Aus dem Bereich der Gestaltung komplexer technischer Systeme sind diverse Methoden, Techniken und Werkzeuge bekannt, die unter dem Sammelbegriff *Systems Engineering* zusammengefasst werden. Dort stehen eher auf den Gestaltungsprozess fokussierte und aus der technischen Perspektive angewandte Methoden im Vordergrund. Unter dem Dach des Sys-

tems Engineering lassen sich die für osGekoS relevanten Rahmen weiter verfeinern. Objekt- und agentenorientierte *Systemmodellierung*, *Gestaltungsmuster* und *Werkzeugketten* sind bei der Anwendung von osGekoS die wichtigsten relevanten techniknahen Gestaltungsrahmen.

2.2.1 Systems Engineering

Als Standardmodell einer Systemgestaltung innerhalb des Systems Engineering^[USD07] wird das V-Modell (Abbildung 2-2) verwendet. Die linke Flanke des entsprechenden V-Diagramms behandelt die Dekomposition und die Definition des Systems. Dazu gehören die Machbarkeitsstudien, die Erfassung von Systemanforderungen und die Systemgestaltung selbst. Das System wird zunächst abstrakt und danach immer detaillierter gestaltet bis, schließlich, die Implementierung des Systems in Soft- und Hardware möglich ist. Die rechte Flanke behandelt die Integration und die Komposition des Systems. Hier wird der besondere Wert auf den Test der Systemkomponenten, auf die Verifikation der Untersysteme und die Validierung des Gesamtsystems gelegt. OsGekoS kann vorwiegend im unteren Bereich des V-Diagramms angeordnet werden. Damit können die detaillierte Gestaltung, Vorabimplementierung (Prototyping), Implementierung und Detailtests des zu gestaltenden CS adressiert werden.

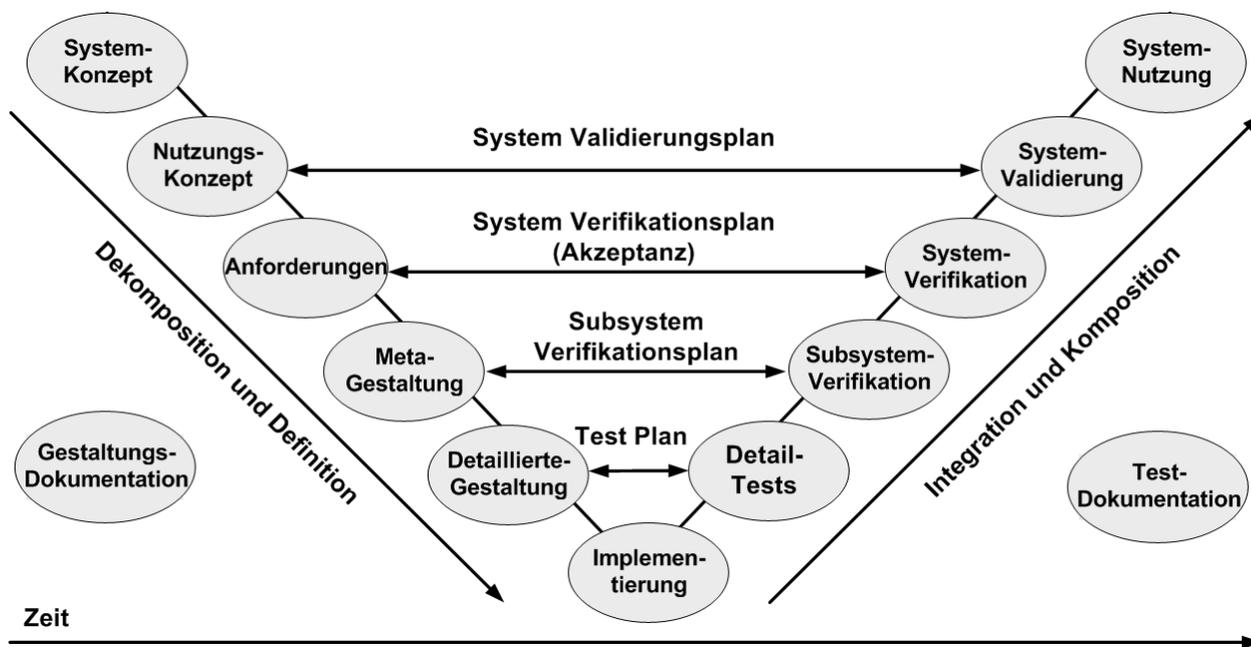


Abbildung 2-2: V-Diagramm: Systemgestaltungsablauf in Systems Engineering (adaptiert nach [USD07])

Wichtige Teilbereiche innerhalb des Systems Engineering sind das Projektmanagement, das Anforderungsmanagement, das Änderungsmanagement und das Risikomanagement.

- *Projektmanagement*^[DOD01] behandelt die grundsätzlichen Abläufe einer komplexen und parallelisierten Systemgestaltung. Mithilfe von Gantt-Diagrammen^[DOD01] z.B., werden die Prozessabläufe innerhalb eines Projektes geplant, visualisiert und entzerrt. Einzelnen voneinander abhängigen Projektschritten können entsprechende Ressourcen, wie Dauer, Finanzmittel, Personal zugewiesen werden. Ein Projektleiter kann sich an dem erstellten Plan orientieren, den Projektfortschritt kontrollieren und, wie im Falle von nicht berücksichtigten Entwicklungen, den Projektplan anpassen.

- *Anforderungsmanagement*^[PoR09] ist ein essenzieller Teil des Systems Engineering. Am Anfang einer Gestaltung werden die Systemanforderungen gesammelt und geklärt. Wichtige Werkzeuge sind dabei z.B. das Lastenheft und das Pflichtenheft, die in einem dezidierten Prozess zwischen dem Leistungsanforderer und dem Leistungserbringer, wie einem Automobilunternehmen und einem Zulieferer, homogenisiert werden. Gegen das Ende des Projektes werden die anfangs aufgestellten und verhandelten Anforderungen für die Verifikation und die Validierung des Systems mitverwendet.
- *Änderungsmanagement*^[Cor93] ist eine projektbegleitende Tätigkeit. Bei der Gestaltung komplexer Systeme ist es wichtig, jederzeit nachvollziehen zu können was, wann, von wem und warum verändert worden ist. Dies erleichtert die mögliche Fehlersuche, den Systemtest, die Erstellung der Dokumentation, der Berichte und sichert die rechtliche Lage des Projektes ab. Ein verbreitetes Werkzeug, das u.a. die Behandlung des Änderungsmanagements anbietet, ist z.B. DOORS^[IBM12].
- *Risikomanagement*^[TiM03] trägt innerhalb des Systems Engineering dazu bei, mögliche Fehler in der Systemgestaltung und in der späteren Systemnutzung frühzeitig zu erkennen. Denn je später in der Systemgestaltung die Fehler erkannt werden, desto kostspieliger ist deren Korrektur^[TiM03]. Als eine effiziente Methode dafür hat sich die FMEA (Fehlermöglichkeits- und Einflussanalyse) bewährt. Dabei wird vor der endgültigen Systemimplementierung bzw. während der Prototyping-Phase systematisch auf mögliche Fehler eingegangen. Das Ziel dabei ist die rechtzeitige Erarbeitung der Lösungsstrategien für die möglichen späteren Systemfehler.

Die Methoden des Systems Engineering haben ihre grundsätzliche Gültigkeit auch auf dem Gebiet des CSE. Dabei muss allerdings beachtet werden, dass der typisch als sequenziell angenommene Gestaltungsablauf nach dem V-Modell sich im Bereich des CSE sehr viel stärker hin zum iterativen Ablauf verändern kann^[MiD09]. Dies ist u.a. durch die nicht präzise erfassbare Komplexität des kognitiven Systemanteils und die daraus notwendigerweise resultierenden schwächer formalisierten Gestaltungsmethoden bedingt. In osGekoS können das Anforderungsmanagement und das Risikomanagement parallel behandelt werden, was mit dem später vorgestellten Konfliktanalysewerkzeug als Thema aufgegriffen wird. Beim Projektmanagement und dem Änderungsmanagement bedient man sich einfach der Standardwerkzeuge, z.B. jeweils Microsoft Project^[MIT13] und TortoiseSVN^[TON13].

2.2.2 Systemmodellierung

Die Modellierung eines Systems ist ein wichtiger Schritt, um mit der Systemkomplexität handhaben zu können. „Ein Modell ist ein beschränktes Abbild der Wirklichkeit“^[Wi13d]. „Auf der Basis von Funktions-, Struktur- oder Verhaltensähnlichkeiten bzw. -analogien zu einem Original werden Modelle zum Zwecke speziell solcher Problemlösungen benutzt, deren Durchführung am Original nicht möglich oder zu aufwendig wäre“^[SP13d]. Während der Gestaltungs- und Implementierungsphase (vgl. Abbildung 2-2) können Systeme mittels der Modelle beschrieben und/oder simuliert werden. Innerhalb des Systems Engineering ist die Systemmodellierung ein wichtiger Gestaltungsschritt, bei dem präziser, als z.B. durch bloße Anforderungsdefinition, auf tatsächliche Systemverhalten und -strukturen eingegangen werden kann.

Zur Unterstützung der Modellierung können semiformale Sprachen, wie UML (Unified Modelling Language)^[OMG11], eingesetzt werden. Mithilfe von UML lassen sich die systeminternen Verhalten, Funktionen und Strukturen genau beleuchten, ohne diese vorher tatsächlich und im Detail implementieren zu müssen. Des Weiteren können Systeme auf der Ebene der Modellierung nicht nur gestaltet, sondern auch mittels formaler Modellierungssprachen, wie Petrietze^[Scr92], simuliert und analysiert werden. Damit können die Struktur und das Verhalten der Systeme frühzeitig getestet und optimiert werden.

UML 2.x ist ein Standard in der heutigen Systemmodellierung und besteht aus einer Sammlung von Diagrammtypen, durch deren Verwendung das System aus unterschiedlichen Blickwinkeln betrachtet, gestaltet und dokumentiert werden kann. Es gibt sieben Verhaltensdiagramme: Das Sequenzdiagramm, das Aktivitätsdiagramm, das Anwendungsfalldiagramm, das Interaktionsübersichtsdiagramm, das Kommunikationsdiagramm, das Zeitverlaufdiagramm und das Zustandsdiagramm. Des Weiteren gibt es sieben strukturbezogene Diagramme: Das Klassendiagramm, das Kompositionsstrukturdiagramm, das Komponentendiagramm, das Verteilungsdiagramm, das Objektdiagramm, das Paketdiagramm und das Profildiagramm.

Für osGekoS am wichtigsten sind das Sequenzdiagramm und das Klassendiagramm. Das Sequenzdiagramm bildet die Grundlage für die Modellierung der Kommunikationsabläufe zwischen den CAs mittels des später vorgestellten Werkzeugs „Syntaktisches Storyboard“. Das Klassendiagramm nutzt man für die Modellierung der Systemstrukturen, wie sie im Werkzeug „Interaktionskontroller“ Eingang findet. Beide Diagramme sowie später eingeführte Werkzeuge des osGekoS greifen die Ideen der objektorientierten Systemmodellierung auf.

2.2.3 Objektorientierte Systeme

Die Grundprinzipien der beiden für osGekoS relevanten UML-Diagrammtypen und der objektorientierten Systemmodellierung sind in der Abbildung 2-3 dargestellt. Mittels der Klassendiagramme können Strukturen und Organisation eines objektorientierten Systems modelliert werden. Grundsätzlich besitzen Objekte einen Zustand (Attribute) und ein Verhalten (Funktionen). Hierarchien von Objektklassen sowie die Schnittstellen dazwischen können damit aufgebaut werden. Bei den Klassen, die eine Art Schablonen für den späteren Objektaufbau sind, werden die grundsätzlich möglichen Objektverhalten und -zustände notiert.

Zwei Hauptmechanismen zum Aufbau von Klassenhierarchien sind die Vererbung und die Implementierung. Diese unterscheiden sich darin, ob nur das Verhalten ohne Struktur (Implementierung) von der Mutterklasse an die Kindklasse vererbt wird oder beides (Vererbung). Damit kann bei unterschiedlichen Objekten ein ähnliches Verhalten über dieselben Schnittstellen (Interfaces) aber unterschiedliche Strukturen modelliert werden. Des Weiteren ist es in einem Klassendiagramm möglich, Beziehungen zu notieren. Objektklassen können zueinander assoziiert werden, wenn sie zueinander in einer Beziehung stehen. Als Spezialfälle der Assoziation können Klassen in anderen Klassen zusammengefasst werden, wenn Klassenstrukturen aus anderen Klassen nicht zwingend (Aggregation) oder zwingend (Komposition) bestehen.

Mittels eines Sequenzdiagramms ist es möglich, das dynamische Verhalten der Objekte zu beschreiben. Objekte werden nach ihren Schablonen (Klassen) durch eine Erzeugungsnachricht innerhalb des Systems instanziiert. Solche Objektinstanzen haben eine bestimmte Lebensdauer.

er, die durch ihre Lebenslinien gekennzeichnet wird. Am Ende des relevanten Zeitraums können diese Instanzen wieder zerstört werden. Während der Lebensdauer ihrer Instanzen können die Objekte synchrone und asynchrone, fremd- und an sich selbst gerichtete Nachrichten austauschen. Diese Nachrichten haben ebenfalls eine bestimmte Lebensdauer (weiße Vierecke auf den Lebenslinien). Der Unterschied zwischen den synchronen und asynchronen Nachrichten ist implementierungsbedingt und bezeichnet die Notwendigkeit oder eben keine, auf die Antwort der Gegenseite zu warten, um das eigene Verhalten fortzusetzen.

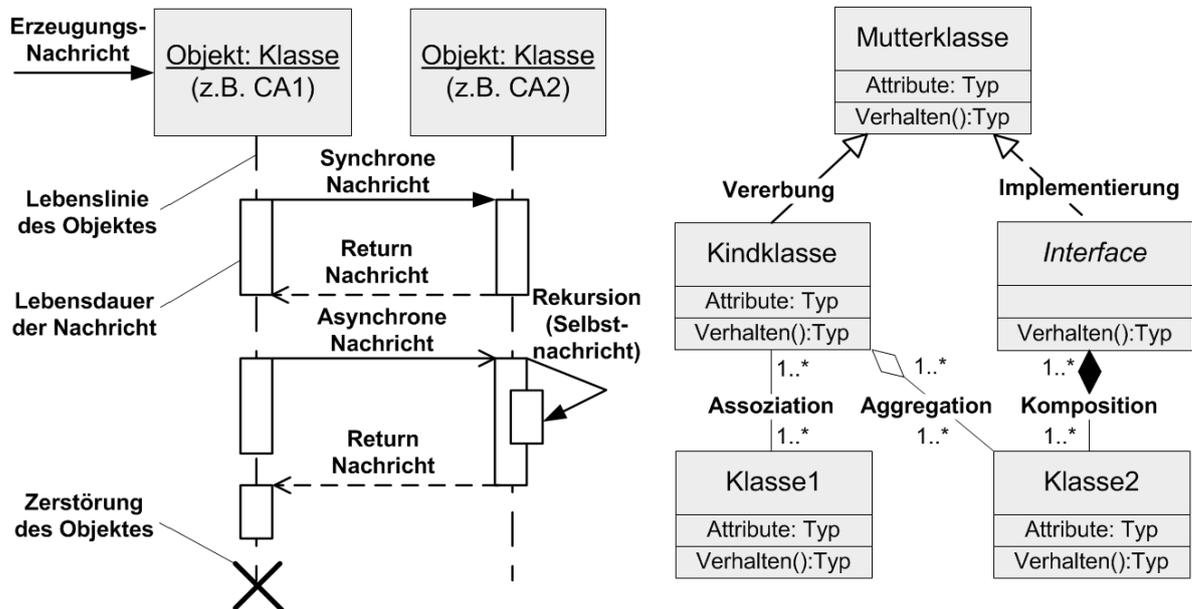


Abbildung 2-3: Sequenzdiagramm (links) und Klassendiagramm (rechts) nach UML

Die vorgestellten Methoden und Techniken der objektorientierten Systemmodellierung finden ihre meiste Anwendung innerhalb des Software Engineering, sind aber nicht darauf beschränkt. Ein Vorteil des UML-gestützten Modellierens für kognitive Systeme, die ebenfalls (kognitive) Verhalten und Zustände aufweisen, ist die große Vielfalt an Softwarewerkzeugen und Werkzeugketten. So kann für die in UML modellierten Systeme eine Codegenerierung für eine Ziel-Programmiersprache verwendet werden. Dies reduziert den Zeitaufwand der tatsächlichen Systemimplementierung. UML ist außerdem eine wichtige Grundlage für einen fachübergreifenden Dialog zwischen den Modellierern unterschiedlicher Fachrichtungen.

2.2.4 Multiagentensysteme

Für eine Systemgestaltung konsequente und sinnvolle Erweiterung der objektorientierten stellt die Gestaltung mittels der Multiagentensysteme (MAS) dar. Dessen Ursprünge gehen zurück zu Anfängen der Forschung an der künstlichen Intelligenz und an zellulären Automaten^[Nen66]. Als Vorreiter der MAS wurden zelluläre Automaten bereits für die Domäne des CSE aufbereitet, z.B. durch den Vorschlag eines Formalismus zur Modellierung verteilter ökologischer Prozesse^[Hog88]. Es gibt aktuellere Meinungen^[Wom02], die den zellulären Automaten und somit den MAS eine große Bedeutung für die Entwicklung der Wissenschaft insgesamt zuteilen.

Bei der Definition eines Agenten existiert in der relevanten Community noch keine allgemeingültige Meinung. Ein Agent kann „von außen betrachtet“ definiert werden. Eine der eingän-

gigsten Definitionen dafür lautet: „an agent is any entity *able to act*“³[Cai98]. Ein Agent kann auch „von innen betrachtet“ definiert werden: „An agent is a highly organized complex system which has intrinsic goals“⁴[Vav09]. Insgesamt kann ein Agent als eine zielgerichtet agierende Einheit mit verschiedenen möglichen Informationszuständen innerhalb seiner komplexen Strukturen verstanden werden. Dies kann gleichermaßen auf Menschen sowie komplexe Maschinen zutreffen, was im späteren Verlauf der Arbeit weiter vertieft werden soll.

Interagierende oder in einer anderen Beziehung zueinander stehende Agenten können Populationen bilden. Beispiele solcher Beziehungen sind Ähnlichkeiten in den äußeren und inneren Strukturen, wie ähnliche Aufgaben, Absichten, Ziele, örtliche oder soziale Nähe. Agenten und Populationen mit „unterschiedlichen Ähnlichkeiten“ lassen sich insgesamt zu Multiagentensystemen zusammenschließen. „Multiagent systems are those systems that include multiple autonomous entities with either diverging information or diverging interests, or both“⁵[ShLO9].

Der Paradigmenwechsel von der objekt- zur agentenorientierten Systemmodellierung ist von einer so grundlegenden Natur, dass damit die Systemgestaltung aus einem prinzipiell neuen Blickwinkel erscheint⁶[LuM05]. Anstatt von gekapselten Objekten aller Art mit heterogenem Verhalten und Zuständen verwendet man bei dem Multiagenten-Konzept die homogenen, ähnlich modellierten Agenten. Diese müssen bestimmte Eigenschaften, wie einen hohen Grad an Autonomie, eine Interaktionsbereitschaft innerhalb lokaler geometrischer und sozialer Umgebungen, eine Robustheit gegenüber offenen Umgebungen etc. aufweisen.

Heute ist der objektorientierte Modellierungsansatz noch deutlich weiter verbreitet und bekannt als der agentenorientierte⁷[LuM05]. Daher erscheint es als wichtig, die beiden Paradigmen miteinander besonders in Bezug auf die Gestaltung von CS zu vergleichen. Zunächst muss man zugeben, dass die Trennlinie zwischen den „klassischen“ Objekten und den Agenten schnell verschwimmen kann, z.B. „agents are objects with an attitude“⁶ oder „objects are agents without these extra agent attributes“⁷[OMG00]. Beide informationstheoretische Konzepte gehen gleichermaßen vom dynamischen Verhalten sowie von statischen Zuständen innerhalb der Strukturen ihrer Entitäten aus. Dennoch gibt es wichtige Unterschiede zwischen dem objekt- und agentenorientierten Gestaltungsparadigma. Deren wichtigsten Gemeinsamkeiten und Unterschiede sind in der Tabelle 2-2 zusammengetragen^{[OMG00][ShLO9][LuM05][WI13f]}.

Durch ihre Ähnlichkeit zu den objektorientierten Systemen können die MAS mit UML-Methoden modelliert werden (Abschnitt 2.2.3). UML-Diagramme können für agentenorientierte Modellierung mittels der Agent UML^[BaM01] geringfügig durch Elemente, wie zugewiesene Rolle, Aufgabe, Kommunikationsprotokolle, Verhaltensmuster etc. angepasst werden. In dieser Arbeit wird Agent UML nicht verwendet, da die angebotenen Erweiterungen keinen wesentlichen Vorteil für die Darstellung der Inhalte sowie für das Erreichen der Ziele bringen.

³ Ein Agent ist eine beliebige agierende Entität. EÜ.

⁴ Ein Agent ist ein hoch organisiertes komplexes System mit inneren Zielen. EÜ.

⁵ Multiagentensysteme enthalten multiple autonome Einheiten mit unterschiedlichen Informationen oder Interessen oder beidem. EÜ.

⁶ Agenten sind Objekte mit einer inneren Einstellung. EÜ.

⁷ Objekte sind Agenten ohne diese zusätzlichen Agentenattribute. EÜ.

Tabelle 2-2: Gemeinsamkeiten und Unterschiede zwischen Objekten und Agenten

Perspektive	Objekte...	Agenten...
Aufbau	Besitzen einen statischen Zustand (Attribute) und ein dynamisches Verhalten (Funktionen), wobei Objekte einen eher heterogenen und Agenten einen eher homogenen Aufbau haben	
Hierarchie	Können hierarchisch organisiert werden: Objekte überwiegend durch Klassenstruktur, Agenten überwiegend durch Zusammenfassung zu Populationen	
Verortung	Können lokal oder verteilt auftreten. Die Verteilung steht bei den Agenten mehr im Vordergrund	
Initialisierung	Werden oft zu Anfang des Programm- bzw. Routinen-Ablaufes explizit initialisiert	Werden meist nach Bedarf initialisiert, können sich selbst reproduzieren, können andere Agenten produzieren und konsumieren
Verhalten	Haben meist ein transparentes, weil z.B. fest einprogrammiertes, Verhalten. Haben kein explizites soziales Verhalten, fokussieren eher auf eigene interne Prozesse, Funktionen und Strukturen	Können ein (von außen betrachtet) nichtdeterministisches Verhalten zeigen, z.B. aufgrund von Lernmechanismen, können Koalitionen, Populationen bilden, fokussieren eher auf Kommunikation und Interaktionen innerhalb des Systems
Wissen Fähigkeiten	Können allwissend sein, z.B. durch die Nutzung von Klassenvariablen und -funktionen, können (begrenzt) selbständig komplexe Probleme lösen	Sind i.d.R. nicht allwissend. Das Systemwissen entsteht durch die Wechselwirkung mit anderen Agenten. Können nicht selbständig komplexe Probleme lösen, sondern nur im Verbund
Interaktion	Können im Prinzip mit allen anderen Objekten interagieren	Können nur mit örtlich und/oder sozial benachbarten Agenten interagieren
Organisation Koordination	Müssen koordiniert werden mit einer eingeschränkten Parallelität des Verhaltens	Können koordiniert werden, nutzen aber Selbstorganisation, Emergenz und Parallelität des Verhaltens
Autonomie	Besitzen keine explizite Autonomie	Besitzen eine explizite Autonomie
Umgebung	Werden in „geschlossenen“ Umgebungen, z.B. innerhalb einer Domäne, eingesetzt	Werden in „offenen“, domänenübergreifenden Umgebungen eingesetzt
Robustheit	Sind weniger robust, können z.B. Wissen verlieren, wenn Systeme kollabieren	Sind robust, können Mechanismen zur Wissens-erhaltung im verteilten System besitzen

Aus Modellierungssicht lässt sich zusammenfassen, dass Agenten eine spezielle Form der Objekte mit einer eingeschränkten Sicht auf die äußeren Systemzustände und -verhalten sowie einer hohen Autonomie sind. Genau diese Eigenschaften, die einer Kognition sehr nahe stehen, wie die Zielgerichtetheit, Autonomie, Emergenz, Selbstorganisation und Robustheit gegenüber offenen Umgebungen, können das Multiagenten-Konzept nützlich für die Gestaltung kognitiver Systeme machen. Daher spielen MAS in osGekoS eine wichtige Rolle.

2.2.5 Techniknahe Gestaltungsleitlinien

Code-Of-Practice, Richtlinien und Normen sind drei miteinander verwandte Gestaltungsrahmen, welche man zur Kategorie *Leitlinien* zählen kann. Der wichtigste Unterschied zwischen ihnen ist der Reifegrad und der Grad der Formalisierung. Ein Code-Of-Practice entsteht oft in forschungs- und vorentwicklungsnahem Umfeld beim gehäuftem Auftreten relativ neuer aber ähnlicher Gestaltungsvorhaben. Seine Empfehlungen sind weniger formal und restriktiv. Richtlinien und Normen sind dagegen mehr ausgereift und strikter formuliert. Die noch restriktivere Stufe bilden die Vorschriften, die in Gesetzen festgehalten sein können.

Code-Of-Practice, Richtlinien und Normen werden in Gremien von Fachleuten auf entsprechenden Gebieten verhandelt und erstellt. Unter Einbeziehung der Expertise auf dem Gebiet der zu gestaltenden Systemart, einigen sich die Fachleute auf die für alle Beteiligten annehmbare Form und den Inhalt dieser Leitlinien. Der Inhalt kann sowohl die meta- als auch die operative Gestaltung, sowohl die technik- als auch menschnahe Perspektiven einnehmen. Die verhandelten Inhalte werden in Dokumenten festgehalten und stehen dann der Öffentlichkeit zur Verfügung. Obwohl Code-Of-Practice, Richtlinien und Normen keine offizielle Rechtsverbindlichkeit haben, sind sie im Bereich der Systemgestaltung und hochautomatisierter Fahrzeuge sehr weit verbreitet und akzeptiert. Dies hat mindestens drei wichtige Gründe:

- Vom Gesetzgeber wird bei eventuellen Rechtsstreitigkeiten auch im Bereich der Systemgestaltung *Stand der Technik* verlangt. „Stand der Technik im Sinne dieses Gesetzes ist der Entwicklungsstand fortschrittlicher Verfahren, Einrichtungen oder Betriebsweisen“^[BUZ13]. Er wird oft in Richtlinien festgehalten. Bei eventuellen Gerichtsverfahren werden Sachverständige eingesetzt, die auf Basis aktueller Richtlinien und Normen zur Beweisaufnahme und damit zum Ausgang des Gerichtsverfahrens beitragen. Dieser Zusammenhang animiert das Befolgen des Stands der Technik.
- Code-Of-Practice, Richtlinien und Normen fördern eine *Standardisierung* der zu gestaltenden Systeme. Solche Standardisierung kann Gestaltungsvorhaben beschleunigen und zwischen verschiedenen Unternehmen abstimmen. Damit können sich wiederholende Gestaltungsprozesse, Systemverhalten und -strukturen effizienter umgesetzt werden. Standardisierung mindert den Mehraufwand bei Abstimmungen der Systemanforderungen und -spezifikation zwischen den unterschiedlichen an der Gestaltung beteiligten Gruppen. Sie ist auch ein Gewinn für den späteren Systemnutzer. Beim Umstieg von einem normierten System auf ein anderes muss der Nutzer nicht viel umlernen, was den möglichen Nutzungsproblemen entgegenwirken kann.
- Der weitere Grund ist die von Code-Of-Practice, Richtlinien und Normen geförderte *Erfahrungsweitergabe*. Das Wissen der beteiligten Experten kann dadurch weitergegeben werden und alle beteiligten Systemingenieure und Unternehmen profitieren davon. Eine Erfahrungsweitergabe ist auch ein Gewinn für den späteren Nutzer der Systeme. Denn durch den Einfluss des Expertenwissens tauchen die Anforderungen des Nutzers in den Richtlinien und Normen indirekt auf.

Neben der Beschreibung der relevanten Theorie bieten Code-Of-Practice, Richtlinien und Normen auch Checklisten und andere Dokumentvorlagen, die bei der Gestaltung entsprechender Systeme verwendet werden können. Für den Bereich der Mensch-Maschine-Systeme, der hochautomatisierter Fahrzeuge und damit bei der Anwendung von osGekoS können für den Systemingenieur folgende Code-Of-Practice und Normen wichtig sein.

Ein Ergebnis des EU-Projektes PReVENT^[W13e] und seines Teilprojekts RESPONSE 3^[ERO13] ist ein Code-Of-Practice, der weiter (zum RESPONSE 4) ausgearbeitet wird^[ACN13]. Darin werden Definitionen, Verfahren und Empfehlungen für die Entwicklung von Fahrerassistenzsystemen in hochautomatisierten Fahrzeugen behandelt. Durch einen expliziten Fokus auf die Usability der Systeme nimmt dieser Code-Of-Practice eine gemischte Perspektive zwischen dem Menschen und der Technik ein. Von der Europäischen Union gibt es auch Empfehlungen und Leitlinien

für die Gestaltung der Mensch-Maschine-Interaktion in Straßenfahrzeugen^[EUN06]. Die ISO 26262^[IS12b] behandelt die funktionale Sicherheit bei der Entwicklung von Straßenfahrzeugen.

Die ISO 9241 beschreibt generelle Anforderungen und Prüfverfahren für die Gestaltung der Mensch/Maschine-Systeme. Ihr Teil ISO 9241-210^[ISO08] behandelt die menschenzentrierte Gestaltung interaktiver Systeme und enthält Empfehlungen, wie man die Nutzeranforderungen erhebt, spezifiziert und evaluiert. Der Teil ISO 9241-110^[ISO04] behandelt die Grundsätze einer Dialoggestaltung. Der Teil ISO 9241-920^[ISO09] behandelt die Gestaltung taktile und haptischer Interaktionen. Die ISO 15006^[IS12a] enthält Normen für die Gestaltung auditiver und die ISO 15008^[ISO3b] visueller Aspekte der Mensch/Maschine-Interaktion in Straßenfahrzeugen. Die ISO 9241-11^[ISO98] behandelt grundsätzliche Usability-Angelegenheiten und die ISO 17287^[ISO3c] Empfehlungen für Usability Studien im Kontext der Fahrerassistenzsysteme.

2.2.6 Techniknahe Gestaltungsmuster

Während einer Systemgestaltung befindet sich der Systemingenieur auf der Suche nach intuitiv verständlichen und technisch implementierbaren *Lösungen* zu *Problemen*, die sein Zielsystem später adressieren soll. Generische Probleme und deren Lösungen bilden Problem/Lösungs-Paare, die in einem solchen Verbund als Muster (engl. pattern) bezeichnet werden können. Grundsätzlich gibt es oft Lösungsvorschläge für gestaltungsrelevante Probleme, so dass es sehr hilfreich ist, sich bei den bereits vorhandenen Gestaltungsmustern umzuschauen. Die Ursprünge solcher Muster lassen sich grundsätzlich in zwei Kategorien einteilen: natürliche und künstliche. Die natürlichen Muster können für den Nutzer etwas intuitiver verständlich sein. Die künstlichen Muster können dann schnell für den Systemnutzer zugänglich sein, wenn sie auf das oft verwendete und/oder gelernte Wissen des Nutzers zurückgreifen.

Design-Pattern sind nützliche, meist künstlich erzeugte Problem/Lösungs-Paare. Sie wurden als eine eigene Pattern-Sprache im Bereich der Architektur entwickelt. „Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice“^{8[All77]}. Das Konzept des Design-Patterns wurde in vielen Gebieten adaptiert, wie Software Engineering oder bei der Gestaltung von kooperativen Mensch/Maschine Interaktionen und Systemen^{[GrP00][D3S12]}. OsGekoS nutzt die Idee der Design-Pattern in seinem „Konfliktanalysewerkzeug“ (vgl. Abschnitte 4.3.3 und 5.1).

2.2.7 Werkzeugketten

Auf der operativen Seite der techniknahen Gestaltung befinden sich Werkzeugketten. Diese bieten eine durchgängige Unterstützung für die Umsetzung der vom Systemingenieur beabsichtigten Gestaltung hin zur Implementierung des Zielsystems. Werkzeugketten können zu operativen Gestaltungsrahmen gezählt werden, da sie ihre eigenen anregenden und einschränkenden Konzepte für die Gestaltung mit sich führen (vgl. Abbildung 1-2).

⁸ Jedes Muster beschreibt ein Problem, das immer und immer wieder in unserer Umgebung auftritt, und den Kern der Lösung für dieses Problem, so dass Sie diese Lösung über eine Million Mal verwenden können, ohne es jemals zweimal auf die gleiche Weise zu tun. EÜ.

Werkzeugketten können in zwei Ausprägungen existieren. Gestaltungsrahmen führen oft neben den übergeordneten Konzepten und Regelwerken für den Ablauf der Gestaltung eigene *geschlossene* Werkzeugketten mit sich, die an die Gestaltungskonzepte des Rahmens angepasst sind. Und es gibt *offene* Werkzeugketten, wie sie im Bereich des Grafikdesigns oder des Software Engineering verfügbar sind. Mithilfe meist universeller Werkzeuge offener Werkzeugketten, wie einer C++ Umgebung oder einer Adobe Creative Suite^[ADE13], lässt sich im Prinzip jede rechnerbasierte Gestaltung umsetzen. Dennoch lässt sich mit der Adobe Creative Suite schlecht programmieren, genauso wie mit einer C++ Umgebung multimodale MMI umsetzen. Solche Einschränkungen sind auch bei geschlossenen Werkzeugketten zu beachten. Wenn man für eine Gestaltung strikt die eigene Werkzeugkette eines bestimmten Rahmens verwendet, kann man nur so viel gestalten, wie die geschlossene Werkzeugkette erlaubt.

Für die Gestaltung von komplexen technischen Systemen und der Mensch/Maschine-Interaktion gibt es zahlreiche offene Werkzeugketten. Es ist nicht das Ziel dieser Arbeit, diese vollständig aufzuzählen. Hier sollen lediglich einige davon erwähnt werden, die bei der Nutzung von osGekoS neben seiner eigenen Werkzeugkette relevant sein können. In der Regelungstechnik, der Signalverarbeitung und der damit verbundenen Datenverarbeitung- und Auswertung hat sich z.B. die Werkzeugkette MATLAB der Firma MathWorks^[MAS13] etabliert. Die kostenfrei verfügbare Alternative zu MATLAB heißt GNU Octave^[GNU13]. In der Automobilindustrie werden für ähnliche Zwecke z.B. die Werkzeugketten ASCET und INCA der Firma ETAS^[ETS13] eingesetzt. Im Bereich der sicherheitsrelevanten Softwareentwicklung im Flugzeug- und Eisenbahnbereich wird die SCADE Suite der Firma ESTEREL Technologies^[ESL13] genutzt. Mit ihrer Erweiterung SCADE Display bietet diese Werkzeugkette eine Möglichkeit, dynamische visuelle Displays zu entwickeln. Eine sehr ähnliche Alternative dazu ist das Werkzeug Adobe Flash von Acrobat^[ADE13] oder die Altia Design-Werkzeugkette^[ALA13].

2.3 Menschnahe Systemgestaltung

Neben der Rahmen der Gestaltung technischer Systeme haben sich im Laufe der letzten drei Jahrzehnte gemischte und menschnahe Rahmen aus der Psychologie und Human Factors herausgebildet. Diese rücken den Menschen, seine Fähigkeiten, Bedürfnisse, aber auch seine inhärenten Verhalten, Funktionen, Organisation und Strukturen in den Vordergrund der Systemgestaltung. Wichtige Gestaltungsrahmen für osGekoS sind z.B. Usability Engineering mit seiner mensch- und nutzerzentrierten Methodik, menschnahe Gestaltungsleitlinien und -Heuristiken, kognitive Architekturen und menschnahe Design-Pattern. Diese Gestaltungsrahmen werden in folgenden Abschnitten erläutert (vgl. auch Abbildung 2-1).

2.3.1 Usability Engineering

Beim Usability Engineering geht es darum, mit unterschiedlichen Empfehlungen, Methoden, Techniken und Werkzeugen eine gute Usability (dt. Gebrauchstauglichkeit) eines Systems zu erreichen. Die ISO 9241-11 definiert die Usability als das „Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und mit Zufriedenheit zu erreichen“^[ISO98]. Drei Leitkriterien werden von der ISO 9241-11 für das Erreichen einer guten Usability formuliert:

- *Effektivität*: „Die Genauigkeit und Vollständigkeit, mit der Benutzer ein bestimmtes Ziel erreichen“^[ISO98].
- *Effizienz*: „Der im Verhältnis zur Genauigkeit und Vollständigkeit eingesetzte Aufwand, mit dem Benutzer ein bestimmtes Ziel erreichen“^[ISO98].
- *Zufriedenstellung*: „Freiheit von Beeinträchtigung und positive Einstellung gegenüber der Nutzung des Produkts“^[ISO98].

Nielsen^[Nin93] nennt weitere Leitkriterien, die für eine gute Usability wichtig sind:

- *Einfachheit und Verständlichkeit*: Man soll nur das Nötige in einer klaren und für den Nutzer verständlichen Art und Weise kommunizieren.
- *Konsistenz*: Unterschiedliche Interaktionselemente, Systemaktionen und Situationskontexte sollen auch voneinander unterschiedlich wahrnehmbare Bedeutungen haben.
- *Fehlerfreiheit*: Die Anzahl schwerwiegender Fehler, insbesondere beim Erstkontakt mit dem System, soll minimiert werden. Die Fehler sind außerdem ein Maß dafür, wie gut und schnell sich der Nutzer nach dem Auftreten eines Fehlers wieder erholt.
- *Erlernbarkeit und Merkbarkeit* bestimmen, wie einfach und schnell der Nutzer das System erlernen kann und über einen längeren Zeitraum im Gedächtnis behalten kann.

Die Usability drückt also aus, wie gut die gestalteten Systeme dem Menschen bei seinen Aufgaben nützen und sich kurz- und langfristig bedienen lassen.

In CSE wird manchmal zwischen der nutzer- und der menschenzentrierten Gestaltung unterschieden (human and user centred design)^[Gan03]. Aus der Sicht der Menschzentrierung wird die in der Industrie weiter verbreitete Nutzerzentrierung auch kritisiert. Bei der letzteren geht es um den Gebrauch und die Interaktion mit einem gegebenen technischen System. Damit sei der Einfluss der Technik bereits zu stark, da man den Menschen nur indirekt berücksichtigt. Dagegen solle es auch um die allgemeine soziale Wirkung und die Anpassbarkeit der Systeme an die tatsächlichen Bedürfnisse des Menschen gehen. Die Nutzerzentrierung sei auch sehr zieleorientiert^{[Boy98][Gan03]}. Dies kann ein Nachteil sein, da von einem intentionalen, rationalen, bewussten Handeln der Nutzer ausgegangen wird, was nicht immer der Fall ist. Das menschliche Handeln enthält einen unbewussten, reaktiven Anteil, der beim zieleorientierten Vorgehen unberücksichtigt bleiben könnte. Bei klar vorgegebenen Gestaltungszielen, Problemen und Systemen kann daher die Nutzerzentrierung die bessere Wahl sein. Dagegen zentriert man sich bei unklaren Anforderungen oder holistischen Gestaltungen eher auf den Menschen. Damit kann der Systemingenieur wichtige, aber vorher nicht definierte Probleme entdecken, neue Ziele setzen und somit zu innovativen Systemlösungen kommen.

Methoden des Usability Engineering lassen sich in nutzer- und menschenzentrierte aufteilen. Dabei sind zwei wichtige Grundsätze zu beachten. 1.) Da der Nutzer meistens ein Mensch ist, können sich die beiden Perspektiven stark überschneiden. 2.) Bei einer operativen Gestaltung von CS dürfen sich diese Perspektiven nicht ausschließen. Vielmehr obliegt es dem Systemingenieur zu entscheiden, bei welchen Teilproblemen er sich eher auf den Nutzungs- oder auf

den Menschenaspekt konzentrieren möchte. Einige gängige und nützliche Ansätze sind in der Tabelle 2-3 zusammengestellt und werden in den nächsten Abschnitten jeweils kurz erläutert.

Tabelle 2-3: Nutzer- und menschenzentrierte Gestaltungsansätze innerhalb des Usability Engineering und des CSE

Zentrierung	nutzerzentriert	menschenzentriert
Beobachten	Aus der Ferne: Nutzertagebücher, Online-Befragungen Teilnehmend: Strukturierte Interviews, lautes Denken, Rapid Prototyping, Card-Sorting etc.	Direkt vor Ort: Am Arbeitsplatz, in der sozialen Umgebung Teilnehmend: Erwartungsabfragen, Kreativitätstechniken mit Fokusgruppen, wie Brainstorming
Messen	Nutzungsbezogene Daten: Effizienz und Qualität der Systembedienung und -Rückmeldung	Menschbezogene Daten: Physiologische Daten, Blickrichtungsmessung, kognitive Metriken
Semiformale Analysen	Nutzungsprozesse: Zielanalyse, wie GOMS (Goals, Operators, Methods and Selection rules), Aufgabenanalyse, wie HTA (Hierarchical Task Analysis)	Kognitive Prozesse: Funktionsanalyse, wie CFA (Cognitive Function Analysis), Arbeitsanalyse, wie CWA (Cognitive Work Analysis)
Heuristiken	Nutzungsrelevant: Benutzerfreundlichkeit, fehlerfreie Bedienung, Zufriedenstellung etc.	Menschrelevant: Situationsbewusstsein, Belastung, Beanspruchung, Human Factors allgemein

Grundsätzlich hat sich beim Usability Engineering der Ansatz des sogenannten „Discount Usability Engineering“ bewährt: „we don't aim at perfection; we just want to find most of the usability problems“⁹[Nin93]. Usability Engineering selbst sollte innerhalb einer Systemgestaltung nützlich sein, aber nicht die Projektkosten oder sonstigen Ressourcenaufwand unnötig in die Höhe treiben. Dies kann z.B. durch schwächere Signifikanzkriterien bei statistischen Auswertungen realisiert werden. So können bereits ab 5-6 Versuchspersonen bis zu 80% der Usability-Probleme aufgedeckt werden^[Nin93]. Ebenso zweckorientiert kann das Usability Engineering bei einer Systemgestaltung unter Verwendung von osGekoS eingesetzt werden.

2.3.2 Nutzerzentrierte Methoden

Nutzerzentrierte Methoden des Usability Engineering sind durch vorher feststehende Ziele und Zweck bestimmt. Der Systemingenieur hat bereits konkrete Ideen für seine Gestaltung oder gar ihre prototypische Umsetzung. Diese möchte er anhand eines oder mehrerer Szenarien mit Nutzern testen. Die Erstellung der Szenarien kann durch Verwendung von Use-Case Diagrammen des UML^[OMG11] unterstützt werden. Damit können Nutzungsszenarien formuliert werden, worin alle Nutzungsfälle des Systems abgebildet sind. Solch ein Diagramm ist eine weitere Perspektive auf das Systemmodell und kann zur systematischen Evaluation zwecks Erreichens einer guten Usability mitverwendet werden. Nutzerzentrierte Methoden folgen der Leitfrage: „Wie kann das System gestaltet werden, damit es am besten nutzbar ist?“

Nutzerzentrierte Methoden können während einer Nutzerbeobachtung angewandt werden. Der Systemingenieur beobachtet die Nutzer seines Systems aus der Ferne, wenn er sie z.B. Nutzertagebücher oder Online-Befragungen ausfüllen lässt. Er kann auch die teilnehmenden Beobachtungen durchführen. Dies sind die klassischen Usability Studien. Der Systemingenieur kann vorher vorbereitete strukturierte Interviews durchführen. Er kann die Nutzer „laut denken“ lassen, um „naive“ Erstreaktionen auf sein System zu erfassen. Mit einer Rapid-

⁹ Wir streben nicht nach Perfektion, wir wollen einfach nur die meisten Usability-Probleme finden. EÜ.

Prototyping Methode kann er mit den Nutzern schnell mehrere Varianten seines Systems testen. Er kann die Card-Sorting Methode anwenden, wobei die verstandenen Bedeutungen und die tatsächlichen Ausprägungen der Interaktionsvarianten auf Karten abgebildet und vom Nutzer sortiert werden. Damit kann die Usability des Systems insgesamt exploriert werden.

Alle Nutzer können generell in naive und erfahrene aufgeteilt werden. Naive Nutzer eignen sich dafür, unerwartete Systemfehler aufzudecken. Mit Experten kann dagegen eine systematische heuristische Evaluation^[Nin94] durchgeführt werden. Dafür definiert der Systemingenieur vorher sogenannte Heuristiken, welche eine Art Bewertungskriterien für sein System darstellen. Anhand solcher Heuristiken bewerten die Experten das dargebotene System.

Neben subjektiven Daten aus Fragebögen, z.B. ob es jederzeit klar ist, in welchem Modus sich das System befindet (mode awareness), können während der Usability Studien auch objektive Daten aufgezeichnet werden. Speziell bei der heuristischen Evaluation sind die Häufigkeit, die Auswirkung und die Beständigkeit der auftretenden Konflikte wichtig^[Nin94]. Zusätzlich kann die auf das Szenario bezogene Leistung der Nutzer aufgezeichnet werden. Im Kontext des hochautomatisierten Fahrens können dies die Güte der Geschwindigkeits- und der Spurhaltung, Reaktionszeiten in Bezug auf die Systembedienung und Kontrollübernahme sein.

Beispiele für semiformale nutzerzentrierte Methoden sind GOMS^{10[CaM83]} und ihre Derivate. Damit zerlegt der Systemingenieur die Nutzerinteraktion mit seinem fertigen oder noch theoretisch modellierten System in elementare Ziele. Solche Ziele können in der Physis, in der Wahrnehmung oder in der Kognition des Nutzers ihren Ursprung haben. Auf diese Weise wird die mögliche Aktivität des Systemnutzers erfasst und analysiert. Daneben sind auch Aktionen wichtig, die der Nutzer überhaupt tätigen darf. Diese sind vom System vorgegeben und heißen Operatoren. Die GOMS-Analyse zielt darauf ab, die elementaren Ziele und Operatoren als Ablaufketten zu identifizieren. Diese Ketten werden Methoden genannt. Des Weiteren sind in GOMS die Auswahlregeln wichtig, wann der Nutzer welche Methode verwendet. Am Ende einer GOMS kommt ein Graph heraus, in dem mehrere Aktionsketten mit ihren Auswahlregeln in Richtung der angestrebten Zielzustände des Systems abgebildet sind.

Eine zu GOMS ähnliche Methode ist die hierarchische Aufgabenanalyse (HTA)^[Bad93]. Dabei werden umfassendere Aufgabenpakete als bei GOMS identifiziert, hierarchisch geordnet und sortiert. Die miteinander in Verbindung (z.B. über die Wahlreihenfolge) stehenden Aufgaben gehören zum selben Plan. Am Ende einer HTA kommt ein Graph heraus, in dem mehrere Aufgaben zu Plänen zusammengefasst und auf unterschiedliche Hierarchieebenen des Graphs verteilt sind. Sowohl die GOMS- als auch die HTA-Graphen kann man anschließend für eine Systemverbesserung oder eine weitere Systemanalyse verwenden. Kritische Pfade und sonstige Hinweise auf mögliche Probleme oder Lösungen können damit identifiziert werden.

Bei einer Systemgestaltung mittels osGekoS können nutzerzentrierte Methoden ebenfalls verwendet werden. So wurden z.B. die wesentlichen Ergebnisse der später vorgestellten Usability Studien für die mit osGekoS gestalteten Fahrerassistenzsysteme mittels strukturierter Interviews während der teilnehmenden Beobachtungen erarbeitet. Ein HTA-Graph wurde verwendet, um typische Fahraufgaben bei einem Fahrstreifenwechsel zu identifizieren.

¹⁰ Goals, operators, methods and selection rules (dt. Ziele, Operatoren, Methoden und Auswahlregeln)

2.3.3 Menschzentrierte Methoden

Mittels menschenzentrierter Methoden wird bei einer Systemgestaltung versucht, den Menschen in seinem möglichst natürlichen Auftreten und seiner Umgebung zu berücksichtigen. Wahrnehmungs- und sonstige kognitive Prozesse der menschlichen Informationsverarbeitung sowie das soziale Umfeld des Menschen spielen dabei eine wichtige Rolle. Menschzentrierte Methoden werden von den Leitfragen bestimmt: „Was kann der Mensch innerhalb seiner natürlichen Umgebung tun, wie macht er das und was braucht er dafür?“

Menschenzentrierte Methoden kommen besonders zur Geltung, wenn der Mensch in seiner gewohnten Umgebung beobachtet wird. Dies kann direkt an seinem Arbeitsplatz oder in seinem sonstigen sozialen Umfeld geschehen. Im Labor können mithilfe von Fokusgruppen Erwartungsabfragen und Kreativitätstechniken, wie Brainstorming^[W13a], Mind-Mapping^[W13c] oder Concept-Mapping (vgl. Anhang A) angewandt werden. Damit können der tatsächliche Bedarf an Automation und ihre Ausprägung exploriert werden. Auch im Kontext des hochautomatisierten Fahrens können menschenbezogene physiologische Daten, wie Puls, Blutdruck, EEG, Blickrichtung etc. gemessen werden. Dadurch entsteht eine Grundlage, um die Kognition berücksichtigende Rückschlüsse für eine Systemgestaltung zu ziehen.

Eine nützliche semiformale menschenzentrierte Methode ist die kognitive Funktionsanalyse (CFA)^[Boy98]. Eine kognitive Funktion (CF) wird als eine Art „Rechenvorschrift“ verstanden, die eine (gestellte) Aufgabe in eine (tatsächliche) Aktivität umwandelt. Während einer CFA können sowohl Experten als auch naive Versuchspersonen entweder durch ihre Äußerungen oder gemessene Aktivität zur Identifikation der CFs beitragen. Es wird iterativ anhand mehrerer Szenarien vorgegangen. Eine CF ist erst dann eindeutig identifiziert, wenn sich eine situative Persistenz zeigt, z.B. mehrere Experten äußern Ähnliches in ähnlichen Situationen. Beispiele der CFs sind: „Entdecke jede Konfliktsituation auf der gewählten Bewegungstrajektorie“ oder „Ändere Bewegungstrajektorie, um Konflikte zu vermeiden.“^[Boy00] Bei der Identifikation von CFs ist es wichtig, wie sie sich auf die Agenten innerhalb eines CS verteilen, welche Rollen existieren und welche kognitiven und physischen Ressourcen den Agenten zur Verfügung stehen. Am Ende einer CFA kommt ein Graph heraus, der die kognitive Komplexität eines CS abbildet und zu den funktionalen Anforderungen eines CS beiträgt.

Eine große Familie semiformalen menschenzentrierter Methoden ist die kognitive Arbeitsanalyse (CWA) und ihre Teilmethoden^{[RaP94][Vie99][Lin13]}. Diese werden meist vor einer tatsächlichen Systemgestaltung mit dem Ziel angewandt, Empfehlungen für eine nachgelagerte Gestaltung zu generieren. Im Unterschied zu der Nutzerzentrierung steht bei der CWA nicht die Nutzung eines konkreten Systems im Vordergrund, sondern der Mensch, sein Umfeld und die Arbeit, die der Mensch zu verrichten hat. Die CWA läuft in fünf Phasen ab: Arbeitsdomänen-, Aktivitäts-, Strategien-, Organisations- und Kompetenzanalyse^{[Vie99][Lin13]}. Auch kognitive Aufgaben und Entscheidungen im Kontext der durchzuführenden Arbeit können analysiert werden^[RaP94]. Am Ende der CWA kommen Empfehlungen und diverse Graphen heraus.

Ein nützlicher Graph aus der CWA ist der Abstraktions-Zerlegungsraum (engl. abstraction-decomposition space)^[Lin13]. Darin werden gleichzeitig eine funktionale und eine grobe strukturelle Systemdekomposition abgebildet. Diese können danach in funktionale Anforderungen für die Gestaltung struktureller Komponenten übersetzt werden. Der Graph entsteht durch

hierarchische und immer konkreter werdende Definitionen von Systemzweck, Systemwerten, Systemfunktionen, technischen Funktionen und physischen Ressourcen auf der vertikalen Dimension. Auf der horizontalen Dimension werden hierarchisch das System, Systemmodule, Komponenten und Teile eingetragen. Man konfiguriert darin gegenseitige Abhängigkeiten, kann abstrakte Konflikte feststellen und durch eine Rekonfiguration auf dieser abstrakten Ebene lösen. Diese Methode kann zusammen mit osGekoS verwendet werden.

2.3.4 Kognitive Architekturen

Im Bereich menschnaher Systemgestaltung findet man kognitive Architekturen. Als Beispiele können SOAR^[LeL06], ACT-R^[AnM97], BDI^[RaG91], CHREST^[GoL01], CLARION^[Sun03] und COSA^[Pu003] genannt werden. Viele dieser Architekturen sind entweder auf einzelne Aspekte einer Kognition, wie Gedächtnis, Lernprozesse und ihre Wechselwirkungen oder speziell auf die menschliche Kognition zugeschnitten. Wenige Architekturen, wie COSA, betrachten CS als Ganzes.

Die bekannten kognitiven Architekturen werden meist zwei differenzierbaren Konzepten zugeordnet. Sie können „symbolisch“ sein, wie SOAR, ACT-R, BDI, COSA. Das bedeutet, dass in solchen Architekturen eher mit Symbolen und deren Manipulation als Wissens- und Prozessrepräsentation gearbeitet wird. Kognitive Architekturen können auch „konnektivistisch“, wie CHREST, sein. In solchen wird eher mit Verknüpfungen zwischen den Systemelementen als Hauptstruktur gearbeitet. CLARION ist ein Beispiel, das beide Konzepte vereinigt.

Insgesamt kann man kognitive Architekturen als relativ komplexe Rahmen im Bereich der CS-Gestaltung verstehen. Sie sind manchmal in konkreten Anwendungen, wie für eine Computersimulation von einzelnen kognitiven Prozessen, oder in sehr spezialisierten Umgebungen entwickelt worden. Dadurch tragen sie oft Artefakte der Entwicklungsumgebung sowie verhaltenssteuernde und strukturelle Annahmen für das, was zum Zeitpunkt und Zweck des jeweiligen Architektorentwurfs aktuell war, unter einer Kognition zu verstehen. Das können festgelegte Regelwerke, wie Lernmechanismen, oder vorgeschriebene Systemelemente mit Verknüpfungen dazwischen sein, wie Langzeit-Kurzzeitgedächtnis. Dies kann sie in der operativen CS-Gestaltung oder einer neuen Umgebung schwerfällig machen.

Darum wird in osGekoS anders vorgegangen, kognitive Systeme zu verstehen und zu beschreiben. Ohne eine neue komplexe kognitive Architektur zu entwickeln, wird versucht, ein leicht nachvollziehbares und skalierbares Konzept einer verteilten Kognition zu erarbeiten.

2.3.5 Menschnahe Gestaltungsleitlinien

Im Bereich der CSE existieren diverse menschnahe Gestaltungsleitlinien und -Heuristiken, die bei der operativen Gestaltung der CS ihren Nutzen entfalten. Diese entstanden nach und nach aus vielen Unfallanalysen, Usability Studien und sonstigen empirischen Untersuchungen und Erfahrungen der Experten mit der CS-Gestaltung. So wird bei einer menschnahen Perspektive beispielsweise vorgeschlagen, eine Automation im Kontext eines CS als einen Teamplayer aufzubauen^[ChW02]. Eine solche Automation muss immer eine Beobachtbarkeit (eng. observability) und eine Steuerbarkeit (eng. directability) vorweisen. Damit ist gemeint, dass der Nutzer solcher Automation das Verhalten und die inneren Zustände seiner Automation beobachten können und seine Automation jederzeit steuern können sollte.

Die Leitkriterien einer guten Usability sind im Abschnitt 2.3.1 bereits zusammengefasst. Um diese Kriterien befolgen zu können, wurden weitere Heuristiken formuliert. Dies sind bestimmte auf Erfahrung basierte Regeln, z.B. „das Systemverhalten muss dem Verhalten der realen Welt entsprechen“, „das System muss über sein Status informieren“, „Verstehen ist besser als Erinnern“, „keine überflüssige Informationen“^[Nin94] etc.

Aus der Theorie zum menschlichen Situationsbewusstsein kommen einige Empfehlungen, wie „Informationen direkt fürs Verstehen (engl. comprehension) darstellen“, „das Projizieren und globales Situationsbewusstsein unterstützen“, „schema-aktivierendes Feedback sichtbar machen“, „parallele Informationsverarbeitung nutzen“ oder „Information nur bedacht filtern“^[Eny03]. Für den Automotivbereich gibt es ebenfalls diverse menschnahe Empfehlungen, wie „den Einfluss der Geschwindigkeit auf die Wahrnehmung berücksichtigen“, „den Fahrer nicht ablenken“, „nicht mehrere parallele Aufgaben überwachen lassen“^[Cao09] etc.

Gestaltungsrelevante Empfehlungen findet man auch im Themengebiet „menschliche Fehler“ (engl. human errors). Dort lernt man z.B., dass Fehler nicht von einzelnen CAs produziert werden, sondern immer von dem Gesamtsystem^[Der03]. Man bekommt Hinweise^[Ra86b] darauf, dass Fehler aufgrund des Lernens und der Adaptierung, durch eine Konkurrenz in Kontrollsituationen, mangels Ressourcen oder durch innere menschliche Variabilität entstehen können.

2.3.6 Menschnahe Gestaltungsmuster

Im Bereich des CSE finden sich diverse menschnahe Gestaltungsmuster. Ähnlich den techniknahen Mustern (vgl. Abschnitt 2.2.6) können sie als Problem/Lösungs-Paare während der operativen Gestaltung von CS eingesetzt werden.

Image Schemata^[Tay83] sind Muster einfacher mentaler Modelle und Prozesse, die durch die Analyse menschlicher Wahrnehmung und Sprache abgeleitet werden. „Image-schemas are notions such as PATH, UP-DOWN, CONTAINMENT, FORCE, PART-WHOLE, and LINK, notions that are thought to be derived from perceptual structure“^{11 [Mar92]}. Solche Muster können als angeborene oder sehr früh erlernte Lösungskonzepte für einfache Probleme innerhalb menschlicher kognitiver Informationsverarbeitung verstanden werden. Diese Konzepte lassen sich bei der Gestaltung von CS gewinnbringend verwenden. Wenn der Systemingenieur ein sich bewegendes System, wie ein Fahrzeug, entwickelt, so könnte er das Image Schema „PATH“ bei der Gestaltung einer pfadfolgenden Automation einsetzen. Das später eingeführte Gestaltungsmuster „Kooperationsdyade“ verwendet das Image-Schema „LINK“.

Design-Metaphern enthalten gleich mehrere Muster bestimmter Problem/Lösungs-Paare und können künstlichen oder natürlichen Ursprungs sein. Eine bekannte künstliche Design-Metapher ist die Desktop-Metapher für die Gestaltung der Interaktion des Menschen mit einem Computer. Für die bessere Verständlichkeit der Mensch/Computer-Interaktion wurde als Vorbild ein künstliches Objekt, der Tisch, gewählt. Die visuelle Interaktion mit einem Computerbildschirm gestattet, darauf virtuelle Dokumente, Ordner abzulegen, zu verschieben etc. Ein Beispiel einer natürlichen Metapher für die Gestaltung hochautomatisierter Fahrzeuge ist

¹¹ Image-Schemata sind Begriffe wie PFAD, OBEN-UNTEN, BEGRENZUNG, KRAFT, TEIL-GANZES und VERKNÜPFUNG, Begriffe, die aus der Wahrnehmungsstruktur abgeleitet werden können. EÜ.

die H-Metapher^[Flh00]. Dabei wurde das System Reiter/Pferd als metaphorisches Vorbild für die Gestaltung von Fahrer/Fahrzeug-Interaktion gewählt.

Bei der Gestaltung von komplexen CS kann eine richtige Design-Metapher sehr hilfreich sein. Sie erfüllt eine doppelte Rolle. Zum einen versorgt sie den Systemingenieur mit anregenden Ideen und Konzepten für seine Gestaltung und fördert dabei die Ganzheitlichkeit des Systems. Zum anderen hilft sie dem späteren Nutzer beim Erlernen und Bedienen des Systems, weil dabei das Wissen des Nutzers über die Metapher ausgenutzt wird. Damit kann die Komplexität des Gesamtsystems reduziert werden und das gestaltete System kann eine holistische Form und Verhalten annehmen. Auch das später ausführlich vorgestellte Gestaltungskonzept der Aktionstension^[KeH12] des osGekoS kann als Design-Metapher dienen.

2.4 Konsistente Interaktionsgestaltung

Bei der operativen CS-Gestaltung sind die Interaktionssignale zwischen den CAs wichtig. Dadurch gewinnt der Systemingenieur Einfluss auf das CS- und CA-Verhalten. Um diesen Einfluss effektiv ausüben zu können, muss der Systemingenieur den CAs in Abhängigkeit von möglichen Situationen etwas mitteilen und die CAs müssen diese Mitteilung verstehen. „Aufgabe des Interaktionsdesigns ist es, Benutzer zu befähigen, Funktionalität und Information zu erkennen, zu dekodieren, zu verstehen und daraufhin geeignet zu kommunizieren und zu handeln“^[Heg06]. Grundlagen der Linguistik, der Wahrnehmungspsychologie, der Ergonomie und der Psychophysik sind daher sehr wichtig für die CS- Gestaltung von CS und für osGekoS.

Das Wichtigste an dieser Stelle ist, dass Sprachen, so auch eine künstliche Interaktionssprache, über ihre Form eine Bedeutung im Kontext ihres Gebrauchs vermittelt. D.h. jedes Interaktionssignal und jede Signalabfolge müssen zunächst eine gewisse Form und eine dazu gehörige Bedeutung besitzen. Es ist auch wichtig zu wissen, warum, wie, wann und von wem diese Signale kommen und wer sie empfängt und interpretiert.

In der Linguistik gibt es auf der Formseite die Phonetik und die Phonologie. Darin werden jeweils die Art und der Aufbau formtragender Einheiten (Phone, Phoneme, Silben) und ihre möglichen Kombinationen untersucht. Auf der Bedeutungsseite gibt es die Morphologie, die Semantik und die Syntax. Darin untersucht man die Art und den Aufbau bedeutungstragender Einheiten (Morphe, Morpheme, Wörter, Phrasen, Sätze), ihre Bedeutungen und mögliche Kombinationen. Die Pragmatik behandelt den Aspekt des Gebrauchs einer Sprache und das sprachliche Handeln^[Sla11]. Bei einer Interaktionsgestaltung muss ein Mapping zwischen der gestalteten Signalform und der möglichst allgemeingültigen Bedeutung des Signals erstellt werden. Linguistik und ihre Teildisziplinen bieten eine solide und erprobte Grundlage dafür.

Um ein solches Mapping erfolgreich zu entwickeln, muss eine weitere Grundlage beachtet werden. Neben den unbewussten bzw. affektiven wird das bewusste Verarbeiten der Interaktionssignale, wie jede Art der kognitiven Aktivität, mittels Symbole und Zeichen vollzogen^{[Nel80][Ra86a]}. Dies hat zur Folge, dass die zu verarbeitenden Interaktionssignale zwangsläufig Symbole und Zeichen bilden, d.h. sie werden vom Nutzer interpretiert. Beim Entwurf der Interaktionssignale gestaltet der Systemingenieur größtenteils also Symbole und Zeichen.

Hier muss die Entstehung eines Zeichens (Semiose) beachtet werden. Diese Entstehung wird in der Zeichentheorie (Semiotik) als eine nicht trennbare Einheit zwischen der syntaktischen, der semantischen und der pragmatischen Dimension eines Zeichens verstanden^[Mos75]. Das bedeutet, dass die Form, die Bedeutung und der Nutzer eines Zeichens in einer komplexen Beziehung zueinander stehen. Diese Theorie hat eine direkte Auswirkung auf die später eingeführte osGekoS-Methode „Semantisches Mapping“, die eine Gestaltung der Form und der Bedeutung eines Interaktionssignals in der Zusammenarbeit mit dem Systemnutzer vorschlägt.

Tabelle 2-4: Vergleich zwischen menschlichen und animalischen Ausdrucksformen

Dimension	menschlich	animalisch
Dualität	Menschlicher Sprechapparat und Kognition sind fähig, sehr viele Laute und ihre semantisch sinnvolle Kombinationen zu produzieren. Zum Verständnis ist sehr viel Kontextwissen erforderlich	Tiere können meistens nur wenige unterschiedliche Laute produzieren mit sehr begrenzten semantisch trennbaren Kombinationsmöglichkeiten
Willkür	Keine eindeutige Verbindung zwischen der Sprachform und der Semantik. Komplexe Wortkonstruktionen mit Mehrdeutigkeiten sind üblich. Es gibt gleichklingende Wörter mit absolut unterschiedlicher Semantik, Betonungsnuancen	Ausdruckweisen der Tiere sind viel mehr determiniert. So entsprechen bestimmte Laute, Bewegungsmuster, Geruchssignale etc. oder ihre Kombinationen oft dem gleichen situativen oder kognitiven Kontext
Reizabhängigkeit	Menschen können unabhängig von dem Zustand aktuell vorhandener Reize kommunizieren	Tiere kommunizieren meistens in direkter Abhängigkeit von vorhandenen Reizen
Deplatziierung	Menschen können indirekt über Orte, Zeiten, Situationen sprechen, in den sie im Moment nicht da oder überhaupt nie gewesen sind	Tiere können zwar Richtungsangaben und nicht sofortige Ereignisse kommunizieren, ihre Ausdrücke sind jedoch sehr orts- und zeitgebunden
Parallelität	Menschen können parallele und stark verschachtelte Argumentationsketten aufbauen	Tiere kommunizieren überwiegend in seriellen Abfolgen von Signalen

Sprache ist ein mächtiges Werkzeug menschlicher Kognition. Allerdings kann die menschliche Sprache sehr kompliziert in ihrer Form, Bedeutung und Gebrauch sein, um sie als alleiniges Vorbild für eine konsistente Interaktionsgestaltung zu nutzen. Kommunikation und sprachähnliche Konstrukte des Tierreichs können für eine konsistente Interaktionsgestaltung ebenfalls wertvoll sein. In der Tabelle 2-4 ist ein Vergleich zwischen den menschlichen und animalischen Kommunikationsformen zusammengestellt^[Trk99]. Sie zeigt, dass animalische Ausdrucksformen zwar nicht so vielfältig und dynamisch sind, wie die menschlichen, dafür sind sie mehr eindeutig, zeitlich und örtlich determiniert. Solche Eigenschaften sind auch für eine konsistente und intuitiv verständliche Interaktionsgestaltung wünschenswert. Darum gelten in osGekoS die animalischen Ausdrucksformen als Vorbild einer konsistenten Interaktionsgestaltung.

Neben den linguistischen Nuancen müssen bei der Gestaltung konsistenter Interaktion weitere Aspekte beachtet werden. Es gibt eine große Anzahl an physiologischen Besonderheiten des menschlichen sensomotorischen Apparates, die sich auf die Verarbeitung von Interaktionssignalen auswirken können. Es ist nicht das Ziel dieses Abschnitts, sie alle aufzuzählen. Hier werden nur die zwei wichtigsten Punkte für die Gestaltung konsistenter Interaktionen kurz unterstrichen: Multimodale Wahrnehmungs- und Koinzidenzschwellen^{[Gon97][Sce93]}.

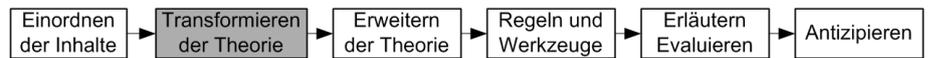
Alle Sinneswahrnehmungen haben ihre eigenen Wahrnehmungs- und Koinzidenzschwellen in Abhängigkeit von der Amplitude des Reizes, seiner Frequenz, Angriffsfläche und Verlaufes.

Diese bestimmen, ob ein Reiz festgestellt wird, ob ein Unterschied zwischen zwei ähnlichen Reizen festgestellt wird, ob zwei zeitlich versetzte Reize als zwei Ereignisse (bzw. Manipulationen) anstatt von einem festgestellt werden und ob, wie und wie schnell auf die Reize reagiert wird. Im Bereich der Psychophysik werden diese Schwellen systematisch untersucht. Neben der Semiose aus Form, Bedeutung und Gebrauchskontext eines Interaktionssignals und der animalischen Kommunikationsformen als Vorbild für die Signalgestaltung müssen diese Randbedingungen bei der Gestaltung konsistenter Interaktionsstrategien beachtet werden. Dies gilt auch bei Anwendung von osGekoS.

2.5 Zusammenfassung

In diesem Kapitel wurde ein Überblick über die Landschaft bereits vorhandener wichtigster Rahmen und Methoden für die CS-Gestaltung gegeben. Dies wurde mit technikhnen Methoden des Systems Engineering begonnen und mit menschnahen Methoden beendet. Alle aufgeführten Methoden können bei Anwendung von osGekoS mitberücksichtigt werden. Des Weiteren wurden die wichtigsten linguistischen Grundlagen für eine konsistente Interaktionsgestaltung präsentiert. Bei der Gestaltung eines Interaktionssignals und seiner Bedeutung sollte der Prozess der Semiose und die Psychophysik beachtet werden. Dabei eignet sich die animalische Art zu kommunizieren gut als Vorbild für eine konsistente Interaktionsgestaltung.

Wie sich feststellen lässt, gehen die vorhandenen Gestaltungsrahmen und Methoden eher auf den Gestaltungsprozess, die Gestaltung des Systemverhaltens und der Funktionen, aber weniger auf die (kognitiven) Strukturen und Organisation ein. Die operative Gestaltung und Implementierung eines CS ist damit nur indirekt möglich. OsGekoS wurde in diese Landschaft als operativer strukturfokussierter Gestaltungsrahmen eingeordnet, was als eine sinnvolle Erweiterung der vorhandenen Rahmen verstanden wird. Im nächsten Kapitel werden die Grundlagen und Konzepte erarbeitet, um CS einfach verstehen und beschreiben zu können. Dafür wird das bereits vorhandene relevante Wissen für die Aufstellung der osGekoS-Theorie präsentiert und mittels zusätzlicher Annahmen transformiert und homogenisiert.



3 Verstehen und Beschreiben kognitiver Systeme

Das Ziel dieses Kapitels ist, strukturelle Ähnlichkeiten der CS zu identifizieren, zu homogenisieren, zueinander in eine eindeutige Beziehung zu setzen und konzeptionell, grafisch und mathematisch für osGekoS aufzubereiten. Dafür werden alle für das Verstehen und das Beschreiben von CS notwendigen Grundlagen eingeführt. Bei der Einführung werden sie als Teil der Eigenleistung ausführlich kommentiert und schrittweise zu der osGekoS-Theorie transformiert. Das passiert mithilfe von fortlaufend entwickelten Annahmen, Schlussfolgerungen, Abbildungen und Formeln. Sie dienen dem Systemingenieur beim holistischen Verständnis und der Beschreibung seines zu gestaltenden CS mit Ideen und Methodenansätzen und dem Autor als Grundlage für die Erarbeitung der Konzepte holistischer CS-Gestaltung in Kapitel 4.

Die Homogenisierung und die Transformation der Grundlagen laufen schrittweise ab. Als erstes werden CS als kognitive MAS dargestellt (Abschnitt 3.1). Mit der Homogenisierung der Grundlagen zu Kommunikationsstrukturen der CS (Abschnitt 3.2) werden kognitive MAS weiter zum MPS-Konzept (Manipulations-Potenzial Systeme) transformiert (Abschnitte 3.3, 3.4) und formalisiert (Abschnitt 3.5). MPS wird als ein osGekoS-eigenes Beschreibungsmittel für CS erarbeitet und in Abschnitt 3.6 zu den bereits vorhandenen kurz positioniert. Am Ende dieses Kapitels werden neu gewonnene Einsichten der osGekoS-Theorie zusammengefasst.

Zur Veranschaulichung wird in der Arbeit ein praxisnahes, aber dennoch generisches und, besonders für die Automotivdomäne, repräsentatives Beispiel verwendet. Es handelt von einem Kontrollkonflikt und einem Entscheidungsvorgang bei der zwischen einem Fahrer und einer Automation geteilten Fahrzeugkontrolle (shared control)^[Shv78]. Das Fahrer/Automation /Fahrzeug/Umwelt-System, der Kontrollkonflikt, der Entscheidungsvorgang und die Lösung des Konfliktes sollen stellvertretend und allgemein die CS und darin vorkommende Vorgänge repräsentieren^[Keh14]. Im Text wird auf das Beispiel mit dem Merker „Y-Beispiel“ verwiesen.

Für das Beispiel stelle man sich folgende Situation vor: Ein Fahrer fährt in einem hochautomatisierten Fahrzeug auf eine Weggabelung zu. Er und seine Automation im Fahrzeug haben unterschiedliche Absichten über die weitere Fahrtrichtung. Der Fahrer möchte nach links abbiegen und die Automation nach rechts. Der Fahrer lenkt nach links, während die Automation nach rechts lenkt. Oder der Fahrer ist abgelenkt, nicht auf das Lenken vorbereitet und lenkt dementsprechend gar nicht. Fährt das Fahrzeug nach links oder nach rechts (Abbildung 3-1)?

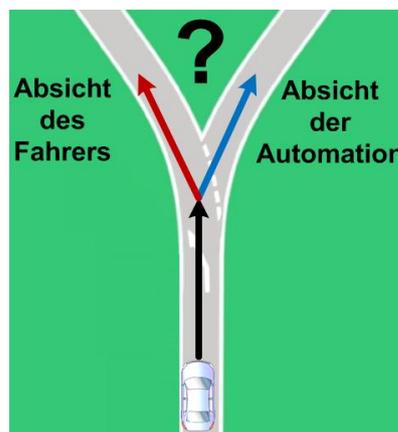


Abbildung 3-1: Beispiel eines Entscheidungsvorgangs in einem generischen kognitiven System

Diese Frage ist schwer zu beantworten, da das Beispiel zwar ein relativ einfaches aber dennoch typisches kognitives System beschreibt. In einem hochautomatisierten Fahrzeug vorhandene kognitive Elemente, wie Fahrer und komplexe Automation, können unterschiedliche Motive, Entscheidungsstrategien, Verhaltensmodelle haben, obwohl sie sich in derselben Verkehrssituation befinden. Sie stehen in einer komplexen Wechselwirkung untereinander und mit der sie umschließenden nahen Umwelt. Um solche relevanten Gestaltungsfragen zu beantworten, muss der Systemingenieur kognitive Systeme verstehen und beschreiben können.

3.1 Kognitive Multiagentensysteme

MAS können nützlich für eine CS-Gestaltung sein (vgl. Abschnitt 2.2.4). Kognitive Strukturen von Menschen und Maschinen bzw. Fahrer und Automation können damit einheitlich verstanden, beschrieben und das MAS-Konzept zum Konzept kognitiver MAS erweitert werden^[Keh14]. Die Erweiterung findet in drei Schritten statt: 1.) Einige Konzepte der Kognitionswissenschaft werden für eine Verknüpfung zwischen Menschen, Maschinen und MAS analysiert. 2.) Die Definitionsvielfalt im Umfeld von CS und MAS wird für kognitive MAS reduziert und homogenisiert. 3.) CS werden als kognitive MAS eingeleitet und mit Beispielen erläutert.

3.1.1 Verknüpfung zwischen Menschen, Maschinen und MAS

Ein Agent eines MAS wurde als eine zielgerichtet agierende Einheit mit verschiedenen möglichen Informationszuständen innerhalb seiner komplexen Strukturen und eine Kognition als „Fähigkeit zur Wahrnehmung und zielgerichteten Interpretation der Lebenswelt“ definiert. Diese Definitionen können sowohl auf Menschen als auch auf komplexe Maschinen zutreffen. Denn beide können in verschiedenen (kognitiven) Zuständen verweilen, diese durch die Wahrnehmung und -interpretation wechseln und dabei ein zielgerichtetes Verhalten zeigen.

Aus Kognitionswissenschaft sind zwei sich ergänzende Konzepte der Entstehung und des Aufbaus der menschlichen Kognition bekannt. In der Entwicklungspsychologie z.B., wird die Entstehung eines Individuums als ein sozialer Prozess definiert. Akquisition und Vermehrung des proximalen Wissens, Regeln und Fertigkeiten wird durch seine Wechselwirkung mit der Umwelt erklärt (Mind in Society^[Vyy78]). Durch Interaktion mit anderen Individuen, wie Lehrern, und Artefakten der Umwelt, wie Symbolen, Sprachen, Werkzeugen, kann das Individuum sich herausbilden und bekommt seine geschlossene autonome Form. Dies geschieht, indem das distale Wissen internalisiert wird. „Mind in Society“ ist kompatibel zu MAS, da sie von multiplen autonomen kognitiven und nichtkognitiven Elementen und ihrer Wechselwirkung handelt. Eine dazu komplementäre Theorie (The Society of Mind^[Miy88]) kann so interpretiert werden, dass die menschliche Kognition mit ihrer Leistung in Form der Intelligenz kein monolithisches Konstrukt ist, sondern durch Wechselwirkung von kleineren kognitiven Elementen entsteht. „The power of intelligence stems from our vast diversity, not from any single, perfect principle“^{12[Miy88]}. Auf „The Society of Mind“ stützend, kann eine Kognition und ihre komplexen Strukturen als ein MAS bzw. eine Agentenpopulation verstanden und beschrieben werden.

Argumente beider Theorien unterstützen die Hypothese, dass mit dem MAS-Konzept es möglich sein sollte, komplexe CS-Strukturen zu verstehen und zu beschreiben. Dies kann mit

¹² Leistung der Intelligenz entstammt unserer enormen Vielfalt und nicht einem einheitlichen, perfekten Prinzip. EÜ.

Kleinsteinheiten innerhalb eines kognitiven Prozesses beginnen und bis zu komplexen verteilten CS fortgeführt werden. So können die einzelnen Wahrnehmungen des Menschen, wie haptische, akustische, visuelle, als Agenten verstanden werden. Für Maschinen können dazu passende technische Agenten gestaltet sein, die durch den Systemingenieur kontrollierte Wechselwirkung mit den beschriebenen Agenten des Menschen die gesamte Wahrnehmungs-, Verarbeitungs-, Entscheidungs- und Handlungsleistung eines Mensch/Maschine-Systems erhöhen können. In einem so gestalteten und implementierten CS ist eine Erhöhung der Gesamtleistung vorstellbar.

Zoomt man weiter auf die CS-Ebene heraus, kann die Wechselwirkung zwischen den einzelnen Menschen und ihnen assistierenden Maschinen untersucht werden. Beispielsweise können komplexe Automation und Fahrer als Agenten (bzw. Populationen) beschrieben werden, um Effekte der geteilten Kontrolle^[ShV78], der Belastung, der Beanspruchung, der Limitierung der einzelnen Agenten innerhalb der Fahrer/Automation/Fahrzeug/Umwelt-Systeme zu untersuchen. Ein weiteres Zoom heraus zeigt die theoretische Möglichkeit, noch komplexere CS als Populationen von Menschen und Maschinen zu untersuchen. Im Bereich hochautomatisierter Fahrzeuge könnten auf diese Weise ganze Verkehrsflüsse inklusive der sozialpsychologischen Effekte, wie Kooperation oder Konkurrenz, verstanden und beschrieben werden.

Zusammenfassend kann angenommen werden, dass sowohl Menschen als auch komplexe Maschinen einheitlich als Agenten eines MAS verstanden werden können. Eine einfache und pragmatische Definition dafür lautet: *Kognitive Agenten sind Agenten mit einer Kognition. Kognitive Systeme sind Multiagentensysteme mit mindestens einem kognitiven Agenten.*

3.1.2 Reduktion und Homogenisierung der Definitionsvielfalt

Oft werden CAs und CS deutlich granularer definiert^{[Cai98][Vav09][RaG91][ShL09][OMG00][BaM01][LuM05]}. Alle diese Definitionen zählen ähnliche Elemente eines CA auf, wie Überzeugungen, Wünsche, Ziele, Pläne, Absichten, *Aktionen* und eine besondere „kognitive Art“ im Umgang mit solchen Elementen. Beispielsweise kann ein CA ihm zugewiesene Aufgaben mittels kognitiver Funktionen^[Boy98] in *Aktionen* abbilden. Je nach Eigenschaft des jeweiligen Agenten kann er etwas lernen, sich adaptieren und/oder ein Situationsbewusstsein durch eine Situationsbewertung aufbauen^[Eny95]. Danach kann dieser CA Entscheidungen treffen und *Aktionen* ausführen. Kognitive Elemente, wie Überzeugungen, Wünsche, Ziele etc. können sowohl als Einflussfaktoren auf kognitive Zustände und Verhalten als auch Produkte und Informationsverbindungen zwischen den Zuständen zwecks Verhaltenszeugung interpretiert werden.

Bei der hier relevanten, auf die System- statt Agentenebene fokussierten Betrachtung der CS, können alle Transformationen der Information, wie Erschaffung, Modifikation, Weiterreichen der o.g. kognitiven Elemente als *Aktionen* zusammengefasst werden. Denn die Überzeugungen, Wünsche, Ziele äußern sich nur durch Aktionen in einer auf der Systemebene beobachtbaren und relevanten Art und Weise. Wenn CAs nicht handeln, beeinflussen sie nicht den CS-Zustand. Äußerungen auf die CS-Ebene, von Zielen, Plänen, Absichten sind nur durch einen aktiven Informationsfluss zu realisieren, was mit einer Aktion gleichzusetzen wäre^[KeT13].

Im Umfeld von MAS, auch im Bereich der hochautomatisierten Fahrzeuge, werden Agenten oft folgendermaßen klassifiziert^{[OMG00][BaM01][LuM05]}.

- *Reaktive Agenten* sind die einfachsten Agenten. Sie können auf ihre Umwelt durch ein einfaches determiniertes Regelwerk reagieren. Objekte aus der Umwelt, wie Hindernisse mit basalen Zuständen und Verhalten, aber auch die Fahrzeuge selbst durch ihre reaktive, fremdbestimmte Natur können zu diesem Agententyp gezählt werden.
- *Kognitive Agenten* sind solche, die eine interne Repräsentation der äußeren Umwelt aufbauen und verwalten können. Damit kann zur komplexen Entscheidungsfindung und zur nachgelagerten Aktionswahl beigetragen werden. Die Fahrer und die komplexen Assistenzsysteme bzw. Automation gehören zu diesem Agententyp.
- *Adaptive Agenten* können eine interne Repräsentation über sich selbst aufbauen und verwalten. Damit kann das eigene Verhalten der Agenten angepasst werden, z.B. durch die Sammlung von Erfahrung, das Lernen (Adaptivität, adaptiv) oder durch die Anpassung von außen (Adaptierbarkeit, adaptierbar). Fahrer und die komplexen Assistenzsysteme bzw. Automation können zu diesem Agententyp gehören.
- *Mobile Agenten* sind solche, die sich innerhalb ihrer geometrischen und/oder sozialen Umwelt bewegen können, z.B. ein herkömmliches Automobil, eine sich im Raum bewegende und/oder sich kooperativ verhaltende Automation oder ein Fahrer.

Wie die Beispiele zeigen, können bestimmte Agenten zu mehreren Kategorien gehören. So sind die Fahrer mobil, kognitiv und adaptiv, ihre Automationen sind mobil und können ebenfalls kognitiv und adaptiv gestaltet sein. Fahrzeuge sind grundsätzlich mobil und reaktiv. Die nahe Umwelt kann ebenfalls als ein reaktiver Agent verstanden und beschrieben werden. Wegen der möglichen Mehrdeutigkeit und der prinzipiellen Ähnlichkeit zwischen den kognitiven und adaptiven Agenten kann für kognitive MAS eine Vereinfachung getroffen werden. Diese besteht darin, dass nur *zwischen den kognitiven (CA) und nichtkognitiven (NA) Agenten unterschieden wird*^[Keh14]. Alle Agenten sind mobil. NAs sind reaktiv und CAs kognitiv und adaptiv. *Jegliche Ressourcen innerhalb eines CS sind ebenfalls als Agenten abbildbar.*

3.1.3 Kognitive Systeme als Multiagentensysteme

Jedes kognitive System kann als ein kognitives MAS verstanden und beschrieben werden. Kognitive MAS enthalten kognitive und nichtkognitive Agenten, die agieren und interagieren.

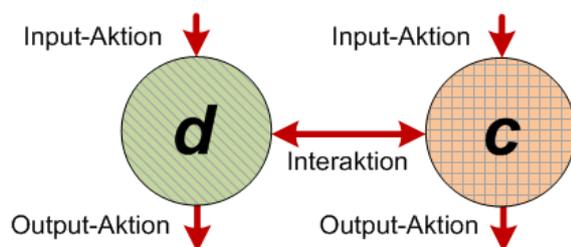


Abbildung 3-2: Modell eines einfachen kognitiven Multiagentensystems bestehend aus einer Population zweier interagierender kognitiven Agenten (d** und **c**). Interaktion ist dabei ein Austausch von Aktionen**

Abbildung 3-2 zeigt diesbezüglich ein einfaches CS mit zwei Agenten: Agent **d** (grün) und Agent **c** (orange) (vgl. Anhang C). Die CAs können Input-Aktionen von außen empfangen, in ihrem Inneren verarbeiten und Output-Aktionen nach außen, auf die CS-Ebene senden (rote Pfeile). Wenn zwei Agenten Aktionen austauschen, handelt es sich um eine Interaktion.

Y-Beispiel: Abbildung 3-3 zeigt eine agentenorientierte Dekomposition des hochautomatisierten Fahrzeuges und seiner nahen Umwelt^[Keh14]. V1 ist ein hochautomatisiertes Fahrzeug auf einer Straße mit eventuellen Fremdfahrzeugen. Population V1 enthält drei Agenten: d , C und v . d ist der Fahrer (*Driver*), C ist die Automation (*Computer*) und v ist das Fahrzeug (*Vehicle*). Fahrer und Automation sind CAs. Das Fahrzeug, auf dieser Ebene betrachtet, ist ein NA, weil es nur auf Aktionen anderer Agenten reagieren kann. Andere relevante Fremdfahrzeuge und Ressourcen der nahen Umwelt, wie Straße, können im e (*Environment*) aggregiert werden. Alle Agenten können Aktionen empfangen, Reaktionen zeigen oder eigene Aktionen senden. Dies ist in der Abbildung 3-3 durch entsprechende Pfeile dargestellt. Alle CAs und NAs bilden ein kognitives MAS bzw. ein CS, das sich innerhalb der offenen Umwelt befindetet.

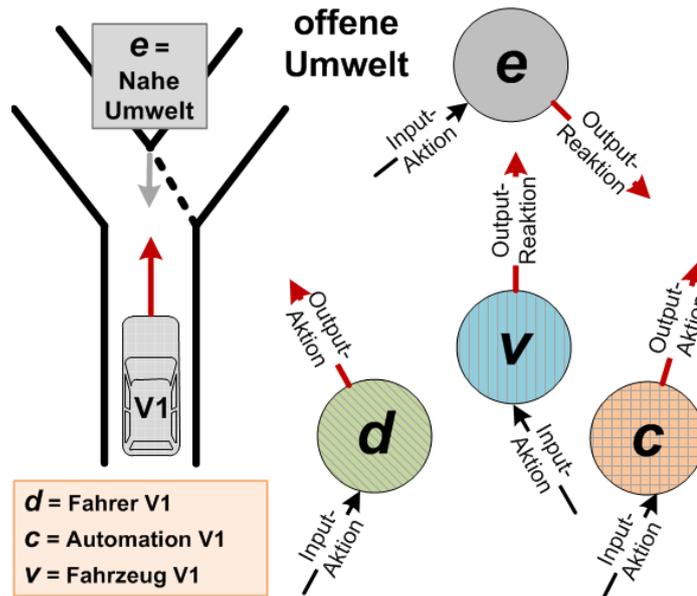


Abbildung 3-3: Dekomposition eines hochautomatisierten Fahrzeuges als kognitives MAS

Würde man die Dekomposition aus einer abstrakteren Sicht anstreben, um z.B. Verkehrsflüsse zu zerlegen, so könnte die V1 als ein einzelner CA erachtet werden, damit ist V1 eine kognitive Population. e kann als eine nichtkognitive Population verstanden werden. Es ist diejenige Population, die der Systemingenieur in einem Nutzungsszenario erzeugen kann^[ScK13], um damit sein CS zu gestalten oder in einer Usability Studie zu testen.

Weil CS komplexe Systeme mit Selbstorganisation, Emergenz etc. sind (vgl. Abschnitt 1.1), kann eine präzise Definition der Systemgrenzen für enthaltene Strukturen, Verhalten und Prozesse erschwert sein. CS können weiträumig verteilt sein, da hier nicht nur die geometrische, sondern auch die soziale Nähe und Ähnlichkeit der CAs und ihrer Relationen zueinander wichtig sind. Andererseits können CS aufgrund ihrer Verwandtschaft mit komplexen Systemen Eigenschaften besitzen, die Systemgrenzen erschließen. *Selbstähnlichkeit und Robustheit gegenüber offenen Umgebungen sind zwei solcher Eigenschaften der CS.*

Die Selbstähnlichkeit kann Systemstrukturen, Verhalten und Prozesse aufzeigen, die sich sowohl im Kleinen als auch im Großen in ähnlicher Form zeigen. Beobachtung der Selbstähnlichkeit im CS kann die Definition der Systemgrenzen unterstützen.

Beispiel: Einzelne Fahrzeuge können sich spontan „entscheiden“, Fremdfahrzeuge im dichten Verkehr durch das Öffnen einer Lücke auf ihren Fahrstreifen auffahren zu lassen. Solche

Bereitschaft zur Ressourcenüberlassung kann im CS ins Kleine und ins Große skaliert sein. Im Kleinen beobachtet man, dass Fahrer bereit sein können, die Fahrzeugkontrolle als Ressource an die Automation abzutreten oder dass im Großen mehrere Fahrzeuge sich spontan zu Einheiten verbinden können, indem sie untereinander ihr Fahrverhalten abgleichen und Ressourcen, wie längere Fahrabschnitte, anderen solchen Fahrzeugverbunden überlassen können.

Die Robustheit gegenüber der offenen Umwelt kann als weitere Hilfe für die Systemgrenzdefinition herangezogen werden. Wie bei jeder Systemgrenzdefinition kann überlegt werden, welche Agenten und Relationen dazwischen im jeweiligen CS für seine Struktur, Verhalten und Prozesse eine Rolle spielen können. Danach kann beobachtet, überlegt oder simuliert werden, ob die definierten Agenten und ihre Relationen als Ganzes eine Stabilität in ihren Strukturen, Verhalten und Prozessen gegenüber den äußeren Einwirkungen haben.

Beispiel: Ein 50 Meter weit entferntes bremsendes Fremdfahrzeug auf dem entgegengesetzten Fahrstreifen wird keinen Einfluss und somit auch keine Rolle für die Gestaltung einer Kontrollübergabe von der Automation zum Fahrer haben, wohl aber ein Fremdfahrzeug mit den genannten Parametern auf dem gleichen Fahrstreifen. Damit gehört das Fremdfahrzeug auf dem gleichen Fahrstreifen in die Grenzen des zu beschreibenden und zu gestaltenden CS.

In osGekoS dient das Konzept der kognitiven MAS als Leitkonzept für die strukturelle CS-Dekomposition. Damit können Nutzungsszenarien erzeugt und in ihre Bestandteile zerlegt werden. Durch solche Dekomposition kann die Wechselwirkung zwischen den CAs, wie Fahrern, Automation, ihren Fahrzeugen und der nahen Umwelt verstanden, beschrieben und gestaltet werden. In der später eingeführten Formalisierung der CS, der gestaltungskompatiblen CS-Architektur und dem Werkzeug für Lösungsexploration findet sich das Konzept kognitiver MAS wieder. Es ist als Denkstütze und Modellierungsvorlage besonders dann vom Nutzen, wenn ein holistisches Verständnis des CS während der Gestaltung angestrebt wird.

Die bisweilen erarbeitete Beschreibung der CS-Strukturen ist ähnlich stark reduziert, wie bei dem abstrakten Sender-Empfänger Kommunikationsmodell der Informationstechnik^[Shn48] und dem „quasi thermodynamischen“ Modell der Brownschen Agenten^[Scr03]. Dem Systemingenieur gibt dies die Möglichkeit, an die vorhandenen informationstheoretischen Modelle und Methoden anzuknüpfen. So können die Konzepte der Sender, Empfänger, Encoder, Decoder, Bandbreite, Kopplung, Mitkopplung etc. auf die Kommunikation und Interaktion der CAs angewandt werden. Dies wird im nächsten Abschnitt 3.2 aufgegriffen. Die weitere Präzisierung der CS-Strukturen und -Verhalten wird in darauf folgenden Abschnitten stattfinden.

3.2 Kommunikationsstrukturen kognitiver Systeme

Eine wichtige Grundlage für die theoretischen Konzepte des Gestaltungsrahmens osGekoS (vgl. auch Abbildung 1-3) ist die parallele Betrachtung der Multiressourcen Theorie (MRT)^[Wis84] und der Kanalstrukturen für die Informationsübertragung, wie sie im Bereich der Informationstechnik^[Shn48] vorgeschlagen sind. Mithilfe beider Konzepte lassen sich die Kommunikations- und Interaktionsstrukturen kognitiver Systeme beschreiben.

Die MRT schlägt vor, dass Menschen kognitive Ressourcen besitzen, die parallel genutzt werden. Wir verarbeiten Informationen auf verschiedenen Ebenen (Wahrnehmung, Kognition,

Aktionswahl), mittels verschiedener Kodierungen (sprachlich, räumlich) und multimodal (visuell, auditiv, haptisch). Dies sind die einzelnen Ressourcen, die uns parallel für die Informationsverarbeitung zur Verfügung stehen. Wir besitzen parallele Ressourcen auch für eine angemessene Reaktion (manuell, sprachlich). Wenn parallele Aufgaben gleiche Ressourcen benötigen, sinkt die Verarbeitungsleistung durch gegenseitige Störungen. Im umgekehrten Fall, wenn bei einer Aufgabenbewältigung parallele Ressourcen verwendet werden, steigt die Leistung^[Wis84].

Ideen der MRT sind ähnlich zur Informationsübertragung und -verarbeitung in technischen Systemen. Dort wird die Informationsverarbeitung durch Bandbreite, Kodierung, Störungen bestimmt und kann durch eine Bandbreitenerhöhung, Komprimierung, Entzerrung erweitert werden. Diese technischen Ansätze stehen der Idee der MRT sehr nah. „One feels, for example, that two punched cards should have twice the capacity of one for information storage, and two identical channels twice the capacity of one for transmitting information“^[Shn48].

Aus der Annahme, dass der Mensch ein typischer CA ist, lässt sich folgendes folgern: Jeder CA sollte parallele Ressourcen für eine Informationsübertragung und -verarbeitung zur Verfügung stellen. Für eine CS-Gestaltung bietet dies die Möglichkeit, für die Kommunikation und Interaktion die „parallele Rechen- und Informationsübertragungsleistung“ der CAs mittels Kanalisierung der Information zu den entsprechenden Ressourcen zu nutzen. Dies kann durch semantische Auftrennung der vom Systemingenieur konzipierten Information in eindeutig definierte Informationspakete geschehen. Diese Pakete können auf getrennten multimodalen Kommunikationskanälen und Unterkanälen zu den Ressourcen der CAs geschickt oder zwischen den CAs ausgetauscht werden. Jeder multimodale Kommunikationskanal, wie der haptische Kanal, kann in Unterkanäle für den Umgang mit verschiedenen Arten der Interaktionssignale und der dazugehörigen Semantik unterteilt werden^[Ket13]. Dadurch können Informationen den CAs gezielt zur Verfügung gestellt, angeregt oder gehemmt werden. Zur Verdeutlichung ist der erarbeitete Zusammenhang in der Abbildung 3-4 dargestellt.

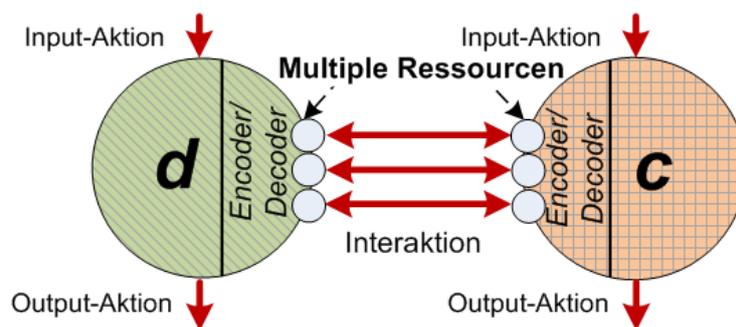


Abbildung 3-4: Kanalisierung der semantisch aufgetrennten Information zwischen den CAs

Beispiel: So kann ein „Hinweis-Unterkanal“ definiert werden, der innerhalb eines CS Hinweise transportiert, z.B. durch diskrete Signale, wie haptische Ticks. Auch ein „Aktionenführungs-Unterkanal“ kann definiert werden, der Aktionen der CAs direkt durch kontinuierliche haptische Kraft- oder Positionsrückmeldung führt. Andere Modalitäten, wie die visuelle und die akustische, können ebenfalls in Unterkanäle mit einer eindeutigen semantischen Zuordnung aufgeteilt werden. Auf diese Weise, mittels der Unterkanäle mit den spezifisch kanalisierten

¹³ Man fühlt zum Beispiel, dass zwei Lochkarten die doppelte Kapazität für die Speicherung und zwei identische Kanäle die doppelte Kapazität zur Übertragung von Informationen haben sollten. EÜ.

und semantisch eingegrenzten Informationspaketen, kann der Systemingenieur die Stärke der parallelen Informationsverarbeitung der CAs bei der CS-Gestaltung ausnutzen.

Solche eindeutige Art der Informationskanalisierung innerhalb des CS erlaubt technische Ansätze zur Verwaltung der Informationsflüsse, wie Priorisierung oder Arbitrierung. Die frühzeitige Berücksichtigung der Semantik bringt Ordnung, Konsistenz und Nutzerkompatibilität in den Gestaltungsprozess sowie in das gestaltete Zielsystem selbst.

Nutzung der semantisch und multimodal aufgetrennten Kanalstrukturen für die Informationsübertragung und -verarbeitung hat in osGekoS einen hohen Stellenwert. Dies wird bei der später vorgestellten Methode „Semantisches Mapping“ und bei dem „Arbitrierungswerkzeug“ aufgegriffen und verwendet (vgl. Kapitel 5).

3.3 Strukturen und Verhalten der Agenten

Nach dem Konzept kognitiver MAS (vgl. Abschnitt 3.1.3) bestehen CS aus Agenten, die Aktionen empfangen, verarbeiten und neue Aktionen senden (vgl. Abbildung 3-4). Die Agentendefinition postuliert komplexe Strukturen und zielgerichtete Informationsumgestaltung im Inneren der Agenten. Folglich müssen von außen empfangene Aktionen zwecks Verarbeitung in die innere Struktur des Agenten und von dort, zielgerichtet umgestaltet, nach außen gelangen können. Diese Feststellung wird zwecks Präzisierung der Strukturen und des Verhaltens von Agenten eines CS weiter verfolgt und ausgearbeitet. Dabei wird das erarbeitete Konzept kognitiver MAS zum MPS-Konzept (Manipulations-Potenzial Systeme) transformiert.

Die oben genannte Feststellung lässt sich verstehen und beschreiben, wenn man zunächst das informationstheoretische Konzept eines Ereignisses nutzt. Ein Ereignis bewirkt eine „Veränderung eines Systemzustands“^[SP13b], die auch als Transition bezeichnet werden kann. Ein Ereignis kann auch in einer engen Verbindung mit der Wahrscheinlichkeit seines Auftretens verstanden werden^{[SP13b][Shn48]}. Ein mit der Input-Aktion empfangenes Ereignis kann den CA anregen, seinen auf der Systemebene beobachtbaren Gesamtzustand zu verändern und die neu produzierte Output-Aktion zusammen mit einem neuen Ereignis auf die Systemebene zu senden. Im Inneren des CA kann ein Ereignis eine Transition seiner kognitiven Zustände triggern.

Wenn man über ein Agentenverhalten im Kontext einer Kognition spricht, muss der Begriff „Aktion“ präzisiert werden. Der Definition nach, produzieren Agenten ihre Aktionen aufgrund zielgerichteter bzw. intentionaler Zustände und Transitionen innerhalb ihrer Strukturen. Für CAs, aufgrund der Zielgerichtetheit der Kognition selbst, lässt sich ein intentionales, beeinflussendes Handeln, also eine *Manipulation* statt einer bloßen gerichteten Aktion, umso mehr annehmen. In der einschlägigen Literatur, die das menschliche Verhalten in Kontrollsituationen behandelt, spricht man oft über Aktion und Manipulation in einem Atemzug^{[ShV78][HuH88][Dör01]}. Des Weiteren wird eine Aufgabe als eine „dauerhaft wirksame Aufforderung an Handlungsträger, festgelegte Handlungen wahrzunehmen“^[SP13f] definiert.

Annahmen: 1.) Das Verhalten von Agenten eines CS ist manipulativ^[Keh14]. 2.) Agenten eines CS können von außen als Manipulations-Potenziale (MP, MPs für Mehrzahl) verstanden werden. 3.) Für jedes CS sind Aktion und Manipulation äquivalent zueinander. Deshalb werden diese Begriffe ab sofort als Synonyme verwendet und Manipulationen in Abbildungen weiter-

hin als rote Pfeile dargestellt. 4.) Eine Aufgabe kann als eine Abfolge von Input-Manipulationen (IM, IMs für Mehrzahl) von der CS-Ebene zum Agenten, eine Handlung als eine Abfolge von Output-Manipulationen (OM, OMs für Mehrzahl) vom Agenten auf die CS-Ebene verstanden und beschrieben werden^[Har86].

Diese Annahmen sind in der Abbildung 3-5 zusammengefasst. d ist ein Agent, der nach außen, auf die CS-Ebene ein MP darstellt (grüner Kreis). Von der CS-Ebene empfängt d eine Aufgabe, die als eine IM-Abfolge ankommt und zur agenteninternen Informationsverarbeitung beiträgt. d kann Ereignisse und IMs empfangen und neue Ereignisse und OMs senden. Ereignisse begleiten die Manipulationen und sind hier als Pfeilspitzen dargestellt. Eine Handlung auf der CS-Ebene ergibt sich dann aus einer OM-Abfolge des Agenten.

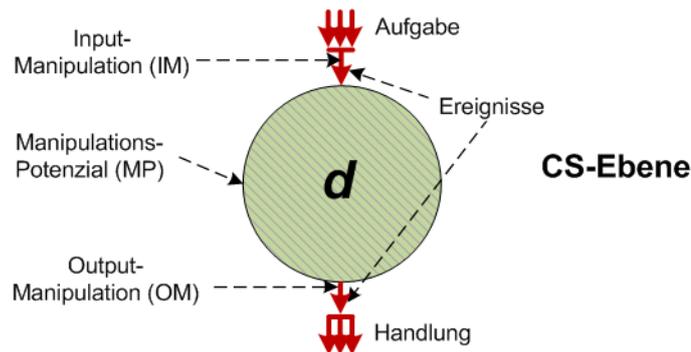


Abbildung 3-5: Äußere Struktur eines Agenten d als MP mit IMs, OMs sowie Ereignissen, die diese Teilstrukturen verbinden. Abfolgen von IMs und OMs auf der CS-Ebene sind jeweils Aufgaben und Handlungen

Y-Beispiel: Der Fahrer kann zwei auf der CS-Ebene beobachtbare MPs darstellen: „Fahren links“ und „Fahren rechts“. Durch den Einfluss (IM) der Umwelt, wie „Weggabelung voraus“, kann der Fahrer sein von außen beobachtbares MP bestimmen, z.B. „Fahren links“. Er wird dabei eine entsprechende OM senden, z.B. „links lenken“. Damit wird er sein beobachtbares MP „Fahren links“ darstellen. Außerdem kann der Fahrer eine beständige Aufgabe (IM-Abfolge) empfangen haben, das Fahrzeug immer nach links zu fahren. Wenn mehrmals eine Gabelung auf seinem Weg auftaucht, wird er unter der beständigen Einwirkung der Aufgabe sich entscheiden, sein äußeres MP als „Fahren links“ zu bestimmen und entsprechend öfter nach links zu lenken als nach rechts. Nach mehreren Weggabelungen wird er als beobachtete Gesamthandlung (OM-Abfolge) deutlich weiter nach links gefahren sein als nach rechts.

Es gibt empirische Nachweise, dass z.B. die Art der Manipulation von Objekten für Menschen von den vorher kognitiv erzeugten Absichten abhängt^{[RoC12][Gon97]}. Wenn man über die Modellierung der (inneren) kognitiven Prozesse des Menschen spricht, ist die Rede von der Manipulation der kognitionseigenen „physikalischen Symbole“^[Nel80]. Eine Entscheidung im Kontext einer Kognition wird als eine „Auswahl einer Aktion aus einer Menge verfügbarer Maßnahmen unter Berücksichtigung möglicher Umweltzustände“^[SP13g] definiert.

Annahmen: 1.) Innere Strukturen der Agenten können ebenfalls als MPs verstanden und beschrieben werden. Die inneren MPs können als Wissen, Überzeugungen, Wünsche, Absichten, Motive interpretiert werden, ersetzen ab sofort den Begriff „kognitiver Zustand“ und werden in Abbildungen weiterhin mit Kreisen dargestellt. 2.) Empfangene IMs wechselwirken im Inneren der Agenten mit den Eigen-Manipulationen (EM, EMs für Mehrzahl) und können ein MP aus der bereits vorhandenen Menge miteinander assoziierter MPs oder ein neu gene-

riertes MP aktivieren, was als eine Entscheidung interpretiert werden kann. Das aktuell aktive MP des Agenten ist als Plan oder Ziel interpretierbar und stellt seine OM nach außen als die beobachtbare OM des Agenten dar. 3.) Die Struktur von Agenten ist manipulierbar^[Keh14].

Abbildung 3-6 zeigt den kognitiven Agenten d . Im Inneren hat er eine Struktur, die aus einer Menge zueinander assoziierter MPs besteht (rote Kreise, verbunden mit roten punktierten Linien). Zusammen mit der Eigen-Manipulation (EM) von d , die die inneren kognitiven Prozesse des Agenten abbildet, kann die empfangene IM ein neu generiertes oder bereits vorhandenes MP aktivieren (schwarzer Punkt). Das jeweils aktive innere MP des Agenten stellt dann seine OM und damit auch die beobachtbare OM des Agenten auf der CS-Ebene dar.

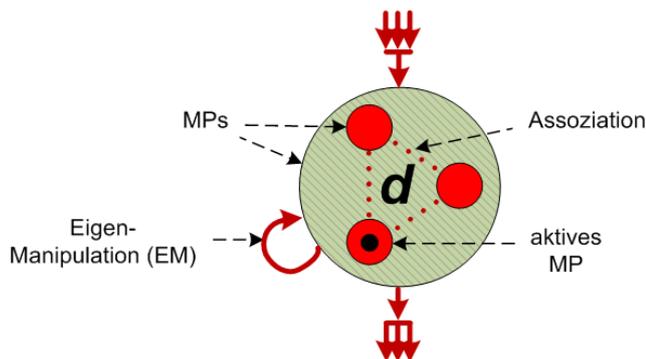


Abbildung 3-6: IMs und EMs des d generieren ein aktives MP, das seine OM nach außen sendet

Y-Beispiel: Der Fahrer kann drei auf der CS-Ebene beobachtbare MPs darstellen: „Fahren links“, „Fahren geradeaus“ und „Fahren rechts“. Durch den Einfluss (IM) der Umwelt, wie „Weggabelung voraus“, und deren Interpretation bzw. Abgleich mit den inneren kognitiven Vorgängen (EM) als z.B. „Gefahr voraus“, wird das bereits vorhandene oder neu generierte MP „Fahren links“ des Fahrers aktiviert. Es sendet dann die entsprechende OM, z.B. „links lenken“. Damit stellt der Fahrer sein auf der CS-Ebene beobachtbares MP „Fahren links“ dar.

Dieses Beispiel und das Beispiel davor zeigen, dass *Agenten zwei dichotome eskalierende Entwicklungen in der Menge der assoziierten und aktivierbaren MPs und ihrer OMs haben können*. In den verwendeten Beispielen wäre eine solche Entwicklung hin zu „Fahren links“ und „links lenken“ bzw. hin zu „Fahren rechts“ und „rechts lenken“. Das bedeutet, dass assoziierte MPs eines Agenten dichotome eskalierende Manipulations-Potenzialketten bilden können, was später beim Erarbeiten des Konzeptes der Aktionstensionen aufgegriffen wird. Für Automotivanwendungen können auf diese Weise „Fahren links und rechts“, „Beschleunigen und Verzögern“, „Kooperieren und Konkurrieren“ in unterschiedlicher Intensität als dichotome MP-Ketten miteinander verknüpft werden.

Abbildung 3-6 macht außerdem den Unterschied zwischen den kognitiven und nichtkognitiven Agenten deutlich. An sich selbst gerichtete Manipulationen (EMs) sind aufgrund kognitiver Prozesse bei den CAs grundsätzlich immer vorhanden. Sie können als eine persistente Berücksichtigung der eigenen Meinung bzw. als eine „kognitive Hysterese“^[NoD86] interpretiert werden. Wenn ein CA frei von IMs ist, sollte er nach der EM handeln. Nichtkognitive Agenten (NAs) besitzen dagegen keine EMs. Sie können sich im Gegensatz zu CAs nicht zielgerichtet oder willkürlich verhalten, sondern lediglich eine „physikalische Hysterese“ zeigen, wie Massträgheit, die ihr Verhalten beeinflussen kann. Daher können sie nur von außen und nicht „eigenmanipuliert“ werden. Wenn NAs nicht manipuliert werden, handeln sie nicht.

3.4 Strukturen und Verhalten kognitiver Systeme

Nach dem aktuellen Stand der Homogenisierung der CS enthalten sie Populationen aus manipulierbaren und manipulierenden Agenten mit MPs. Führt man Abbildung 3-2 und Abbildung 3-6 zusammen, kommt man zur Abbildung 3-7. Diese zeigt ein CS bestehend aus einer kognitiven Population. d und c führen eine gemeinsame Aufgabe aus und generieren dabei eine gemeinsame Handlung. Die Interaktion zwischen d und c lässt sich nun als ein gegenseitiger Manipulationsaustausch zwischen ihren momentan aktiven MPs verstehen und beschreiben.

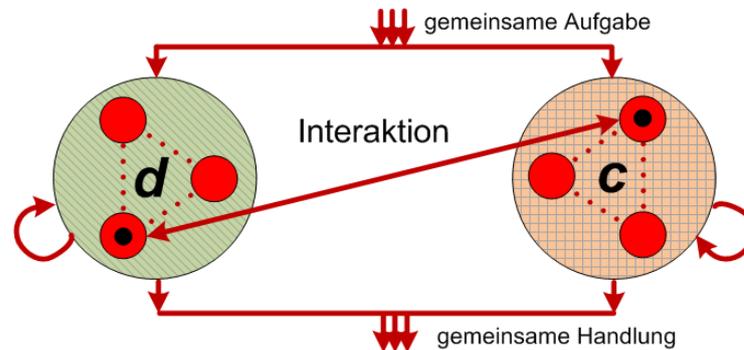


Abbildung 3-7: Struktur eines einfachen CS: d und c führen eine gemeinsame Aufgabe aus und interagieren, was sich als gegenseitige Manipulation zwischen den aktiven MPs der beiden CAs verstehen und beschreiben lässt

Y-Beispiel: Der Fahrer und die Automation können drei unterschiedliche MPs auf der CS-Ebene darstellen: „Fahren links“, „Fahren geradeaus“ und „Fahren rechts“. Um den jeweiligen Partner zu überzeugen, seine Meinung und damit sein aktives MP und die OM zu ändern und stattdessen den eigenen MP und OM zu folgen, werden der Fahrer und die Automation interagieren. Dies geschieht, indem sie versuchen werden, sich gegenseitig zu manipulieren, indem sie ihre momentan aktive MPs, z.B. „Fahren links“ und „Fahren rechts“ austauschen.

Durch ähnliche Behandlung von Populationen, Agenten und MPs wird die Selbstähnlichkeit innerhalb der CS-Strukturen deutlich. Populationen können je nach Detailgrad der Betrachtung als Agenten oder MPs, Agenten als MPs oder Populationen und MPs als Agenten oder Populationen verstanden und beschrieben werden. Darum werden die Begriffe Population, Agent und MP ab sofort synonym verwendet (CA, NA: \Leftrightarrow MP).

Die Abbildung 3-7 zeigt einen weiteren Unterschied zwischen den kognitiven und nichtkognitiven Agenten. *MPs der NAs sind nichtkognitiv*, da sie, wie bereits erklärt, einem einfachen determinierten Regelwerk folgen. Sie können also nicht als CAs, sondern nur wieder als NAs beschrieben werden. Dies kann die CS-Grenzdefinition „ins Innere“ des CS unterstützen.

Definition: Kognitive Systeme sind selbstähnliche, robuste Strukturen von manipulierenden und manipulierbaren kognitiven und nichtkognitiven Manipulationspotenzialen (CS: \Leftrightarrow MPS).

Im Folgenden wird erarbeitet, wie das Verhalten von CS vom Systemingenieur zwecks Gestaltung verstanden, beschrieben und modifiziert werden kann. Durch Manipulierbarkeit und Manipulationen können örtlich und sozial benachbarte CAs auf der CS-Ebene eine *negative*, *unbestimmte* oder *positive* Wechselwirkung erzeugen. Die Agentenwechselwirkung definiert das CS-Verhalten und kann durch MP-Transitionen modifiziert werden. Dies kann der Systemingenieur bei der CS-Gestaltung ausnutzen. Durch das Erweitern der Abbildung 3-7 ist dieser Gedankengang in der Abbildung 3-8 grafisch veranschaulicht.

Abbildung 3-8 zeigt die Struktur eines CS bestehend aus interagierenden CAs d und c . Mittels der Interaktion und OMs können die CAs auf der CS-Ebene eine negative (-), unbestimmte (o) oder positive (+) Wechselwirkung erzeugen. Diese kann verschoben werden, indem z.B. das aktive MP von d bei dessen Generieren bzw. Aktivieren verschoben wird. Die möglichen CA-Wechselwirkungen (-, o, +) sind als farbige Kreise an den Hilfslinien mit der Darstellung der immer kleiner werdenden räumlichen Abstände zwischen den korrespondierenden MPs der Agenten schematisiert. Die MP- und Wechselwirkungstransition von negativ (-) zu positiv (+) sind durch schwarze Pfeile dargestellt. Beispiele dazu folgen weiter unten.

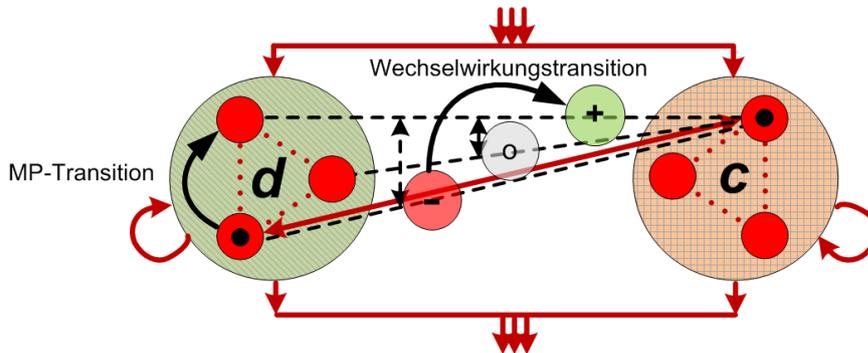


Abbildung 3-8: Struktur eines CS bestehend aus zwei CAs (d und c): IMs, OMs und EMs der Agenten können insgesamt zur positiven (+), unbestimmten (o) oder negativen (-) Wechselwirkung auf der CS-Ebene führen. Diese Wechselwirkung bestimmt das CS-Verhalten und lässt sich durch MP-Transitionen modifizieren

Es ist wichtig zu betonen, dass die Interaktion im Kontext von MPS und osGekoS ungleich der Wechselwirkung ist. Interaktion ist eine direkte, gegenseitige wertfreie Manipulation der Agenten. Ein Interaktionsprozess ist ein eben solcher Manipulationsaustausch zwischen ihnen. Die CA-Wechselwirkung kann dagegen aufgrund von Manipulationen und Handlungen entstehen, die Agenten nicht nur direkt, sondern über weiträumige Ketten von CAs und NAs austauschen können. Sie ist auf der übergeordneten CS-Ebene verortet, im Gegensatz zur Interaktion bewertet (-, o, +), z.B. in Bezug auf die Performanz, keinem der CAs eigen und kann ein emergenter, selbstorganisierter, generativer Systemprozess sein.

Unbestimmte CA-Wechselwirkung ist in kognitiven Systemen grundsätzlich immer vorhanden und kann spontan zu einer positiven oder einer negativen Wechselwirkung umschlagen. Sie kann mit einer angespannten „Ignoranz“ oder auch „Willkür“ verglichen werden. Sie ist wichtig bei einer CS-Gestaltung als eine Gedankenstütze. Sie gibt einen beständigen Hinweis darauf, dass der Systemingenieur entweder etwas nicht berücksichtigt haben könnte oder dass seine CAs, z.B. aufgrund ihrer Kognition, sich nicht erwartungskonform verhalten.

Y-Beispiel: Vertraut der Fahrer der Automation, folgt er ihren Hinweisen. Wenn er ihr nicht vertraut, überstimmt er sie und setzt seinen Willen durch. Beide Varianten sind höchst situations- und personenabhängig, instabil, schwer vorher bestimmbar, können sowohl zu positiven als auch negativen Effekten auf der CS-Ebene führen oder gar nicht auftreten. Aus der Sicht der Systemgestaltung handelt es sich dabei um eine unbestimmte CA-Wechselwirkung.

Außerdem gibt es grundsätzlich zwei zueinander duale Arten der CA-Wechselwirkung: negative und positive. Der Zusammenhang zu möglicher negativen und positiven Wechselwirkung innerhalb von CS ist in der einschlägigen Literatur ebenfalls erkannt^[Cai98]. Es wurde argumentiert, dass CAs innerhalb eines CS miteinander wechselwirken und dass *durch die CA-Wechselwirkung das Verhalten eines CS definiert werden kann.*

Negative CA-Wechselwirkung ist ein emergenter Prozess eines komplexen CS, der die Performanz des CS während der CA-Wechselwirkung negativ beeinflusst^[KeT13]. Sie ist charakteristisch für fehlabgestimmte Handlungen, potenzielle oder akute Konflikte und eine Konkurrenz.

Y-Beispiel: Der Fahrer möchte nach links abbiegen und die Automation nach rechts, der Fahrer lenkt nach links und die Automation nach rechts. Diese Situation kann zu einer gefährlichen und daher negativ zu bewertenden Destabilisierung des Fahrer-Automation-Fahrzeug-Umwelt-Systems führen. Es handelt sich um eine negative CA-Wechselwirkung.

Positive CA-Wechselwirkung ist ein emergenter Prozess eines komplexen CS, der die Performanz des CS während der CA-Wechselwirkung positiv beeinflusst^[KeT13]. Sie ist dual zu der negativen. Die positive Wechselwirkung ist charakteristisch für abgestimmte Handlungen, konfliktfreie Situationen und eine Kooperation^{14[Cai98][Hoc01]}.

Y-Beispiel: Der Fahrer ist abgelenkt und nicht auf das Lenken vorbereitet. Die Automation weist ihn frühzeitig auf die richtige Abbiegerichtung hin oder folgt einer vorher berechneten Route. Dies kann zum positiven Effekt des effizienten, sicheren und komfortablen Verhaltens des Gesamtsystems Verkehr beitragen. Es handelt sich um eine positive CA-Wechselwirkung.

Eine Verschiebung der Manipulationen bzw. MPs kann die CA-Wechselwirkung von der positiven zur negativen und umgekehrt entwickeln. Sie kann in allen Schritten der Informationsverarbeitung der CAs stattfinden. Wenn CAs einen Situationsbewertungsprozess besitzen^[Eny95], dann wechselwirken sie in ihrem Wahrnehmen, Verstehen, Projizieren, Entscheiden und Agieren. Wenn CAs kognitive Elemente, wie Überzeugungen, Wünsche, Ziele etc. besitzen, dann wechselwirken sie mittels dieser Elemente^{[Cai98][Hoc01]}. Auf allen Ebenen der kognitiven Informationsverarbeitung können Konkurrenz, Willkür und Kooperation auftreten. Diese können durch eine vom Systemingenieur initiierte Beeinflussung der MPs bzw. der Manipulationen der CAs gehemmt bzw. angeregt werden (vgl. Abschnitt 4.1.1).

Explizite Betrachtung der CA-Wechselwirkung als gegebenes CS-Verhalten kann ein Vorteil für die CS-Gestaltung sein. Denn die meisten Probleme innerhalb eines CS können aufgrund schlechter Gestaltung der Interaktion zwischen den Agenten entstehen^[Non90]. Dies kann die negative CA-Wechselwirkung (fehlabgestimmte Handlungen, Konkurrenz) bedingen und mittels vorheriger Anregung der positiven (abgestimmte Handlungen, Kooperation) gelöst oder umgangen werden. In osGekoS wird Wert darauf gelegt, das zu gestaltende CS auf die CA-Wechselwirkung hin zu untersuchen und erst dann die Interaktionsmodelle zu entwickeln.

Alle bisherigen Einsichten zu MPS können bei einem holistischen Verständnis und der Beschreibung von CS während ihrer Gestaltung unterstützen. Außerdem werden sie bei der Formalisierung der CS (vgl. Abschnitt 3.5) und für die Erarbeitung der Konzepte der Aktions-tension und der Arbitrierung nützlich sein (vgl. Abschnitte 4.2 und 4.3).

¹⁴ Es muss erwähnt werden, dass hier der Einfluss sozialpsychologischer Effekte auf der Ebene der Agenten, wie Steigerung der Performanz einzelner Agenten in einem Konflikt, Wettkampf etc., oder bei einer Kooperation, wie Selbstlosigkeit oder Opportunismus (vgl. auch [Mar92]), bewusst außen vor gelassen ist.

3.5 Semi-Formale Repräsentation Kognitiver Systeme

Bisherige Einsichten zu Strukturen und Verhalten von CS lassen sich formalisieren. Der Nutzen davon wäre die Möglichkeit, ein CS formal beschreiben und es zwecks effektiver Gestaltung modellieren und implementieren zu können. Nützlich wäre auch, das CS formal analysieren und simulieren zu können. Allerdings ist es weder im Fokus noch im Rahmen des Durchführbaren für diese Arbeit, eine vollständige und allgemeine formale Methode für CS-Modellierung zu entwickeln. Diese (Semi-)Formalisierung soll dem Systemingenieur vorrangig Ideen und Methodenansätze liefern, wie er seine CS bei der Gestaltung formal behandeln könnte und dient als Grundlage für die Erarbeitung noch fehlender Konzepte holistischer CS-Gestaltung (vgl. Kapitel 4) und einiger wichtiger osGekoS-Werkzeuge (vgl. Abschnitt 5.4).

Die bisherigen theoretischen Überlegungen und fortlaufend entwickelten Grafiken zu Strukturen und Verhalten von CS legen nahe, sich zwecks Formalisierung der CS für ein auf Graphentheorie basierendes Beschreibungsmittel zu entscheiden. Dies ist damit begründet, dass CS als bestehend aus zustandshaltenden, wie Agenten (MPs), und verhaltensführenden Strukturen, wie Manipulationen, beschrieben wurden. Mit Knoten und Kanten aus der Graphentheorie sollten diese Strukturen gut abbildbar sein. Nach einer intensiven Überlegung (vgl. Anhang B) wurde entschieden, synchrone Automaten als Beschreibungsmittel auszuwählen.

Für die Semantik der Formalisierung kognitiver Systeme sei folgendes definiert (1):

$$\begin{aligned}
 &\text{MP-Indikatoren "spezifisch": } \{d, c, a_2, \dots\} \in L : \{\text{Zeichen- und Ziffernketten} \mid n \notin L\} \\
 &\text{MP-Indikatoren "alle", "beliebig": } \{\alpha, \beta\} \in G : \{\text{griechische Kleinbuchstaben} \mid \eta \notin G\} \\
 &\text{Diskreter Zeitpunkt, Ereignis "spezifisch": } n \in Z : \{\text{ganze Zahlen}\} \\
 &\text{Diskreter Zeitpunkt, Ereignis "beliebig": } \eta \in Z \\
 &\text{Dichotome kontinuierliche Potenzialkette: } R_{norm} := [-1; 1] \in R : \{\text{reelle Zahlen}\}
 \end{aligned} \tag{1}$$

Mittels dieser Semantik lassen sich als selbstständig definierte Strukturen von CS (MPS) als zustandshaltende Agenten (MPs) sowie ihre Hierarchie formal beschreiben, z.B. (2):

$$\text{Agent eines Ober- mit allen Unteragenten zum } n: \quad {}^d_\alpha c_n \in A_\eta : \{a_\eta \mid \forall a_\eta \in R_{norm}\} \tag{2}$$

Die (2) beschreibt einen spezifischen Agenten c mit allen Unteragenten und einem spezifischen Oberagenten d zum Ereignis n . Er gehört der Menge aller Agenten an, die ihr aktives Manipulationspotenzial zum beliebigen Ereignis η auf dem R_{norm} zeigen.

Formalisierung der CS wird mit dem bereits bekannten repräsentativen Beispiel (vgl. Abbildung 3-1) vorangetrieben. Die Abbildung 3-9 zeigt dafür einen Zustandsübergangsgraphen des entsprechenden CS am Beispiel eines hochautomatisierten Fahrzeuges (vgl. auch Abbildung 3-3). Agenten sind als Knoten und Kommunikationskanäle dazwischen sind als Kanten modelliert. d ist der Fahrer, c ist die Automation. Aus Vereinfachungsgründen sind im v die NAs das Fahrzeug und die nahe Umwelt zusammengefasst. Jeder Agent hat hier nur einen gerichteten Kommunikationskanal zu jedem anderen Agenten. Bei mehreren, z.B. multimodalen Kommunikationskanälen, können diese als zusätzliche Kanten oder als weitere Agenten (Knoten) modelliert werden, wie in Abschnitt 3.1 bereits vorgeschlagen wurde.

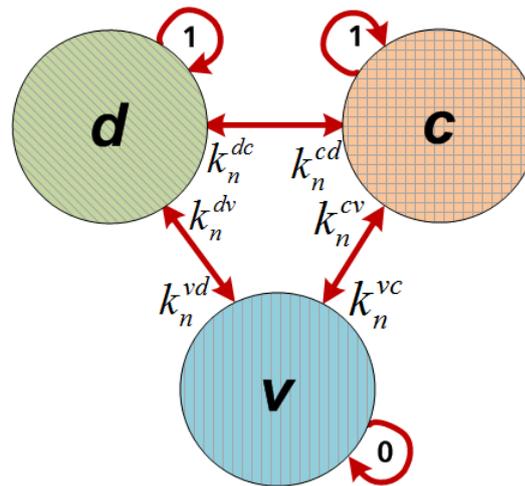


Abbildung 3-9: Dekomposition eines hochautomatisierten Fahrzeuges als Zustandsübergangsgraph

Mittels Manipulationen können Agenten über Kommunikationskanäle gekoppelt und entkoppelt werden^[FK10]. Dies lässt sich an den Kanten des Graphen als Kopplungsgrade angeben. In einem Fahrzeug beispielsweise, kann der Fahrer sich ein- und abkoppeln, indem er die Stellteile für die Fahrzeugkontrolle (Lenkrad, Pedale) anfasst oder loslässt.

Ein Kopplungsgrad: $k_n^{\beta\alpha} \in \mathbb{R}_{norm}$ stellt eine Beziehung zwischen beliebigen Sender- und Empfängeragenten zum Ereignis n her. Eine Manipulation zwischen zwei Agenten kann dann als Produkt des Kopplungsgrades mit auf dem entsprechenden Kommunikationskanal geführtem aktiven MP des Senderagenten angegeben werden. Die (3) beschreibt exemplarisch eine spezifische Manipulation m zwischen spezifischen Agenten d und c zum Ereignis n :

$$\text{Manipulation von } d \text{ nach } c \text{ zum } n: k_n^{dc} \cdot d_n = m_n^{dc} \in M_n : \{m_n \mid \forall m_n \in \mathbb{R}_{norm}\} \quad (3)$$

Solche Formalisierung der Manipulationen erfüllt folgende logische Merkmale:

- Der Multiplikationsoperator (*) stellt eine anregende, hemmende oder ignorierende Beziehung zwischen dem Sender- und dem Empfängeragenten her.
- Je größer der Betrag der Manipulation ist, umso „stärker“ ist die jeweilige Manipulation in ihrem anregenden oder hemmenden Charakter. Sonderfälle sind bei:
 - $k_n = 1$: Der Kommunikationsanal ist vollständig offen und Agenten sind positiv gekoppelt, z.B. Fahrer lenkt aktiv und regt sein Fahrzeug damit zum Lenken an.
 - $k_n = -1$: Der Kommunikationsanal ist vollständig offen und Agenten sind negativ gekoppelt, z.B. Fahrer lenkt nach links, Automation auch. Fahrer empfindet das zusätzliche Lenken der Automation als unangenehm und lenkt dagegen.
- Je kleiner der Betrag der Manipulation ist, umso „schwächer“ ist die jeweilige Manipulation in ihrem anregenden oder hemmenden Charakter. Sonderfall ist bei:
 - $k_n = 0$: Der Kommunikationsanal ist vollständig geschlossen und Agenten sind entkoppelt, z.B. Fahrer lässt die Stellteile los und steuert das Fahrzeug nicht.

- Der Betrag der Manipulation wächst nie über 1 an, denn ein Agent kann einen anderen Agenten nur zu einem maximal möglichen MP anregen oder hemmen. Der Betrag eines solchen aktiven MP kann nach der Definition höchstens 1 sein.

Eigen-Manipulationen (EMs) sind aufgrund kognitiver Prozesse bei den CAs grundsätzlich immer vorhanden (vgl. Abschnitt 3.3). Die EMs der CAs (Fahrer und Automation) werden daher im Zustandsübergangsgraphen als „Mitkopplungen“ mit 1 angenommen. NAs haben in der Regel keine EMs. Beim NA (Fahrzeug in naher Umwelt) ist die EM daher mit 0 angegeben.

Ein Kopplungsgrad ist als Produkt des Nachdruck- bzw. Salienzkoeffizienten $s_n^\beta \in R_{norm}$ und des Hinwendungs- bzw. Aufmerksamkeitskoeffizienten $h_n^\beta \in R_{norm}$ zerlegbar, z.B. (4):

$$k_n^{cd} := s_n^c \cdot h_n^d \quad (4)$$

Solche Formalisierung mittels des Multiplikationsoperators (*) und die angegebenen Wertebereiche für Salienz und Aufmerksamkeit erfüllen folgende logische Merkmale:

- Bei steigendem Betrag der Aufmerksamkeit gegenüber einer IM oder der Salienz einer OM steigen auch der Betrag der Kopplung und damit die Wirkung der Manipulation.
- Bei sinkendem Betrag der Aufmerksamkeit gegenüber einer IM oder der Salienz einer OM sinken auch der Betrag der Kopplung und damit die Wirkung der Manipulation.
- Die Beträge der Aufmerksamkeit und der Salienz können nicht größer als 1 werden, da mit 1 (=100%) die Sättigung dieser Werte bereits erreicht wäre.

Für das Beispiel aus Abbildung 3-9 und (4) lassen sich die Kopplungen wie folgt aufschlüsseln:

- $h_n > 0; s_n > 0 \Rightarrow k_n > 0$: d ist hörig gegenüber der Anregung des C und nimmt sie positiv in die MP-Generierung bzw. -Aktivierung auf. Z.B. Fahrer lenkt nach links \rightarrow Automation regt das Rechtslenken an \rightarrow Fahrer lenkt aus Hörigkeit nach rechts um.
- $h_n < 0; s_n > 0 \Rightarrow k_n < 0$: d ist trotzig gegenüber der Anregung des C und nimmt sie negativ in die MP-Generierung bzw. -Aktivierung auf. Z.B. Fahrer lenkt nach links \rightarrow Automation regt das Rechtslenken an \rightarrow Fahrer lenkt aus Trotz weiter nach links.
- $h_n > 0; s_n < 0 \Rightarrow k_n < 0$: d ist hörig gegenüber der Hemmung des C und nimmt sie negativ in die MP-Generierung bzw. -Aktivierung auf. Z.B. Fahrer lenkt nach links \rightarrow Automation hemmt das Rechtslenken \rightarrow Fahrer lenkt aus Hörigkeit weiter nach links.
- $h_n < 0; s_n < 0 \Rightarrow k_n > 0$: d ist trotzig gegenüber der Hemmung des C und nimmt sie positiv in die MP-Generierung bzw. -Aktivierung auf. Z.B. Fahrer lenkt nach links \rightarrow Automation hemmt das Rechtslenken \rightarrow Fahrer lenkt aus Trotz nach rechts um.

CAs können ihre Kopplungsgrade zu anderen Agenten jederzeit selbst bestimmen. Dies liegt darin begründet, dass CAs aufgrund der kognitiven Prozesse ihre Aufmerksamkeit und den Manipulationsnachdruck selbst steuern können. Kopplungsgrade der NAs können dagegen nur von außen, z.B. von den damit gekoppelten CAs, bestimmt werden.

Y-Beispiel: Ein Fahrer (CA) kann jegliche IMs ignorieren und seinen Willen in der Fahrzeugkontrolle (OM) durchsetzen wollen. Dies gilt für jeden CA. Denn eine komplexe Automation mit dem Zugriff auf die Fahrzeugsteuerung wird ihren „Willen“ ebenfalls durchsetzen wollen. Ein Fahrzeug (NA) kann sich nicht vom Fahrer abkoppeln, wenn der Fahrer es nicht will.

Die gesamten Kopplungsgrade zum Ereignis n können in einer Adjazenz- bzw. Kopplungsmatrix K_n zusammengefasst, analysiert und explorativ gestaltet werden. Für das CS aus dem Beispiel der Abbildung 3-1 und Abbildung 3-9 ergibt sich die Kopplungsmatrix zu (5).

$$K_n = \begin{pmatrix} k_n^{dd} & k_n^{cd} & k_n^{vd} \\ k_n^{dc} & k_n^{cc} & k_n^{vc} \\ k_n^{dv} & k_n^{cv} & k_n^{vv} \end{pmatrix} = \begin{pmatrix} 1 & k_n^{cd} & k_n^{vd} \\ k_n^{dc} & 1 & k_n^{vc} \\ k_n^{dv} & k_n^{cv} & 0 \end{pmatrix} \quad (5)$$

Multiplikation der Kopplungsmatrix K_n mit dem Vektor der aktiven MPs aller Agenten des CS \vec{a}_{n-1} des vorherigen Ereignis $n-1$ beschreibt in allgemeiner Form den eingangsseitigen Wahrnehmungsschritt, den kognitionsinternen Bewertungsschritt und den ausgangsseitigen Manipulationsschritt der Agenten und ergibt den neu generierten Vektor der aktiven MPs \vec{a}_n (6).

$$\vec{a}_n = K_n \cdot \vec{a}_{n-1} \quad (6)$$

Y-Beispiel: Fahrer und Automation stellen zum Ereignis $n-1$ MPs d_{n-1} und c_{n-1} dar. Das Fahrzeug stellt das MP v_{n-1} dar. Der resultierende Vektor der aktiven MPs zum Ereignis n kann mit (7) angegeben werden. Dabei bildet der Additionsoperator (+) bei dem Matrix-Vektor-Produkt dieser linearen Abbildung den Prozess des „gewichteten Bewertens“ ab.

$$\begin{pmatrix} d_n \\ c_n \\ v_n \end{pmatrix} = \begin{pmatrix} 1 & k_n^{cd} & k_n^{vd} \\ k_n^{dc} & 1 & k_n^{vc} \\ k_n^{dv} & k_n^{cv} & 0 \end{pmatrix} \cdot \begin{pmatrix} d_{n-1} \\ c_{n-1} \\ v_{n-1} \end{pmatrix} = \begin{pmatrix} d_{n-1} + k_n^{cd} \cdot c_{n-1} + k_n^{vd} \cdot v_{n-1} \\ k_n^{dc} \cdot d_{n-1} + c_{n-1} + k_n^{vc} \cdot v_{n-1} \\ k_n^{dv} \cdot d_{n-1} + k_n^{cv} \cdot c_{n-1} \end{pmatrix} \quad (7)$$

Bereits bei einem relativ einfachen CS, wie es in der Abbildung 3-9 und in (7) gezeigt ist, liegt eine relativ hohe Komplexität der möglichen Ausprägung einer Agentenwechselwirkung vor. Es wird deutlich, dass die aktiven MPs der Agenten sich als Zeitreihen mit multiplen Abhängigkeiten von den EMs und OMs jeweils anderer Agenten entwickeln können. Die Auswirkungen der Interaktion zwischen den Agenten tragen durch komplexe Rückkopplungen zur CA-Wechselwirkung und damit zum CS-Verhalten bei.

Außerdem zeigt diese integrierte Beschreibung der Agentenwechselwirkung, dass die eingekoppelten Manipulationen die aktiven MPs und damit auch die MP-Generierung bzw. Aktivierung und die OMs der Agenten entlang der dichotomen kontinuierlichen Potenzialkette (R_{norm}) verschieben können (vgl. Abschnitt 3.3). Dies wird in Kapitel 4.2 aufgegriffen und zur Definition der allgemeinen Steuerstrukturen in CS (Aktionstensionen) mitverwendet.

Weil nach (6) die neuen aktiven MPs durch die Summenbildung beim Matrix-Vektor-Produkt in ihrem Betrag größer 1 sein könnten, müssen sie anschließend z.B. mit der Funktion (8) auf den Wertebereich R_{norm} wieder begrenzt werden. Es ist bereits argumentiert worden, dass MPs und Manipulationen im Betrag nie über 1 anwachsen, sondern in eine Sättigung laufen.

$$a: \mathbb{R} \mapsto R_{norm}, x_j \in R \mapsto \left\{ \begin{array}{ll} -1; & \text{für } x_j \leq -1 \\ 1; & \text{für } x_j \geq 1 \\ x_j; & \text{sonst} \end{array} \right\}, \text{ für } j \in N_0 \quad (8)$$

Y-Beispiel: Wenn der Fahrer nach links lenkt und von der Automation zum Rechtslenken angeregt wird, kann er maximal nach rechts umlenken. Dann würde seine OM eine Sättigung erreichen (mehr nach rechts lenken als maximal nach rechts lenken kann er nicht). Damit wäre der Fahrer durch den Bewertungsschritt vollständig „überzeugt“ bzw. „umgestimmt“.

Die vorgeschlagene Begrenzung stellt eine Vereinfachung des Modells dar. Näher an der Wirklichkeit kann eine agenteninterne Wechselwirkung in ihren selbstähnlichen Strukturen sein. Die möglichen „Manipulationsüberläufe“, die hier begrenzt werden, können z.B. zum Modellieren agenteninterner Lernprozesse oder Aufmerksamkeit beitragen, was in der vorliegenden Arbeit nicht weiter vertieft wird.

Die angenommene einfache unstetige Begrenzungsfunktion kann während der Anpassung des CS-Modells durch eine stetige Begrenzungsfunktion ersetzt werden, deren Grenzwert in ihrem oberen bzw. unteren Bereich gegen 1 bzw. -1 strebt, wie (9) oder ähnlich.

$$b: \mathbb{R} \mapsto R_{norm}, x_j \in R \mapsto \frac{2}{\pi} \cdot \arctan(x_j), \text{ für } j \in N_0 \quad (9)$$

Als sehr praktisch hat sich folgende Begrenzungsfunktion erwiesen (11). Diese begrenzt das aktive MP in Abhängigkeit von der Anzahl der zu ihm verbundenen MPs anderer Agenten.

$$f: \mathbb{R} \mapsto R_{norm}, x_j \in R \mapsto \left\{ x_j \mapsto x_j \cdot \frac{1}{k} \right\}, \quad (10)$$

für $j \in N_0, k := \text{Anzahl verbundener MPs}$

CS-Verhalten lässt sich mittels Zeitreihenanalysen untersuchen. Dabei sollte erwähnt werden, dass bei der hier vorgeschlagenen Art der Formalisierung von CS die IMs, die OMs und die aktiven MPs äquivalent zueinander sind. Nach (6) werden die aktiven MPs aller Agenten und somit auch ihre IMs und OMs zum Ereignis $n-1$ durch die Multiplikation mit der Kopplungsmatrix zu aktiven MPs, IMs und OMs des aktuellen Ereignis n transformiert. Dieses lässt sich mithilfe von (6) und (10) als allgemeine Berechnungsvorschrift (11) angeben^[Keh14].

$$\vec{a}_n = f(K_n \cdot \vec{a}_{n-1}) \quad (11)$$

Es sollte beachtet werden, dass in (11) für die Generierung des Vektors der aktiven MPs \vec{a}_n eine einfache lineare Abbildung (Multiplikation mit K_n) angenommen wurde. In sehr komplexen CS wird diese Transformation wahrscheinlich als komplexere nichtlineare Abbildung auftreten, so dass K_n entsprechend angepasst werden muss. Diese Anpassung kann durchgeführt werden, indem Kopplungsgrade z.B. selbst als MPs modelliert werden, was in der vorliegenden Arbeit nicht weiter vertieft wird.

Die (11) berücksichtigt nur die bekannten und modellierbaren Agenten (MPs). Die unbekannt, nicht exakt modellierbaren „Störeinflüsse“ lassen sich bei Bedarf durch dynamische Berechnung der sonst konstanten Kopplungsgrade in das CS-Modell integrieren. Störeinflüsse können z.B. durch eine Addition deren Erwartungswerte mit der EM des betroffenen Agenten

oder durch Anwendung einer Entropiefunktion auf die Kopplungsgarde (Kommunikationskanäle) in die (11) integriert werden, was in der vorliegenden Arbeit nicht weiter vertieft wird. Die (11) zeigt eine Rekursion, die eine Komplexitäts- oder Stabilitätsanalyse des CS-Modells erlauben kann, z.B. mittels der Eigenwertanalyse der Kopplungsmatrix, CS-Analyse mittels der Lyapunov-Exponenten etc., was in der vorliegenden Arbeit nicht weiter vertieft wird. Außerdem deutet diese Rekursion auf eine enge Verwandtschaft der erarbeiteten Formalisierung von CS mit zellulären Automaten^[Wom02] und somit einer bereits etwas breiteren Sammlung nützlicher Methoden und Konzepte aus dem Bereich „generative Systeme“.

Insgesamt lassen sich CS-Strukturen und -Verhalten allgemein als (12) angeben:

$$\text{Struktur eines CS: } S_\eta := \{A_\eta, M_\eta, K_\eta\} \quad (12)$$

$$\text{Verhalten eines CS: } V_\eta := S_{\eta-1} \rightarrow S_\eta$$

Die (12) definiert das Verhalten von CS als zeitliche Veränderung seiner Struktur. Damit das Verhalten des CS beeinflusst werden kann, muss also eine CS-Reorganisation vorgenommen werden, z.B. durch das Hinzufügen oder Entfernen von Agenten (MPs), MP-Transitionen oder Kopplungsgradänderungen. Diese Feststellungen sind nützlich für die Erarbeitung noch fehlender Konzepte der holistischen Gestaltung von CS in Kapitel 4 und das osGekoS-Werkzeug zur Lösungsexploration in Abschnitt 5.4.

Die vorgeschlagene Formalisierung der CS kann eine sinnvolle Unterstützung des holistischen Verständnisses und der Beschreibung von CS bieten. Dennoch sollte sie ein Hilfsmittel und kein alleiniges Werkzeug der CS-Gestaltung sein. Bei der Gestaltung von CS und der CA-Wechselwirkung bleiben immer noch die unbestimmte CA-Wechselwirkung und der Nichtdeterminismus des kognitiven Verhaltens. Diese sollten Hauptmotive für eine sorgfältige Gestaltung und können Gründe für eine mögliche Fehlgestaltung eines CS sein.

3.6 Verwandte Konzepte und Diskussion

Die bekannten Beschreibungsmittel für CS in Form von kognitiven Architekturen (vgl. Abschnitt 2.3.4) sind entweder auf einzelne Aspekte einer Kognition, wie Gedächtnis, Lernprozesse, ihre Wechselwirkungen oder speziell auf die menschliche Kognition zugeschnitten. Wenige Architekturen betrachten CS allgemein und als Ganzes. Sie tragen auch einige bis viele domänenspezifische und von der jeweiligen Entwicklungsumgebung abhängige Artefakte mit sich. MPS wurde entwickelt, um diesen Umständen mit einer möglichst allgemeinen Beschreibung der CS zu begegnen, um seinen Nutzen besonders beim Verstehen der CS zu entfalten. Als vorgeschlagenes Beschreibungsmittel des osGekoS ist es holistisch und behandelt durch die vorgeschlagene Selbstähnlichkeit der CS sowohl CAs als auch CS gleichermaßen.

Die bekannten Beschreibungsmittel für CS werden meist zwei unterschiedlichen Konzepten zugeordnet. Sie können symbolisch, konnektivistisch oder beides sein (vgl. Abschnitt 2.3.4). MPS betrachtet beides: Symbole und Verknüpfungen. Agenten (MPs) können als Symbole oder Subsymbole verstanden werden, die sich durch eine Wechselwirkung zu Symbolen entwickeln können. Gleichzeitig spielen konnektivistische Ansätze eine große Rolle, wie die kopplenden Manipulationen zwischen den Agenten (MPs) auf expliziten Kommunikationskanälen.

Der größte Unterschied (und ein womöglich wichtigster Vorteil) des MPS gegenüber den bekannten Beschreibungsmitteln liegt in der Gestaltungskompatibilität des Konzeptes, die später erarbeitet wird (vgl. Abschnitte 4.2.3 und 4.4). Durch eine entsprechende Schnittstelle bietet eine auf dem MPS-Konzept basierende gestaltungskompatible CS-Architektur für den Systemingenieur einen direkten Zugriff auf die zu gestaltenden CS-Struktur und -verhalten als Wechselwirkung seiner Agenten (vgl. Abschnitt 4.4). Der Gedanke der holistischen System- und Interaktionsgestaltung ist bei MPS dadurch etwas mehr im Vordergrund.

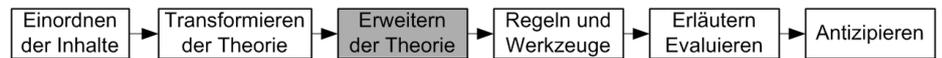
Die bekannten Beschreibungsmittel mögen einen erheblichen Vorteil in der genaueren Nachbildung der Agentenspezifika haben, wie aktuell in der Kognitionswissenschaft geltende Wahrnehmungs- oder Gedächtnismodelle. MPS ist dagegen deutlich allgemeiner und deskriptiver an diesen Stellen, was hier aber als Vorteil gesehen wird. Denn beim MPS-Konzept geht es vorrangig um das Verstehen und das Beschreiben von weiträumig verteilter allgemeiner Kognition und nicht von einzelnen Menschen oder ähnlichen CAs. Dennoch ist MPS kompatibel zu anderen Beschreibungsmitteln, insbesondere aufgrund seiner MAS-Basis. Von seinen Ideen her kommt MPS am nächsten den BDI-Architekturen und CHREST (vgl. Abschnitt 2.3.4).

3.7 Zusammenfassung

In diesem Kapitel wurden Grundlagen für das strukturfokussierte Verstehen und Beschreiben von CS erarbeitet. Diese wurden durch Transformation und Homogenisierung bekannter Grundlagen mit zusätzlichen Annahmen als strukturelle Ähnlichkeiten der CS formuliert. Demnach lassen sich alle CS als selbstähnliche robuste Strukturen von manipulierbaren und manipulierenden kognitiven und nichtkognitiven Agenten zerlegen. Dabei lassen sich sowohl die Agenten als auch ihre lokale Strukturen (Zustände) und globale Strukturen (Populationen) als MPs verstehen, die in dichotomen eskalierenden Ketten angeordnet sein können.

Über multiple multimodale Kommunikationskanalstrukturen können Agenten IMs empfangen, sie zusammen mit den EMs verarbeiten und OMs senden. Beim Verarbeiten von IMs und EMs werden neue aktive MPs generiert und als OMs auf die CS-Ebene gesendet, was einer CS-Reorganisation gleicht. Manipulationen aller Agenten wirken auf der CS-Ebene und erzeugen dort eine Wechselwirkung. Drei Arten der möglichen Wechselwirkung wurden eingeführt: Die negative, charakteristisch für Konflikte, fehlabgestimmtes Handeln und Konkurrenz, unbestimmte, charakteristisch für Ignoranz und Willkür und die positive, charakteristisch für abgestimmtes Handeln und Kooperation. Diese Wechselwirkungen definieren das CS-Verhalten und können durch MP-Transitionen vom Systemingenieur gestaltet werden.

Vorgestellte Grundlagen wurden anhand mehrerer Annahmen, Schlussfolgerungen, Abbildungen und Formeln aufgearbeitet, die im Verlauf der Arbeit aufgegriffen und weiterentwickelt werden. Der Mehrwert dieser Wissenstransformation und -homogenisierung liegt im vereinfachten Verständnis und der Beschreibung der sonst hochkomplexen CS. Wie sich das CS-Verhalten operativ gestalten lässt, wird im nächsten Kapitel anhand zweier Konzepte erarbeitet: Aktionstension mit der Nutzung der externalisierten Motivation der CAs und Arbitrierung als allgemeine Methode der Beeinflussung der CA-Wechselwirkung von fehlabgestimmten Handlungen und Konkurrenz zu abgestimmten Handlungen und Kooperation.



4 Gestalten kognitiver Systeme

Nachdem erarbeitet wurde, wie CS verstanden und beschrieben werden können, soll in diesem Kapitel auf das Gestalten von CS eingegangen werden. Durch die Definition und die Formalisierung der CS als MPS in Abschnitt 3 wurde dafür eine wichtige Grundlage gelegt. CS können als MPS dekomponiert, modelliert und analysiert werden, was drei wichtige Schritte der Gestaltung sind (vgl. Abschnitt 2.2.1). In diesem Kapitel soll es darum gehen, wie eine CS-Gestaltung weiter und konkreter bis zum Gestaltungserfolg geführt werden kann.

Eines der Ziele und Nutzen dieser Arbeit ist das Formulieren noch fehlender Konzepte für osGekoS. Wie in Abschnitt 3.4 erklärt, wechselwirken CAs über auf die CS-Ebene gesendete Manipulationen. Damit diese Manipulationen und die resultierende CA-Wechselwirkung (Konkurrenz, Willkür, Kooperation) effektiv beeinflusst werden können und zwecks holistischer Gestaltung von CS, sollen zwei weitere Konzepte erarbeitet werden. Diese vervollständigen das MPS-Konzept für die Formulierung der gesamten osGekoS-Theorie.

Zum einen ist es das Konzept der Aktionstensionen, die die inhärente Motivation aller CAs externalisieren und homogenisieren. Sie können als allgemeine Steuerstrukturen innerhalb eines CS die Manipulationen verschiedener CAs miteinander verbinden und zeitlich abstimmen. Mit Motivation ist hier der Prozess der Zielsetzung der CAs gemeint. Diesen gilt es über alle Agenten des CS hinweg zu verstehen, zu beschreiben und als Aktionstensionen zu externalisieren. Operationalisierte Aktionstensionen können nützlich sein, um technische Verfahren und maschinelle Agenten für die Beeinflussung des CS-Verhaltens zu entwickeln und einen gemeinsamen Referenzraum der Agentenhandlungen zu beschreiben.

Zum anderen ist es das Konzept der Arbitrierung, die eine allgemeine Methode der holistischen Gestaltung von CS darstellt. Aktionstensionen nutzend, kann durch die Arbitrierung eine Reorganisation eines CS herbeigeführt werden, entweder durch eine überwachte Selbstorganisation der CAs (implizit) oder durch eine Fremdorganisation (explizit). Mit Organisation (auch Struktur) ist hier die Gesamtheit aller Manipulationen und der damit gekoppelten Agenten gemeint. Fehlabgestimmte Manipulationen im CS gilt es abzustimmen und Kopplungen zwischen den Agenten gilt es zu verändern mit dem Ziel, die CA-Wechselwirkung von der negativen (Konkurrenz) zur positiven (Kooperation) zu verschieben. Dafür können maschinelle Agenten, die Arbitrer, entwickelt und dem kognitiven System hinzugefügt werden.

Aufbauend auf grundlegenden Gedanken zum Gestalten von CS (Abschnitt 4.1), werden beide Konzepte in Abschnitten 4.2 (Aktionstension) und 4.3 (Arbitrierung) erarbeitet. In Abschnitt 4.4 wird die gesamte bis dahin zur Verfügung stehende Theorie verwendet, um eine gestaltungskompatible CS-Architektur sowie eine dezidierte Schnittstelle für die Gestaltung der Arbitrer und der Arbitrierung zu entwickeln. Im letzten Abschnitt dieses Kapitels wird der gesamte neue theoretische Hintergrund von osGekoS kurz zusammengefasst.

4.1 Grundlegendes zum Gestalten kognitiver Systeme

In diesem Abschnitt soll auf Grundlagen eingegangen werden, wie das CS-Verhalten als Wechselwirkung seiner Agenten durch den Systemingenieur beeinflusst werden kann. Die po-

sitive CA-Wechselwirkung (Kooperation) kann angeregt und die negative (Konkurrenz) gehemmt werden. Dies kann unter der Berücksichtigung des zeitlichen Horizonts des Agentenverhaltens geschehen und das Ziel einer solchen Gestaltung sollte sein, eine konsistente und gut verständliche Interaktion zwischen allen Agenten des CS zu entwickeln.

4.1.1 Beeinflussung des Agentenverhaltens

Das Verhalten eines CS kann durch die Wechselwirkung seiner Agenten definiert werden. Mittels spezieller Strategien, Signale und zusätzlicher Agenten, wie HMI-Agenten^[Boy94], kann der Systemingenieur bereits bei der Gestaltung dafür sorgen, dass die mögliche negative CA-Wechselwirkung gehemmt und die positive angeregt wird. Dafür muss er die Handlungen bzw. die Manipulationen der CAs in seinem CS beeinflussen können. In Abschnitt 3.4 wurde gezeigt und in Abschnitt 3.5 formalisiert, wie die CAs über Kommunikationskanäle sich gegenseitig manipulieren können. Drei grundlegende Arten der möglichen Beziehungen zwischen den CAs wurden beschrieben: Anregung, Hemmung und Ignoranz. Diese Beziehungen können für die Beeinflussung des Agentenverhaltens verwendet werden.

- **Anregung:** Objektiv richtige Handlungen können *angeregt*, *bestärkt* oder *erzungen* werden. Richtig agierende Agenten, wie Fahrer oder fremdüberwachte Automationen, können z.B. in die geteilte Fahrzeugkontrolle *eingekoppelt* oder durch adäquate Maßnahmen *belohnt* werden. Mit angenehm wirkender Haptik, Farben, Tönen (positive Salienz) können richtige Fahrerhandlungen angeregt oder bestärkt werden. Durch eine steigende *Rigidität* der Automation können sie im Notfall erzwungen werden. Fahrer könnten ihr sicheres und effizientes Fahrverhalten beibehalten lernen. Eine Automation kann mittels des maschinellen Lernens einüben, wie sie zu handeln hat.
- **Hemmung:** Objektiv falsche Handlungen können *abgestellt*, *gehemmt* oder *verhindert* werden. Falsch agierende Agenten, wie Fahrer oder fremdüberwachte Automationen, können von der geteilten Fahrzeugkontrolle *entkoppelt* oder durch adäquate Maßnahmen *bestraft* werden. Mit unangenehm wirkender Haptik, Farben, Tönen (negative Salienz) können falsche Fahrerhandlungen gehemmt werden, bis der Fahrer von der Fahrzeugkontrolle komplett *entkoppelt* ist. Bei einer Unaufmerksamkeit des Fahrers z.B., kann auf diese Weise sein Kontrolleinfluss gegenüber dem der Automation reduziert werden. Fahrer könnten so ein sicheres und effizientes Fahrverhalten erlernen. Eine Automation kann mittels des maschinellen Lernens einüben, wie sie zu handeln hat.
- **Ignoranz:** Objektiv unverständliche oder nicht erfassbare Handlungen können aus der Sicht der Verhaltensbeeinflussung ignoriert und unter Beobachtung mit dem Zweck der „Richtig-Falsch-Klassifizierung“ gestellt werden. Dies kann entweder durch die komplette Auskopplung der Automation von der Fahrzeugkontrolle, Unsicherheitsrückmeldung^[HeD14] der Automation oder mit schwach salienten Signalen geschehen.

Objektive Einschätzung, ob eine Handlung richtig oder falsch ist, ist eine anspruchsvolle Aufgabe für den Systemingenieur und für das Zielsystem. Solche Einschätzungen können nur in Bezug auf einen *gemeinsamen Referenzrahmen* (engl. Common Frame of Reference - CoFoR^[Hoc01]) gemacht werden. Einerseits muss der Systemingenieur geltende und relativ abstrakte Bestimmungen, wie die Wiener Konvention („Every driver shall at all times be able to con-

trol his vehicle or to guide his animals"^{15[UNE68]}) und die Straßenverkehrsordnung beachten. Diese lassen sich für konkrete Situationen schwer operationalisieren. Andererseits werden Fahrzeuge zunehmend automatisiert und der Systemingenieur muss alle Einschränkungen und Möglichkeiten der technischen und der menschlichen Komponenten seines CS beachten. Diese lassen sich ebenfalls schwer operationalisieren und auf eine operative Systemgestaltung anwenden. Abhilfe kann ein sorgfältig entwickeltes CoFoR schaffen. Allgemeine Metriken dafür können die Aktionstensionen sein, die im Abschnitt 4.2 ausführlich erarbeitet werden.

Die Rollenverteilung innerhalb des zu gestaltenden CS ist ebenfalls wichtig^[FH12], wenn man eine Handlung als richtig oder falsch bewertet und sie daraufhin anregt, hemmt oder ignoriert. Wenn die Rollen unklar oder missverständlich kommuniziert sind, kann die Akzeptanz des Systems sinken. Dies kann sich auf die Willkür der Handlungswahl, beeinflusst durch Aufmerksamkeits-Komponente der CA-Kopplungen (4), negativ auswirken. Die vom Systemingenieur gewünschte Verhaltensbeeinflussung kann sich ins Gegenteil der beabsichtigten Gestaltung entwickeln und das kognitive System wäre instabil.

Um solche Probleme zu adressieren, sollten die Rollenverteilung und die damit verbundene Autorität vorher explizit gestaltet sein. Die Höhe der Autorität könnte von sehr wenig bis sehr viel variieren. Bei wenig Autorität dürfte die Automation den Fahrer lediglich informieren oder Vorschläge unterbreiten. Bei viel Autorität könnte die Automation selbständig in die Fahrzeugführung eingreifen, ohne den Fahrer darüber in Kenntnis zu setzen oder ihm gar Anweisungen erteilen. Dies muss durch fest zugewiesene und möglichst konstant bleibende Rollenverteilung bzw. Kooperationsmodi^[Hoc07] bekannt sein oder deutlich gemacht werden.

Agentenverhalten kann *direkt* oder *indirekt*, z.B. auf dem Umweg über andere Agenten, beeinflusst werden. Dies eröffnet zwei Möglichkeiten in der operativen CS-Gestaltung:

1.) *Direkte Manipulationen*^[HuH88], wenn Agenten einander über reelle, wie mechanische Stellteile, oder virtuelle, wie Steer-by-Wire, Stellteile direkt manipulieren. Direkte Manipulationen können eine bessere Wahl in CS mit nur einem CA sein. Beispielsweise nutzt man bei der Desktop-Metapher für die Gestaltung der Nutzerinteraktion mit einem Computer (vgl. Abschnitt 2.3.6) vorwiegend direkte Manipulationen. Damit kann die Rückkopplungskomplexität der Manipulationen deutlich reduziert werden, da das Manipulationsmedium (Computermaus, Mauszeiger) vom Nutzer gut „ausgeblendet“ werden können. Manuelle Fahrzeugführung ist ein Beispiel der direkten Manipulation, wo über Stellteile das Fahrzeug direkt vom Fahrer manipuliert wird. In hochautomatisierten Fahrzeugen mit mindestens zwei CAs (Fahrer und Automation) stoßen direkte Manipulationen allerdings an ihre Grenzen.

2.) *Indirekte Manipulation*, wenn Agenten einander über den Umweg zusätzlicher Agenten manipulieren. Solche zusätzlichen Agenten können potenzielle oder akute Konflikte rechtzeitig erkennen, umgehen oder lösen (vgl. dazu Abschnitt 4.3).

Y-Beispiel: Für das hochautomatisierte Fahrzeug und seine Zerlegung als Zustandsübergangsgraph aus Abbildung 3-9 sei angenommen, dass der Fahrer d und die Automation C das Fahrzeug V zum Ereignis $n-1$ direkt ($k_n^{dv} = k_n^{cv} = 1$) mittels der OMs mit dem Wert M steu-

¹⁵ Jeder Fahrer muss jederzeit in der Lage sein, sein Fahrzeug zu kontrollieren oder seine Tiere führen. EÜ.

ern, z.B. beide CAs lenken gleichzeitig nach rechts. Fahrer und Automation manipulieren sich nicht direkt ($k_n^{dc} = k_n^{cd} = 0$), aber es gibt ein Feedback vom Fahrzeug zum Fahrer und zur Automation hin ($k_n^{vd} = k_n^{vc} = 1$), z.B. über das Lenkrad. Der v (Fahrzeug) besitzt noch keine OM ($v_{n-1} = 0$). Der dazugehörige Zustandsübergangsgraph ist in der Abbildung 4-1 dargestellt.

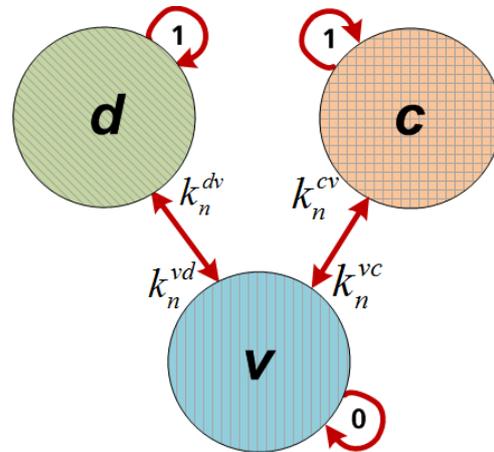


Abbildung 4-1: Dekomposition eines hochautomatisierten Fahrzeuges als Zustandsübergangsgraph

Mit (7) ergibt sich die Zeitreihenentwicklung für den Vektor der aktiven MPs \vec{a}_n zu (13):

$$\begin{pmatrix} d_n \\ c_n \\ v_n \end{pmatrix} = \begin{pmatrix} 1 & k_n^{cd} & k_n^{vd} \\ k_n^{dc} & 1 & k_n^{vc} \\ k_n^{dv} & k_n^{cv} & 0 \end{pmatrix} \cdot \begin{pmatrix} d_{n-1} \\ c_{n-1} \\ v_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} M \\ M \\ 0 \end{pmatrix} = \begin{pmatrix} M \\ M \\ 2M \end{pmatrix} \quad (13)$$

Die IM am Fahrzeug (v_n) zum Ereignis n wird doppelt so groß sein, wie sie bei einer rein manuellen Steuerung des Fahrzeuges wäre. Wenn der Fahrer eine OM mit dem Wert $(-M)$ und die Automation mit dem Wert (M) sendet, z.B. der Fahrer lenkt nach links und die Automation nach rechts, wäre die IM des Fahrzeuges zum Ereignis n gleich 0 . Solche Manipulationseffekte können zur negativen CA-Wechselwirkung bis hin zu einer CS-Instabilität führen. Neben dem relativ trivialen Ergebnis macht diese Berechnung die Notwendigkeit eines zusätzlichen Agenten deutlich, der mittels geeigneter Strategien und Signale die IMs des Fahrzeuges stabilisieren kann. Komplexere Rechenbeispiele hierfür werden in Abschnitt 4.3 behandelt.

4.1.2 Zeithorizont der Beeinflussung des Agentenverhaltens

Eine wichtige Grundlage für die CS-Gestaltung ist die Berücksichtigung des zeitlichen Horizonts im CA-Verhalten^[KeT13]. Rasmussen^[Ra86a] schlägt z.B. vor, dass das menschliche Kontrollverhalten auf drei Ebenen liegt: Fertigungs-, regel- und wissensbasierte Ebene. Entsprechend diesen Ebenen, schlägt er für die Interaktionsgestaltung die Verwendung von Signalen, Zeichen und Symbolen vor. Das Argument dafür ist, dass die Verarbeitung der genannten Signalarten unterschiedliche Verarbeitungszeit benötigt. Sie wird länger, wenn sie sich von der Signal- zur Symbolverarbeitung und von der fertigungs- zur wissensbasierten Ebene verlagert.

In der Theorie über das Situationsbewusstsein und -bewertung^[Eny95] gibt es ebenfalls drei Ebenen: Wahrnehmen, Verstehen und Projizieren. Diese werden als eine ineinander greifende Abfolge der kognitiven Informationsverarbeitung angenommen. Aufgrund der Notwendigkeit, Informationen der inneren Ebenen (Wahrnehmen und Verstehen) zu verarbeiten, würde

die gesamte Informationsverarbeitung, inklusive der nächsten Ebene (Projizieren), im Schnitt und insgesamt ein längeres Zeitfenster benötigen. Das würde bedeuten, dass Prozesse, die das Projizieren innerhalb der Situationsbewertung enthalten, mit längeren Zeitkonstanten behaftet sein sollten. Etwas kürzere Zeitkonstanten würde man bei Prozessen bis zur Ebene des Verstehens vorfinden. Am kürzesten wären dann die Prozesse des Wahrnehmens.

Es gibt ähnliche Überlegungen mit dem Schwerpunkt der CA-Kooperation^{[Cai98][Hoc01]}. Dort findet man ebenfalls drei Ebenen vor: Aktions-, Plan- und Meta-Ebene. Diese können ebenfalls mit steigenden Zeitkonstanten auf dem Weg von der Aktions- zu der Metaebene verstanden werden. Ähnliche Tripel in Bezug auf den Zeithorizont finden sich z.B. in der Wirtschaftswissenschaft, Militärwesen und Spieltheorie^[SP13e]. Das sind: Operative, taktische und strategische Ebene, die auch steigende Zeitkonstanten auf dem Weg von der operativen zu der strategischen Ebene haben, z.B. bei der Planung oder der Ausführung von Aufgaben.

Bei einer Gestaltung von CS und CA-Wechselwirkung interessiert sich der Systemingenieur für Effekte, die durch eine korrekte Wahl einer CA-Verhaltensbeeinflussung und eine korrekte Parametrierung der dazu gehörigen Interaktionssignale erreicht werden kann. Darum erscheint es als sinnvoll, alle diskutierten Begriffe nach ihrem zeitlichen Horizont zu gruppieren. So können drei Ebenen der erwarteten Effekte des Einflusses auf das Agentenverhalten definiert werden. Es sind Ebenen der langfristigen, der kurzfristigen und der unmittelbaren Effekte der Wirkung der Interaktions- und der CA-Wechselwirkungsstrategien (Abbildung 4-2).

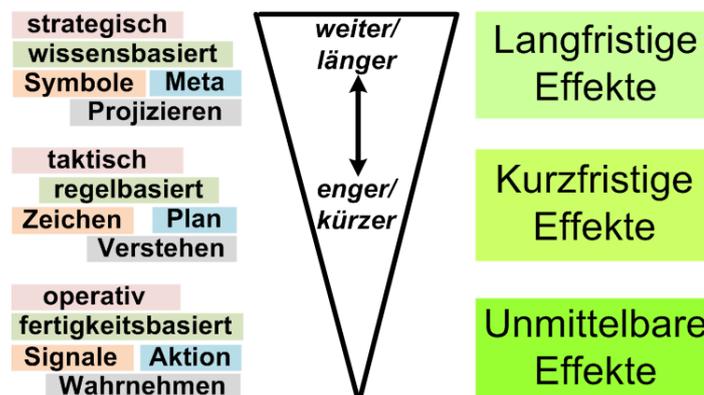


Abbildung 4-2: Zeithorizont, Zeitkonstanten und Effekte der Beeinflussung des CA-Verhaltens können durch eine Gruppierung der Begriffe der Konzepte der kognitiven Informationsverarbeitung ermittelt werden

Beispiel: Im Bereich hochautomatisierter Fahrzeuge können sich auf der Ebene der langfristigen Effekte die Verhandlung und die Entscheidung über die Wahl der Assistenzmodi, wie Automationslevel^[Pa500] oder der Kooperationsmodi^[Hoc01], befinden. Hier kann auch in die nahe Zukunft projizierte Manöverwahl bei geteilter Kontrolle^[ShV78] zwischen dem Fahrer und der Automation befinden. Des Weiteren kann hier die strategische Koordination der CAs in Richtung einer Kooperationsanregung oder einer Konkurrenzhemmung stattfinden. Auf der Ebene der kurzfristigen Effekte kann sich die Verhandlung des regelbasierten Verhaltens, wie die Wahl der Trajektorien, der seitlichen Abweichung, der Geschwindigkeit und des Abstandes zu anderen Fahrzeugen befinden. Auf der Ebene der unmittelbaren Effekte kann sich die Verwaltung des unmittelbaren Zugriffs auf die Fahrzeugkontrolle durch die CAs platziert werden.

Berücksichtigung des zeitlichen Horizonts im CA-Verhalten gibt dem Systemingenieur Hinweise darauf, wie und mit welchen Zeitkonstanten er sein CS dekomponieren und auf Agenten

verteilen sollte. Er bekommt auch Hinweise darauf, welche Art und welche semantische und syntaktische Ausprägung der Signale er für die Gestaltung der CA-Wechselwirkung wählen sollte. Diese Überlegungen sind ein wichtiger Teil der osGekoS-Theorie und finden ihren Platz in der später vorgestellten gestaltungskompatiblen Architektur der kognitiven Systeme.

4.1.3 Zusammenfassung

In diesem Abschnitt ging es um die Grundlagen, wie die operative CS-Gestaltung nach dem MPS-Konzept weiter präzisiert werden kann. Es wurde argumentiert, dass kognitives Verhalten wie auch Effekte des Einflusses auf ein CS über drei Ebenen verteilt werden können: Langzeit-, Kurzzeit- und unmittelbare Ebene der Effekte. Dort kann das CA-Verhalten durch die Beeinflussung der CA-Manipulationen gehemmt, ignoriert oder angeregt werden. Dies kann mittels konsistent gestalteter Interaktionsstrategien und -signale bewerkstelligt werden. Wie man diese Grundlagen operativ anwenden kann, darum geht es in weiteren Abschnitten.

4.2 Aktionstension

In Abschnitt 3.1 wurde gezeigt, dass jedes CS sich strukturell in CAs und NAs zerlegen lässt. Alle Agenten stellen ihre aktiven Manipulationspotenziale durch OMs nach außen auf die CS-Ebene dar. Dabei kann es zur negativen, unbestimmten oder positiven Wechselwirkung kommen, die der Systemingenieur gestalten möchte. CAs sind aber autonom und können sich nichtdeterministisch, willkürlich verhalten. Die Frage an dieser Stelle lautet: Wie können Manipulationen unterschiedlicher CAs so miteinander verknüpft werden, dass es möglich wird, sie gemeinsam zu beeinflussen? Die Anforderung an die Behandlung dieser Frage ist: Eine allgemeine Steuerstruktur muss gefunden werden, durch deren Nutzung eine Verknüpfung und eine gleichzeitige Beeinflussung von mehr als einer CA-Manipulation möglich ist.

4.2.1 Problemstellung: Zeitlich fehlabgestimmte Agentenhandlungen

Laut der Definition (vgl. Abschnitt 3.1) bestehen CS, wie Mensch/Maschine- oder Fahrer /Automation/Fahrzeug/Umwelt-Systeme, aus mindestens einem CA, wie einem Fahrer (d) und einer Automation (c). Gehörend zum selben CS, können diese CAs gleichzeitig ein technisches Gerät steuern, wie ein in die nahe Umwelt eingebettetes Fahrzeug (v). Dies kann mit einem Zustandsübergangsgraphen beschrieben werden (vgl. Abschnitt 3.5). Der Fahrer und die Automation führen eine gemeinsame Aufgabe aus, z.B. ihr Fahrzeug unfallfrei durch die Umwelt zu bewegen. Die Fahrzeugkontrolle kann zwischen ihnen geteilt sein. In einer solchen Situation gibt es mindestens fünf Wirkungskreise (blaue Pfeile) (Abbildung 4-3)^[KeH12].

d und c stellen ihre aktiven MPs als Manipulationen am Fahrzeug (m_n^{dv}, m_n^{cv}) dar. Dort wird die gemeinsame Handlung bestimmt, die das Fahrzeug innerhalb der Umwelt steuert. Des Weiteren werden dort die Rückmeldungen an d und c (m_n^{vd}, m_n^{vc}) bestimmt, was zwei innere Wirkungskreise schließt. d und c können miteinander interagieren (m_n^{dc}, m_n^{cd}), was zwei weitere innere Wirkungskreise schließt. Der äußere Wirkungskreis wird geschlossen, wenn die gemeinsame Handlung die gemeinsame Aufgabe der beiden CAs beeinflusst.

Der Fahrer ist meistens fähig, ein hochautomatisiertes Fahrzeug zu kontrollieren, z.B. mittels mentaler Modelle über die Aufgabe, das Verhalten der Automation und der Fertigkeit der

Fahrzeugführung. Die Automation kann so gestaltet sein, dass sie den Fahrer, die Fahrer /Automation-Schnittstelle, die Fahrzeugdynamik berücksichtigt^[Lök08] und die Fahrzeugkontrolle beherrscht. Die genannten im CS verteilten CAs kontrollieren gleichzeitig das Fahrzeug und können aufgrund ihrer Autonomie *zeitlich fehlabgestimmt* bzw. asynchron handeln, was die Fahrzeugkontrolle destabilisieren kann. Die Synchronisierung der beiden „Fahrzeugkontroll-einheiten“ geschieht gewöhnlich nur indirekt über die Wahrnehmung der Situation durch die beiden CAs. Dies kann zur suboptimalen Systemstabilität, Robustheit und Usability bei der geteilten Systemkontrolle führen¹⁶. Ein auf der Systemebene, d.h. außerhalb des Fahrers und der Automation, vorhandenes, operationalisierbares und technisch nutzbares Synchronisierungskonzept wird benötigt. Dies können allgemeine Steuerstrukturen in CS übernehmen.

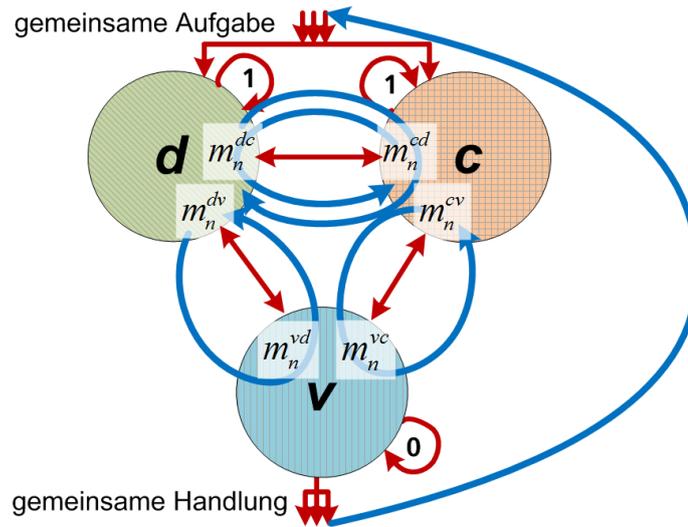


Abbildung 4-3: Graph einer Fahrer/Automation/Fahrzeug -Systemkonfiguration bei geteilter Kontrolle

4.2.2 Lösungsansatz: Steuerstrukturen in kognitiven Systemen

Wechselt man von der technischen zur psychologischen und zur Human Factors Perspektive, entdeckt man folgendes: Theorien und Modelle aus den Bereichen Kognitions- und Motivationsforschung stellen eine Reihe von nützlichen Konzepten für die Beschreibung der menschlichen Kognition, seiner Informationsverarbeitung, seines Verhaltens und der Verbindungen dazwischen. Einige Theorien legen Wert auf das Verständnis und die Beschreibung der im Menschen inhärenten Motivations- bzw. Zielsetzungsprozesse. Es wird mit Metriken gearbeitet, die die Motivation zu beschreiben und zu operationalisieren versuchen.

Lewin^{[Len38][HeV02]} behauptet, dass Menschen in bestimmten kognitiven Zuständen „äußere psychologische Kräfte“ erleben. Diese verursachen eine Lokomotion zu anderen Zuständen. Lewin nennt die „inneren Kräfte“ für die Lokomotion *Tensionen* (engl. tension). Wenn wir z.B. im Zustand zu langsames Fahren sind, verursacht dieser eine Anspannung. Wir führen gezielt Handlungen aus, um schneller zu werden, in den Zustand optimales Fahren zu wechseln und eine Entspannung zu erreichen. Mit dem „ökologischen Ansatz“ führt Gibson^[Gin77] *Aufforderungen* (engl. affordance) als eine Objektqualität ein, die Handlungsmöglichkeiten eröffnet und mit dem aktuellen Zustand des Menschen wechselwirkt. Wenn wir also zu lang-

¹⁶ Wegen des Fokus der Arbeit auf die Systemebene muss erwähnt werden, dass an dieser Stelle die psychologischen und technischen Ursachen der Asynchronität auf Agentenebene, wie unterschiedliche mentale Modelle, Wissen, Regeln, Fähigkeiten und ihre informationstechnische und kognitive Verarbeitung, noch bewusst außen vor gelassen sind.

sam sind und ein Gaspedal sehen oder fühlen, wird es uns anziehen, es zu benutzen. Sinnhaftigkeit und Verfügbarkeit einer Handlung tragen hierbei zu der Handlungswahl bei.

Beide Konzepte (Tensionen und Aufforderungen) können nützlich für die CS-Gestaltung sein. Sie operieren mit Begriffen, die sich von der Bedeutung her in der Nähe von physikalischen und technischen Begriffen befinden, wie Kraft, Druck, Spannung und Tension. Das eröffnet die Möglichkeit, das Verhalten der beiden Teilsysteme Mensch und Maschine bzw. Fahrer und Automation mit gemeinsamen Begriffen und Konzepten zu verstehen und zu beschreiben.

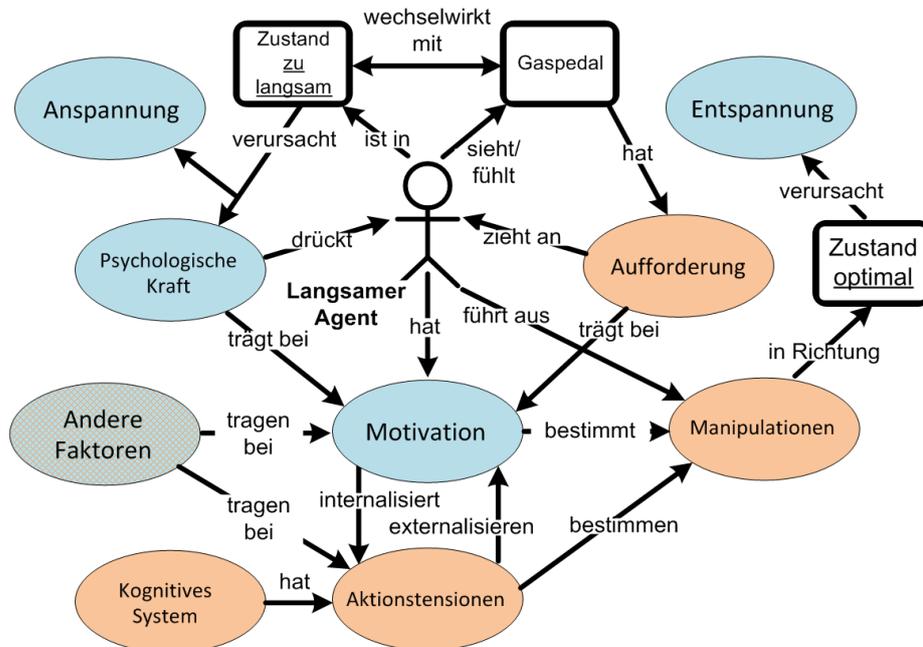


Abbildung 4-4: Concept-Map zum Zusammenhang zwischen der psychologischen Kraft, der Aufforderung, der Aktionstension und der Manipulation eines „langsamen Agenten“. Blaue ovale sind agentenintern, orangen sind extern

Ideen beider Konzepte können in einer prospektiv operationalisierbaren und technisch nutzbaren Steuerstruktur *Aktionstension* verbunden werden. Damit können die in Menschen inhärenten Motive und auf bestimmte Ziele gerichtete Motivation zum Zweck der Gestaltung auf die CS-Ebene externalisiert und mit denen der Maschinen homogenisiert werden. Der Zusammenhang zwischen den genannten Konzepten und Begriffen ist an dem verwendeten „Beschleunigungsbeispiel“ in der Concept-Map der Abbildung 4-4 grafisch veranschaulicht.

Definition: Aktionstensionen sind Steuerstrukturen auf der CS-Ebene, die in jedem Agenten inhärente Motivationsstrukturen externalisieren und homogenisieren. Sie bilden dichotome Ketten angespannter MPs mit Entwicklungstendenzen zu den entspannten MPs. Alle Agenten des CS internalisieren MPs verschiedener Aktionstensionen als ihre Motivationsstrukturen.

Annahmen: 1). Jedes CS kann als getriggert durch Aktionstensionen in Richtung entspannter MPs verstanden und beschrieben werden. 2). Operationalisierte Aktionstensionen lassen sich bei einer CS-Gestaltung für zeitliche und qualitative Abstimmung des CS-Verhaltens verwenden. 3.) Menschen halten ihre inhärente Motivation in entspannten Zuständen und mittels daraus externalisierter Aktionstensionen können Maschinen so gestaltet werden, dass sie dies ebenfalls und kompatibel zum Menschen tun^[KeH12]. 4). Die Anspannung der MPs einer Aktionstension nimmt ab, wenn CAs in Richtung des dazugehörigen entspannten MP handeln und sie nimmt zu, wenn CAs entgegen des entspannten MP handeln.

positive verschoben werden. Dafür wird an das CS von der auf die CS-Ebene externalisierten Aktionstension eine Synchronisierungsaufgabe gesendet, die den C weiterhin sein entspanntes MP (zo) und den d anstatt (zo) das MP (z+) generieren bzw. aktivieren lässt.

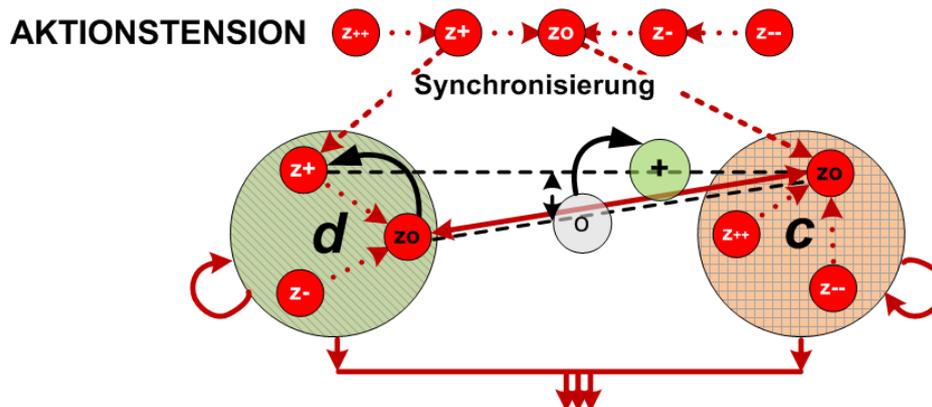


Abbildung 4-6: Erweiterung des MPS-Konzeptes durch die Aktionstension. Aktionstension kann die MP-Generierung bzw. -Aktivierung und damit die OMs aller Agenten, sowie ihre Interaktion zeitlich untereinander abstimmen

Y-Beispiel: Die gezeigte Situation lässt sich wie folgt aufschlüsseln: d ist der Fahrer und kann grundsätzlich MPs „Fahren links“ (z-), „Fahren nicht (selbst)“ (zo) und „Fahren rechts“ (z+) mit dazugehörigen OMs „links lenken“, „nicht lenken“ und „rechts lenken“ darstellen. Aktuell stellt er „Fahren nicht (selbst)“ dar und lenkt nicht. c ist die Automation und kann grundsätzlich MPs „Ausweichen links“ (z--), „Fahren nicht (selbst)“ (zo) und „Ausweichen rechts“ (z++) mit dazugehörigen Manipulationen „stark links lenken“, „nicht lenken“ und „stark rechts lenken“ darstellen. Aktuell stellt sie auch das MP „Fahren nicht (selbst)“ dar, was bedeuten kann, dass es sich um eine Ausweichautomation in Notfällen handelt.

Wenn das Fahrzeug weiterhin von keinem CA gelenkt wird, ist ungewiss, was passieren wird. Denn sowohl der Fahrer als auch die Automation kann plötzlich merken, dass das Fahrzeug doch gelenkt werden muss. Es handelt sich um eine unbestimmte CA-Wechselwirkung. Nun passiert folgendes: Die Automation wird im MP „Fahren nicht (selbst)“ belassen (zo), der Fahrer bekommt aber eine IM als Hinweis auf die Notwendigkeit, das MP „Fahren rechts“ (z+) zu aktivieren und nach rechts zu lenken, z.B. durch eine akustische Ansage. Der Fahrer aktiviert das MP „Fahren rechts“, lenkt nach rechts und verhindert eine mögliche Destabilisierung des CS. Die Synchronisierungsaufgabe, die Automation „nicht lenken“ und den Fahrer „rechts lenken“ zu lassen, kann ein maschineller Agent übernehmen (vgl. Abschnitt 4.3).

Beim Verarbeiten der IMs mit EMs kann die MP-Transition und damit die OM der Agenten auch ein angespanntes MP generieren bzw. aktivieren. Trotzdem haben die EMs der Agenten eine Tendenz zum entspannten MP. Dies kann bei der Operationalisierung von Aktionstensionen helfen, die entspannten MPs als Referenz für die dazugehörigen angespannten und in dichotomen Ketten angeordneten MPs zu finden, zu erarbeiten oder festzulegen. Außerdem greift die Abbildung 4-6 bereits etwas vor und zeigt ein Beispiel einer expliziten Arbitrierung (vgl. Abschnitt 4.3).

Strategisch betrachtet, können Aktionstensionen eine der „vorkognitiven Strukturen“^[Cai98] sein, die für eine erfolgreiche Formalisierung der CSE-Prozesse gefordert werden. Aufgrund der gleichzeitigen Kompatibilität der Aktionstensionen zur menschlichen Motivation und zu technischen Begriffen und Regelwerken kann der Systemingenieur damit effektiv gebrauchts-

taugliche und technisch robuste CS gestalten. Aktionstensionen bilden hierfür ein nützliches strukturfokussiertes Bindeglied zwischen allen CA-Manipulationen eines CS.

Taktisch betrachtet, können Aktionstensionen für die Analyse und die Modifikation des CS-Verhaltens durch die Verschiebung der MPs der Agenten und der CA-Wechselwirkung nützen. Dafür kann der Systemingenieur eine Reorganisation des zu gestaltenden CS entwickeln und der Anfälligkeit der CS für willkürliches Verhalten entgegenwirken. Mittels Aktionstensionen kann auch ein operativ nutzbares CoFoR der Agentenhandlungen entwickelt werden.

Aktionstensionen können aus CAs durch den induktiven Ansatz einer Exploration oder eine Deduktion der möglichen MPs und damit verknüpften Manipulationen externalisiert und operationalisiert werden. Angespannte MPs können einen CoFoR der CA-Manipulationen beschreiben. Durch das Auffinden der entspannten MPs können allgemeine Bezugspunkte in dieses System gesetzt werden. Die operationalisierten Transitionen zwischen den MPs können als Trigger für zusätzlich in das CS injizierbare Manipulationen nützen. Die Operationalisierbarkeit der Aktionstensionen bleibt die wichtigste Einschränkung ihrer Verwendbarkeit.

Operativ kann für das zu gestaltende CS ein Modell entwickelt werden. Durch die Analyse des bestehenden CS-Verhaltens am Modell können Konkurrenz- und Kooperationspotenziale aufgedeckt werden. Mit externalisierten Aktionstensionen kann eine explorative Reorganisation des Modells vorgenommen und damit Strategien für Konkurrenzhemmung und Kooperationsanregung entwickelt werden. Mit operationalisierten Aktionstensionen können dann für entwickelte Strategien entsprechende Signale gestaltet werden, die das bestehende CS-Verhalten positiv beeinflussen können. Diese Signale können technisch umgesetzt werden, indem der Systemingenieur dem CS zusätzliche Agenten hinzufügt, die die Manipulationen der CAs des ursprünglichen CS zeitlich und qualitativ abstimmen.

4.2.3 Semi-Formale Repräsentation der Aktionstensionen

Aktionstensionen lassen sich semi-formalisieren und können für die Analyse und die Gestaltung von CS mitverwendet werden. Sie sind als Ketten dichotomer angespannter MPs mit einer gemeinsamen Referenz in Form der entspannten MPs definiert (vgl. Abschnitt 4.2.2). Das macht sie kompatibel zu der sonstigen Formalisierung von CS mit dem MPS-Konzept (vgl. Abschnitt 3.5). Der Selbstähnlichkeit von CS nach, können Aktionstensionen als verkettete, assoziierte MPs bzw. Agenten verstanden und beschrieben werden. Unter Verwendung von (1) und (12) sei demnach folgendes definiert (14):

$$\text{Aktionstension: } at_n := \{ {}^{at}A_n, {}^{at}M_n, {}^{at}K_n \} \in T_\eta : \{ t_\eta \mid \forall t_\eta \in R_{norm} \}$$

$$\text{Struktur eines (gestaltungskompatiblen) CS: } {}_D S_\eta := \{ T_\eta \cup S_\eta \} \quad (14)$$

$$\text{Verhalten eines (gestalteten) CS: } {}_D V_\eta := {}_D S_{\eta-1} \rightarrow {}_D S_\eta$$

Die (14) sagt aus, dass eine spezifische Aktionstension at aus allen dazugehörigen Agenten bzw. externalisierten und homogenisierten MPs, deren Manipulationen und einer in das CS verbindenden Kopplungsmatrix besteht. Die Aktionstension at gehört der Menge aller Aktionstensionen an und hat die dichotome kontinuierliche Potenzialkette R_{norm} als Wertebereich.

Gestaltungskompatibles CS erreicht man demnach durch das Hinzufügen der Menge aller Tensionen in das ursprüngliche CS. Das Verhalten eines auf diese Weise gestalteten CS ergibt sich aus der zeitlichen Veränderung der Struktur des gestaltungskompatiblen CS.

Die Anspannungen innerhalb einer Aktionstension ergeben sich (in Analogie zur elektrischen Spannung) als Differenzen der Manipulationspotenziale. Sie können sowohl relativ zwischen den beliebigen MPs als auch absolut zwischen einem beliebigen MP und dem entspannten MP angegeben werden (15). Damit lassen sich fehlabgestimmte Manipulationen, Konflikte und Konkurrenzen in einem CS beschreiben und behandeln (vgl. Abschnitt 4.3.5).

$$\begin{aligned} \text{Relative Anspannung von } d_n \text{ zum } c_n \text{ auf } at \text{ zum } n: {}^{at}u_n^{dc} &= {}^{at}d_n - {}^{at}c_n \\ \text{Absolute Anspannung von } d_n \text{ zum } o_n \text{ auf } at \text{ zum } n: {}^{at}u_n^{do} &= {}^{at}d_n - {}^{at}o_n \end{aligned} \quad (15)$$

Aktive MPs aller Agenten im CS zum n bezogen auf at können in einem Vektor \vec{at}_n zusammengefasst werden. In Abschnitt 3.3 wurde argumentiert, dass alle Agenten des CS ihre aktiven MPs mittels OMs auf die CS-Ebene senden und damit zur CA-Wechselwirkung beitragen. Alle Manipulationen eines gestaltungskompatiblen CS führen die aktiven MPs aller Agenten bezogen auf alle Tensionen. Mittels dieser Feststellungen lässt sich (11) als (16) schreiben:

$$\vec{at}_n = f({}^{at}K_n \cdot \vec{at}_{n-1}) \quad (16)$$

Mithilfe von (16) lassen sich die entsprechenden Zeitreihen der Manipulationsverläufe im CS entwickeln und untersuchen. Dies kann für die Analyse von auf Aktionstensionen basierenden maschinellen Agenten, der Arbitr (vgl. Abschnitt 4.3), gewinnbringend eingesetzt werden.

Im Kontext der Aktionstensionen und der Formalisierung der CS lässt sich die (4) als das Produkt der „Psychologischen Kraft = Aufmerksamkeit“ und der „Aufforderung = Saliens“ interpretieren. Wenn man das entspannte MP mit dem Wert 0 auf dem R_{norm} annimmt, dann können die EMs der CAs anstatt mit 1, wie ursprünglich angenommen, mit $0 < k_n^{\beta\beta} < 1$ angegeben werden, z.B. 0,9. Dies würde die geforderte Tendenz der angespannten MPs der CAs in Richtung des entspannten MP ($:=0$) abbilden. Ein kontinuierlicher, bidirektionaler, auf den Bereich R_{norm} beschränkter und weiter in diskrete Bereiche aufgeteilter Wert kann als eine *allgemeine Datenrepräsentation* für die Gestaltung der CS und der CA-Wechselwirkung in einer gestaltungskompatiblen CS-Architektur verwendet werden (vgl. Abschnitt 4.4). Dieser Wert beschreibt das kontinuierliche quantitative Maß und die Richtung der Anspannung der MPs einer Aktionstension. Diskrete Punkte bzw. Bereiche darauf können mit qualitativen bzw. klassifizierenden, deskriptiven Metriken für MPs verknüpft werden.

4.2.4 Verwandte Konzepte und Diskussion

In der einschlägigen Literatur sind einige zur Aktionstension verwandten Konzepte zu finden. Für die Gestaltung der Fahrer/Automation-Systeme verwendet Onken^[Onn94] z.B., ein „Gefahrenmodell“, das einen nicht gerichteten Wert namens „Zeitreserve“ enthält. Dieser Wert ergibt sich aus auf einem Fahrermodell basierter Gefahrenschätzung und aus auf Umwelt- und Fahrzeug-Randbedingungen basierter Gefahrenberechnung. Die Zeitreserve definiert ein Zeitfenster für eine mögliche Aktion zur Vermeidung der Gefahr. Unter dem Stichwort „Warnbaukasten“ gibt es Überlegungen^[RhW11] zu aktionsorientierten Warnstrategien für die

Interaktionsgestaltung innerhalb hochautomatisierter Fahrzeuge. Diese basieren auf der Aufteilung des Zeitablaufes der Mensch/Maschine-Interaktion in Stufen mit dem Ziel, von der Situationskritikalität abhängige Eskalationsstrategien zu entwickeln.

Die eben beschriebenen Konzepte sind in hohem Maße kompatibel zu dem Ansatz der Systemgestaltung mittels der Aktionstensionen, obwohl diese Konzepte überwiegend deskriptive Metriken zu liefern scheinen. Dagegen sind Aktionstensionen prospektive Strukturen. Der prospektive und Strukturcharakter der Aktionstensionen sind wichtige Merkmale, mit den verschiedene mögliche und sinnvolle Manipulationen in Richtung der gleichen entspannten Systemzustände modelliert werden können. Außerdem ist es wichtig anzumerken, dass das prospektive Konzept der Aktionstensionen mehr gestaltungs-, modell- und lösungsorientiert ist als seine verwandten deskriptiven Metriken, wie Kritikalitäts- oder Risikobetrachtung.

Nachteile: Die Nutzung abstrakter Steuerstrukturen, wie Aktionstensionen, als Leitprinzip für die Gestaltung dynamischer CS kann die Systemkomplexität sowie die Komplexität der Gestaltung selbst reduzieren und linearisieren. Nichtsdestotrotz ist zu beachten, dass die Operationalisierung der Aktionstensionen ein relativ aufwendiger Prozess werden kann. Denn bevor es zu einer konsistenten Definition der MPs, Transitionen, dazu gehöriger Manipulationen und ihrer Ausgestaltung als Interaktionssignale kommt, müssen viele Systemparameter und -prozesse berücksichtigt werden. Dabei sind es nicht nur die deterministischen technischen Prozesse, sondern auch die kognitiven Prozesse des Menschen. Diese lassen sich oft nicht mit festen Werten und Kategorien fassen, wie es in der Technik möglich ist, sondern geben eher Wahrscheinlichkeiten und Hinweise auf die richtigen Werte- und Kategorienbereiche vor.

Bei der CS-Gestaltung benötigt man deswegen zusätzliche Methoden, um mit solchen unkonkreten Werten umgehen zu können. Mit neuronalen Netzen, Fuzzy-Logik, Bayesschen Netzen etc. gibt es eine große Methodenvielfalt, um dieses Problem adressieren zu können. Eine andere erfolgsversprechende Herangehensweise könnte sein, keine genauen Werte bestimmen zu wollen, sondern sie bei guten Schätzungen und explorativ-empirisch erarbeiteten Bereichen zu belassen. Dies kann absolut ausreichen und zum Gestaltungserfolg führen, da der menschliche Systemnutzer aufgrund seines inhärent adaptiven und selbstorganisierten kognitiven Verhaltens bereits gut damit zurechtkommen könnte. Dies unterstreicht die Wichtigkeit des Usability Engineering für die Gestaltung von CS (vgl. auch Abschnitt 2.3.1).

Vorteile: Unter Verwendung von Aktionstensionen kann der Systemingenieur prospektiv jene Manipulationen berücksichtigen, die gesendet werden müssen, um bestimmte Konkurrenzen zu hemmen oder Kooperationen anzuregen. Damit können sowohl die positive als auch die negative CA-Wechselwirkung behandelt werden. Im Unterschied zu den bekannten Konzepten, kann das Tensionskonzept als integrativer Ansatz für die Gestaltung verschiedener Arten von dynamischen CS betrachtet werden, anstatt eine Beschreibung einer Gestaltungsmethode und domänenspezifischer Referenzwerte zu sein. Aktionstensionen können bei einer Systemgestaltung auch als Leitmetaphern verwendet werden. Damit wird die Gestaltung der dynamischen CS holistisch und für den menschlichen Systemnutzer nachvollziehbar. Insgesamt gestattet die Verwendung der Aktionstensionen zusammen mit dem MPS-Konzept, das gesamte CS holistisch zu betrachten und zu gestalten, was im weiteren Verlauf der Arbeit anhand einiger Beispiele für entwickelte Assistenzsysteme noch gezeigt und erläutert werden soll.

Aktionstensionen besitzen eine konzeptionelle Nähe gleichermaßen zu der psychologischen als auch zu der technischen Perspektive. Dies fördert den Dialog zwischen den Fachleuten verschiedener Disziplinen, die an der Gestaltung von Mensch/Maschine-Systemen beteiligt sind, wie Ingenieure, Psychologen, Informatiker, Systemergonomen etc. Solche fachübergreifende Art der Gestaltung komplexer dynamischer CS kann sehr effektiv sein, da viele fachspezifische Probleme auf einer gemeinsamen Grundlage diskutiert und gelöst werden können.

4.2.5 Aktionstensionen in hochautomatisierten Fahrzeugen

Bei einer Fahrer/Automation/Fahrzeug/Umwelt-Systemgestaltung können Aktionstensionen ebenfalls verwendet werden. Einige Beispiele dazu wurden bereits in Abschnitt 4.2.2 erläutert. Im Folgenden wird das Konzept der Aktionstensionen einen weiteren Schritt näher dem Automotivbereich und der darin geforderten Anwendungsspezifika gebracht.

Die Operationalisierung der Aktionstensionen ist abhängig von externen und internen Faktoren, bezogen auf die Fahrer/Automation/Fahrzeug-Systemgrenze^[FK08]. Die externen Faktoren sind Effekte und Randbedingungen, die in der Regel außerhalb des Fahrer/Automation/Fahrzeug-Systems sind, z.B. die Gefährlichkeit der Situation, gesetzliche und andere externe Randbedingungen und -effekte. Die internen sind innerhalb des Fahrer/Automation/Fahrzeug-Systems. Dies sind tatsächliche und mögliche Fahrer- und Automationsaktionen, Zustände, Randbedingungen und Effekte aus den Bereichen Human Factors, Sensorik und Fahrdynamik.

Speziell im Automotivbereich kann es mehrere Aktionstensionen geben, bezogen auf die Fahrzeugführungs- sowie Koordinationsmanipulationen. Damit kann das entsprechende CS-Verhalten zeitlich abgestimmt werden, wie Längsführungs-, Querführungs- sowie das Verhalten bei der Ausführung von Koordinationsaufgaben, z.B. zwecks einer Kooperation. Ausführliches Beispiel der Operationalisierung und Verwendung der lateralen Aktionstension zum Zweck der Gestaltung hochautomatisierter Fahrzeugführung befindet sich im Abschnitt 5.5. Das Beispiel mit der longitudinalen Aktionstension ist in den Abschnitten 6.1.5 und 6.1.6. Das Beispiel der kooperativen Aktionstension befindet sich in den Abschnitten 6.2.5 und 6.2.6.

Die Arbeitshypothese bei der Verwendung von Aktionstensionen für die Synchronisierung der Mensch/Maschine-Systeme ist, dass Menschen ihre Motivation, beschrieben durch MPs und ihre Wechselwirkung, in dem entspannten MP halten sollten und dass die Maschinen mittels daraus externalisierter Aktionstensionen so gestaltet sein können, dass sie dies ebenfalls tun.

Y-Beispiel: Weder der Fahrer noch die Automation kollidieren in der Regel mit Hindernissen, weil beide CAs Manipulationen senden, die zum sicheren CS-Verhalten führen. Sie folgen bestimmten Regeln und versuchen, gleichzeitig das entspannte MP während der Ausführung von beabsichtigten oder zugewiesenen Aufgaben zu generieren oder es zu aktivieren.

Das entspannte MP kann durch eine Gestaltungsentscheidung bestimmt oder dynamisch arbitriert werden (vgl. Abschnitt 4.3). Die Bestimmung des entspannten MP für hochautomatisierte Fahrzeuge kann beispielsweise unter Berücksichtigung des Yerkes-Dodson Gesetzes geschehen^[YeD08]. Nach dem dort formulierten Zusammenhang kann man annehmen, dass Fahrer ein Optimum zwischen der Anspannung und der Effektivität ihrer Handlungen anstreben sollten, was bei der Definition des entspannten und der angespannten MPs helfen kann.

Abbildung 4-3 zeigt einen modifizierten Zustandsübergangsgraphen des Fahrer/Automation/Fahrzeug/Umwelt-Systems (vgl. Abbildung 4-3). Agent t ist als Systemrepräsentation der Aktionstension der Abbildung hinzugefügt worden. Die Aktionstension kann für die Synchronisierung von d und c verwendet werden. Die Transitionen zwischen den MPs des t können als Synchronisierungssignale im gesamten System verteilt werden.

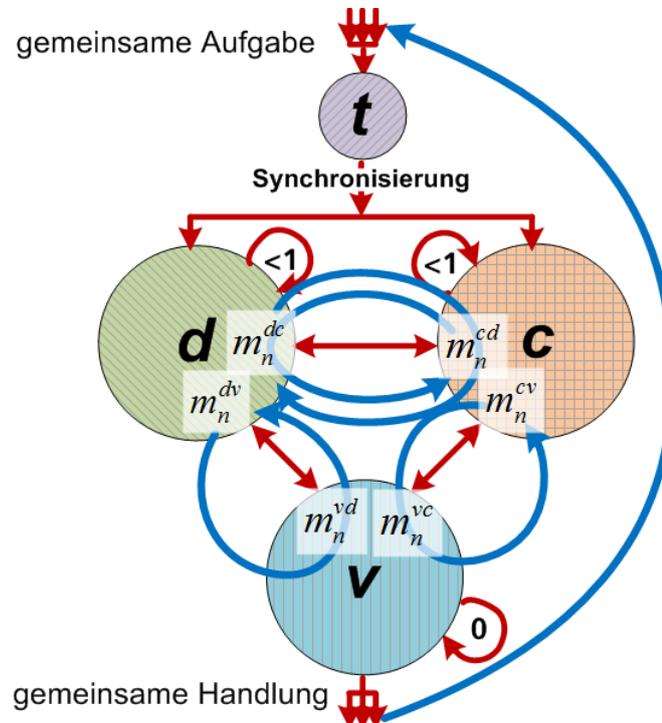


Abbildung 4-7: Zustandsübergangsgraph des Fahrer/Automation/Fahrzeug/Umwelt-Systems modifiziert nach der Anwendung des Aktionstensionskonzepts

Verwendung von Aktionstensionen als allgemeine Steuerstrukturen in der Gestaltung eines hochautomatisierten Fahrzeuges kann die Systemkomplexität deutlich reduzieren. Sie kann Steuerungs- und Regelungsprozesse in die vom Systemingenieur präferierte Richtung linearisieren und zu denen der Fahrer kompatibel machen. Aktionstensionen können für generelle Fahrzeugführungsaufgaben benutzt werden, wie Geschwindigkeits- und Abstandsregelung, aber auch für die Gestaltung und Behandlung von Human Factors Prozessen, wie kognitive Lastverteilungen, Aufmerksamkeitssteuerung oder Unterstützung des Situationsbewusstseins. Komplexe Aufgaben, wie die Verbesserung der Sicherheit, der Effizienz oder des Komforts eines hochautomatisierten Fahrzeuges können damit ebenfalls adressiert werden.

4.2.6 Zusammenfassung

Aufgrund ihrer Kognition können CAs sich willkürlich verhalten. Dies führt dazu, dass CA-Manipulationen im CS zeitlich fehlabgestimmt sein können, was eine negative CA-Wechselwirkung bis hin zu einer Systeminstabilität verursachen kann. Techniknahe Konzepte der Motivationsforschung, wie „psychologische Kraft“ und „Aufforderung“, zeigen einen Weg auf, Motivationsprozesse in einem CS mittels operationalisierbarer Metriken beschreiben zu können. Das Konzept der Aktionstensionen setzt diese Ideen fort. Für die holistische CS-Gestaltung stellt es externalisierbare, homogenisierbare und CA-kompatible Strukturen bereit. Damit kann der Systemingenieur der zeitlichen Willkür des CA-Verhaltens entgegenwirken.

Aktionstensionen können Manipulationen unterschiedlicher CAs miteinander verknüpfen, so dass es möglich wird, sie gezielt und gemeinsam zu beeinflussen. Eine operationalisierte Aktionstension beschreibt eine dichotome Kette von MPs und dazugehöriger Manipulationen mit der gemeinsamen Referenz auf ein entspanntes MP. Transitionen der MPs stellen Trigger-Ereignisse bereit, an den Synchronisationssignale platziert werden können. Damit kann die Frage nach allgemeinen Steuerstrukturen unter der Anforderung der Beeinflussbarkeit von mehr als einer CA-Manipulation beantwortet werden. Dies wurde anhand einiger Beispiele und Erläuterungen dazu gezeigt. Mehr Beispiele folgen weiter im Text dieser Arbeit.

Das Konzept der Aktionstensionen bietet insgesamt einen nützlichen Rahmen für die Gestaltung gut nachvollziehbarer und nutzbarer CS. Aktionstensionen können nicht nur in ihrer operationalisierten Form als ein CoFoR der CA-Manipulationen bei der operativen CS-Gestaltung verwendet werden, sondern auch in Form einer Design-Metapher. Dies kann das Finden und das Umsetzen neuer Gestaltungsideen unterstützen. Die Operationalisierbarkeit der Aktionstensionen bleibt die wichtigste Einschränkung ihrer Verwendbarkeit.

4.3 Arbitrierung

In Abschnitt 3.4 wurde gezeigt, dass innerhalb von CS eine negative, unbestimmte oder positive Wechselwirkung zwischen den CAs aufgrund gegenseitiger Manipulationen entstehen kann. Die negative Wechselwirkung ist charakteristisch für Konflikte, Konkurrenzen, die positive für Kooperationen. In der Regel sollte die negative CA-Wechselwirkung gemieden oder gehemmt werden. Positive CA-Wechselwirkung sollte angeregt und unterstützt werden. Dies liegt darin begründet, dass die Kooperation in vielen Fällen effektiver (z.B. energiesparender), komfortabler (z.B. angenehmer) und sicherer (z.B. konfliktfreier) als die Konkurrenz sein sollte.

Die Frage an dieser Stelle lautet: Wenn alle Arten der CA-Wechselwirkung denselben theoretischen Ursprung in den CA-Manipulationen haben, gibt es eine allgemeine Methode, um diese CA-Wechselwirkung einheitlich beeinflussen zu können? Die Anforderung an die Behandlung dieser Frage ist: Eine allgemeine Methode muss gefunden werden, mit der alle Arten von CA-Wechselwirkungen beeinflusst werden können. Mit einer solchen Methode wäre es möglich, sowohl fehlabgestimmte Handlungen und Konkurrenzen zu hemmen als auch abgestimmte Handlungen und Kooperation innerhalb von CS anzuregen.

4.3.1 Problemstellung: Qualitativ fehlabgestimmte Agentenhandlungen

Der Fahrer in einem hochautomatisierten Fahrzeug und seine Automation sind CAs. Sie können unterschiedliche MPs darstellen und daher unterschiedliche Manipulationen auf die CS-Ebene senden, obwohl sie objektiv in derselben Verkehrssituation sind (vgl. Abbildung 4-3 und Kommentare dazu). Dies kann zur negativen CA-Wechselwirkung, also einem Konflikt bzw. Konkurrenz führen. Auch bei einer bereits intakten Kooperation zwischen dem Fahrer und der Automation kann es jederzeit zu *qualitativ fehlabgestimmten* CA-Manipulationen kommen. Wenn diese nicht kontrolliert ablaufen, können sie zur Instabilität und zur suboptimalen Usability bei der geteilten Systemkontrolle führen. Wegen der fortwährend vorhandenen Gefährlichkeit der Umwelt, wie im Straßenverkehr, müssen derartige konfliktbehaftete Systemzustände und -Eigenschaften aufgelöst werden oder dürfen gar nicht auftreten.

4.3.2 Lösungsansatz: Allgemeine Methode der Handlungsabstimmung

Im Abschnitt 4.2 wurde ein Ansatz gezeigt, wie zeitlich fehlabgestimmte CA-Manipulationen zwecks CS-Gestaltung mit Aktionstensionen als einer Art „Referenztaktgeber“ für alle Manipulationen behandelt werden können. Neben der zeitlichen Abstimmung müssen die CA-Manipulationen auch qualitativ abgestimmt sein, damit es nicht zu einer negativen Wechselwirkung bzw. einem Konflikt kommt. Damit ist nicht gemeint, dass alle CA-Manipulationen identisch sein müssen, vielmehr geht es darum, eine Kompatibilität dazwischen zu schaffen.

Y-Beispiel: Sowohl der Fahrer als auch die Automation können gleichzeitig die MPs „Fahren links“ mit der OM „links lenken“ darstellen. Die MPs und die OMs beider CAs wären qualitativ identisch, es kann trotzdem zu einem Konflikt kommen, wenn der Fahrer das additive „links lenken“ der Automation beispielsweise missbilligt (negative Aufmerksamkeit, vgl. Abschnitt 3.5) und dagegen nach rechts lenkt. Um solche Fälle gestalterisch abzudecken, sollte es darum gehen, CA-Manipulationen nicht immer identisch, sondern kompatibel zueinander zu gestalten bzw. qualitativ abzustimmen.

Eine qualitative Manipulationsabstimmung zwischen den CAs in einem CS kann mit der Methode der *Arbitrierung* erreicht werden^{[KeF06][Ke12a][KeT13]}. Sie basiert auf einer einfachen Idee: Wenn CA-Manipulationen in *Konflikte* laufen, müssen sie zügig wieder abgestimmt werden, so dass eine negative und unbestimmte CA-Wechselwirkung sich zur positiven entwickelt.

Definition: Arbitrierung ist eine determinierte Verhandlung mittels geeigneter Strategien und Signale mit dem Ziel, eine zeitliche und qualitative Manipulationsabstimmung zwischen den Agenten eines CS innerhalb der verfügbaren Ressourcen zu erreichen^[Ke12a]. Mit Arbitrierung kann der Systemingenieur der Willkür der Manipulationswahl der Agenten entgegenwirken.

Annahme: Bei jeder CS-Gestaltung ist es möglich, jede Art der CA-Wechselwirkung und somit des CS-Verhaltens und der CS-Struktur durch Arbitrierung vollständig zu behandeln.

Die Arbitrierung kann implizit in einer Selbstorganisation der CAs^[WaH67] (Analogie: Gerichtlicher Vergleich) oder explizit durch ihre Fremdorganisation (Analogie: Gerichtliche Entscheidung) gestaltet sein. Arbitrierungssignale einer Arbitrierungsstrategie können sinngemäß durch direkte CA-Manipulationen oder spezielle Agenten (Abschnitt 4.1.1), die *Arbiter*, im CS verteilt werden. Die gesamten CA-Manipulationen müssen dafür innerhalb eines CoFoR vorliegen, das mit operationalisierten Aktionstensionen gegeben sein kann (Abschnitt 4.2.2).

Mit direkten Manipulationen können Strategien entwickelt werden, die keine explizite Entscheidung des Arbiters benötigen. CAs können sich dabei selbst abstimmen, z.B. durch eine Entscheidungskonvergenz^[Keh14]. Solches Vorgehen lässt sich als *implizite Arbitrierung* bezeichnen. Eine explizite Entscheidung des Arbiters wird nur dann notwendig, wenn innerhalb der verfügbaren Ressourcen, wie Zeit, keine implizite Abstimmung zwischen den CAs stattfand. Dann können Arbiter handeln, indem sie die Abstimmung zwischen den CAs moderieren und nach Erschöpfung der zur Verfügung stehenden Ressourcen eindeutige, möglichst für alle CAs optimale Entscheidungen treffen. Solches Vorgehen lässt sich als *explizite Arbitrierung* bezeichnen. Beide Arbitrierungsarten müssen vom Systemingenieur vorher innerhalb des CoFoR und im Sinne des *Vorteils für das Gesamtsystem* gestaltet sein.

Diese Ideen lassen sich in der Abbildung 4-8 als Modifikation der Abbildung 4-6 darstellen. d und c können hier unterschiedliche MPs darstellen ($z--$ bis $z++$) und haben eine gemeinsame Tendenz zu den entspannten MPs ($z0$). Ihre aktiven MPs sind schwarz markiert. Die Aktionstension homogenisiert alle MPs und EMs, die in den CAs als Motivation internalisiert sind. Im Unterschied zur Abbildung 4-6 ist die Aktionstension nun ein Teil eines neu in das CS hinzugefügten Agenten, des Arbiters (a). Die Abbildung zeigt, dass die Aktionstension in einem CS über die Arbitrierung dazu verwendet werden kann, die CAs insgesamt so zu beeinflussen, dass sie zeitlich und qualitativ abgestimmte Manipulationen auf die CS-Ebene senden.

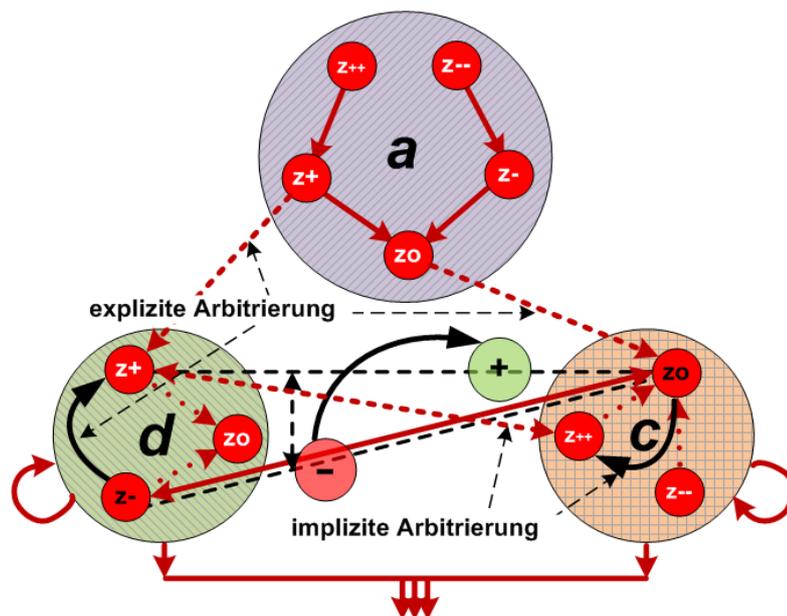


Abbildung 4-8: Erweiterung des MPS-Konzeptes durch die Arbitrierung. Der Arbitrer nutzt die Aktionstension und kann auf die CAs wirken, damit sie die ihre OMs für die positive Wechselwirkung zeitlich und qualitativ abstimmen

Aktuell kommt es auf der CS-Ebene zu einer negativen CA-Wechselwirkung (-). Es gibt aber mindestens zwei Möglichkeiten, durch MP-Transitionen die CA-Wechselwirkung von der negativen zur positiven zu entwickeln. Dafür kann an das abgebildete CS 1.) Von a eine Aufgabe gesendet werden, die c weiterhin das entspannte MP ($z0$) und d das MP ($z+$) aktivieren lässt, wobei es sich um die explizite Arbitrierung handelt, oder 2.) Der c kann selbständig das MP ($z++$) aktivieren und den d damit versuchen, in das MP ($z+$) zu zwingen oder ein neues, MP ($z++$) zu generieren und zu aktivieren, wobei es sich um die implizite Arbitrierung handelt.

Y-Beispiel: Die gezeigte Arbitrierung lässt sich wie folgt aufschlüsseln: d ist der Fahrer und kann grundsätzlich MPs „Fahren links“ ($z-$), „Fahren nicht (selbst)“ ($z0$) und „Fahren rechts“ ($z+$) mit dazugehörigen Manipulationen „links lenken“, „nicht lenken“ und „rechts lenken“ darstellen. Aktuell ist er im „Fahren links“ und lenkt links. c ist die Automation und kann grundsätzlich „Ausweichen links“ ($z--$), „Fahren nicht (selbst)“ ($z0$) und „Ausweichen rechts“ ($z++$) mit dazugehörigen Manipulationen „stark links lenken“, „nicht lenken“ und „stark rechts lenken“ darstellen. Aktuell stellt sie das MP „Fahren nicht (selbst)“ dar, was bedeuten kann, dass es sich um eine Ausweichautomation in Notfällen handelt.

Die explizite Arbitrierung lässt sich im Beispiel wie folgt zeigen: Dem Arbitrer kann bekannt sein, dass es suboptimal ist, nach links zu fahren, weil z.B. ein langer Stau voraus ist. Wenn das Fahrzeug weiterhin nach links gelenkt werden würde, kommt es zum negativen CS-Verhalten, da zu viel Zeit, Sprit etc., verbraucht wäre. Nun passiert folgendes: Der Arbitrer op-

timiert das CS-Verhalten. Er lässt die Automation weiterhin das MP „Fahren nicht (selbst)“ (zo) generieren und der Fahrer bekommt von ihm eine IM mit der Aufforderung (z+), das MP „Fahren rechts“ (z+) zu generieren und nach rechts zu lenken, z.B. durch eine laute akustische Ansage. Der Fahrer lenkt nach rechts um und verbraucht deutlich weniger Zeit und Sprit.

Die implizite Arbitrierung lässt sich im Beispiel wie folgt zeigen: Im Gegensatz zum Fahrer kann es der Automation selbst bekannt sein, dass es suboptimal ist, nach links zu fahren. Kurz vor der Weggabelung kann sie daher einfach rechts Ausweichen (z++) und den Fahrer „übersteuern“, z.B. direkt über das Lenkrad. Dabei kann der Arbitrer sich beobachtend zurückhalten und die Selbstorganisation der Agenten zwecks Optimierung des CS-Verhaltens ausnutzen.

Beide Varianten der Arbitrierung (explizit und implizit) führen in diesem Beispiel zum selben Ergebnis: Das Fahrzeug fährt anschließend nach rechts. Dennoch kann die implizite Arbitrierung (das Ausweichen der Automation) eine suboptimale Arbitrierungsstrategie sein, da es sich nicht um einen Notfall handelt und die Funktion der Automation damit zweckentfremdet wäre. Es ist wichtig anzumerken, dass obwohl die Automation es bei der impliziten Arbitrierung schafft, das Fahrzeug nach rechts umzulenken, ist es damit nicht gegeben, dass der Fahrer dem mental folgt. Er kann auch dann weiter das (z-) aktiv halten und dagegen nach links lenken, was das CS destabilisieren kann. Außerdem merkt man an der impliziten Arbitrierung, dass durch eine Ausweichmanipulation insgesamt eine deutlich kritischere Situation erzeugt worden wäre. Der Arbitrer sollte die Anzahl der Anspannungswechsel (bzw. die Anzahl der Manipulationen) und der gesamten Anspannungen der Agenten minimieren. Diese Einsichten nützen bei der Formalisierung der Arbitrierung (vgl. Abschnitt 4.3.5).

Strategisch betrachtet, ist Arbitrierung eine allgemeine Methode für die zeitliche und qualitative Optimierung des CS-Verhaltens auf allen Ebenen der kognitiven Informationsverarbeitung. Das Hauptziel der Arbitrierung ist, alle CA-Manipulationen robust und konfliktfrei in die Nähe der entspannten MPs der Agenten zu entwickeln. Das Nebenziel der Arbitrierung ist, die Anzahl der CA-Manipulationen insgesamt zu reduzieren. Die dafür notwendigen MPs der Agenten können durch die vom Arbitrer kontrollierte MP-Aktivierung synchron angeregt oder unterstützt werden, die suboptimalen können verhindert oder gehemmt werden. Damit kann der Systemingenieur langfristig, kurzfristig und unmittelbar die CA-Manipulationen abstimmen, damit das CS-Verhalten sich positiv (zur Kooperation) entwickelt.

Taktisch betrachtet, kann die Arbitrierung Kooperationsmodi^[Hoc01], Automationslevel^[Pa500] und ihre Transitionen^[ScF08] nutzen. Sie kann durch eine Aufgabenumverteilung zwischen den CAs, eine Adaptierung der CAs untereinander^[GrK12], eine Abstimmung des Situationsbewusstseins zwischen den CAs, eine Sicherheits- bzw. Unsicherheitsrückmeldung^[HeB11] etc. gestaltet sein. Durch die Analogie der Arbitrierung zu einer gerichtlichen Verhandlung kann auf Taktiken der Dialektik zurückgegriffen werden, wie „lasse ausreden“, „nutze deutliche, transparente und konsistente Formulierungen“, „kommuniziere höflich, aber mit Nachdruck“.

Operativ betrachtet, kann die CA-Wechselwirkung als ein bereits laufender Prozess im CS angenommen werden. Diesen Prozess kann der Systemingenieur analysieren und gestalten, indem er Arbitrer für eine Strukturveränderung des CS entwickelt und dem CS hinzufügt. Die Arbitrer nutzen Aktionstensionen und wenden Arbitrierungsstrategien mittels Arbitrierungssignale an, mit dem Zweck durch Aktivierung bzw. Generierung neuer MPs die OM der

Agenten zu verschieben. Arbitrierungsstrategien können in mehrere *zeitlich begrenzte Phasen* unterteilt sein, in denen moderiert und entschieden wird^[Ke12a]. Diese Phasen können sich in der Intensität und der Art der Verhandlung unterscheiden. Sie können über *räumlich begrenzte Schnittstellen* zwischen den CAs ablaufen. Arbitrierungssignale können dafür entlang der Aktionstensionen platziert werden. Transitionen zwischen den MPs dienen dafür als Trigger.

Operative Anwendung der Arbitrierung sollte insgesamt zu intuitiv verständlichen und gut erlernbaren CS führen. Der Systemingenieur kann dabei den bewährten minimalistischen Gestaltungsprinzipien folgen, wie „keep it simple, but not simpler“ oder „never touch a running system“. Nur eine vorherige Aufdeckung von potenziellen Konflikt- oder Kooperationspotenzialen rechtfertigt den Einsatz neuer Arbitrierungsstrategien und -signale. Wenn keine potenziellen oder akuten Konflikte, Kooperationen entdeckt wurden, dann ist meistens auch keine Systemgestaltung notwendig. Jede Arbitrierung benötigt daher eine Vor- und Laufzeitidentifikation der fehlabgestimmten CA-Manipulationen, Anspannungen bzw. Konflikte im CS.

Der Abschnitt 4.3.3 beschäftigt sich mit möglichen Problemen und Lösungen in CS, um Hinweise für eine Voridentifikation der Konflikte im CS zu erarbeiten. Der Abschnitt 4.2.5 beschäftigt sich mit der Formalisierung der Arbitrierung, die Hinweise und Berechnungsvorschriften auf eine mögliche Implementierung der Laufzeitidentifikation der Konflikte liefert.

4.3.3 Zu Problemen und Lösungen in kognitiven Systemen

Dieser Abschnitt beschäftigt sich mit möglichen Problemen und Lösungen in CS, um Hinweise zum Auffinden adäquater Arbitrierungsstrategien und -signale zu erarbeiten. Neben impliziter und expliziter Arbitrierung lassen sich zwei weitere grundsätzliche Arten der Arbitrierungsstrategien als Lösungen für Probleme in CS definieren: Konfliktlösung und Konfliktprävention^[GrK12]. Dies ist in der Abbildung 4-9 schematisch dargestellt.

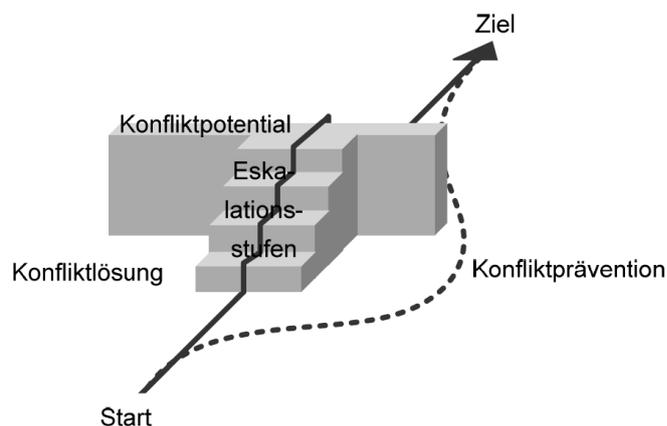


Abbildung 4-9: Zwei grundlegende Arbitrierungsstrategien: Konfliktlösung und Konfliktprävention^[GrK12]

Die Abbildung 4-9 zeigt, dass zu Anfang einer CS-Gestaltung (Start) bis zu ihrer erfolgreichen Implementierung (Ziel) ist es für den Systemingenieur notwendig, Konfliktpotentiale im CS aufzudecken, zu umgehen oder zu lösen. Das gleiche Schema gilt auch für das bereits implementierte Zielsystem und die Arbiter. Zur Laufzeit des CS müssen die Arbiter den Job des Systemingenieurs übernehmen und Konfliktpotentiale aufdecken, umgehen oder lösen.

Die aufgedeckten Konfliktpotentiale können als Anspannungen zwischen den CAs operationalisiert (vgl. Abschnitte 4.2.3 und 4.3.5) und mit geeigneten Arbitern und Arbitrierungsstra-

tegien prospektiv und mit Arbitrierungssignalen zur Laufzeit durch eine CS-Reorganisation umgangen oder gelöst werden. Dabei können der Systemingenieur prospektiv und die Arbitrierer zur Laufzeit die MP-Aktivierung der Agenten zu abgestimmten CA-Manipulationen mittels künstlich injizierten Arbitrierungssignale moderieren. Solche Signale sind ebenfalls Manipulationen und müssen innerhalb des CoFoR (Aktionstensionen) definiert sein.

Beispiele: 1.) Durch eine strategische Gestaltung der Wahlmöglichkeiten und ihrer Anzahl unter Berücksichtigung von Dialogregeln, wie klarer, logischer Formulierungen, Denkpausen etc., können potenzielle Konflikte präventiv umgangen werden. 2.) Durch das Steuern der Aufmerksamkeit, der Motivation und durch Vermeidung von hohen Belastungen können akute Konflikte präventiv umgangen werden. 3.) Die Abbildung 4-9 zeigt außerdem ein Beispiel einer konkreteren Arbitrierungsstrategie, die als eine Abfolge eskalierender Arbitrierungssignale akute Konflikte lösen kann (vgl. „Geschwindigkeitsassistent“ aus Abschnitt 4.2.2).

Weil die Arbitrierung eine Beeinflussung fehlabgestimmter CA-Manipulationen in den Vordergrund stellt, ist es ein streng problem- und konfliktorientiertes Konzept. Darum stehen bei der Gestaltung der Arbitrierung das zu lösende Problem bzw. der akute und der potenzielle Konflikt zwischen den CAs stets im Vordergrund. Dies ermöglicht, die entdeckten Probleme und die dafür entwickelten Lösungen mittels Gestaltungsmuster zu behandeln. Eine Arbitrierung kann also Strategien und Signale beinhalten, die sich aus Gestaltungsmustern zusammengesetzt, entwickelt und hinterlegt werden können (vgl. Abschnitte 2.2.6 und 2.3.6). So lassen sich die Kombinationen der vier grundlegenden Arbitrierungsstrategien (implizite, explizite, Konfliktlösung und -prävention) präzisieren und ordnen.

Akute und potenzielle Konflikte, die bei einer CS-Gestaltung aufgedeckt und gelöst werden müssen, lassen sich entsprechend der drei Ebenen der erwarteten Effekte aufteilen: Langfristige, kurzfristige und unmittelbare Konflikte (vgl. Abschnitt 4.1.2 und Abbildung 4-2). So können Konflikte auf der strategischen, taktischen oder operativen Ebene auftreten. Es sind wissens-, regel- und fertigkeitsbasierte Konflikte möglich. Des Weiteren können Konflikte im Wahrnehmen, Verstehen und Projizieren einer Situation entstehen. Zusätzlich können Konflikte bei den Absichten, Entscheidungen, Koordination und bei den Handlungen der Agenten auftreten. Die hier beschriebenen Konfliktmöglichkeiten sind in der Tabelle 4-1 ohne einen expliziten Anspruch auf Vollständigkeit zusammengetragen^{[MoS98][PaR97][PaS00][Sa95a][Sa95b]}.

Die Tabelle 4-1 ist über zwei Dimensionen aufgespannt. Auf der vertikalen Dimension ist der Zeithorizont der Konflikte und der dadurch hervorgerufenen negativen Effekte aufgetragen. Je kurzfristiger der Zeithorizont des akuten oder des potenziellen Konfliktes ist, desto schneller und mit kürzeren Zeitkonstanten sollte und kann er gelöst oder umgangen werden. Auf der horizontalen Dimension der Tabelle sind Konfliktkategorien vom Situationsbewusstsein der einzelnen Agenten über ihre Absichten und Entscheidungen, Interaktion und Koordination der Agenten bis zu den konkreten Agentenhandlungen aufgespannt. Mögliche Konflikte auf den genannten Dimensionen sind in den Zellen der Tabelle eingetragen.

In der

Tabelle 4-2 ist eine Zusammenstellung der Lösungsstrategien für Konflikte gezeigt, die innerhalb eines CS auftreten können. Sie ist in ihrer Form an die Tabelle 4-1 angelehnt, so dass die übereinstimmenden Zellenpaare als Gestaltungsmuster betrachtet werden können.

Tabelle 4-1: Mögliche Probleme und Konflikte innerhalb eines kognitiven Systems

Konflikt-Ebenen	Situationsbewusstsein	Absichten, Entscheidungen	Interaktion, Koordination	Handlung
langfristig	Falsche Antizipation der Situation, z.B. Mode Confusion, Pilot-out-of-the-loop, Ressourcenkonflikte	Abweichende, nicht synchronisierte Überzeugungen, Wünsche, Absichten	Rollen-, Autoritätskonflikte, Unter- oder Übervertrauen, Inkompatibilität der Schnittstellen	Unerkannte Systemgrenzen, zu hohe Aufgabenkomplexität, Monotonie, Nachlässigkeit
kurzfristig	Falsches Verstehen bestimmter Situations-elemente oder der Situation insgesamt	Abweichende, nicht synchronisierte Pläne, Ziele, falsch erlernte oder angewandte Regeln	Fehlende oder fehlerhafte Aufgabenverteilung, falsche mentale Modelle über sich selbst oder andere Agenten	Fehlerhafte oder verlernte Handlungsmuster, wie Fehlgebrauch des Systems oder Skill Degradation
unmittelbar	Negative Aufmerksamkeitseffekte, schlechte Wahrnehmung, fehlende Motivation, störende Emotionen, Affekte	Zu viele, konkurrierende, zu wenige oder keine unmittelbare Wahlmöglichkeiten	Schlechte Signalqualität, Informationsüberfluss bzw. -mangel, das unmittelbare „Gegeneinander-Arbeiten“	Motorische Probleme, z.B. Unstetigkeit im Verhalten, zu hohe Belastung, Beanspruchung, unangepasste Reaktionszeiten

Tabelle 4-2: Mögliche Lösungen für Probleme und Konflikte innerhalb eines kognitiven Systems

Lösung-Ebenen	Situationsbewusstsein	Absichten, Entscheidungen	Interaktion, Koordination	Handlung
langfristig	Antizipation der Situation fördern, z.B. durch eine Systemtransparenz oder „In-Loop Simulation“, bei der der CA im simulierten Kontrollloop gehalten wird	Überzeugungen, Wünsche, Absichten abstimmen und synchronisieren, z.B. durch einen verständlichen konsistenten und logisch gestalteten Dialog	Rollen möglichst früh klären und konstant halten, richtiges Maß an Systemvertrauen aufbauen, kompatible Schnittstellen gestalten	Systemgrenzen deutlich aufzeigen, einfache Aufgaben formulieren, Monotonie vermeiden und Nachlässigkeit unterbinden, z.B. durch persistente Aktivierung der CAs
kurzfristig	Korrekte mentale Modelle über die Situation fördern, z.B. durch ein Feedback verdeckter Situationselemente	Pläne, Ziele durch eindeutige Regeln abstimmen und synchronisieren. Regeln richtig und nachhaltig beibringen z.B. durch Konditionierung	Aufgaben adäquat verteilen, richtige mentale Modelle über Agenten aufbauen, z.B. über vorheriges „Kennenlernen“ oder „Selbsttest“	Richtige Handlungsmuster anregen, falsche Muster hemmen, z.B. durch Konditionierung, Skill Degradation durch „In-Loop“-Halten verhindern
unmittelbar	Aufmerksamkeit explizit steuern, z.B. über Salienzen, Wahrnehmung unterstützen, Motivation und Emotionen berücksichtigen und ggf. steuern, z.B. mit Aktionstensionen	Eindeutige Auswahlmöglichkeiten geben, die Auswahl unterstützen, z.B. durch das Unterstreichen der Vorteile und Nachteile der jeweiligen Auswahl	Signalqualität verbessern, mit Information sparsam umgehen, Konkurrenzen unterbinden, z.B. durch ein Bewertungs- und Belohnungssystem	Motorik und Stetigkeit des Verhaltens aktiv unterstützen, hohe Belastung und Beanspruchung vermeiden, Reaktions- und Antwortzeiten bei der Agenten angleichen

Konflikte und Lösungen im CS können auf allen Ebenen der Informationsverarbeitung der CAs entstehen. Konflikte können durch eine FehlAbstimmung entsprechender kognitiver Ele-

mente (Überzeugungen, Wünsche, Ziele etc.) entstehen und die Lösung dafür wäre die Beeinflussung dieser Elemente mittels Arbitrierung. Dafür müssen die genannten Elemente innerhalb des CoFoR (Aktionstensionen) vorliegen. Dabei gilt es für den Systemingenieur und für die von ihm gestalteten Arbitrer, die MP-Generierung aller Agenten abzustimmen und über alle Agenten optimale Entscheidungen zu treffen, so dass der Gesamtkonflikt bzw. die gesamte Anspannung im CS minimiert wird (vgl. Abschnitt 4.3.5).

Diese zusammengestellte Kategorisierung zeigt auch die möglichen Entwicklungspfade der Konflikte und der Lösungen. So können Konflikte innerhalb des Situationsbewusstseins der Agenten entstehen, sich weiter über die falschen Entscheidungen der Agenten, über die fehlerhafte Koordination der Agenten bis hin zu den falschen Handlungen fortpflanzen. Konflikte können sich auch in die umgekehrte Richtung entwickeln. Fehlerhafte Handlungen können falsche Interaktion hervorrufen oder unpassende mentale Modelle aktivieren. Konflikte können auch spontan auftreten, ohne auf der Ebene davor entstanden zu sein. Damit können die passenden Lösungen auch auf verschiedenen Ebenen und Entwicklungspfaden der kognitiven Informationsverarbeitung gefunden und implementiert werden.

Eine konkrete Anwendung der erarbeiteten Konfliktkategorisierung findet in Abschnitt 5.1 als „Konfliktanalysewerkzeug“ statt. Die konkrete Anwendung der aufgezählten möglichen Arbitrierungsstrategien findet in Abschnitt 4.3.5 und in Kapiteln 5 und 6 statt.

4.3.4 Verwandte Konzepte und Diskussion

Das Wort „arbitrare“ und „arbitrer“ kommen aus dem Lateinischen und bedeuten „beobachten“ bzw. „verhandeln“ und „Willkür“ bzw. „Schiedsrichter“. In der Literatur wird von einer abgestimmten Art der Interaktion („collaborative art of interaction“) und der Notwendigkeit für Verhandlungen zwischen dem Menschen und der Maschine gesprochen^[GrG05]. Dabei schlagen die Autoren eine bestimmte Art der Verhandlung zwischen dem Menschen und der Maschine über ein gemeinsames haptisches Stellteil vor.

In der Literatur verwendet man an manchen Stellen Begrifflichkeiten für künstlich initiierte Verhandlungen innerhalb agentenbasierter Systeme, die synonym zu Begriffen der Arbitrierung verwendet werden können. Auch bei verwandten Konzepten, wie in der MAS^[LuM05], künstlicher Intelligenz (AI), Sozialpsychologie etc. werden oft Begriffe „Koordination“, „Mediation“, „Organisation“ oder auch einfach „Verhandlung“ verwendet. „While any forms of social interaction (including negotiation, cooperation, conflict, etc.) might be called “coordination”^{17, [MaC94]}. Damit lässt sich die implizite Arbitrierung als „Selbstorganisation“ und die explizite als „Fremdorganisation“ des Systems bezeichnen. Der Arbitrer könnte dann als „Kordinator“, „Mediator“, „Organisator“ oder „Verhandlungsführer“ bezeichnet werden. Die Wahl des Begriffes „Arbitrierung“ ist aber aus zwei Gründen vorteilhaft:

- Die semantische Nähe zu den Begriffen „beobachten“, „verhandeln“, „Willkür“, „Schiedsrichter“ ist hilfreich, um das Konzept und die Metapher der *gerichtlichen Verhandlung* für die Gestaltung kognitiver Systeme zu aktivieren.

¹⁷ Jegliche Form der sozialen Interaktion (einschließlich der Verhandlung, Kooperation, Konflikt, etc.) könnte "Koordination" genannt werden. EÜ.

- Arbitrierung umfasst mehr als nur den Prozess einer Verhandlung bzw. einer Reorganisation, da er explizit die Entscheidung miteinschließt. In der Literatur werden oft Entscheidungsfindung (engl. decision making) und Verhandlung (engl. negotiation) getrennt behandelt. Eine Verhandlung lässt sich aber nicht von einer Entscheidung trennen. Sie gehören als ein holistisches Konstrukt zusammen. Manchmal fällt die Entscheidung während einer Verhandlung und manchmal muss sie explizit getroffen werden. „Arbitrierung“ vereinigt diese Nuancen in einem Begriff.

Das Konzept der Arbitrierung ist kompatibel sowohl zu menschnahen als auch zu techniknahen Systemgestaltungsansätzen. In der Handlungsregulationstheorie^[Har86], z.B. mit dem TOTE-Modell (test-operate-test-exit), findet man Gedanken zu kognitiven Regulationsvorgängen, die ähnlich der Arbitrierung zum Zweck der Gestaltung von CS verwendet werden können.

Der Begriff „Arbitrierung“ wird im ähnlichen Kontext bei der Gestaltung von konflikthanfälligen technischen Systemen bereits verwendet. Die Bandbreite eines Datenübertragungsbusses ist beschränkt. So sind im Bereich der Datenbusentwicklung (CAN-, SCSI-Bus etc.) die Begriffe Arbitrierung und Arbitrer ebenfalls bekannt, nämlich als koordinierte, priorisierte Ressourcenverwaltung und -zuordnung in Konfliktfällen bei mehreren autonomen technischen Entitäten. Die Methode der Arbitrierung ist, die CA-Manipulationen zu beeinflussen, was technisch als ein Spezialfall der Steuerungs- und Regelungstechnik interpretiert werden kann.

Nachteile: In seiner ursprünglichen hier vorgestellten Form kann das Konzept der Arbitrierung als „zu abstrakt“ erscheinen. Dies kann das Erkennen des unmittelbaren Nutzens für den Systemingenieur bei der konkreten Implementierung des Konzeptes erschweren. Durch die Nutzung der Analogien zu technikfernen Konzepten, wie Gerichtsverhandlung, Willkür etc. kann die konkrete Umsetzung eines Arbiters auf der einen Seite erschwert sein.

Vorteile: Auf der anderen Seite eröffnet die Arbitrierung für die Gestaltung von CS einen speziell auf den Konflikt und auf eine strukturelle Umorganisation fokussierten Blickwinkel mit der Methode der Arbitrierung als Konfliktlösungsansatz. Es ist ein streng problemorientierter und systematischer Ansatz. Die gezielte und differenzierte Analyse und Kanalisierung der Konflikte erleichtert deren Behandlung und gestattet von der Anzahl her überschaubare und transferfähige Lösungskonzepte in Form von Gestaltungsmustern zu erarbeiten.

Das Konzept der Arbitrierung stellt eine allgemeine Methode für die Gestaltung gebrauchstauglicher CS bereit. Damit gestaltete Systeme können kompatibel zum Menschen implementiert werden, da sie für den Menschen auf bekannten Grundlagen basieren, wie Konflikt, Kooperation, Motivation, Verhandlung und Entscheidung in einer sehr integrierten Form. Arbitrierung wurde in nationalen und internationalen Projekten (H-Mode^[KeH09], HAVEit^[HAE14], Interactive^[INE14], Adaptive^[SPR14]) im Bereich hochautomatisierter Fahrzeugführung verwendet.

4.3.5 Semi-Formale Repräsentation der Arbitrierung

Arbitrierung lässt sich semi-formalisieren und kann für die Analyse und die Gestaltung von CS mitverwendet werden. In Abschnitt 4.2.3 wurde definiert, dass eine Aktionstension aus externalisierten und homogenisierten MPs, Manipulationen und einer sie verbindenden Kopplungsmatrix besteht und dass ein gestaltungskompatibles CS sich durch das Hinzufügen der

Menge aller Tensionen ins ursprüngliche CS ergibt (14). Das Verhalten eines so gestalteten CS ist dann die zeitliche Veränderung der Struktur des gestaltungskompatiblen CS.

Die Strukturveränderung des CS kann „zur Laufzeit“ des CS geschehen. Das wäre das CS-Verhalten bzw. die CA-Wechselwirkung, wie oben beschrieben. Solche Strukturveränderung kann auch prospektiv geschehen. Das wäre dann die CS-Gestaltung. Beide Strukturveränderungen (die zur Laufzeit und die prospektive) haben eines gemeinsam: Beides kann als Arbitrierung bezeichnet werden. Zur Laufzeit des CS sorgen Arbitrer (explizit) und die CAs selbst (implizit) für die Strukturveränderung. Prospektiv ändert der Systemingenieur die CS-Struktur, indem er z.B. Arbitrer in das CS einfügt, Manipulationen der Agenten oder Kopplungen zwischen den Agenten modifiziert (vgl. (12)). Das macht deutlich, dass die Arbitrierung insgesamt eine gewollte, gestaltbare CS-Strukturveränderung darstellt. Demnach sei (17) definiert:

$$\text{CS-Gestaltung} :\Leftrightarrow \text{Arbitrierung: } D_\eta := T_{\eta-1} \rightarrow T_\eta; \quad (17)$$

Die (17) sagt aus, dass die CS-Gestaltung und Arbitrierung als äquivalent definiert und eine zeitliche Strukturveränderung der Menge aller Aktionstensionen sind. Das bedeutet, dass bei der Arbitrierung es darum geht, Strukturen der Aktionstensionen zu beeinflussen, also nach (14) alle relevanten Agenten (MPs), Manipulationen und Kopplungen dazwischen.

Das Haupt- und das Nebenziel der Arbitrierung waren, die CA-Manipulationen so zu beeinflussen, dass sie sich robust und konfliktfrei in die Nähe der entspannten MPs entwickeln und dass die Anzahl der CA-Manipulationen insgesamt reduziert wird (vgl. Abschnitt 4.3.2). Dies lässt sich als eine Optimierungsaufgabe auffassen und mittels (15) als (18) schreiben:

$$\begin{aligned} \text{CS-Anspannung} :\Leftrightarrow \text{Konflikt: } U_\eta &:= \sum | {}^\alpha u_\eta^{\alpha\alpha} | + | {}^\alpha u_\eta^{\alpha o} | \\ \text{CS-Gestaltungsziele} :\Leftrightarrow \text{Arbitrierungsziele: } U_\eta &\rightarrow 0; \Delta U_\eta \rightarrow 0; M_\eta \rightarrow \emptyset \end{aligned} \quad (18)$$

Die (18) sagt aus, dass die CS-Anspannung bzw. der Konflikt sich als Summe aller äußeren und inneren Konflikte der Agenten, d.h. als Summe der Anspannungsbeträge zwischen allen Agenten bezogen auf alle Aktionstensionen (äußerer Konflikt) und zwischen allen MPs der Agenten und ihren entspannten MPs (innerer Konflikt) ergibt. Kooperation bzw. Konkurrenz gäbe es z.B., wenn der äußere Konflikt unterhalb bzw. oberhalb einer bestimmten Schwelle läge. Die Ignoranz läge bei komplett fehlendem Konflikt vor. Die CS-Gestaltungsziele sind als Arbitrierungsziele definiert und lassen sich 1.) als Minimierung der CS-Anspannung, 2.) Als Minimierung aller Anspannungsveränderungen und 3.) als Erreichen der leeren Menge aller Manipulationen im CS begreifen. Diese Optimierungsziele können bei einer erfolgreichen Operationalisierung der Aktionstensionen mit formalen Optimierungsmethoden adressiert werden. Die (18) ist insgesamt als Minimierung der CS-Gestaltung interpretierbar und nach dem Motto „keep it simple but not simpler“ und „never touch a running system“ umsetzbar.

4.3.6 Arbitrierung in hochautomatisierten Fahrzeugen

Bei der Gestaltung hochautomatisierter Fahrzeuge kann die Arbitrierung genutzt werden. Der Fahrer und die Automation eines solchen Fahrzeuges sind CAs, die mit dem Fahrzeug und der nahen Umwelt ein CS bilden. Alle Agenten wechselwirken und können in eine Konkurrenz oder eine Kooperation treten. Damit dies vom Systemingenieur kontrolliert ablaufen kann,

benötigt das CS einen zusätzlichen CA, den Arbitrer. Während der Gestaltung kann dieser dem CS hinzugefügt werden. Er kann ein Modul des hochautomatisierten Fahrzeuges sein, das im Konfliktfall die Verhandlung zwischen dem Fahrer und der Automation moderiert und gegebenenfalls eine optimale Entscheidung im Sinne des Vorteils für das Gesamtsystem trifft.

Der Zustandsübergangsgraph eines hochautomatisierten Fahrzeugs lässt sich zum Zweck der Anwendung des Konzeptes der Arbitrierung aus der Abbildung 4-7 als Abbildung 4-10 modifizieren. OMs des Fahrers und der Automation werden nun von dem hinzugefügten Arbitrer empfangen und interpretiert. Der Arbitrer kann die gemeinsame Handlung kontrollieren, die das Fahrzeug innerhalb der nahen Umwelt steuert. Er berechnet die multimodalen Feedbacks an den Fahrer und an die Automation. Währenddessen kann der Arbitrer eine Optimierung der CS-Anspannung und eine Reduktion seiner OMs durchführen (vgl. (18)).

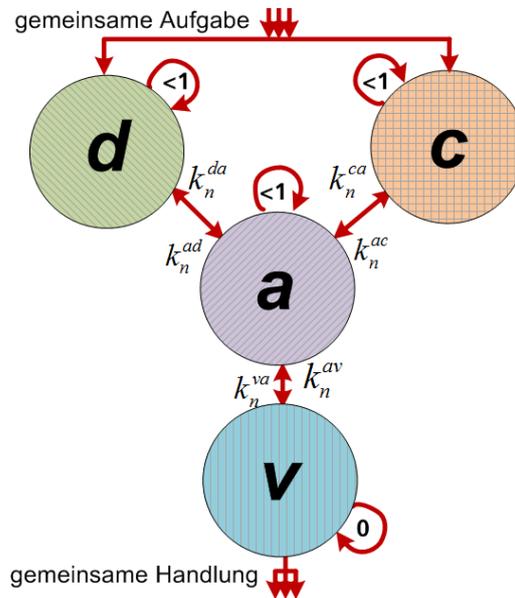


Abbildung 4-10: Zustandsübergangsgraph eines Fahrer/Automation/Fahrzeug-Systems modifiziert nach der Anwendung des Aktionstensions- und Arbitrierungskonzeptes

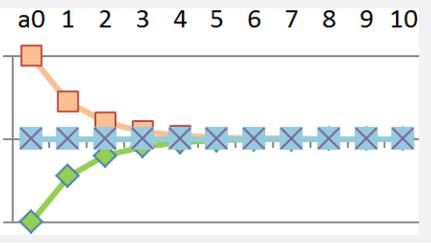
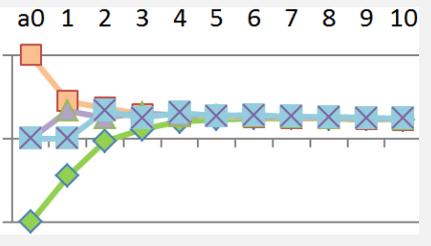
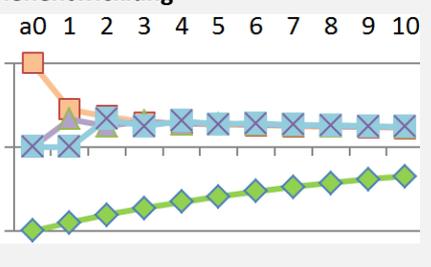
Eine Konfiguration eines solchen Fahrzeugs kann wie folgt aussehen: Der Fahrer, die Automation und der Arbitrer sind CAs mit Eigenmanipulationen ($0 < k_n^{dd} = k_n^{cc} = k_n^{aa} = 0,9 < 1$) (vgl. Abschnitt 4.2.3). Das Fahrzeug ist ein NA ohne eine EM ($k_n^{vv} = 0$). Sowohl der Fahrer als auch die Automation können den Arbitrer zwecks Fahrzeugsteuerung manipulieren (k_n^{da}, k_n^{ca}), können aber weder das Fahrzeug, noch sich gegenseitig direkt manipulieren, noch das Feedback des Fahrzeuges direkt empfangen ($k_n^{dv} = k_n^{cv} = k_n^{dc} = k_n^{cd} = k_n^{vd} = k_n^{vc} = 0$). Der Arbitrer kann die CAs (Fahrer, Automation) manipulieren ($k_n^{ad} = k_n^{ac}$), das Fahrzeug steuern ($k_n^{av} = 1$) und das Feedback des Fahrzeuges empfangen ($k_n^{va} = 1$). Die aktiven MPs und daher die OMs des Fahrers, der Automation, des Arbitrers und des Fahrzeuges zum Ereignis n können als Vektor \vec{a}_n angegeben und nach (11) auf die Entscheidungskonvergenz untersucht werden. Mit dazugehöriger Kopplungsmatrix nach (5) zum Ereignis n ergibt sich bei allen Annahmen (19).

$$K_n = \begin{pmatrix} k_n^{dd} & k_n^{cd} & k_n^{ad} & k_n^{vd} \\ k_n^{dc} & k_n^{cc} & k_n^{ac} & k_n^{vc} \\ k_n^{da} & k_n^{ca} & k_n^{aa} & k_n^{va} \\ k_n^{dv} & k_n^{cv} & k_n^{av} & k_n^{vv} \end{pmatrix} = \begin{pmatrix} 0,9 & 0 & k_n^{ad} & 0 \\ 0 & 0,9 & k_n^{ac} & 0 \\ k_n^{da} & k_n^{ca} & 0,9 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}; \quad \vec{a}_n = \begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad (19)$$

Y-Beispiel: Unter Nutzung von (11) lassen sich Zeitreihen für das aufgestellte Modell entwickeln. Damit lässt sich zeigen, dass durch eine gezielte Reorganisation des CS durch den Arbitrator und seine Kopplungen k_n^{da} , k_n^{ca} , k_n^{ad} , k_n^{ac} drei Entscheidungskonvergenzen auftreten können: 1.) Die Entscheidung konvergiert nicht, 2.) Sie konvergiert in Richtung des MP des Fahrers oder 3.) Sie konvergiert in Richtung des MP der Automation. Es lässt sich zeigen, wie damit die drei Arten der Beeinflussung der Agentenhandlungen (Anregung, Hemmung, Ignoranz, vgl. Abschnitt 4.1.1) auf ein hochautomatisiertes Fahrzeug abgebildet werden können.

Die Tabelle 4-3 hat folgenden Aufbau: Sie zeigt Beschreibungen der vom Arbitrator modifizierbaren Strukturvarianten (oben links), die vom Autor prospektiv erwarteten Entscheidungskonvergenzen innerhalb des Modells (oben rechts), Anfangswerte für K_n und \bar{a}_n (unten links), die entsprechenden Zeitreihen mit der MP-Aktivierung (Entscheidungskonvergenz) der Agenten bis zum 10. Berechnungsschritt (unten Mitte) und ihre Interpretationen (unten rechts).

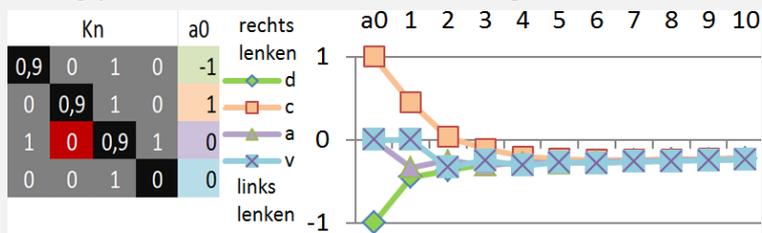
Tabelle 4-3: Restrukturierungsvarianten des Modells eines hochautomatisierten Fahrzeuges, erwartete Entscheidungskonvergenzen, entsprechende Anfangsparameter, Zeitreihenentwicklungen und ihre Interpretationen

<p>Struktur 1 (Konflikt): Fahrer lenkt links, Automation rechts, Fahrer und Automation sind in die Fahrzeugkontrolle über den Arbitrator positiv eingekoppelt und bekommen das positive Feedback von ihm</p>	<p>Erwartete Entscheidungskonvergenz: Instabile Entscheidung sollte zustande kommen. Das Fahrzeug sollte geradeaus fahren</p>																																			
<p>Anfangsparameter und Zeitreihenentwicklung</p> <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><th>K_n</th><td>0,9</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><th>0,9</th><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><th>0,9</th><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><th>0</th><td>0</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><th>a_0</th><td>rechts</td><td>-1</td></tr> <tr><td>lenken</td><td>d</td><td>1</td></tr> <tr><td>c</td><td>a</td><td>0</td></tr> <tr><td>v</td><td>links</td><td>0</td></tr> <tr><td>lenken</td><td></td><td>-1</td></tr> </table> 	K_n	0,9	0	1	0	0	0,9	1	0	0	1	1	0,9	1	0	0	0	1	0	0	a_0	rechts	-1	lenken	d	1	c	a	0	v	links	0	lenken		-1	<p>Interpretation: CS-Entscheidung konvergiert gegen „nicht lenken“. Fahrer und Automation entspannen sich „synchron“ aufgrund ihrer EMs zum entspannten MP „Fahren nicht (selbst)“. Fahrzeug fährt gerade aus, wie erwartet. Entscheidung ist zwar stabil, aber wegen Weggabelung stark konfliktbehaftet. Handlungsbedarf für Arbitrator ist gegeben</p>
K_n	0,9	0	1	0																																
0	0,9	1	0	0																																
1	1	0,9	1	0																																
0	0	1	0	0																																
a_0	rechts	-1																																		
lenken	d	1																																		
c	a	0																																		
v	links	0																																		
lenken		-1																																		
<p>Struktur 2 (Anregung des Fahrers): Wie Struktur 1, Fahrer wird von der Kontrolle durch den Arbitrator oder durch Eigenwillen entkoppelt</p>	<p>Erwartete Entscheidungskonvergenz: Stabile Entscheidung sollte zustande kommen. Das Fahrzeug sollte nach rechts fahren</p>																																			
<p>Anfangsparameter und Zeitreihenentwicklung</p> <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><th>K_n</th><td>0,9</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><th>0,9</th><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><th>0,9</th><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><th>0</th><td>0</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><th>a_0</th><td>rechts</td><td>-1</td></tr> <tr><td>lenken</td><td>d</td><td>1</td></tr> <tr><td>c</td><td>a</td><td>0</td></tr> <tr><td>v</td><td>links</td><td>0</td></tr> <tr><td>lenken</td><td></td><td>-1</td></tr> </table> 	K_n	0,9	0	1	0	0	0,9	1	0	0	0	1	0,9	1	0	0	0	1	0	0	a_0	rechts	-1	lenken	d	1	c	a	0	v	links	0	lenken		-1	<p>Interpretation: CS-Entscheidung konvergiert gegen „rechts lenken“. Fahrer adaptiert sich [GrK12]. Automation bleibt im „rechts lenken“. Fahrzeug fährt nach rechts, wie erwartet. Entscheidung ist stabil und nutzbar für kritische Szenarien, wie Notausweichen^[HeD14], oder Transition von niedrigen in hohe Automationslevel^[ScF08]</p>
K_n	0,9	0	1	0																																
0	0,9	1	0	0																																
0	1	0,9	1	0																																
0	0	1	0	0																																
a_0	rechts	-1																																		
lenken	d	1																																		
c	a	0																																		
v	links	0																																		
lenken		-1																																		
<p>Struktur 3 (Ignoranz des Fahrers): Wie Struktur 2, Fahrer bekommt zusätzlich kein Feedback vom Arbitrator</p>	<p>Erwartete Entscheidungskonvergenz: Stabile Entscheidung sollte zustande kommen. Das Fahrzeug sollte nach rechts fahren</p>																																			
<p>Anfangsparameter und Zeitreihenentwicklung</p> <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><th>K_n</th><td>0,9</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><th>0,9</th><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><th>0,9</th><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><th>0</th><td>0</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><th>a_0</th><td>rechts</td><td>-1</td></tr> <tr><td>lenken</td><td>d</td><td>1</td></tr> <tr><td>c</td><td>a</td><td>0</td></tr> <tr><td>v</td><td>links</td><td>0</td></tr> <tr><td>lenken</td><td></td><td>-1</td></tr> </table> 	K_n	0,9	0	0	0	0	0,9	1	0	0	0	1	0,9	1	0	0	0	1	0	0	a_0	rechts	-1	lenken	d	1	c	a	0	v	links	0	lenken		-1	<p>Interpretation: CS-Entscheidung konvergiert nicht vollständig gegen „rechts lenken“. Automation bleibt im „rechts lenken“. Fahrzeug fährt nach rechts, wie erwartet. Entscheidung ist „quasi stabil“, da bei einer eventuellen notwendigen Kontrollübernahme durch Fahrer das CS wegen seiner Anspannung zu den restlichen Agenten instabil werden kann</p>
K_n	0,9	0	0	0																																
0	0,9	1	0	0																																
0	1	0,9	1	0																																
0	0	1	0	0																																
a_0	rechts	-1																																		
lenken	d	1																																		
c	a	0																																		
v	links	0																																		
lenken		-1																																		

Struktur 3 (Anregung der Automation): Wie Struktur 1, Automation wird von der Kontrolle durch den Arbitrer entkoppelt

Erwartete Entscheidungskonvergenz: Stabile Entscheidung sollte zustande kommen. Das Fahrzeug sollte nach links fahren

Anfangsparameter und Zeitreihenentwicklung



Interpretation: CS-Entscheidung konvergiert zum „links lenken“. Automation adaptiert sich^[GrK12]. Fahrer bleibt im „links lenken“. Fahrzeug fährt nach links, wie erwartet. Entscheidung ist stabil und nutzbar als Transition von hohen in niedrige Automationslevel^[ScF08] oder Übersteuern der Automation durch Entkopplung^[HeD14]

Wie man den Interpretationen der Zeitreihenentwicklungen entnehmen kann, existieren für die zwei grundlegenden Arbitrierungsstrategien (Konfliktlösung und Konfliktprävention) drei entsprechende operativ umsetzbare explizite Arbitrierungsstrategien (Anregung, Hemmung Ignoranz), die allesamt durch Veränderung der Kopplungsgrade des Arbiters abbildbar sind. Des Weiteren zeigt die Tabelle 4-3 eine Methode der Analyse hochautomatisierter Fahrzeugführung, die für die operative CS-Gestaltung nützlich sein kann (vgl. auch Abschnitt 5.4).

Die weiter zu präzisierenden und operativ umsetzbaren Arbitrierungsstrategien können auf mehreren Ebenen nach dem Konzept des Zeithorizonts der Beeinflussung des Agentenverhaltens (vgl. Abschnitte 4.1.2 und 4.3.3) implementiert werden. Unmittelbare Effekte können zu der Stabilisierungsebene, die kurzfristigen der Bahnführungsebene und die langfristigen der Navigationsebene der Fahrzeugführung zugeordnet werden^[DoN96]. Die geeigneten Arbitrierungsstrategien und Beispiele der Arbitrierungssignale für den haptischen Interaktionskanal sind ohne Anspruch auf Vollständigkeit als Heuristiken in der Tabelle 4-4 zusammengefasst.

Tabelle 4-4: Beispiele von Arbitrierungsstrategien und -signalen für die drei Ebenen der Fahrzeugführung

Ebene	Zeithorizont	Arbitrierungsstrategien	Arbitrierungssignale
Stabilisierung	Unmittelbare Effekte (0...1s)	Schnell umsetzbare Strategien, wie (partielle) Entkopplung des Fahrers oder der Automation, direkter stabilisierender Eingriff in die Fahrzeugsteuerung, Hysterese	Direkt spürbare Signale, wie Kopp-lungs-/Entkopplungsfeedback auf den Stellteilen, kontinuierliche Kraft, Vibrationen, Steifigkeiten
Bahnführung	Kurzfristige Effekte (1...3s)	Schnell erfassbare Strategien, wie rasche Eskalationen, Richtungs-, Manöver-, und Trajektorien-Hinweise, Warnungen	Abrupte, diskrete Signale, wie kurze Vibrationsmuster, Rucks, deutlich spürbare Kraftschwellen
Navigation	Langfristige Effekte (>3s)	Komplexere Strategien, wie Handshakes, informierende Eskalationen mit begleitenden Erklärungen, Dialog, Adaptierung	Komplexe Kombinationen von diskreten und kontinuierlichen Signalen

Für die operative Gestaltung der Arbitrierung in hochautomatisierten Fahrzeugen ist es wichtig, einen Arbitrierungsraum in Bezug auf den Ort der Arbitrierung zu definieren. Wenn die Arbitrierung zwischen dem Fahrer und der Automation mittels eines haptischen Stellteils durchgeführt wird, sollten geeignete Amplituden, Kräfte und Zeitintervalle für die Arbitrierung definiert werden. Aufgrund möglicher störender Interferenz zwischen der Fahrzeugsteuerung und den Arbitrierungssignalen sollte dafür ein störungsunempfindliches oder ein spezielles Arbitrierungsstellteil gewählt werden. Auch der Arbitrierungsraum in Bezug auf den Zeitpunkt der Verhandlung bzw. der Entscheidung muss richtig gewählt werden, damit keine

negativen zeitlichen Interferenzen mit anderen dynamischen Vorgängen der Fahrzeugführung auftreten. Zur Aktivierung der Arbitrierungssignale kann ein zeitliches Fenster geöffnet und danach wieder geschlossen werden.

4.3.7 Zusammenfassung

Arbitrierung ist bei einer CS-Gestaltung notwendig und nützlich, da CAs unterschiedliche MPs darstellen und eine Koordination benötigen können. Die Arbitrierung stellt dafür eine allgemeine Methode zur Verfügung. Sie eignet sich für die Gestaltung aller Arten von CA-Wechselwirkung (fehlabgestimmte Handlungen, Konkurrenz und abgestimmte Handlungen, Kooperation). Das Konzept der Arbitrierung ordnet die bekannten Prinzipien der CS-Gestaltung unter dem Gesichtspunkt der Restrukturierung von CS mittels künstlich hinzugefügter Arbitrer mit dem Ziel, Anspannungen und ihre Veränderungen im CS zu minimieren.

Arbitrer nutzen die Aktionstensionen und beeinflussen die CA-Wechselwirkung mittels vorher gestalteter Arbitrierungsstrategien und -signale. Dies kann implizit geschehen, wenn die CA-Wechselwirkung passiv beobachtet wird, um auf potenzielle oder bereits akute Konflikte zu schließen. Wenn das Konfliktpotenzial zu hoch ist oder nur wenige Ressourcen, wie Zeit, zur Verfügung stehen, dann kann die explizite Arbitrierung eingreifen. Der Arbitrer kann dabei über das ihm zur Verfügung stehende HMI die aufgedeckten Konflikte lösen oder umgehen.

Am repräsentativen Beispiel eines Konfliktes bei der Fahrtrichtungswahl an einer Weggabelung wurde exemplarisch gezeigt, wie die Arbitrierung im Bereich hochautomatisierter Fahrzeuge ihre Anwendung finden kann. Über den Arbitrer und seine Kopplungen in das CS hat der Systemingenieur einen unmittelbaren Zugriff auf das hochautomatisierte Fahrzeug und sein Verhalten. Er kann die ebenfalls vorgeschlagenen Arbitrierungsstrategien operativ umsetzen und gebrauchstaugliche und nützliche Art des hochautomatisierten Fahrens entwickeln.

Das Konzept der Arbitrierung wurde nahtlos in das MPS-Konzept integriert (vgl. Abbildung 4-8 und Abschnitt 4.3.5), so dass gemeinsam mit dem Konzept der Aktionstensionen holistische strukturfokussierte Verständnis, Beschreibung und Gestaltung von CS möglich ist. In Abschnitt 4.4 wird die gesamte aufgestellte Theorie dazu verwendet, um eine gestaltungskompatible CS-Architektur und die dazugehörige Gestaltungsschnittstelle zu erarbeiten.

4.4 Gestaltungskompatible Architektur kognitiver Systeme

Das Ziel dieses Abschnitts ist, einen Weg von den theoretischen Konzepten für das strukturfokussierte Verstehen, Beschreiben und Gestalten von CS in die Praxis der operativen Implementierung aufzuzeigen. Dafür wird die erarbeitete Theorie von osGekoS in einer gestaltungskompatiblen Systemarchitektur zusammengeführt. Die Architektur integriert die Theorie in einer ebenen-basierten Struktur, die als Vorlage für den Entwurf einer konkreten Software/Hardware-Architektur für die operative Gestaltung und Implementierung von CS verwendet werden kann. Sie ermöglicht einen Zugriff auf das Verhalten des gestalteten CS durch die prospektive Entwicklung der Arbitrer über eine dazugehörige Gestaltungsschnittstelle.

4.4.1 Anforderungen an Architektur und Gestaltungsschnittstelle

In der zu erarbeitenden Architektur sollten bereits vorintegrierte Grundlagen von osGekoS abbildbar sein. Konkret sollte es möglich sein, sowohl das MPS-Konzept als auch die darin enthaltenen zusätzlich erarbeiteten Konzepte der Aktionstension und der Arbitrierung zu verwenden. Dabei sollte die Architektur möglichst einfach und überschaubar bleiben. Des Weiteren sollte es möglich sein, innerhalb der Architektur Arbitrer zu platzieren. Die Architektur sollte eine Schnittstelle für die operative Gestaltung von Arbitern bieten. Diese Schnittstelle sollte ebenfalls kompatibel zu den Konzepten von osGekoS sein. Sowohl die Architektur als auch die Gestaltungsschnittstelle sollten als Software-Werkzeuge abbildbar sein.

4.4.2 Zur Architektur

Alle autonomen Entitäten in einem CS sind Agenten (CAs, NAs) mit MPs. Dies kann mit dem MPS-Konzept (vgl. Kapitel 3) verstanden und beschrieben werden. Agenten zeigen ihre MPs durch OMs auf die CS-Ebene und können dort insgesamt eine negative, unbestimmte oder positive Wechselwirkung erzeugen. Durch das Berücksichtigen der Aktionstensionen in einem so beschriebenen CS kann eine gestaltungskompatible Architektur erzeugt werden (18). Gestaltbare Arbitrer können dann Aktionstensionen nutzen, um zwischen den Agenten entstehende akute oder potenzielle Anspannungen (Konflikte) zu lösen oder zu umgehen. Diese Konzepte können für eine gestaltungskompatible Architektur verwendet werden.

Das bedeutet, dass auf der *ersten Ebene* der Architektur (blau) Agenten (MPs), Manipulationen und Aktionstensionen erfasst und geordnet werden müssen (Abbildung 4-11)^[KeT13]. Dies kann mittels interner Agentenrepräsentationen geschehen. Die Agentenrepräsentationen sammeln Daten der „realen“ Agenten, wie Fahrer und Automation. Hier kann eine Fahrer- und Automationszustands- und Absichtserkennung platziert werden. Alle Aktionen der Agenten, wie Lenken, Beschleunigen, Bremsen, Betätigen der Knöpfe, Blick-, Risikoverhalten etc. können hier in MPs übersetzt werden. Arbitrer können diese Daten verwenden, um Anspannungen im CS in Bezug auf operationalisierte Aktionstensionen festzustellen und aufzulösen.

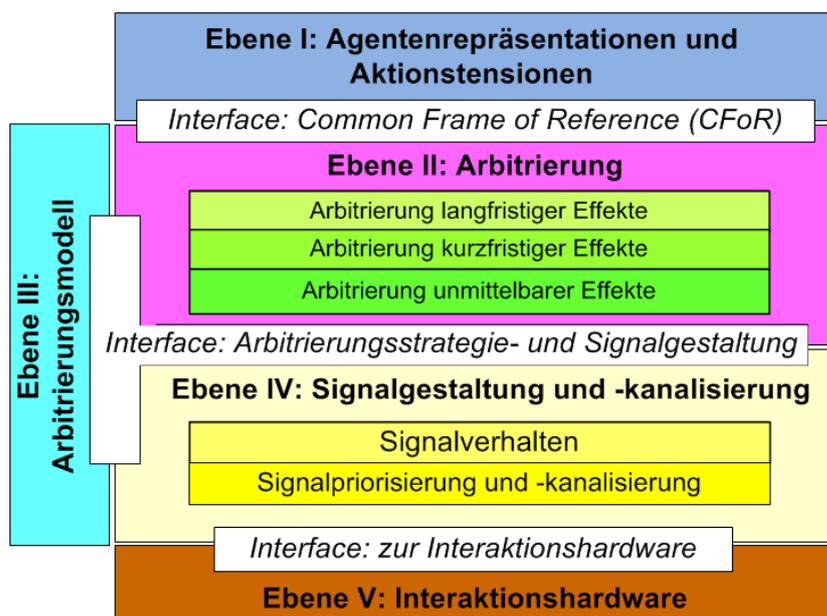


Abbildung 4-11: Generische gestaltungskompatible Architektur kognitiver Systeme

Y-Beispiel: Vor einer Weggabelung stellt der Fahrer das MP „Fahren links“ und die Automation das „Fahren rechts“ dar. Aus dem jeweiligen Lenkmoment können die Agentenrepräsentationen das jeweilige MP des Agenten bestimmen, wie stark er nun nach links bzw. nach rechts fahren möchte. Ein Modul, das für die Aktionstension der Querverführung zuständig ist, ordnet die MPs der Agenten der Berechnung der Anspannung bezogen auf diese Aktionstension zu. Solche Daten können über eine standardisierte Schnittstelle (CoFoR) an die nachfolgenden Ebenen weitergereicht werden. Dort können sie zum Zweck der Gestaltung der CA-Wechselwirkung weiterverarbeitet werden. Alle CA-Manipulationen und damit alle MPs können in nachgelagerten Ebenen mittels Arbitrierungsstrategien und -signale, die entlang der operationalisierten Aktionstensionen platziert sind, zeitlich und qualitativ abgestimmt werden.

Auf der *zweiten Ebene* „Arbitrierung“ (magenta) kann die CA-Wechselwirkung durch Arbitrer beeinflusst werden. Nach dem Konzept des Zeithorizonts der Beeinflussung des Agentenverhaltens (vgl. Abschnitt 4.1.2) können unterschiedliche Arbitrer für die Arbitrierung langfristiger, kurzfristiger und unmittelbarer Konflikte eingesetzt werden. Auf dieser Ebene können akute oder potenzielle Konflikte aus Daten der ersten Ebene entdeckt, und mittels passender Arbitrierungsstrategien (vgl. Abschnitt 4.3.3) gelöst oder umgangen werden.

Wenn man eine Analogie zu den Begriffen der Sprachwissenschaft zieht (vgl. Abschnitt 2.4), befindet man sich hier auf der morphologischen (sinnvolle Wortbildung) und syntaktischen (sinnvolle Satzbildung) Ebene der Informationsgestaltung. Diese Gestaltung wird auf der *dritten Ebene* (Arbitrierungsmodell) (cyan) über die Schnittstelle „Arbitrierungsstrategie und -Signalgestaltung“ vollzogen (vgl. Abschnitt 4.4.3). Insgesamt können auf der zweiten und dritten Ebene der Architektur durch Arbitrierung die negativen (Konflikte) und unbestimmten CA-Wechselwirkungen mittels der Beeinflussung des Agentenverhaltens (vgl. Abschnitt 4.1.1) zur positiven CA-Wechselwirkung (Kooperation) entwickelt werden.

Y-Beispiel: Der Konflikt (Anspannung) könnte mit der Arbitrierung unmittelbarer Effekte gelöst werden, z.B. durch die teilweise oder vollständige Entkopplung des objektiv falsch agierenden CA durch den Arbitrer (vgl. Abschnitt 4.3.6). Auf der Unterebene der kurzfristigen Effekte könnte der Konflikt schon vorher erkannt gewesen sein, z.B. durch die Manövererkennung des Fahrers. Dann kann der dazugehörige Arbitrer an den Fahrer rechtzeitig die Lenkpräferenz der Automation kommunizieren. Der Konflikt könnte auch auf der Unterebene langfristiger Effekte durch die Anregung des korrekten Rollenverhaltens umgangen werden.

Auf der *vierten Ebene* „Signalgestaltung und -kanalisierung“ (gelb) kann eine Umsetzung der gestalteten Arbitrierungsstrategien in Form von konkreten Arbitrierungssignalen platziert werden. Hier kann die Definition des Verhaltens, die Parametrierung, die Berechnung, die Priorisierung und die Kanalisierung (vgl. Abschnitt 3.2) der Signale stattfinden.

Wenn man hier eine Analogie zu den Begriffen der Sprachwissenschaft zieht, befindet man sich auf dieser Ebene auf der phonologischen (sinnfreien Lautbildung) Ebene der Informationsgestaltung. Die Definition, welche Arbitrierungssignale es gibt, wie und wann sie in das CS injiziert werden, geschieht auf der dritten Ebene (Arbitrierungsmodell, Abschnitt 4.4.3).

Y-Beispiel: Ein Konflikt ist festgestellt und muss gelöst werden. Der Systemingenieur hat sich für eine Arbitrierungsstrategie entschieden, an den Fahrer rechtzeitig die Lenkpräferenz der

Automation zu senden. Dies könnte mittels eines richtungsaufgelösten haptischen, akustischen oder visuellen Reizes geschehen. Der Reiz bekommt auf der dritten Ebene seine semantische und syntaktische Bedeutung, erhält auf der vierten Ebene seine konkrete Form, wie Frequenz, Amplitude, und wird über den zugeordneten multimodalen Kanal (vgl. Abschnitt 3.2) innerhalb des Informationsflusses priorisiert und zum Ort der Darstellung kanalisiert.

Über eine Schnittstelle können ausgestaltete Arbitrierungssignale auf die *fünfte Ebene* (Interaktionshardware) (braun) der gestaltungskompatiblen Architektur kanalisiert werden. Dabei bedeutet die Interaktionshardware die CA-kompatiblen aktiven multimodalen Eingabe- und Ausgabevorrichtungen. Das können aktive haptische Stellteile, visuelle und akustische Displays aber auch eine nicht näher spezifizierte Interaktionshardware zum kognitiven Automationsagenten sein. Damit wird in der Architektur der Wirkungskreis zwischen der ersten Ebene mit als Agentenrepräsentationen und Aktionstensionen beschriebenen CS-Strukturen und der Beeinflussung des CS-Verhaltens durch Arbitrierungssignale auf der fünften Ebene geschlossen.

Y-Beispiel: Der Systemingenieur hat sich für die Arbitrierungsstrategie entschieden, an den Fahrer rechtzeitig die Lenkpräferenz der Automation zu senden. Er hat sich entschieden, einige Sekunden vor der Weggabelung einen haptischen Hinweis an den Fahrer zu kommunizieren. Dieser Hinweis kommt auf der fünften Ebene an und wird dort als ein kurzer Ruck nach rechts auf dem aktiven Lenkrad dargestellt. Der Fahrer nimmt diesen Ruck wahr und passt sein Lenkverhalten dem der Automation an. Damit aktiviert er sein MP für die erste Ebene.

Insgesamt hat die Architektur eine relativ einfache und ganzheitliche Grundstruktur, wobei trotzdem alle osGekoS-Konzepte, angefangen mit MPS über Aktionstension bis zur Arbitrierung, verwendet werden. Auf der ersten Ebene der Architektur befinden sich die Agentenrepräsentationen, ihre CA-Manipulationen und Aktionstensionen, auf der zweiten Ebene die Arbitrierer. Durch den Bezug zu sprachwissenschaftlichen Begriffen und Konzepten ist die Architektur für die Aufnahme einer gestaltungskompatiblen Schnittstelle vorbereitet. Diese Schnittstelle wird in Abschnitt 4.4.3 ausführlich erarbeitet und eingeführt.

4.4.3 Zur Gestaltungsschnittstelle

Operative Gestaltung und Nutzung eines CS lässt sich mit dem Konzept des Senders und des Empfängers^[Shn48] beschreiben^[Ket13]. Der Systemingenieur ist während der Gestaltung und der Arbitrierung zur Laufzeit der Sender der Information. Der Systemnutzer ist der Empfänger. Zwischen dem Systemingenieur, der während der Systemgestaltung über Arbitrierer zur Laufzeit eine beabsichtigte Information senden möchte, und einem Nutzer, der diese Information während der Systemnutzung empfängt, besteht ein Informationsfluss (Abbildung 4-12).

Bei der Informationsgestaltung steht für den Systemingenieur die erste zentrale Frage im Vordergrund: „*Wann, wem, welche* Information muss mitgeteilt werden und *was* bedeutet sie?“ Die Antwort auf diese Frage impliziert die Antworten auf die Teilfragen: „Welche Aktionstensionen können im CS identifiziert und operationalisiert werden? Welche MPs können damit angeregt oder gehemmt werden? Wann und welche Konflikte (Anspannungen) gibt es, die durch die Arbitrierung gelöst oder umgangen werden können? Was sind die geeigneten Arbitrierungsstrategien dafür? Was bedeuten einzelne Arbitrierungssignale und ihre bestimmte Anordnungen? Wie sehen diese Signale auf einem konkreten HMI aus?“

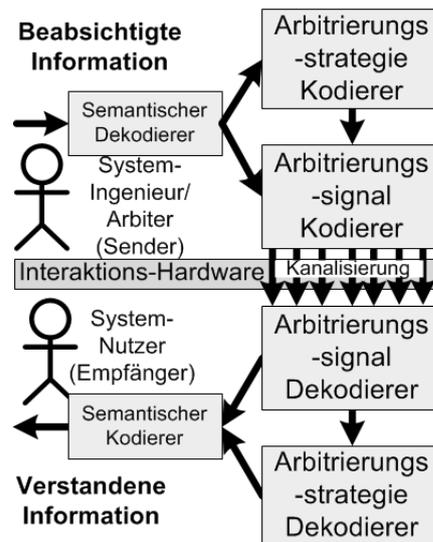


Abbildung 4-12: Informationsfluss zwischen dem Systemingenieur/Arbiter und dem Systemnutzer

Antworten auf die gestellten Fragen können durch die Modellierung von Konflikten mit Hilfe passender Nutzungsszenarien (engl. use case)^[OMG11] und durch das vorherige Verstehen der Wirkung der Arbitrierungssignale gefunden werden (vgl. auch Abschnitt 5.3). Die Analyse der Nutzungsszenarien kann mittels deduktiver und/oder induktiver Methoden vollzogen werden. Deduktive Methoden können die Literaturrecherche, kognitive Aufgaben-, Funktions-, Arbeitsanalyse etc. sein (vgl. Abschnitt 2.3). Induktive Methoden können die systematische Definition eines Gestaltungsraums und seine Exploration sein^[FIS08].

Streng problemorientiert wird die beabsichtigte Information in ihre morphologische (Bedeutung, Symbole), syntaktische (Anordnung, Sätze) und phonetische (Form, Zeichen) Komponenten zerlegt. Bereits bekannte Problem-Lösungspaare bzw. Gestaltungsmuster können dabei verwendet werden (vgl. Abschnitte 2.2.6 und 2.3.5). Solche Informationsdekomposition ist in der Abbildung 4-12 als „semantischer Dekodierer“ gezeigt. Das Ergebnis der semantischen Dekodierung können deutlich formulierte und konsistente Informationseinheiten, wie Phrasen und Wörter sein. Die konsistente Interaktionsgestaltung ist dabei wichtig (vgl. Abschnitt 2.4). Die Verknüpfung zwischen der Bedeutung einer Informationseinheit und deren Form bzw. Signalverlauf kann mit dem Werkzeug und Methode „Semantisches Mapping“ bewerkstelligt werden. Alle Informationseinheiten können am Ende als ein „Syntaktisches Storyboard“ vorliegen. Beide genannten Werkzeuge werden in Kapitel 5 genauer eingeführt.

Unter der Annahme, dass es möglich ist, die CA-Wechselwirkung mithilfe der operationalisierten Aktionstensionen (vgl. Abschnitt 4.2.2) und der zusätzlich durch Arbitrer in das System injizierten Signale zu beeinflussen, können Arbitrierungsstrategien und -signale entwickelt werden. Die morphologischen und syntaktischen Komponenten können für die Definition der Arbitrierungsstrategien und die phonetische für die Definition und die Parametrierung der Arbitrierungssignale verwendet werden. Dieses Vorgehen ist in der Abbildung 4-12 als zwei dezidierte Kodierer dargestellt. Diese Kodierer integrieren sich jeweils in die zweite und in die vierte Ebene der gestaltungskompatiblen CS-Architektur (vgl. Abschnitt 4.4.2).

Nach der Definition der Arbitrierungsstrategien und -signale kann die zweite zentrale Frage beantwortet werden: „Wie kann die beabsichtigte Information zu den CAs kommuniziert werden und welche multimodalen Kanäle und Unterkanäle können dafür verwendet wer-

den?" Nach der Beantwortung dieser Frage können die gestalteten Arbitrierungssignale auf multimodale Kanäle und Unterkanäle verteilt werden und in Richtung der Empfänger-CAs über die Interaktionshardware kanalisiert werden. Während der Systemnutzung empfängt der CA die kanalisierte Information, dekodiert sie auf phonetischer, morphologischer und syntaktischer Ebene und setzt sie am Ende zu „verständener Information“ zusammen.

Y-Beispiel: Der Systemingenieur löst den Konflikt bereits bei der Systemgestaltung. Er identifiziert eine Aktionstension, die die dichotomen Handlungen (links lenken - rechts lenken) steuert. Er operationalisiert diese Aktionstension mit dem Lenkmoment und seiner Aufteilung in MPs, wie „stark lenken“, „schwach lenken“, „nicht lenken“. Zwischen den MPs sind Trigger-Ereignisse für das Versenden der Arbitrierungssignale platziert. Der Systemingenieur entscheidet sich für die Arbitrierungsstrategie, rechtzeitig vor einer Weggabelung an den Fahrer einen Richtungshinweis über die bevorstehende Automationshandlung zu geben. Dies soll dann passieren, wenn der Fahrer „schwach“ in die der bevorstehenden Automationshandlung entgegengesetzte Richtung lenkt. Der Hinweis soll ein „Ruck“ mit der Bedeutung „Ich, Automation, will nach rechts“ auf dem haptischen Kanal des Lenkrades auf dem Unterkanal der diskreten haptischen Elemente mit dem maximalen Moment von 2 Nm und der Dauer von 0,3 s sein. Vor einer Weggabelung nimmt der Fahrer den „Ruck“ auf dem Lenkrad wahr, dekodiert ihn als Richtungshinweis und passt sein Verhalten dem der Automation an.

Eine solche ganzheitliche Interaktionsgestaltung benötigt oft die Teilnahme und die Diskussion zwischen Personen aus verschiedenen Fachdisziplinen: Ingenieuren, Designern, Psychologen und Informatiker. Um die Gestaltung für alle beteiligten Fachleute leicht verständlich und nutzbar zu machen, ist es notwendig, die Interaktion in einer übersichtlichen Art und Weise zu diskutieren und zu implementieren. Abbildung 4-13 links zeigt ein Konzept für eine Gestaltungsschnittstelle, die sich zwischen der zweiten (Wechselwirkungsmanagement) und der vierten (Elementgestaltung und -kanalisierung) Ebene der bereits vorgestellten gestaltungskompatiblen CS-Architektur (vgl. Abbildung 4-11) integrieren lässt.

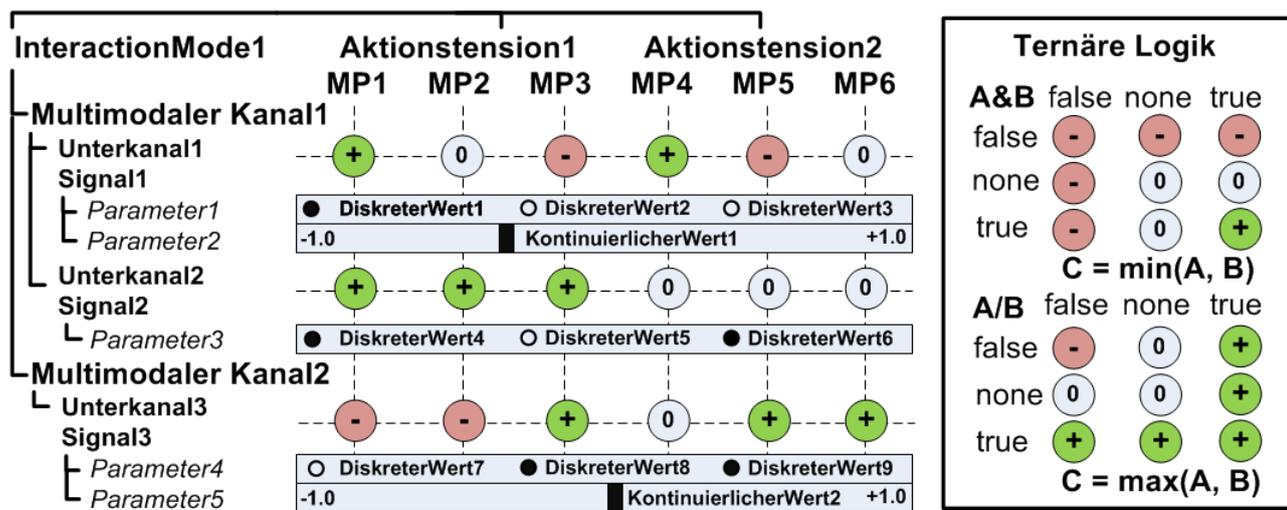


Abbildung 4-13: Schnittstelle zur gestaltungskompatiblen Architektur der kognitiven Systeme über ein gestaltbares Arbitrierungsmodell (links). Wahrheitstabellen für AND und OR der ternären Logik (rechts)

Die Abbildung 4-13 zeigt ein Konzept für eine GUI, die vom Systemingenieur für die Gestaltung der Arbitrierung verwendet werden kann. Waagrecht sind die Trigger-Ereignisse aller Aktionstensionen, wie laterale, longitudinale oder kooperative Aktionstension (vgl. Abschnitte

5.5, 6.1.5, 6.2.5) platziert. Senkrecht ist eine Hierarchie, die sich über multimodale Kanäle, Unterkanäle (vgl. Abschnitt 3.2), Arbitrierungssignale bis hin zu deren Parametern aufspannt. In der Mitte sind die logischen Verknüpfungen zwischen den Trigger-Ereignissen und Arbitrierungssignalen, sowie manipulierbare Einstellungsfelder für die Signalparameter platziert.

Nach der Definition der Aktionstensionen, der Morphologie und der Syntax der Arbitrierungsstrategien kann ein Arbitrierungsmodell mittels der ternären Logik^[Kle38] erstellt werden^[Ket13]. Das Verhalten der multimodalen Unterkanäle, die die Arbitrierungssignale führen, und die von den operationalisierten Aktionstensionen abgeleiteten Trigger-Ereignisse sind logisch miteinander verknüpft. Die verwendete ternäre Logik besteht aus drei logischen Zuständen: Einschalten (engl. „enable“, +), ausschalten (engl. „disable“, -) und „egal“ (engl. „don't care“, 0). Die Wahrheitstabellen der Logik sind in der (Abbildung 4-13 rechts) dargestellt. Aktivierung eines multimodalen Unterkanals und damit die Aktivierung eines Arbitrierungssignals sind logisch durch die ternäre UND-Verknüpfung bestimmt. Die Signalpriorisierung auf dem gleichen multimodalen Unterkanal ist durch die ternäre ODER-Verknüpfung bestimmt.

Damit ermöglicht die ternäre Logik einerseits die Implementierung des Gedanken der multiplen Ressourcen (vgl. Abschnitt 3.2) und der Konzepte der Beeinflussung des Agentenverhaltens (vgl. Abschnitt 4.1.1). Und andererseits zeigt sie, dass die binäre Logik nicht ausreichend und eine höherwertigere Logik bereits zu viel wäre. Denn in den genannten osGekoS-Konzepten war nicht mehr als Aktivieren (Einschalten), Hemmen (Ausschalten) oder Ignorieren von Signalen auf parallelisierbaren Kommunikationskanälen vorgeschlagen und gefordert.

Arbitrierungssignale selbst können durch die entsprechende Elementparameter konfiguriert werden. Auf diese Weise kann innerhalb der vorgeschlagenen Gestaltungsschnittstelle ein eindeutiges Arbitrierungsmodell erzeugt werden, das nahtlos in die gestaltungskompatible CS-Architektur integriert wird. Dieses Modell kann innerhalb des jeweils entwickelten CS auf die verwendete Simulationsumgebung abgestimmt und in Usability Studien geprüft werden.

Die vorgeschlagene Gestaltungsschnittstelle kann eine einfache, offene und ganzheitliche Gestaltung von Arbitern bieten. Eine schnelle Übertragbarkeit der Arbitrierungsmodelle zwischen den verschiedenen CS-Umgebungen, eine Wiederverwendbarkeit von Arbitrierungsmodellen und ein geordneter Dialog zwischen den beteiligten Fachleuten verschiedener Disziplinen sind weitere Vorteile dieser Schnittstelle. Damit werden die Gestaltung der Arbitrierung und auch der gesamte CS-Gestaltungsprozess ganzheitlich, nachvollziehbar, gut dokumentierbar, archivierbar und konsistent. Das osGekoS-„Arbitrierungswerkzeug“ nutzt sie (vgl. Abschnitt 5.6). Ebenda findet sich auch ein konkretes Beispiel, für den Einsatz der Gestaltungsschnittstelle.

4.4.4 Zusammenfassung

Das Konzept der gestaltungskompatiblen CS-Architektur besteht aus einer vertikalen Anordnung von Ebenen mit einer horizontalen Gestaltungsschnittstelle. In seiner integrierten Form bietet es Platz für alle Konzepte von osGekoS. So ist es möglich, innerhalb der Architektur die Schlüsselkonzepte, wie MPS, Aktionstension und Arbitrierung, zwecks einer holistischen Systemgestaltung zu verwenden. Die Architektur bleibt dabei einfach, übersichtlich, flexibel und skalierbar. Durch die fest verankerte Basis in osGekoS trägt die vorgeschlagene Architektur auch seine möglichen Nachteile, wie schwierige Operationalisierbarkeit der Aktionstensionen.

Das Architekturkonzept basiert auf formalen Überlegungen (14). Es ermöglicht genau das, was (14) vorschlägt: Man gibt damit die Aktionstensionen und die damit arbeiteten Arbitrierer in das ursprüngliche CS. Nach (14) sollte die Architektur ihren Zweck erfüllen, nämlich das CS gestaltbar bzw. gestaltungskompatibel zu machen. Mittels des vorgeschlagenen Architekturkonzeptes kann daraus eine konkrete Software/Hardware Architektur abgeleitet werden (vgl. Abschnitt 5.7). Zu der vertikalen Architektur gehört eine horizontale Gestaltungsschnittstelle. Unter Berücksichtigung der osGekoS-Konzepte und deren Formalisierungen können operativ holistische, skalierbare und übertragbare Arbitrierungsmodelle entwickelt werden.

4.5 Zusammenfassung der Gestaltungskonzepte

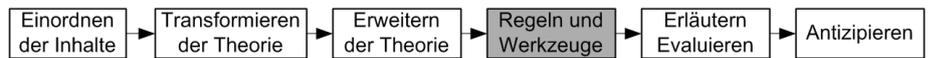
Innerhalb von osGekoS kann das CS-Verhalten mit dem MPS-Konzept als eine Veränderung der CS-Struktur verstanden und beschrieben werden. Die formale Basis dazu ist in Abschnitt 3.5 erarbeitet. In diesem Kapitel wurden noch fehlende theoretische und formale Grundlagen von osGekoS erarbeitet. Durch eine Gestaltung kann das CS-Verhalten vom Systemingenieur beeinflusst werden. Effekte des Einflusses können über drei Ebenen verteilt werden: Langzeit-, Kurzzeit- und unmittelbare Effekte. Das negative CS-Verhalten kann durch die Beeinflussung der CA-Manipulationen gehemmt, ignoriert und das positive CS-Verhalten angeregt werden.

Für die Beeinflussung der CA-Manipulationen können Steuerstrukturen, die Aktionstensionen, aus inhärenten Motivationsstrukturen der CAs auf die CS-Ebene externalisiert werden. Aktionstensionen sind dichotome Ketten von MPs, die für die Steuerung von CS verwendet werden können. Dafür müssen sie operationalisiert werden und nützen dann mit Trigger-Ereignissen für gestaltbare und in das CS injizierbare Manipulationen zum Zweck der Synchronisierung und der Abstimmung. Die Formale Basis dazu ist in Abschnitt 4.2.3 erarbeitet.

Mittels der Aktionstensionen kann eine moderierte Verhandlung mit der impliziten oder expliziten Entscheidung innerhalb der verfügbaren Ressourcen, die Arbitrierung, entwickelt werden. Durch die Arbitrierung kann das negative und unbestimmte CS-Verhalten gehemmt und in Richtung des positiven CS-Verhaltens entwickelt werden. Dafür können Arbitrierungsstrategien und -signale prospektiv gestaltet und an Trigger-Ereignissen der Aktionstensionen durch technische Module, die Arbitrierer, zwecks Konfliktlösung oder -prävention in das CS zur Laufzeit injiziert werden. Die Formale Basis dazu ist in Abschnitt 4.3.5 erarbeitet.

Alle erarbeiteten Konzepte von osGekoS können in einer gestaltungskompatiblen CS-Architektur zusammengeführt werden. Sie besteht aus einer vertikalen Anordnung von Ebenen, wo die Konzepte von MPS, Aktionstension und Arbitrierung ihre Anwendung finden. Arbitrierungsstrategien und -signale können hier mittels horizontaler Gestaltungsschnittstelle und den genannten formalen Basen der osGekoS-Konzepte als Arbitrierungsmodelle entworfen und in einer Usability Studie getestet werden.

Neben dem theoretischen und holistischen Verständnis der CS, ihrer Strukturen, Verhalten und Prozesse bietet osGekoS nun einen Mehrwert in Form einer deskriptiven und einer formalen Basis für die operative Gestaltung der CS. Konkrete Prozesse, Regeln und Werkzeuge für die Gestaltung werden im nächsten Kapitel behandelt.



5 Prozess, Regeln und Werkzeuge der Gestaltung

Jeder Gestaltungsrahmen besteht aus einer Theorie, einem Regelwerk und einer Werkzeugkette (vgl. Abschnitt 1.3). In Kapiteln 3 und 4 wurde ausführlich die komplette Theorie der operativen strukturfokussierten Gestaltung kognitiver Systeme (osGekoS) erarbeitet und eingeführt. Das Ziel dieses Kapitels ist, den Prozess, die Regeln und die Werkzeugkette von osGekoS einzuführen. Dabei ist zu berücksichtigen, dass dieses Regelwerk und die Werkzeugkette keine vollständig in sich geschlossene Sammlung von Vorschriften, Checklisten etc. darstellt. Vielmehr handelt es sich um eine Erweiterung der bestehenden Gestaltungsrahmen (vgl. Kapitel 2) um die operative und strukturfokussierte Perspektive auf die CS-Gestaltung.

5.1 Konfliktanalysewerkzeug

Der erste Schritt der Gestaltung eines CS sollte seine strukturelle und funktionale Dekomposition sein. Dafür eignet sich das MPS-Konzept (vgl. Abschnitte 3.1, 3.2, 3.3, 3.4). Das kognitive System kann als eine Menge von Agenten (MPs) zerlegt werden, die Handlungen aufgrund von Aufgaben ausführen. Regel: *Jedes handelnde, Aufgaben ausführende Element eines CS ist ein Agent (MP)*. Die Dekomposition des CS kann durch die kognitive Arbeitsanalyse (CWA), speziell durch die Methode „Abstraktions-Zerlegungsraum“ (vgl. Abschnitt 2.3.3) unterstützt werden. Handlungen und Aufgaben, die die Agenten auszuführen haben, können mittels der GOMS, der HTA und der CFA (vgl. Abschnitte 2.3.2 und 2.3.3) erarbeitet und als CoFoR und Aktionstensionen operationalisiert und festgehalten werden (vgl. Abschnitt 4.2).

Nach der Zerlegung des CS in Agenten und Aktionstensionen kann ein problem- bzw. konfliktorientiertes Vorgehen stattfinden. Denn alle Agenten stehen in einer komplexen Wechselwirkung miteinander und können Konfliktsituationen erzeugen. Regel: *Bei der Gestaltung von CS müssen Konflikte (Anspannungen) innerhalb des Systems aufgedeckt und aufgelöst werden* (vgl. (15)). Dazu gehören sowohl akute Konflikte, wie zeitlich und qualitativ fehlabgestimmte Handlungen der CAs (vgl. Abschnitte 4.1 und 4.3), als auch potenzielle und abstrakte Konflikte im System, wie fehlende Kooperation, Effizienz, Sicherheit oder Komfort. Solche „Konfliktsuche“ kann mittels des Konfliktanalysewerkzeugs^[D3S12] stattfinden. Die Tabelle 5-1 zeigt dieses Werkzeug mit dem eingetragenen Beispiel des Kontrollkonfliktes an einer Weggabelung bei ihrem Durchfahren in einem hochautomatisierten Fahrzeug (vgl. Abbildung 3-1).

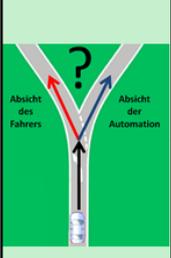
Das Werkzeug ist in drei Abschnitte aufgeteilt. Im linken (grünen) Abschnitt können die Nutzungsszenarien definiert und abgebildet werden. Dort wird auch die aus allen Agenten und ihnen zugeordneten Aufgaben bestehende Grundkonfiguration platziert. Im mittleren (roten) Abschnitt des Werkzeugs findet die Konfliktanalyse statt, wobei mögliche Konflikte bereits vorsortiert sind (vgl. Tabelle 4-1). Diese Vorsortierung sieht agenten- und umweltbezogene Konflikte vor. Agentenbezogen sind Konflikte „innerhalb“ der Agenten, wie durch den Fehlgleich von Situationsbewusstsein, mentaler Modelle, Aufmerksamkeit etc. bedingte Konflikte. Diese Konfliktkategorien gelten sowohl für menschliche als auch technische Agenten.

Des Weiteren können Konflikte auf der Ebene der Intentionbildung und Entscheidungen auftreten, wie Zielsetzungs- oder Aufgabenzuordnungskonflikte. Außerdem können Konflikte in

der Koordination und der Handlung der Agenten stattfinden. Diese Kategorien sind besonders für die Operationalisierung der Aktionstensionen wichtig. Im linken (blauen) Abschnitt können schließlich erste Lösungsideen für gefundene Konflikte notiert werden (vgl.

Tabelle 4-2). Diese bilden die Grundlage für die Gestaltung der Arbitrierungsstrategien.

Tabelle 5-1: Konfliktanalysewerkzeug des osGekoS mit dem eingetragenen Beispiel des Kontrollkonfliktes an einer Weggabelung beim Durchfahren in einem hochautomatisierten Fahrzeug. Es ist in MS-Excel umgesetzt

SZENARIO	ANFANGS-SYSTEM-ZUSTAND		KONFLIKTBESCHREIBUNG					LÖSUNG
			Agentenbezogene Konflikte				Umgebungs-konflikte	
			Bewusstsein	Intention Entscheidung	Koordination	Handlung		
Agenten	Aufgaben	Aufmerksamkeit, Vertrauen, Situationsbewusstsein: Wahrnehmen, Verstehen, Antizipieren	Zielsetzung, Aufgabenzuordnung und -verteilung, unklare oder keine Alternativen etc.	Kommunikation, Interaktion, Arbitrierung, trust, mental models of partner behaviour, interface	motoric constraints, lack of skills, physical constraints, contact	weather, infrastructure conditions, environmental limits, physical limits, legal issues etc.		
	Fahrer	Durchfahren der Weggabelung	Abgelenkt	Alternative Fahrtrichtung nicht bekannt	Erkennt nicht rechtzeitig, was die Automation vorhat	lenkt (schwach, stark) nach links	Konfliktvermeidung: Mittelung der Automationsintention an den Fahrer rechtzeitig vor der Weggabelung Konfliktlösung: Entkopplung der Agenten von der Fahrzeugkontrolle	
	Automation	Durchfahren der Weggabelung	Sensorfehler	Alternative Fahrtrichtung nicht bekannt	Erkennt nicht rechtzeitig, was der Fahrer vorhat	lenkt (schwach, stark) nach rechts		

Das Konfliktanalysewerkzeug kann auch zum Erarbeiten von Systemanforderungen und zum Risikomanagement verwendet werden (vgl. Abschnitt 2.2.1). Weil hier die möglichen Konflikte explizit in den Vordergrund der Gestaltung gerückt werden, bilden sie eine solide Grundlage für die Formulierung möglicher Risiken und Systemanforderungen.

5.2 Lösungsexploration

Für die Präzisierung von Lösungen zu den aufgedeckten Konflikten in Form von Arbitrierungsstrategien kann die Methode und das skalierbare Werkzeug „Lösungsexploration“ verwendet werden. Es ist ein wichtiges Werkzeug des osGekoS, da es den Fokus direkt auf die Organisation bzw. die Struktur des CS legt. Regel: *Systemverhalten ergibt sich aus der Veränderung der Systemstruktur* (vgl. (12)). Nach der Dekomposition der CS-Struktur in Aktionstensionen und CAs und der CA-Strukturen in MPs, kann ein Modell des CS erstellt werden. Die Kopplungsmatrix (5) und der Anfangsvektor der aktiven MPs können definiert werden. Das Werkzeug beinhaltet die Anwendung von (11) zwecks einer explorativen Lösungsfindung.

Die Abbildung 5-1 zeigt einen Ausschnitt des Werkzeuges, mit dem die Lösungsexploration aus dem Abschnitt 4.3.6 mit dem Ergebnis der Tabelle 4-3 für das Beispiel des Kontrollkonfliktes an einer Weggabelung (vgl. Abbildung 3-1) durchgeführt wurde.

Oben im Werkzeug befindet sich die jeweilige Zeitreihenentwicklung des CS-Modells mit der entsprechenden Systemkonfiguration in Form der Kopplungsmatrix und des Anfangsvektors

der aktiven MPs (oben links). In der Mitte wird das Diagramm der Zeitreihenentwicklung automatisch gezeichnet. Darunter können zu explorierende Systemkonfigurationen mit den dazugehörigen Strategiebeschreibungen eingetragen und sortiert werden. Mit der Lösungsexploration kann die Strukturveränderung des CS und damit sein Verhalten systematisch und ohne Einsatz von Gestaltungsheuristiken untersucht werden. Dies kann bei der operativen Gestaltung funktionaler Arbitrer, Arbitrierungsstrategien und -signale für die Beeinflussung der CA-Wechselwirkung nützen und den Gestaltungsprozess beschleunigen.

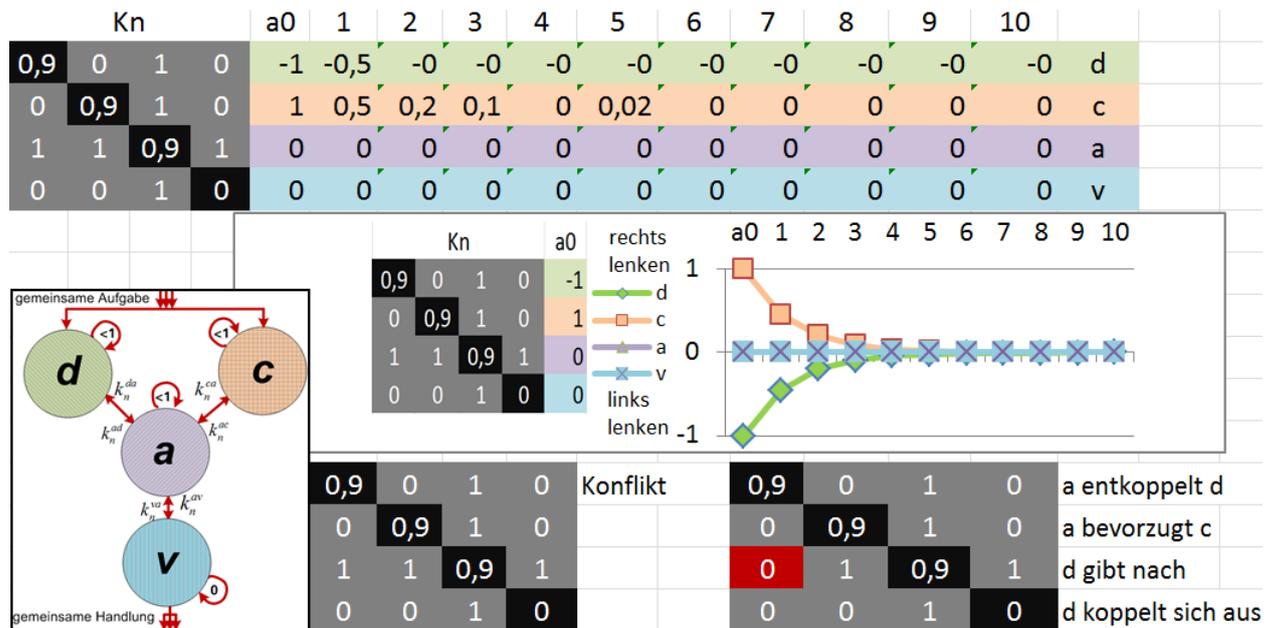


Abbildung 5-1: Werkzeug zur Lösungsexploration, mit dem Entscheidungskonvergenzen innerhalb des CS und damit das CS-Verhalten exploriert werden können. Es ist in MS-Excel umgesetzt

5.3 Semantisches Mapping

Bevor Arbitrierungsstrategien für die Lösung entdeckter Konflikte eindeutig festgelegt werden können, ist es wichtig, die dazugehörigen Arbitrierungssignale gestaltet zu haben. In Abschnitt 2.4 wurde bereits erklärt, dass jedes Interaktionssignal vom Systemnutzer als ein Zeichen interpretiert und verwendet wird. Daher muss die Entstehung eines Zeichens (Semiose) beachtet werden. Die Form, die Bedeutung und der Nutzer eines Zeichens stehen zueinander in einer komplexen Beziehung. Genau diesen Sachverhalt kann der Systemingenieur mit der Methode und dem Werkzeug „Semantisches Mapping“ ausnutzen.

In Vorstudien mit potenziellen Systemnutzern kann ein wiederverwendbares und vom Nutzer gut verstandenes „Alphabet“ der Arbitrierungssignale entwickelt werden. Dies kann mittels nutzerzentrierter Gestaltungsmethoden stattfinden, wie Nutzerbeobachtungen, Befragungen, Fragebögen und das Ausprobieren von bereits erstellten Prototypen der Zeichen und ihrer Varianten (vgl. Abschnitt 2.3.2). Dabei kann mit verschiedenen Modalitäten, wie haptisch, akustisch, visuell, experimentiert werden. Wichtig ist dabei die Beachtung und die Verwendung der Wahrnehmungs- und der Koinzidenzschwellen in Abhängigkeit von der Amplitude des Signals, seiner Frequenz, Angriffsfläche und Verlaufes. Auch bereits bekannte Gestaltungsmuster, wie Image-Schemata (vgl. Abschnitt 2.3.6), können hier sehr nützlich sein. Eine Regel der Erstellung eines solchen Alphabets ist: *Die Konsistenz und Eindeutigkeit in der Zuordnung*

der Bedeutungen zu den jeweiligen Arbitrierungssignalen ist gefordert. Sie sollte bei einer Ablehnung an kurze, prägnante animalische Ausdrucksformen gegeben sein (vgl. Tabelle 2-4).

Beispiel: Wenn eine Vibration für „Gefahr“ bereits verwendet wird, dann sollte sie nicht in einem anderen Zusammenhang, wie für „Hinweis“ oder „Frage“ verwendet werden. Dies hilft dem Nutzer, das System besser zu verstehen, das Vertrauen in das System aufzubauen, zu erhalten und das entwickelte „Alphabet“ der Arbitrierungssignale schnell zu erlernen.

In der Tabelle 5-2 ist ein Beispiel aus dem Alphabet für haptische Signale als entsprechendes Werkzeug für semantisches Mapping abgebildet. Es zeigt von links nach rechts, wie Arbitrierungssignale, auf haptische Unterkanäle aufgeteilt, ihre von Variationsmöglichkeiten abhängigen Bedeutung und Form bekommen können. Es ist bei der Gestaltung der in Kapitel 6 präsentierten Beispiele für Assistenzsysteme aus dem Bereich hochautomatisierter Fahrzeugführung entstanden. Abgebildet ist das Mapping zwischen einfachen haptischen Signalformen (links) und ihrer Bedeutung (rechts). Dazwischen sind die Elementnamen und Variationsmöglichkeiten der Signale notiert. Regel: *Komplexe Signale sollten aus Kombinationen einfacher Signale erstellt werden, ohne ihre eigenen Bedeutungen zu verlieren.* Dies sollte bei einer Ablehnung an animalische Ausdrucksformen gegeben sein (vgl. Tabelle 2-4).

Tabelle 5-2: Auszug aus dem Werkzeug „Semantisches Mapping“ haptischer Signale für die Gestaltung hochautomatisierter Fahrzeugführung. Das Werkzeug ist in MS-Excel umgesetzt

Verlauf	Element	Variationsmöglichkeiten	Generische Bedeutung
Diskret	SingleTick	Richtung, Amplitude, Dauer des Ticks	Gerichtete Handlungsanregung
Diskret	DoubleTick	Richtung, Amplitude, Dauer des einzelnen Ticks, Dauer zwischen den einzelnen Ticks	Gerichtete Handlungsanregung mit Nachdruck
Diskret	N-Ticks	Richtung, Amplitude, Dauer des Ticks Dauer zwischen den einzelnen Ticks: konstant, ansteigend, absteigend	Handlungsaufforderung (z.B. als "Schlackern")
Mischform	Vibrationspulse	Frequenz: konstant, ansteigend, absteigend Amplitude: konstant, ansteigend, absteigend	"Ich funktioniere!"-Signal Zustandstransition
Kontinuierlich	Vibration	Frequenz: konstant, ansteigend, absteigend Amplitude: konstant, ansteigend, absteigend	Ungerichtete Hemmung der Handlung
Kontinuierlich	Kraftschwelle	Anfangskraft, geometrische Maße, Flankensteilheit, Endkraft	Gerichtete Hemmung der Handlung
Kontinuierlich	Kraft	Richtung, Betrag	Handlungskorrektur "weg von der negativen Valenz"
Kontinuierlich	Position	Absolute Position auf dem Stellteil	Handlungskorrektur „hin zu positiver Valenz“
Kontinuierlich	Steifigkeit	Flankensteilheit der Federkraft um den Nullpunkt	Anspannung/Entspannung
Zusammengesetzt	Virtuelles Kiesbett	vgl. Vibration + Kraftschwelle + Kraft	Bildet das Verhalten eines realen Kiesbetts nach

Die Methode „Semantisches Mapping“ ist die Erfassung der Bedeutung der einzelnen Arbitrierungssignale und eine Grundlage für die erfolgreiche Moderation und Entscheidung (Arbitrierung) bei Konflikten. Diese Methode und das Werkzeug können neben der haptischen auf jede andere Modalität der Kommunikation und der Interaktion angewendet werden.

5.4 Arbitrierungsmatrix

Nachdem das CS dekomponiert ist, alle Einzelkonflikte, noch grob formulierte, explorierte Lösungswege und die Interaktionssprache der CAs bekannt sind, kann der Systemingenieur mit der weiteren Detaillierung der Arbitrierungsstrategien beginnen. Dafür eignet sich das Werkzeug „Arbitrierungsmatrix“^[INY12]. Für die im Konfliktanalysewerkzeug erfassten Konflikte können in der Arbitrierungsmatrix detaillierte Lösungen erarbeitet werden.

Y-Beispiel: Tabelle 5-3 zeigt die Arbitrierungsmatrix mit dem eingetragenen Beispiel des Kontrollkonfliktes an einer Weggabelung bei ihrem Durchfahren in einem hochautomatisierten Fahrzeug (vgl. Abbildung 3-1). Konflikt auf der Handlungsebene (Fahrer lenkt links, Automation lenkt rechts) kann nach dem Konzept des Zeithorizonts der Beeinflussung des CA-Verhaltens (vgl. Abschnitt 4.1.2) auf mehreren Ebenen der Fahrzeugführung behandelt werden. In der Arbitrierungsmatrix können für unterschiedliche Kombinationen der MPs des so zerlegten (möglichen) Lenkverhaltens des Fahrers und der Automation separate Strategien entwickelt werden. Das unterschiedlich starke Lenken des jeweiligen CA kann als Folge des vom CA geplanten Abbiege- bzw. Ausweichmanövers angenommen werden (vgl. Aktionstension). Bei entsprechend gesteigener Intensität des Lenkverhaltens des jeweiligen CA, kann dies als Intention zum Wechsel des Automationsgrades interpretiert werden.

Tabelle 5-3: Arbitrierungsmatrix mit vier Arbitrierungsstrategien für unterschiedlich starke Kontrollkonflikte zwischen Fahrer und Automation beim Durchfahren einer Weggabelung. Das Werkzeug ist in MS-Excel umgesetzt

Automation	Manöver: Abbiegen rechts	Manöver: Ausweichen rechts
Fahrer	Handlung: lenken rechts	Handlung: lenken stark rechts
Manöver: Abbiegen links	Systemverhalten: Automationsgrad: Beibehalten Manöver: Abbiegen links Handlung: lenken links <i>Interaktion:</i> Rückmeldung über die Beibehaltung des aktuellen Automationsgrades und Änderung des Automationsmanövers, ändere die Richtung des Lenkmomentes der Automation	Systemverhalten: Automationsgrad: Transition zu „Vollautomatisiert“, nach Entspannung zu „Hochautomatisiert“ Manöver: Ausweichen nach rechts Handlung: lenken stark rechts <i>Interaktion:</i> Rückmeldung über die Transition zum neu gewählten Automationsgrad und Manöver, setze das Aufbringen des Lenkmomentes der Automation fort, Kopple den Fahrer von der Fahrzeugkontrolle ab, nach Entspannung kopple den Fahrer ein
Manöver: Ausweichen links	Systemverhalten: Automationsgrad: Transition zu „Manuell“, nach Entspannung zu „Assistiert“ Manöver: Ausweichen links Handlung: lenken stark links <i>Interaktion:</i> Rückmeldung über die Transition zum neu gewählten Automationsgrad und Manöver, kein Aufbringen des Lenkmomentes der Automation, Kopple die Automation von der Fahrzeugkontrolle ab, nach Entspannung kopple die Automation ein	Systemverhalten: Automationsgrad: Transition zu „Manuell“, nach Entspannung zu „Assistiert“ Manöver: Ausweichen links Handlung: lenken stark links <i>Interaktion:</i> Rückmeldung über die Transition zum neu gewählten Automationsgrad und Manöver, kein Aufbringen des Lenkmomentes der Automation, Kopple die Automation von der Fahrzeugkontrolle ab, nach Entspannung kopple die Automation ein
Handlung: lenken links		
Handlung: lenken stark links		

In der Arbitrierungsmatrix können die Handlungen und die beabsichtigten Manöver des Fahrers (waagrecht) mit denen der Automation (senkrecht) gekreuzt werden. Dies ermöglicht eine systematische Analyse und Gestaltung dessen, was direkt auf den Stellteilen bzw. Displays zwischen dem Fahrer und der Automation passieren soll. Jede Matrixzelle enthält die spezielle Ausprägung der Arbitrierungsstrategie und ist in zwei Abschnitte unterteilt. Im ersten Abschnitt „Systemverhalten“ wird das gewünschte Gesamtsystemverhalten beschrieben. Dies kann auf den Ebenen der Automationsgrade, der Manöver und der Handlung geschehen. Im zweiten Abschnitt „Interaktion“ wird die Fahrer/Automation-Interaktion beschrieben. Obwohl hier zwischen Verhalten und Interaktion unterschieden wird, ist dies nur der Nutzungsbequemlichkeit der Arbitrierungsmatrix geschuldet. Regel: *Jedes Verhalten eines CA sind Manipulationen, jede Manipulation beeinflusst das Verhalten des CS* (vgl. Abschnitt 3.3).

5.5 Syntaktisches Storyboard

Während und nachdem Arbitrierungsstrategien mittels aller vorher eingeführten Methoden und Werkzeuge erarbeitet worden sind, kann das „Syntaktische Storyboard“ erzeugt werden. Es ist eine integrierte Notationsmethode für das Verhalten einzelner CAs, inklusive Arbitrer, und für die Interaktion zwischen den CAs. Syntaktisches Storyboard beschreibt den Interaktions- und Verhaltensfluss der Agenten in Bezug auf eine bestimmte operationalisierte Aktions-tension. Es ist eine diskutierbare und dokumentierbare Vorstufe des später implementierbaren und evaluierbaren Arbitrierungsmodells (vgl. auch Abschnitt 4.4.3 und Abbildung 4-13).

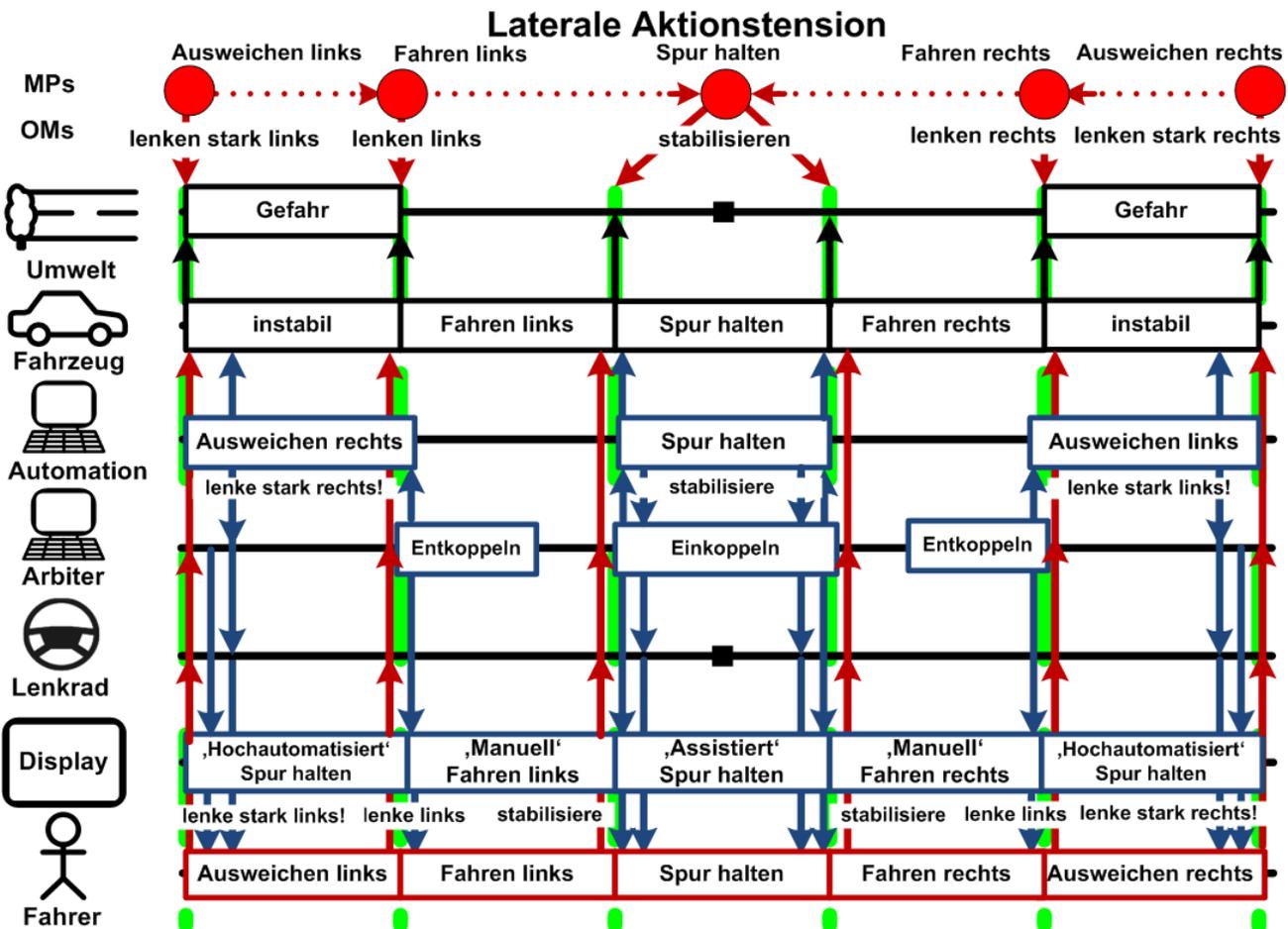


Abbildung 5-2: Syntaktisches Storyboard für den Konflikt an einer Weggabelung beschreibt den Interaktions- und Verhaltensfluss der Agenten des CS. Das Werkzeug ist in MS-Visio umgesetzt

Syntaktisches Storyboard ist von UML-Sequenzdiagrammen (vgl. Abschnitt 2.2.3) abgeleitet. Der Unterschied zu Sequenzdiagrammen ist, dass die Interaktions- und Verhaltensflüsse der notierten Arbitrierungsstrategien nicht (immer) dem Zeitverlauf, sondern der Entwicklung der Aktionstension folgen. Dementsprechend sind Sprünge vorwärts oder rückwärts entlang der ansteigenden oder abklingenden Anspannungen der Aktionstensionen vorgesehen und möglich. Die Abbildung 5-2 zeigt das Syntaktische Storyboard mit dem eingetragenen Beispiel des Kontrollkonfliktes an einer Weggabelung (vgl. Abbildung 3-1 und Abbildung 4-5).

Im oberen Bereich des Syntaktischen Storyboards ist die verwendete „Laterale Aktionstension“ mit ihren operationalisierten MPs und dazu gehörigen OMs abgebildet. Im linken Bereich sind die im CS vorhandenen Agenten als Symbole dargestellt. Rechts von den Agentensymbolen befindet sich der Interaktions- und Verhaltensfluss für den Kontrollkonflikt zwischen dem Fahrer (rot) und der Automation (blau). Die Interaktionssignale sind als Pfeile zwischen den Lebenslinien der Agenten und die aktiven MPs der Agenten sind die Rechtecke auf den Linien. Die Konfliktanalyse, die Arbitrierungsmatrix und die Lösungsexploration für diesen Konflikt wurden entsprechend in der Tabelle 5-1, der Tabelle 5-3 und der Abbildung 5-1 gezeigt.

Das Storyboard kann von links nach rechts für alle Konflikte „Fahrer fährt links und Automation rechts“ gelesen werden. Für den Konflikt „Fahrer fährt rechts und Automation links“ kann das Storyboard von rechts nach links gelesen werden. Als erstes tritt der Konflikt ein, indem der Fahrer das MP „Ausweichen links/rechts“ mit der OM „lenke stark links!/rechts!“ und die Automation „Ausweichen rechts/links“ mit „lenke stark rechts!/links!“ annehmen können. Das Fahrzeug wäre „instabil“ und die Umwelt in „Gefahr“. Das wird an den Arbiter gemeldet, der noch den aktuellen Automationsgrad „Hochautomatisiert“ und das letzte Manöver „Spur halten“ auf dem Display anzeigt. An der nächsten Transition der Aktionstension (Trigger-Ereignis) wird die Automation durch den Arbiter entkoppelt und der Fahrer lenkt allein nach links/rechts. Das Fahrzeug fährt nach links/rechts, der Arbiter zeigt auf dem Display den neuen Automationsgrad „Manuell“ und das aktuelle Manöver „Fahren links/rechts“. Am nächsten Trigger wird die Automation wieder eingekoppelt und hilft dem Fahrer, das Fahrzeug zu stabilisieren. Das Fahrzeug hält die Spur, der Arbiter zeigt auf dem Display den Automationsgrad „Assistiert“ und das Manöver „Spur halten“. Mit dieser symmetrischen Arbitrierungsstrategie kann der Kontrollkonflikt in jeder Intensität gelöst werden.

Regel: Die Nachrichten der Arbitrierungssignale im Storyboard sollten möglichst umgangssprachlich formuliert werden. Aus Erfahrungen mit dem Einsatz des Storyboards hilft dies, sich auf das Wesentliche der Interaktion zu fokussieren, wichtige Stichwörter herauszuarbeiten und dem Konzept der konsistenten Interaktionsgestaltung zu folgen (vgl. Abschnitt 2.4).

Mittels des Syntaktischen Storyboards können bereits (vor-)entwickelte Arbitrierungsstrategien diskutiert, notiert und dokumentiert werden. Außerdem bildet das Storyboard die Grundlage für die operative Umsetzung der Prototypen im Arbitrierungswerkzeug.

5.6 Arbitrierungswerkzeug

Wenn die Syntaktischen Storyboards mit Arbitrierungsstrategien für alle Konflikte vorliegen, können sie in das Arbitrierungswerkzeug übertragen werden. Das Werkzeug stellt eine Im-

plementierung der Schnittstelle zur gestaltungskompatiblen Systemarchitektur dar (vgl. Abschnitt 4.4.3 und Abbildung 4-13). In der Abbildung 5-3 ist das Werkzeug mit dem eingetragenen Beispiel des Kontrollkonfliktes an einer Weggabelung (vgl. Abbildung 3-1) gezeigt. Das Werkzeug ist in der objektorientierten Programmiersprache C++ mittels der plattformunabhängigen GUI-Bibliothek FLTK^[FAT12] umgesetzt. Das mit dem Werkzeug erstellte Arbitrierungsmodell wird in der textbasierten hierarchischen Sprache XML^[W313b] gespeichert.

Die Idee des Werkzeuges ist, mit möglichst wenig Aufwand, die vorliegenden Syntaktischen Storyboards in die sowohl für einen Computer als auch für den Systemingenieur lesbare und handhabbare Form zu überführen. Deswegen besitzt das Werkzeug eine Geometrie, die nahezu identisch zu der eines Syntaktischen Storyboards ist. Im oberen Bereich können Trigger-Ereignisse der Aktionstensionen und andere relevante Systemereignisse platziert werden. Dies geschieht mittels eines dynamischen Hinzufügen, Entfernen, Modifizieren etc. von Bedienschnittflächen. Die Trigger sind als Knöpfe implementiert, hinter denen Wertebereiche auf der Aktionstension zwecks Prototypgestaltung und -tests frei parametrisiert werden können.

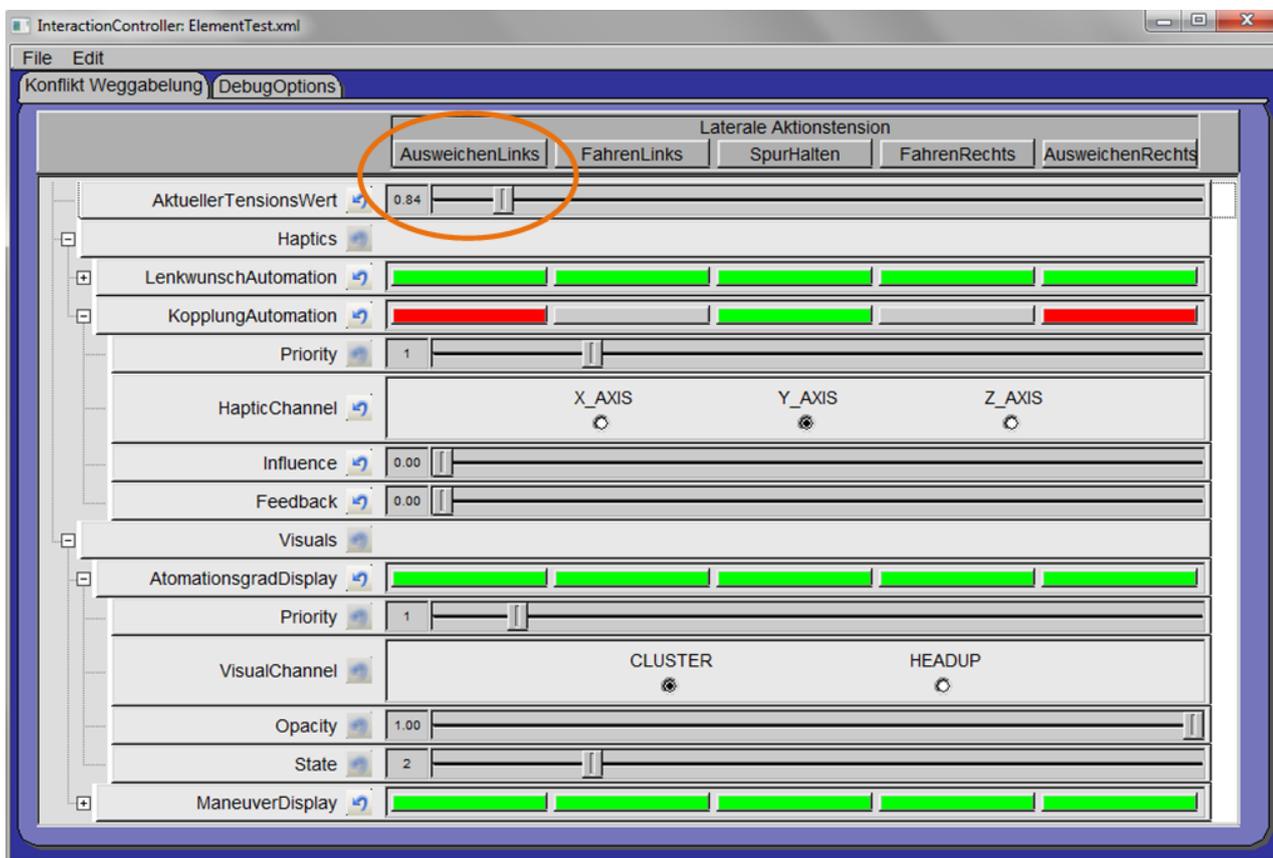


Abbildung 5-3: Arbitrierungswerkzeug für den Konflikt an einer Weggabelung implementiert die Arbitrierungsstrategie und -signale. Das Werkzeug ist mittels C++ und FLTK-Bibliothek umgesetzt

Links von den Aktionstensionen können beliebige Arbitrierungssignale, wie „KopplungAutomation“ oder „AutomationsgradDisplay“, hinzugefügt werden, die auf beliebige Hierarchien von multimodalen Kanäle (Haptics, Visuals, Acoustics) verteilt werden können. Arbitrierungssignale können Parameter besitzen, wie „Priority“ oder „HapticChannel“, die während der operativen Gestaltung verändert werden können. Der senkrechte Ereignisfluss und der waagerechte Arbitrierungssignalfloss sind durch als Knöpfe ausgeführte Farbflächen in Rot (Kanal abschalten), grün (einschalten) und grau (ignorieren) mittels ternärer Logik verknüpft (vgl.

Abbildung 4-13). Neben den Arbitrierungssignalen können beliebige Schieberegler, Knöpfe und andere GUI-Elemente für eine Parametereingabe oder -ausgabe hinzugefügt werden. Unterschiedliche Arbitrierungsstrategien können durch das Hinzufügen zusätzlicher „Reiter“, wie „DebugOptions“, im Werkzeug und im Arbitrierungsmodell berücksichtigt werden.

In dem gezeigten Beispiel ist das Semantische Storyboard aus der Abbildung 5-2 implementiert. Gezeigt ist der momentane Zustand des Prototyps beim Ausweichen nach links (Anspannung 0,84, MP „AusweichenLinks“ [oranger Kreis]). Die Automation möchte zwar lenken (LenkwunschAutomation ist „grün“), wird aber gerade ausgekoppelt (KopplungAutomation ist „rot“). Die Clusterdisplays (Automationsgrad- und ManeuverDisplay) sind aktiv (grün) und zeigen entsprechende Nachrichten an (State 2 bedeutet z.B. die Schrift „Hochautomatisiert“ im Clusterdisplay an der entsprechenden Stelle).

Das Arbitrierungswerkzeug stellt einen „weichen“ Übergang von der Diskussion einer Gestaltung mittels Syntaktischen Storyboards zur operativen prototypischen Umsetzung der Gestaltung mittels der Software „Interaktionskontroller“. In den dahinterliegenden Parameter-GUIs des Arbitrierungswerkzeugs werden beispielsweise Variablennamen verwendet, die durch das Parsen des XML-Codes der Arbitrierungsmodelle und der entsprechenden C++-Headerdateien direkt im Code des Interaktionskontrollers verwendet werden.

5.7 Interaktionskontroller

Im Interaktionskontroller wurde die gestaltungskompatible Architektur von CS für den Bereich hochautomatisierter Fahrzeugführung abgebildet (vgl. Abschnitt 4.4.2). Die mit dem Arbitrierungswerkzeug erstellten Modelle können im Interaktionskontroller ausgeführt und in einer Simulationsumgebung mit der entsprechenden Interaktionshardware und Systemnutzern als vollständiges gestaltungskompatibles CS aufgebaut, zur Laufzeit parametrisiert und getestet werden. Die Abbildung 5-4 zeigt den Aufbau des Interaktionskontrollers. Die Farben der Module entsprechen den Ebenen der bereits eingeführten gestaltungskompatiblen CS-Architektur^[Ket13] (vgl. Abbildung 4-11).

„Ebene I: Agentenrepräsentationen und Aktionstensionen“ (blau) der gestaltungskompatiblen Architektur kognitiver Systeme ist in dem Interaktionskontroller als eine Menge der Agentenrepräsentationen implementiert. Jede Agentenrepräsentation ist dafür verantwortlich, Informationen, wie Agentenmanipulationen und MPs, über „ihren Agenten“ zu sammeln, auf den R_{norm} (vgl. (1)) zu normieren und dem Wechselwirkungsmanagementmodul zur Verfügung zu stellen. In den Agentenrepräsentationen, wie Fahrer-, Automation- und Situationsrepräsentation, werden auch die Anspannungen der Agenten bezogen aufeinander und auf die entspannten MP der Aktionstensionen (vgl. Abschnitt 4.1) berechnet.

„Ebene II: Arbitrierung“ (magenta) besteht aus einer Menge von Arbitern, die für die Arbitrierung langfristiger, kurzfristiger und unmittelbarer Effekte verantwortlich sind (vgl. Abschnitt 4.1.2). Bei der hochautomatisierten Fahrzeugführung können die langfristigen Effekte Konflikte bzw. Kooperationen auf der Ebene der Automationslevel^[GaA12], der Kooperationsmodi^[Hoc01] und der Transitionen^[ScF08] dazwischen sein. Kurzfristige Effekte können Konflikte bzw. Kooperationen auf der Ebene der Manöver, Trajektorien der Bewegung des Fahrzeuges

oder kurzfristige Aushandlungen der Querablage oder der Fahrzeuggeschwindigkeit sein. Dies können auch kurzfristige Prozesse sein, wie die Zuordnung, Umverteilung primärer und sekundärer Fahraufgaben während der Fahrzeugführung. Die Arbitrierung unmittelbarer Effekte kann die Behandlung von Konflikten und Kooperationen bei sehr schnellen Operationen beinhalten, wie die Ein- bzw. Entkopplung der Agenten in bzw. von der Fahrzeugkontrolle.

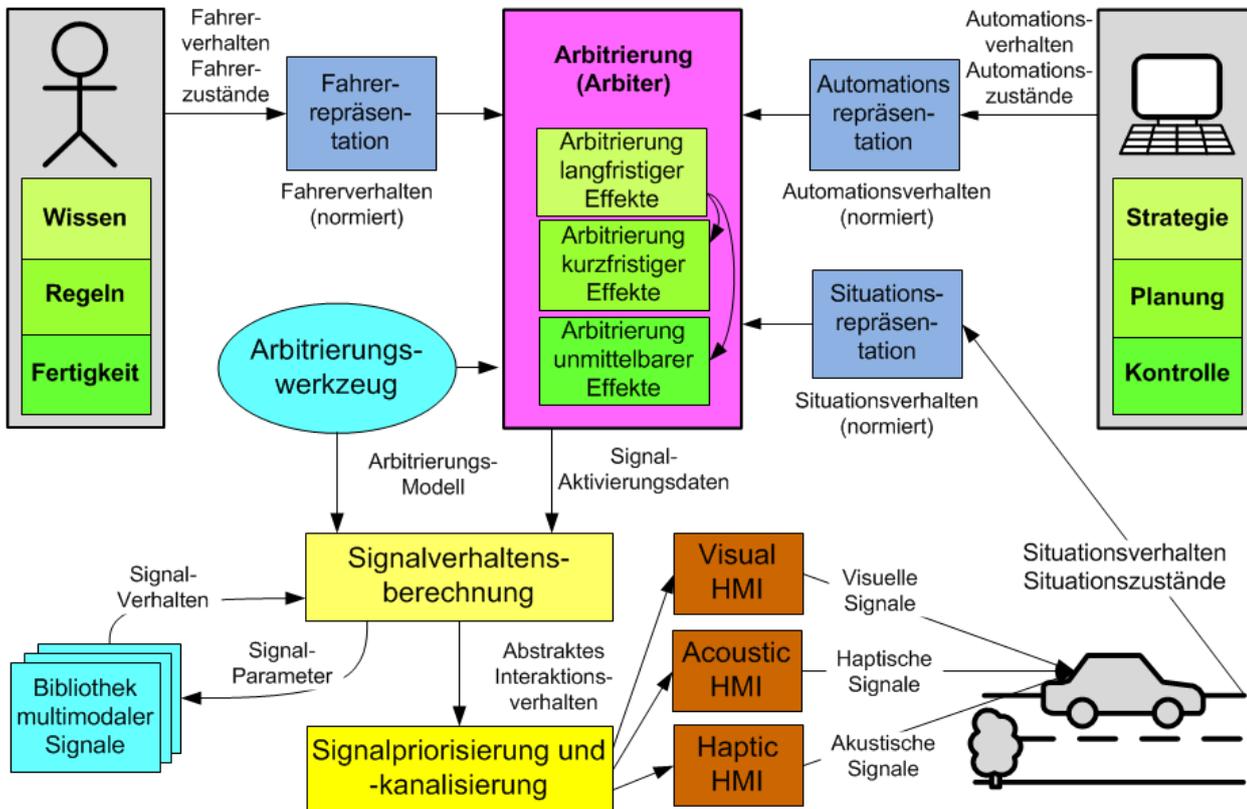


Abbildung 5-4: Aufbau des Interaktionskontrollers bildet die gestaltungskompatible CS-Architektur für den Bereich hochautomatisierter Fahrzeugführung nach. Sie ist als Software-Framework „Interaktionskontroller“ mittels C++ und des Software-Modellierungswerkzeuges BOUML 4.2.3 Patch 6^[Pas12] umgesetzt

Mittels der normierten Verhaltensdaten der Agenten in den Agentenrepräsentationen und der im Arbitrierungsmodell kodierten Arbitrierungsstrategien können Arbitrierer Signalaktivierungsdaten berechnen. Sie entscheiden, ob auf den Kommunikationskanälen zwischen den Agenten Signale ausgetauscht werden sollen und welche es sind. Damit können Arbitrierer Verhandlungen zwischen den Agenten moderieren, falls nötig, eindeutige Entscheidungen treffen und deren Ausführung unterstützen. Bei der impliziten Arbitrierung (vgl. Abschnitt 4.3.2) arbeiten die Arbitrierer als Beobachter und bei der expliziten Arbitrierung übernehmen sie eine aktive Rolle, da sie vollständig den Kommunikationsfluss zwischen den Agenten kontrollieren.

„Ebene III: Arbitrierungsmodell“ (cyan) enthält neben dem im Arbitrierungswerkzeug vorliegenden und vom Systemingenieur implementierten Arbitrierungsmodell eine objektorientierte Bibliothek mit der vollständigen Sammlung aller multimodaler Arbitrierungssignale. Dies sind genau die Signale, die im Werkzeug „Semantisches Mapping“ erarbeitet wurden (vgl. Abschnitt 5.3). D.h. alle visuellen, akustischen, haptischen etc. Signale liegen hier und können bei Bedarf ihr Signalverhalten an die Signalverhaltensberechnung als Antwort auf die von ihr zugeschickten Parametersätze liefern.

„Ebene IV: Signalgestaltung und -kanalisierung“ (gelb) besteht aus zwei hierarchisch angeordneten Modulen. Oben in der Hierarchie steht die Signalverhaltensberechnung. Dies ist die zentrale Stelle, wo aus dem Arbitrierungsmodell, dem Signalverhalten aus der Bibliothek und Signalaktivierungsdaten der Arbitrierer das komplette Interaktionsverhalten berechnet wird. Da einige Arbitrierungssignale auf gleichen multimodalen Kanälen und Unterkanälen (vgl. Abschnitt 3.2) laufen können, müssen sie anschließend von der Signalpriorisierung und -kanalisierung eindeutig determiniert und auf die richtige HMI-Hardware verteilt werden.

„Ebene V: Interaktionshardware“ (braun) besteht aus drei Modulen: „Visual HMI“, „Acoustic HMI“ und „Haptic HMI“, wobei auch andere Arten von Modalitäten hinzugefügt werden können. Diese Module sorgen dafür, dass das determinierte und noch in einer abstrakten Form gelieferte Interaktionsverhalten auf den richtigen Stellteilen, Displays etc. ankommt. Solche Aufteilung in abstraktes und konkretes Interaktionsverhalten ist insofern hilfreich, dass die Interaktionshardware (HMI) ausgetauscht werden kann, ohne das Arbitrierungsmodell zu verändern. Bei einem Wechsel der Versuchsträgers oder der Simulationsumgebung müssen dann nur die Parameter der Arbitrierungssignale im Modell geringfügig angepasst werden.

Insgesamt stellt der Interaktionskontroller ein nützliches und abschließendes Werkzeug des osGekoS. Bei einer CS-Gestaltung aus dem Bereich hochautomatisierter Fahrzeugführung gestattet er, die theoretische Konzeptphase der Gestaltung nahtlos in die Praxis, den Test und in einen erlebbaren (fahrbaren) Prototypen des gestalteten CS zu überführen.

5.8 Zusammenfassung

In diesem Kapitel wurden die Regeln, die Werkzeuge von osGekoS und durch das Beispiel eines Kontrollkonfliktes an einer Weggabelung auch der Prozess der Gestaltung von hochautomatisierter Fahrzeugführung als Spezialfall der Gestaltung kognitiver Systeme vorgestellt.

Die operative strukturfokussierte Gestaltung kann mit einer strukturellen Dekomposition mittels des MPS-Konzeptes beginnen. Die funktionale Dekomposition sollte mittels bekannter Methoden des CSE ebenfalls stattfinden. Nach der strukturellen und funktionalen Dekomposition des Systems in Agenten und Aktionstensionen sollten potenzielle und akute Konflikte innerhalb des Systems aufgedeckt und grobe Ideen für die Lösungen dieser Konflikte generiert werden. Das kann durch das *Konfliktanalysewerkzeug* unterstützt werden. Dabei sollten Nutzungsszenarien definiert und Aktionstensionen operationalisiert werden.

Detaillierte Erarbeitung von Arbitrierungsstrategien kann durch eine *Lösungsexploration* begleitend unterstützt werden. Das Ziel dabei ist, immer präziser die Arbitrierungsstrategien zur Behandlung aufgedeckter Konflikte zu erarbeiten. Bis zu dieser Ebene kann noch ohne Gestaltungsheuristiken gearbeitet werden. Danach beginnt man nach und nach Gestaltungsheuristiken in Bezug auf die Semantik und die Syntax der Interaktionselemente in den Gestaltungsprozess mit aufzunehmen.

Semantisches Mapping vorausgesetzt, können Detaillösungen für entdeckte Konflikte in Form von Arbitrierungsstrategien erarbeitet werden. Dieser Prozess kann durch das Werkzeug *Arbitrierungsmatrix* unterstützt werden. Die besten Detaillösungen lassen sich anschließend in *Syntaktische Storyboards* integrieren. Wenn die *Syntaktischen Storyboards* vorliegen, können

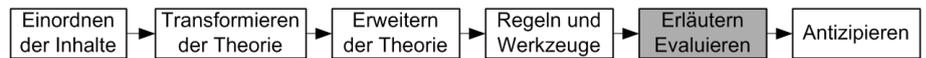
diese in das *Arbitrierungswerkzeug* zwecks Erstellung eines Arbitrierungsmodells übertragen werden. Das Arbitrierungsmodell kann im *Interaktionskontroller* ausgeführt werden. Zusammen mit dem auf diese Weise entstandenen Interaktionsprototyp in einer entsprechenden Simulationsumgebung und mit Versuchspersonen kann das komplette gestaltungskompatible CS in einer Usability Studie evaluiert und in einem Usability Experiment validiert werden.

Die wichtigsten Regeln wurden in der Tabelle 5-4 zusammengefasst. Diese Regeln sind in der Tabelle mit den dazugehörigen Werkzeugen bzw. Methoden sowie den Schritten des Gestaltungsprozesses verknüpft.

Der Mehrwert des eben beschriebenen Gestaltungsprozesses, Regeln und Werkzeuge wird im nächsten Kapitel anhand von zwei umfassenden Beispielen für neu entwickelte Fahrerassistenzsysteme weiter erläutert. Ergebnisse der durchgeführten Nutzerbewertungen dieser Assistenzsysteme werden zur Evaluation von osGekoS mitverwendet.

Tabelle 5-4: Zusammenfassung des osGekoS Gestaltungsprozesses, dazugehöriger Werkzeuge und Regeln

Gestaltungsschritt	Einsatz Gestaltungsheuristiken	Werkzeug/Methode	Regeln
Systemdekomposition	Ohne Gestaltungsheuristiken	Anwendung der Theorie	
Konfliktanalyse		Konfliktanalysewerkzeug	1. Jedes handelnde und Aufgaben ausführende Element eines kognitiven Systems ist ein Agent bzw. Manipulationspotenzial. 2. Konflikte bzw. Anspannungen müssen innerhalb des Systems aufgedeckt, vermieden oder aufgelöst werden.
Lösungsexploration		Lösungsexploration Arbitrierungsmatrix	3. Das Systemverhalten ergibt sich aus der Veränderung der Systemstruktur. 4. Jedes Verhalten eines Agenten sind Manipulationen, jede Manipulation beeinflusst das Verhalten des kognitiven Systems.
Semantisches Mapping	Mit Gestaltungsheuristiken	Semantisches Mapping	5. Die Konsistenz und Eindeutigkeit in der Zuordnung der Bedeutungen zu den jeweiligen Arbitrierungssignalen ist gefordert. 6. Komplexe Signale sollten aus Kombinationen einfacher Signale erstellt werden, ohne ihre eigenen Bedeutungen zu verlieren.
Interaktionsgestaltung		Syntaktisches Storyboard	7. Die Nachrichten der Interaktionssignale im Storyboard sollten möglichst umgangssprachlich formuliert werden.
Implementierung		Arbitrierungswerkzeug Interaktionskontroller	



6 Erläuterung und Evaluation durch Anwendung

In diesem Kapitel werden die osGekoS-Theorie, -Regeln und -Werkzeuge anhand von zwei damit entwickelten Fahrerassistenzsystemen erläutert und evaluiert. Die Erläuterungen werden schrittweise mit dem Ziel stattfinden, den im Kapitel 5 eingeführten Gestaltungsprozess wiedererkennbar zu machen. Das Ziel der Evaluation von osGekoS in dieser Arbeit ist nicht der vollständige Nachweis der Validität des erarbeiteten Gestaltungsrahmens, sondern seine prinzipielle Anwendbarkeit für die Gestaltung gebrauchstauglicher kognitiver Systeme mit hoher Akzeptanz der Nutzer im Bereich hochautomatisierter Fahrzeugsysteme.

- Das erste Assistenzsystem (Beispielsystem 1) sollte mit einer zwischen einem Fahrer und einer Automation geteilten Längs- und Querführung ausgestattet sein.
- Das zweite Assistenzsystem (Beispielsystem 2) sollte eine Kooperation zwischen zwei Verkehrsteilnehmern bei Fahrstreifenwechseln anregen und unterstützen.

Die Assistenzsysteme sind in dieser speziellen Konstellation ausgewählt, weil sie exemplarisch den Raum primärer Fahraufgaben bei geteilter Fahrzeugkontrolle abdecken. Dazu gehören im Beispielsystem 1 sowohl die Längs- und Querführung des Fahrzeuges als auch die Koordination mehrerer Fahrzeuge untereinander im Beispielsystem 2.

6.1 Beispielsystem 1: Annähern, Bremsen und Ausweichen

Dieses Fahrerassistenzsystem wurde im Verlauf des DFG-Projektes „H-Mode“ entwickelt^{[KeH09][Flh00]}, in dem es um die Exploration einer Designmetapher für hochautomatisierte Fahrzeuge ging. Das natürliche Vorbild einer Reiter/Pferd-Interaktion wurde in diesem Projekt als Leitmetapher für die Systemgestaltung verwendet.

6.1.1 Anforderungen

Für Nutzungsfälle Annähern, Bremsen und Ausweichen eines hochautomatisierten Fahrzeuges auf einer Autobahn soll ein integriertes und vom Nutzer akzeptiertes System gestaltet werden. Die Fahrer/Automation-Interaktion soll mittels haptischer Signale geschehen. Zwei unterschiedliche Automationsgrade („Assistiert“ und „Hochautomatisiert“) sind ohne Transitionen dazwischen zu berücksichtigen. Im Automationsgrad „Assistiert“ fährt der Fahrer aktiv, die Automation gibt ihm eine schwache Unterstützung bei Spur-, und Geschwindigkeitshaltung und durch diskrete haptische Hinweise. Im Automationsgrad „Hochautomatisiert“ fährt die Automation aktiv, der Fahrer bleibt durch seine Unterstützung der Spur-, und Geschwindigkeitshaltung und durch diskrete haptische Befehle in die Fahrzeugführung involviert.

6.1.2 Szenario und Systemdekomposition

Das Hauptszenario für die Systemgestaltung ist eine dreispurige gerade Autobahn (Abbildung 6-1 links). Auf der mittleren Fahrspur fährt das hochautomatisierte Eigenfahrzeug. Seine Anfangsgeschwindigkeit beträgt 100 km/h. Vor dem Eigenfahrzeug erscheint plötzlich in unterschiedlichen Abständen ein Fremdfahrzeug, das eine konstante Geschwindigkeit von 50 km/h

fährt. Das Eigenfahrzeug muss entsprechend reagieren und eine kollisionsfreie und für den Fahrer akzeptable Fahrt bewerkstelligen. V1 ist ein hochautomatisiertes Fahrzeug mit drei Agenten: d ist der Fahrer, c ist die Automation, v ist das Fahrzeug. Das erscheinende Fremdfahrzeug ist mit der nahen Umwelt als e zusammengeführt (Abbildung 6-1 rechts).

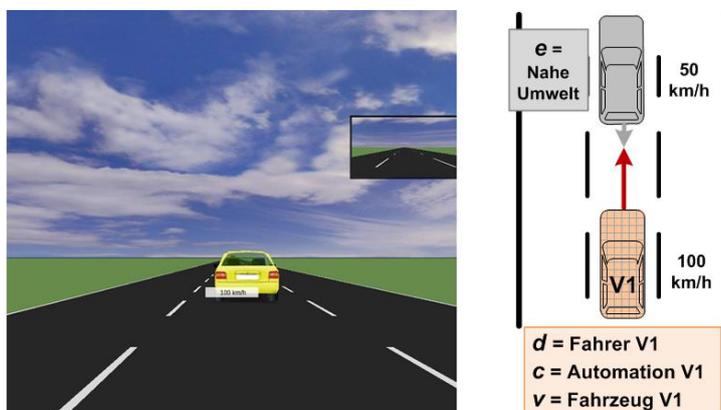


Abbildung 6-1: Simulationsansicht und die Dekomposition des Szenarios für das Beispielsystem 1

6.1.3 Konfliktanalyse und Semantisches Mapping

Für dieses System wurde angenommen, dass die Automation konfliktfrei funktioniert. Der Auszug der fahrer- und der umgebungsbezogenen Konfliktanalyse des gegebenen Anfangszustandes ist in der Tabelle 6-1 dargestellt.¹⁸

Tabelle 6-1: Auszug aus Konfliktanalysewerkzeug für das Beispielsystem 1

ANFANGS-SYSTEM-ZUSTAND		KONFLIKTBESCHREIBUNG					LÖSUNG
		Agentenbezogene Konflikte				Umgebungs-konflikte	
		Bewusst-sein	Intention Entscheidung	Koordination	Handlung		
Fahrer V1	Fahre, meide Kollisionen	Abgelenkt und/oder reagiert nicht	Entscheidet sich anders als Automation	Erkennt nicht, was Automation vorhat	lenkt oder bremst zu schwach	langsames Fremdfahrzeug taucht plötzlich vorne auf der gleichen Spur auf Konfliktvermeidung: Mitteilung der Automationsintention an den Fahrer rechtzeitig vor der Kollision Konfliktlösung: Übersteuern der Fahrereingabe	
Automation V1	Fahre, meide Kollisionen						

Das Semantische Mapping für haptische Arbitrierungssignale wurde in mehreren kurzen Vorstudien und mittels Analyse bereits vorliegender Interaktionsprototypen durchgeführt. Das Ergebnis des Semantischen Mapping wurde in der Tabelle 5-2 bereits vorgestellt.

6.1.4 Lösungsexploration

Die Systemdekomposition kann mit der Aufstellung eines Modells für die Untersuchung der Entscheidungskonvergenz fortgesetzt werden. Für das vorliegende Beispielsystem 1 aus Ab-

¹⁸ Weil die Konfliktanalysewerkzeug und -analyse zur Zeit der Systemgestaltung in der aktuellen Form noch nicht zur Verfügung standen, wurden sie für diese Arbeit aus begleitenden Dokumentationsunterlagen rekonstruiert.

bildung 6-1 lassen sich mit den Konzepten aus Kapitel 3 folgende Zustandsübergangsgraphen aufstellen (Abbildung 6-2):

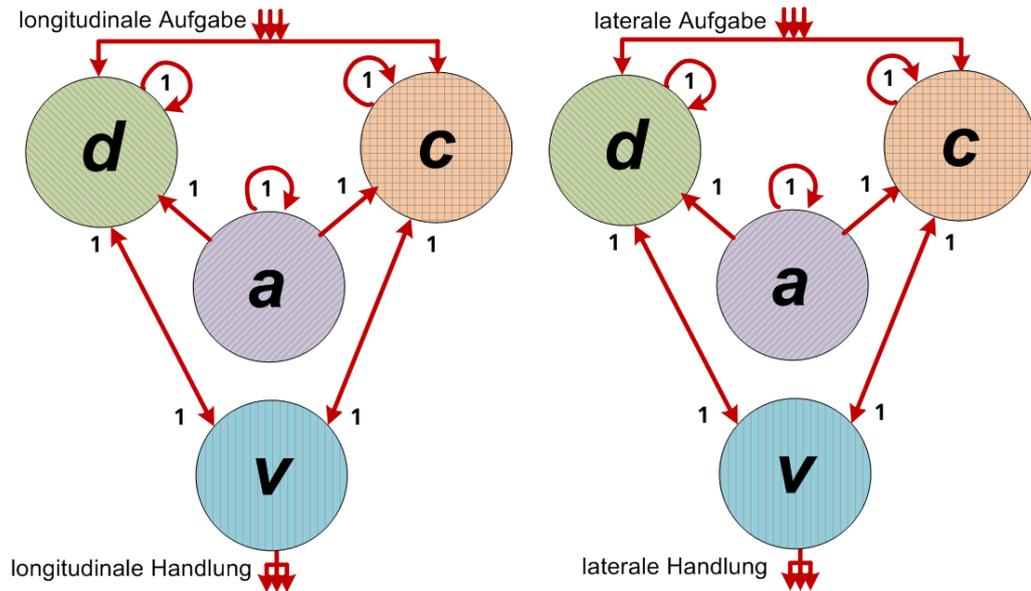


Abbildung 6-2: Zustandsübergangsgraphen des Szenarios für das Beispielsystem 1

Die Abbildung zeigt jeweils für die laterale und die longitudinale Handlung innerhalb des CS die Manipulationen zwischen den einzelnen Agenten im hochautomatisierten Fahrzeug des Beispielsystems 1. Die Konfiguration eines solchen Fahrzeugs kann für die Quer- und Längsführung wie folgt aussehen: Der Fahrer (d), die Automation (c) und der Arbitrer (a) sind CAs mit Eigenmanipulationen ($k_n^{dd} = k_n^{cc} = k_n^{aa} = 1$) (vgl. Abschnitt 4.2.3). Das Fahrzeug (v) ist ein NA ohne eine EM ($k_n^{vv} = 0$). Sowohl der Fahrer als auch die Automation können sich gegenseitig nicht ($k_n^{dc} = k_n^{cd} = 0$), dafür aber das Fahrzeug zwecks Fahrzeugsteuerung manipulieren ($k_n^{dv} = k_n^{cv}$) und empfangen ein Feedback vom Fahrzeug ($k_n^{vd} = k_n^{vc} = 1$), z.B. spüren bzw. messen, wie es lenkt oder bremst. Fahrer, Automation und Fahrzeug können den Arbitrer nicht direkt beeinflussen ($k_n^{da} = k_n^{ca} = k_n^{va} = 0$), können aber mit Ausnahme von Fahrzeug ($k_n^{av} = 0$) vom Arbitrer beeinflusst werden ($k_n^{ad} = k_n^{ac} = 1$), z.B. in dem der Arbitrer dem Fahrer Hinweise und der Automation Anweisungen gibt. Die aktiven MPs und daher die OMs des Fahrers, der Automation, des Arbitrers und des Fahrzeuges zum Ereignis n können als Vektor \vec{a}_n angegeben und nach (11) auf die Entscheidungskonvergenz untersucht werden. Mit dazugehöriger Kopplungsmatrix nach (5) zum Ereignis n ergibt sich bei allen Annahmen (20).

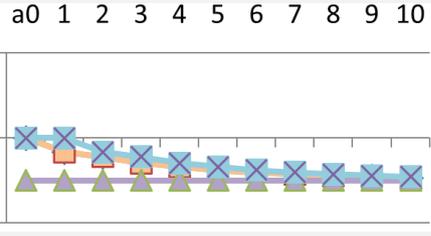
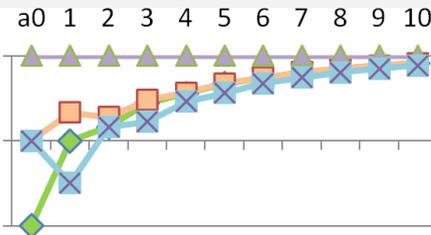
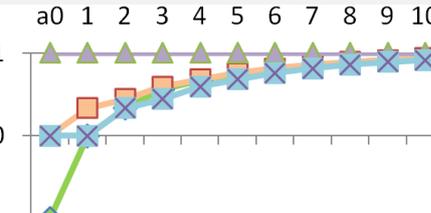
$$K_n = \begin{pmatrix} k_n^{dd} & k_n^{cd} & k_n^{ad} & k_n^{vd} \\ k_n^{dc} & k_n^{cc} & k_n^{ac} & k_n^{vc} \\ k_n^{da} & k_n^{ca} & k_n^{aa} & k_n^{va} \\ k_n^{dv} & k_n^{cv} & k_n^{av} & k_n^{vv} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ k_n^{dv} & k_n^{cv} & 0 & 0 \end{pmatrix}; \quad \vec{a}_n = \begin{pmatrix} a_n^d \\ a_n^c \\ a_n^a \\ 0 \end{pmatrix} \quad (20)$$

Mit (11) lassen sich Zeitreihen für das aufgestellte Modell entwickeln. Damit kann man z.B. zeigen, wie Fahrer, Automation und danach das Fahrzeug den Empfehlungen/Anweisungen des Arbitrers in der Längs- und Querführung folgen, oder wie man durch Entkopplung des Fahrers bei kritischen Manövern mögliche Fahrzeuginstabilität vermeiden kann.

Die Tabelle 6-2 zeigt einige relevante und vom Arbitrer modifizierbare Strukturvarianten für die Querführung (oben links), die vom Autor prospektiv erwarteten Entscheidungskonvergenzen

innerhalb des Modells (oben rechts), Anfangswerte für K_n und \bar{a}_n (unten links), die entsprechenden Zeitreihen mit der MP-Aktivierung bzw. Entscheidungskonvergenz der Agenten bis zum 10. Berechnungsschritt (unten Mitte) und ihre Interpretationen (unten rechts). Die Längsführung kann aufgrund der Modellähnlichkeit als analog dazu betrachtet werden.

Tabelle 6-2: Restrukturierungsvarianten des Modells für die Querführung des Beispielsystems 1

<p>Struktur 1 (Aktivierung bei Inaktivität): Fahrer und Automation lenken nicht ($a_n^d = a_n^c = 0$) und sind in die Fahrzeugkontrolle positiv eingekoppelt ($k_n^{dv} = k_n^{cv} = 1$), der Arbitrer gibt einen Hinweis (Aktivierung) zum links Fahren (Spurwechsel) ($a_n^a = -0,5$)</p>	<p>Erwartete Entscheidungskonvergenz: Fahrer, Automation lenken nach links. Fahrzeug fährt nach links (macht einen Spurwechsel). Die Entscheidung sollte stabil bleiben.</p>																																			
<p>Anfangsparameter und Zeitreihenentwicklung</p> <table border="1" style="display: inline-table; margin-right: 10px;"> <thead> <tr> <th></th> <th colspan="4">K_n</th> <th>a_0</th> <th>rechts lenken</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>d</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>c</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>-0,5</td> <td>0</td> <td>a</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>v</td> </tr> </tbody> </table> 		K_n				a_0	rechts lenken	1	0	1	1	0	0	d	0	1	1	1	0	0	c	0	0	1	0	-0,5	0	a	1	1	0	0	0	0	v	<p>Interpretation: CS-Entscheidung konvergiert gegen „links lenken“ und folgt damit der Empfehlung des Arbiters, die z.B. durch einen dezenten haptischen Hinweis am Lenkrad dargestellt werden kann. Fahrzeug fährt nach links, wie erwartet. Die Entscheidung ist stabil. Die Arbitrierungsstrategie ist ebenso auf die Längsführung anwendbar.</p>
	K_n				a_0	rechts lenken																														
1	0	1	1	0	0	d																														
0	1	1	1	0	0	c																														
0	0	1	0	-0,5	0	a																														
1	1	0	0	0	0	v																														
<p>Struktur 2 (Konflikt beim Ausweichen): Fahrer lenkt stark links ($a_n^d = -1$), Automation lenkt nicht ($a_n^c = 0$), beide sind in die Fahrzeugkontrolle positiv eingekoppelt ($k_n^{dv} = k_n^{cv} = 1$), der Arbitrer gibt eine Anweisung zum Ausweichen nach rechts ($a_n^a = 1$)</p>	<p>Erwartete Entscheidungskonvergenz: Stabile Entscheidung sollte zustande kommen. Das Fahrzeug sollte nach rechts ausweichen.</p>																																			
<p>Anfangsparameter und Zeitreihenentwicklung</p> <table border="1" style="display: inline-table; margin-right: 10px;"> <thead> <tr> <th></th> <th colspan="4">K_n</th> <th>a_0</th> <th>rechts lenken</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>-1</td> <td>0</td> <td>d</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>c</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>a</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>v</td> </tr> </tbody> </table> 		K_n				a_0	rechts lenken	1	0	1	1	-1	0	d	0	1	1	1	0	0	c	0	0	1	0	1	0	a	1	1	0	0	0	0	v	<p>Interpretation: CS-Entscheidung konvergiert gegen das „rechts Ausweichen“ und folgt der Anweisung des Arbiters, die z.B. durch einen deutlichen, zeitlich ausgedehnten Ruck am Lenkrad dargestellt werden kann. Fahrzeug weicht nach rechts aus, wie erwartet. Die Entscheidung ist aber nur bedingt stabil, da das Fahrzeug einen starken Schlenker vom links nach rechts macht.</p>
	K_n				a_0	rechts lenken																														
1	0	1	1	-1	0	d																														
0	1	1	1	0	0	c																														
0	0	1	0	1	0	a																														
1	1	0	0	0	0	v																														
<p>Struktur 3 (Ignoranz des Fahrers): Wie Struktur 2, Fahrer wird vom Arbitrer von der Fahrzeugkontrolle ausgekoppelt ($k_n^{dv} = 0$)</p>	<p>Erwartete Entscheidungskonvergenz: Stabile Entscheidung sollte zustande kommen. Das Fahrzeug sollte nach rechts ohne Schlenker ausweichen.</p>																																			
<p>Anfangsparameter und Zeitreihenentwicklung</p> <table border="1" style="display: inline-table; margin-right: 10px;"> <thead> <tr> <th></th> <th colspan="4">K_n</th> <th>a_0</th> <th>rechts lenken</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>-1</td> <td>0</td> <td>d</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>c</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>a</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>v</td> </tr> </tbody> </table> 		K_n				a_0	rechts lenken	1	0	1	1	-1	0	d	0	1	1	1	0	0	c	0	0	1	0	1	0	a	0	1	0	0	0	0	v	<p>Interpretation: CS-Entscheidung konvergiert gegen „rechts ausweichen“ und folgt der Anweisung des Arbiters. Der bei der Struktur 2 noch vorhandene Schlenker wird durch die rechtzeitige Auskoppelung des Fahrers ausgeglichen, so dass das System stabil bleibt.</p>
	K_n				a_0	rechts lenken																														
1	0	1	1	-1	0	d																														
0	1	1	1	0	0	c																														
0	0	1	0	1	0	a																														
0	1	0	0	0	0	v																														

Die auf diese Weise explorierten Arbitrierungsstrategien und Schwellwerte für Manipulationspotenziale können im weiteren Schritt präzisiert und operationalisiert werden.

6.1.5 Operationalisierung der Aktionstension

Für die Operationalisierung der Aktionstension wurde zunächst ein deduktiver Ansatz mittels einer Literaturrecherche nach relevanten Daten und Konzepten verwendet^{[KeH12][KeH09]}. Anschließend wurde das konzeptionelle Ergebnis der Recherche in einem induktiven Ansatz mit Probanden innerhalb einer Usability Studie überprüft.

Die Aktionstension ist eine Steuerstruktur und eine dichotome Kette aus MPs und dazugehörigen Manipulationen, die in Richtung des entspannten MP zeigen (vgl. Abschnitt 4.2.2). Für das Beispielsystem ist das entspannte MP die kollisionsfreie Fahrt. Außer den gewöhnlichen Fahrzeugführungsaktionen zum Zweck der Spur- und Geschwindigkeitshaltung sind in diesem MP weder weitere Manipulationen noch Interaktion zwischen dem Fahrer und der Automation notwendig. Die zu den anderen MPs gehörenden Manipulationen sind das zusätzliche Bremsen und Lenken in unterschiedlichen Intensitäten.

Nach einer Literaturrecherche wurde entschieden, den Kehrwert der Time-To-Collision (TTC) für die grundsätzliche Beschreibung des kontinuierlichen Wertes der Aktionstension zu verwenden. TTC ist definiert als "the time required for two vehicles to collide if they continue at their present speed and on the same path"^{19[Had72]}. Die TTC ist ein kontinuierlicher Wert, der auf einer abstrakteren Ebene die Notwendigkeit zu handeln beschreibt. Wenn TTC hoch genug ist, gibt es keinen Grund zu handeln und den aktuellen Fahrzustand zu verändern. Wenn die TTC aber gegen Null läuft, ist die Notwendigkeit zu handeln und damit die Notwendigkeit für die Interaktion zwischen dem Fahrer und der Automation am höchsten. Der Kehrwert der TTC hat das umgekehrte und zu der Logik der Aktionstension kompatible Verhalten. Er läuft gegen Null, wenn kein Handlungsbedarf besteht und wird unendlich bei einer Kollision.

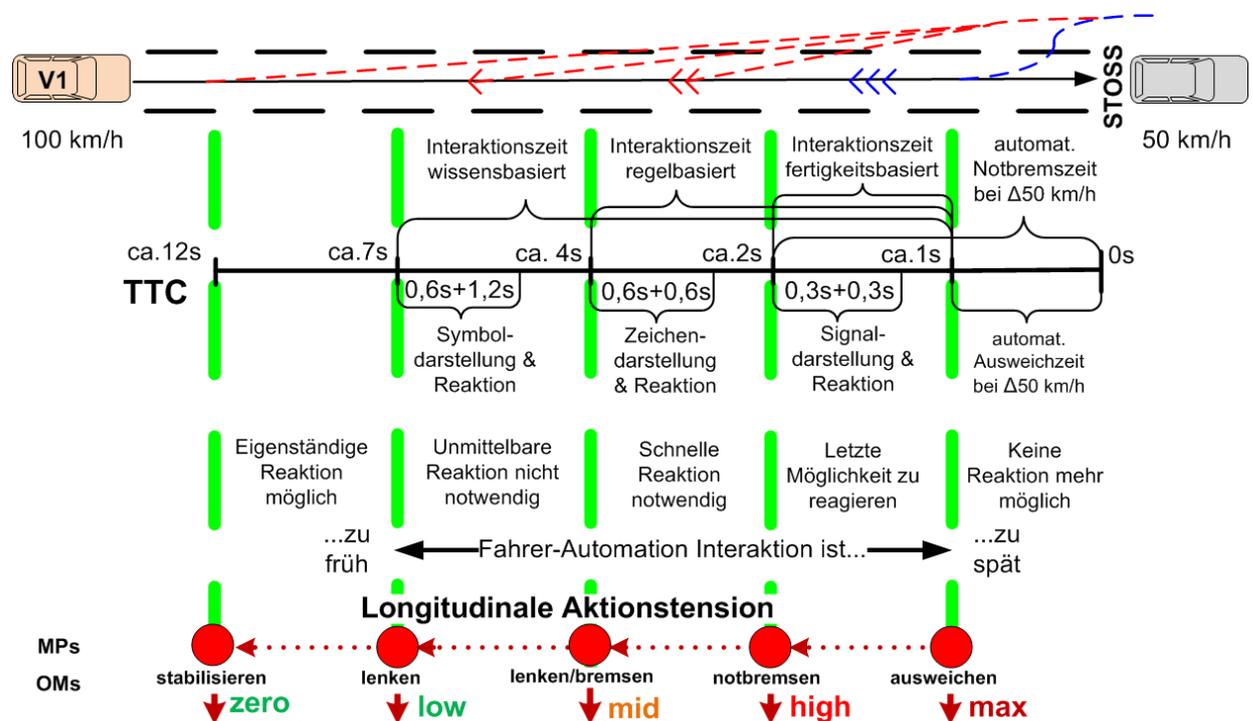


Abbildung 6-3: Operationalisierung der Aktionstension für das Beispielsystem 1

¹⁹ ... die Zeit, die von zwei Fahrzeugen für eine Kollision benötigt wird, wenn sie sich mit ihrer gegenwärtigen Geschwindigkeit auf den ursprünglichen Trajektorien bewegen. EÜ.

Der Verlauf des Kehrwertes der TTC wurde auf fünf MPs („zero“, „low“, „mid“, „high“ und „max“) aufgeteilt (Abbildung 6-3 unten). Die Transitionen zwischen den Bereichen (grüne vertikale gestrichelte Linien) sollten als Trigger-Ereignisse für die Interaktion zwischen dem Fahrer und der Automation und gleichzeitig als Trigger für die Einleitung des entsprechenden Automationsverhaltens dienen, wie das Ausführen bestimmter Manöver. Dabei war die angestrebte Manipulation immer die in Richtung des entspannten MPs „zero“. In dem entspannten MP sollte das Verhältnis zwischen der Leistung und der Aktivierung des Fahrers optimal^[YeD08] sein und jegliche Handlungsänderung der Agenten sowie die Interaktion zwischen den Agenten zu früh wäre.

Die MPs „low“, „mid“ und „high“ wurden unter Berücksichtigung des Konzeptes des Zeithorizonts der Beeinflussung des Agentenverhaltens (vgl. Abschnitt 4.1.2) definiert. Das MP „max“ definierte den Bereich, in dem keine Reaktion des Fahrers mehr möglich war und das Erreichen des entspannten MP „zero“ nur durch autonomes Handeln der Automation möglich war. Die MP-Transitionen wurden durch die vom Fahrer und von der Automation abhängigen Parameter bestimmt. Vom Fahrer kam die Berücksichtigung seiner Reaktionszeiten im aufmerksamen und unaufmerksamen Zustand^[Grn00] und abhängig von seiner Involvierung in die primäre Fahraufgabe aufgrund des aktuell verwendeten Automationslevels. Des Weiteren wurden Symbol-, Zeichen- und Signalverarbeitungszeiten und typische Zeiten bei relevanten Entscheidungen des Menschen berücksichtigt, z.B. wann die meisten Fahrer sich für einen Spurwechsel bei einer Annäherung an ein langsamerer Fahrzeug entscheiden^[FaH01]. Auf der Automationsseite waren die Reaktionszeiten der verwendeten Automation in bestimmten Situationen berücksichtigt, wie das autonome Notbremsen oder Ausweichen, und Zeiten für die Darbietung bestimmter Arbitrierungssignale, wie Empfehlungen oder Warnungen.

Die deduktiv erarbeitete Definition der longitudinalen Aktionstension wurde mit Probanden in einem induktiven Prozess in einer Usability Studie evaluiert. Eines der Ziele der Evaluation war es, herauszufinden, ob die theoretisch identifizierten Tensionsbereiche dem subjektiv wahrgenommenen Risiko und Handlungsbedarf durch eine zusätzliche Automation entsprechen.

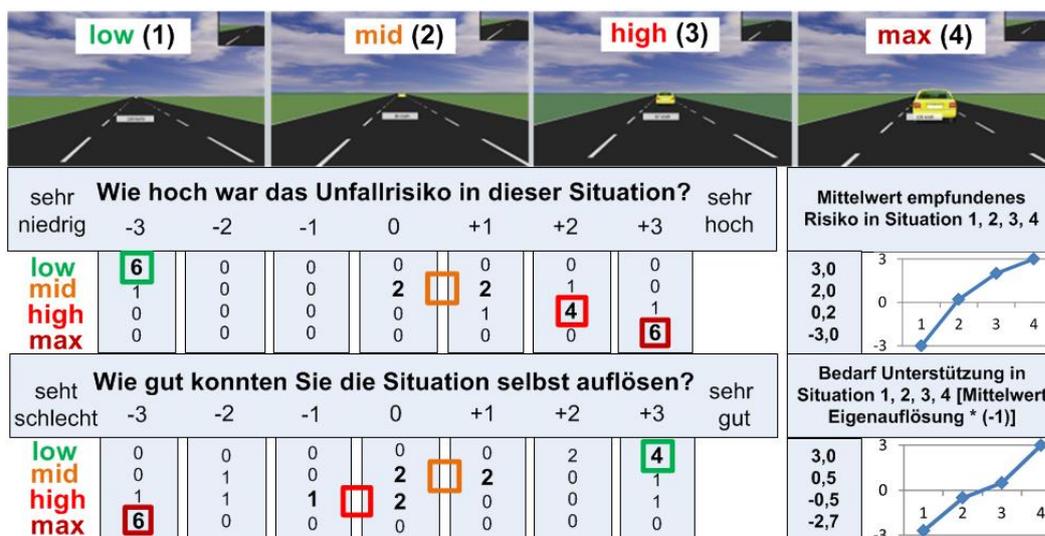


Abbildung 6-4: Auszug der Auswertungsergebnisse für deduktiv operationalisierte longitudinale Aktionstension

Nach dem zweckorientierten Usability Paradigma (vgl. Abschnitt 2.3.1) wurden sechs Probanden (3 Männer, 3 Frauen, 23-50 Jahre alt) mit Situationen konfrontiert, die als prototypisch

für die operationalisierten MPs angesetzt wurden. Die Probanden führen ein simuliertes Fahrzeug auf der mittleren Fahrspur einer geraden dreispurigen Autobahn für eine randomisierte Zeit in verschiedenen Bedingungen, z.B. ohne Zusatzverkehr und die linke Fahrspur war frei oder mit Zusatzverkehr und die linke Fahrspur war blockiert. Ein vorausfahrendes Fahrzeug erschien plötzlich auf der gleichen Fahrspur in unterschiedlichen Abständen und TTCs (Abbildung 6-4 oben). Dies sollte eine Fahrerreaktion in Form eines Fahrspurwechsels, Bremsens oder Ausweichens provozieren. Die Probanden mussten reagieren und danach konnten sie die eben erlebte Situation auf einem semantischen Differential bewerten.

Ein Ausschnitt der Ergebnisse ist in der Abbildung 6-4 gezeigt. Zahlen unter der Skala zeigen die Anzahl der Antworten der Probanden. Die farbigen Quadrate markieren die Medianwerte. Rechts in der Abbildung sind die Mittelwerte des empfundenen Risikos und des Bedarfs an Unterstützung dargestellt. Im Allgemeinen wurde beobachtet, dass das empfundene Risiko sowie der wahrgenommene Bedarf an Unterstützung mit der steigenden Aktionstension nahezu linear ansteigen. Dies entspricht der deduktiv hergeleiteten Definition der MPs.

6.1.6 Arbitrierungsstrategie

Mithilfe der operationalisierten und evaluierten longitudinalen Aktionstension wurden eine integrierte Arbitrierungsstrategie und ein dazugehöriges Interaktionsdesign entwickelt. Da die Funktion des Fahrerassistenzsystems in zwei unterschiedlich hohen Automationsgraden gefordert wurde, wurden zwei minimal verschiedene Versionen der Interaktion mit angepassten Schwellwerten für die MPs entwickelt. Die Arbitrierungsstrategie für den Automationsgrad „Hochautomatisiert“ ist im Syntaktischen Storyboard der Abbildung 6-5 gezeigt.

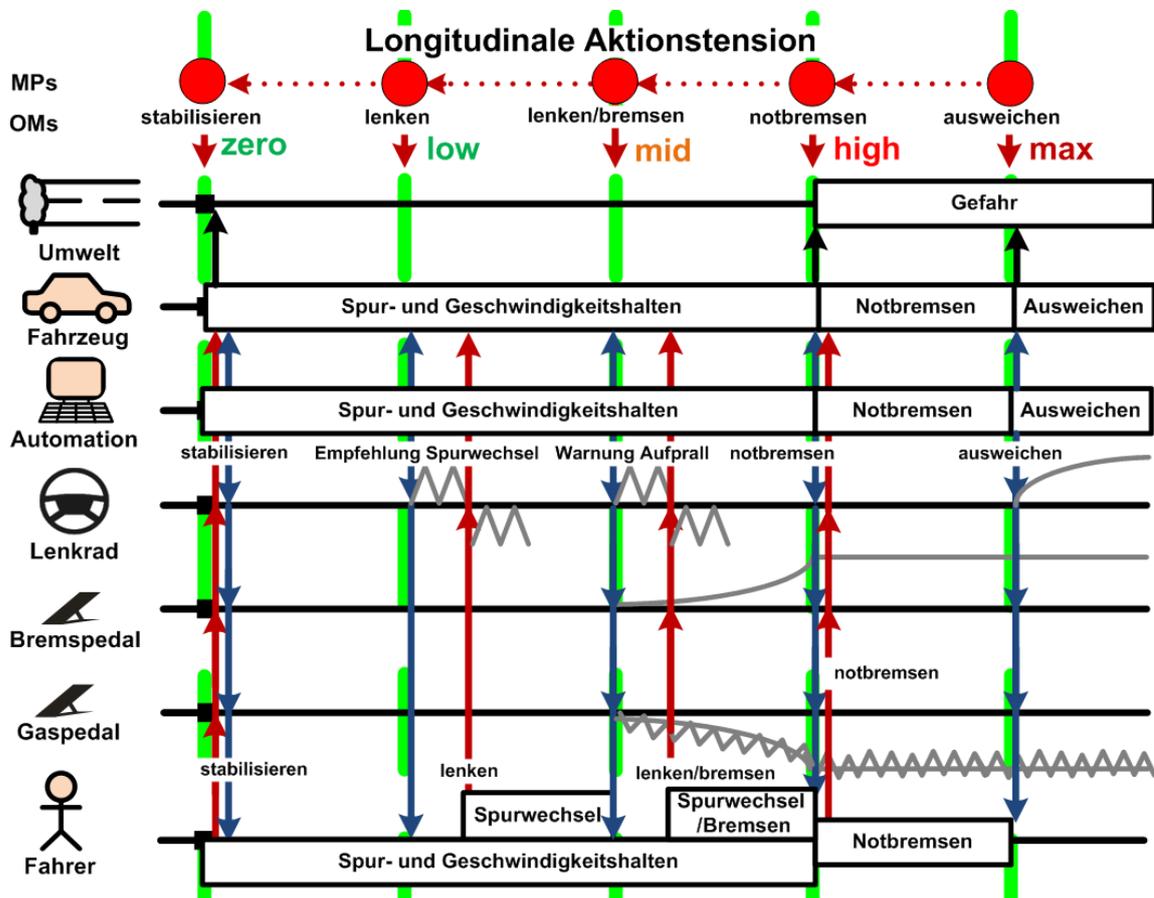


Abbildung 6-5: Arbitrierungsstrategie für das Beispielsystem 1 im Syntaktischen Storyboard

Das Storyboard kann von links nach rechts oder von rechts nach links gelesen werden. Beliebige Startpunkte auf dem Storyboard sind möglich. Für die nachfolgende Erklärung wird der Pfad vom linken zum rechten Ende des Storyboards verwendet. Im MP „zero“ steuern die Automation und der Fahrer gleichzeitig das Fahrzeug über das Lenkrad und Pedale zwecks einer Spur- und Geschwindigkeitshaltung. Sobald das MP „zero“ nach „low“ generiert wird, schickt die Automation eine Spurwechselempfehlung an den Fahrer in Form eines Doppelticks auf dem Lenkrad. Der Fahrer kann darauf mit einem Spurwechselbefehl reagieren, indem er lenkt oder einen Doppeltick seinerseits am Lenkrad ausführt.

Wenn der Fahrer nicht reagiert, schickt die Automation am Trigger von „low“ nach „mid“ eine Stoßwarnung. Diese besteht aus einem komplexen Arbitrierungssignal in Form von einem weiteren Doppeltick auf dem Lenkrad, einer ansteigenden Kraft auf dem Bremspedal und einer ansteigenden Gegenkraft mit einer Vibration auf dem Gaspedal. Es handelt sich dabei um das „virtuelle Kiesbett“ (vgl. Tabelle 5-2). Der Fahrer hat die Möglichkeit, darauf mit einem Bremsen bzw. einem Spurwechsel zu reagieren. Wenn der Fahrer nicht reagiert, steigert sich die Stoßwarnung zu einer Notbremsung am Trigger von „mid“ nach „high“, indem auf dem Bremspedal die maximale Bremskraft und am Gaspedal die maximalen Gegenkraft und Vibration anliegen. Dies ist die letzte Möglichkeit für den Fahrer zu reagieren. Reagiert er nicht, weicht die Automation selbständig dem Hindernis aus, indem sie stark lenkt. Die symbolisierte Interaktion, wie Doppelticks, Kräfte und Vibrationen, ist zur Veranschaulichung der haptischen Interaktion auf den Lebenslinien der aktiven Stellteile gezeigt.

Die Abbildung 6-3 und die Abbildung 6-5 zeigen zusammen, wie sich die verwendete Aktionsdimension vom Szenario über ihre Operationalisierung bis zum integrierten Interaktionsdesign auswirkt. Die Transitionen zwischen den MPs gelten als Trigger für das Automationsverhalten, wie Notbremsen oder Ausweichen, aber auch für Interaktionsverhalten, wie Empfehlungen oder Warnungen. Dies sollte das Verhalten der Automation und des Fahrers zeitlich abstimmen. Außerdem gestattet es, die Komplexität des Systems zu reduzieren, zu linearisieren und insgesamt ein holistisches System zu gestalten.

6.1.7 Umsetzung als Prototyp

Das mit osGekoS entwickelte Interaktionsdesign wurde mittels des Interaktionskontrollers (vgl. Abschnitt 5.7) implementiert. Dafür wurde das in der Abbildung 6-5 gezeigte Storyboard in das Arbitrierungswerkzeug übertragen (vgl. Abschnitt 5.6).

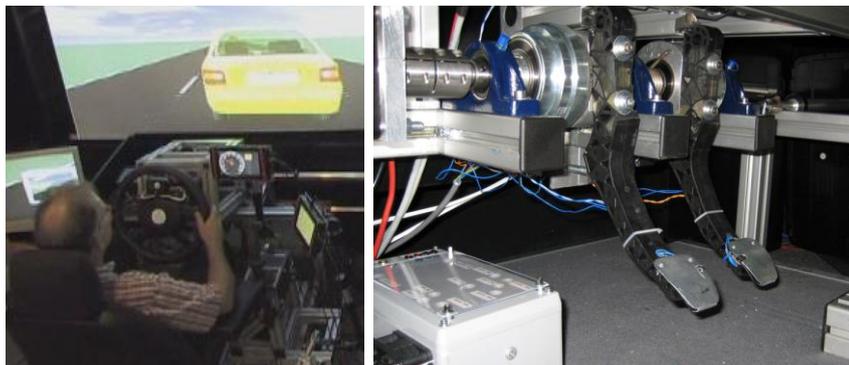


Abbildung 6-6: Versuchsanordnung für die Usability Studie für das Beispielsystem 1 und die aktiven Pedale, die speziell für die Versuche mit haptischer Modalität entwickelt wurden

Als Automationsagent wurde die manöverbasierte Automation verwendet^[Lök08]. Diese stellt eine Kompatibilität zum Fahrer durch interne Einteilung ihrer Algorithmen auf wissens-, regel- und fertigkeitbasierten Ebenen bereit. Als Simulationsumgebung wurde ein statischer Fahr-Simulator des SMPLab verwendet. Neben einer Fahrsimulation und Automation war dieser Simulator mit aktiven Pedalen und einem aktiven Lenkrad ausgestattet (Abbildung 6-6).

6.1.8 Usability Studie

Eines der Ziele der Usability Studie war die Beurteilung, ob das mit osGekoS entwickelte Interaktionsdesign von potenziellen Systemnutzern gut verstanden und akzeptiert wird. Diese Studie war nötig, da sie eine objektive Bewertung sowohl des vorher vom Systemingenieur mit einigen subjektiven Anteilen entwickelten Interaktionsdesigns als auch indirekt von osGekoS gestatten würde. Im Falle der negativen Bewertung ließe sich Verbesserungspotenzial für das Interaktionsdesign aber auch für osGekoS ableiten. Im Falle der positiven Nutzerwertung würde das auch eine positive Bewertung der Nützlichkeit und des Mehrwertes des Interaktionsdesigns und auch von osGekoS erlauben. Weil die übrigen Ziele der Usability Studie, wie die Evaluation des Fahrsimulators, der entwickelten Fahrszenarien und der HMI-Hardware nicht zur Evaluation von osGekoS beitragen, werden sie hier nicht weiter thematisiert.

Die eingeladenen Probanden befuhren mit dem aktiven Lenkrad und aktiven Pedalen eine dreispurige Autobahn auf der mittleren Fahrspur mit entweder freier linker und rechter Spur (Ausweichmöglichkeit) oder belegter linker und rechter Spur (keine Ausweichmöglichkeit) (vgl. Abbildung 6-1). Ihre Aufgabe war, während der Fahrt Zweitaufgaben zu bearbeiten und dabei Kollisionen mit auf der eigenen Spur erscheinenden Fahrzeugen zu vermeiden.

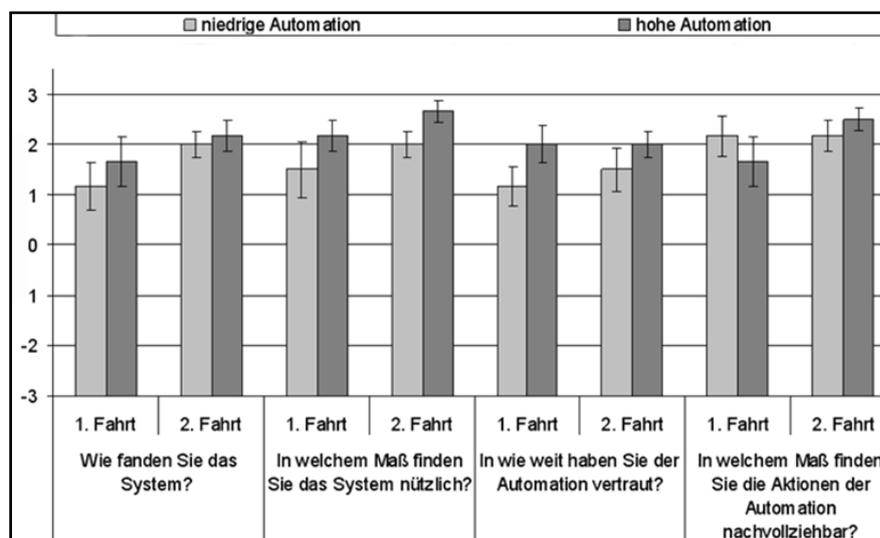


Abbildung 6-7: Ergebnisse der subjektiven Befragung nach der ersten (naiven) und der zweiten (nach Training) Fahrt des Prototyps für das Beispielsystem 1 in zwei unterschiedlichen Automationsgraden

Nach dem zweckorientierten Usability Paradigma (vgl. Abschnitt 2.3.1) nahmen sechs Probanden (3 Männer, 3 Frauen, 23-50 Jahre alt) an der Usability Studie teil. Im ersten Durchgang trafen die Probanden ohne Systemwissen auf den Prototyp. Sie fuhren alle Kollisionssituationen durch und beurteilten das System. Im zweiten, segmentierten Durchgang wurde nach jeder Kollisionssituation ein Fragebogen vorgelegt, um die Systembeurteilung für diese Situation zu erfassen. Gemessene Variablen waren empfundene Nützlichkeit und Sicherheit,

Kontrollierbarkeit, Kontrollverteilung etc. Im letzten Durchgang mit vorher durch Training vermitteltem Systemwissen wurde der Prototyp nochmals beurteilt.

Insgesamt zeigte sich, dass Probanden das System in der Gesamtbeurteilung sowohl nach Erstkontakt und noch ausgeprägter nach einem Training positiv bezüglich Gefallen, Nützlichkeit und Nachvollziehbarkeit bewerteten. Je höher die Automationsunterstützung war, desto positiver war die Bewertung. Ebenso führte das System zu einer Verbesserung der subjektiv eingeschätzten Kontrollierbarkeit der Kollisionssituationen. In der Einzelbetrachtung fiel in den MPs „low“ und „mid“ auf, dass Probanden trotz Ablenkung durch Zweitaufgabenbearbeitung die kritische Situation durch einen selbständigen Eingriff auflösten bevor die Eskalationsstrategie fortschritt (14 von 18 Situationen). Die Interaktion beinhaltete einen Doppeltick (Abbildung 6-5) als Automationsempfehlung, dass ein Spurwechsel sinnvoll ist. Dieser wurde bereits im Erstkontakt in 5 von 12 Situationen erkannt und mit der richtigen Bedeutung assoziiert. Von keinem Probanden wurde der Doppeltick als störend empfunden. Als verbesserbar stellte sich eine mit zunehmender Aktionstension ansteigende Gegenkraft auf dem Gaspedal heraus, die von einigen Probanden nicht konsistent erkannt wurde.

6.1.9 Zusammenfassung

Das Beispielsystem 1 für Nutzungsfälle Annähern, Bremsen und Ausweichen wurde mittels der osGekoS-Theorie, Regeln und Werkzeuge gestaltet. Während der Usability Studie wurde im Allgemeinen beobachtet, dass das mit osGekoS gestaltete CS von den Probanden gut verstanden und akzeptiert wird. Dies wurde durch anschließende Datenauswertung bestätigt. Daraus ergibt sich, dass mit osGekoS akzeptable und gebrauchstaugliche kognitive Systeme im Bereich hochautomatisierter Fahrzeugführung gestaltet werden können.

6.2 Beispielsystem 2: Kooperation bei Spurwechseln

Dieses Fahrerassistenzsystem wurde im EU-Projekt D3CoS^[D3S12] entwickelt^{[KeT13][KeD15]}. In dem Projekt ging es im Allgemeinen um Methoden, Techniken und Werkzeuge für die Gestaltung dynamischer verteilter kooperativer Mensch/Maschine Systeme. Allgemeine Prinzipien der (Mensch/Mensch) Kooperation und des kooperativen Verhaltens^[Man02] wurden im Projekt erforscht und in Methoden, Techniken und Werkzeuge für eine Systemgestaltung überführt.

6.2.1 Anforderungen

Für den Nutzungsfall „Kooperativer Fahrspurwechsel“ soll ein Fahrerassistenzsystem entwickelt werden. Das Assistenzsystem sollte im schnellen dichten Verkehr auf einer zweispurigen Autobahn einen Kooperationsvorgang zwischen zwei Verkehrsteilnehmern initiieren, beibehalten und beenden können. Das Ziel der Kooperation sollte sein, dem Fahrer auf der rechten Fahrspur, einen Spurwechsel auf die linke Fahrspur zu ermöglichen. Die Fahrer/Automation Interaktion sollte über den visuellen Kanal auf den Fahrzeugcockpitdisplays und auf den Lenkrädern der beiden Fahrzeuge ohne einen aktiven Eingriff durch die Automation stattfinden.

6.2.2 Szenario und Systemdekomposition

Als Hauptszenario für die Systemgestaltung ist eine zweispurige gerade Autobahn ausgewählt worden (Abbildung 6-8 links). Auf der linken Fahrspur fährt das erste Eigenfahrzeug V1 in einer Kolonne von Fremdfahrzeugen. Seine konstante Geschwindigkeit sowie die Geschwindigkeit der Kolonne auf seiner Fahrspur betragen 80 km/h. Auf der rechten Fahrspur fährt das zweite Eigenfahrzeug V2 in der dichten Kolonne von Fremdfahrzeugen. Seine konstante Geschwindigkeit sowie die Geschwindigkeit der Kolonne auf seiner Fahrspur betragen 120 km/h. Ab einem bestimmten Zeitpunkt fängt das Fremdfahrzeug vor dem V1 langsam abzubremsen und provoziert damit einen Ressourcenkonflikt und die Notwendigkeit eines Spurwechsels für V1 von rechts nach links. Damit wird die Möglichkeit für V2 geschaffen, V1 auf seine (linke) Fahrspur fahren zu lassen, wenn V2 eine Lücke für V1 aufmacht. V1 und V2 sind hochautomatisierte Fahrzeuge, die jeweils aus $d1, c1, v1$ und $d2, c2, v2$ bestehen. $d1$ und $d2$ sind die Fahrer, $c1$ und $c2$ sind die Automationen, $v1$ und $v2$ sind die Fahrzeuge selbst. Alle Fremdfahrzeuge sind mit der nahen Umwelt als e zusammengeführt.

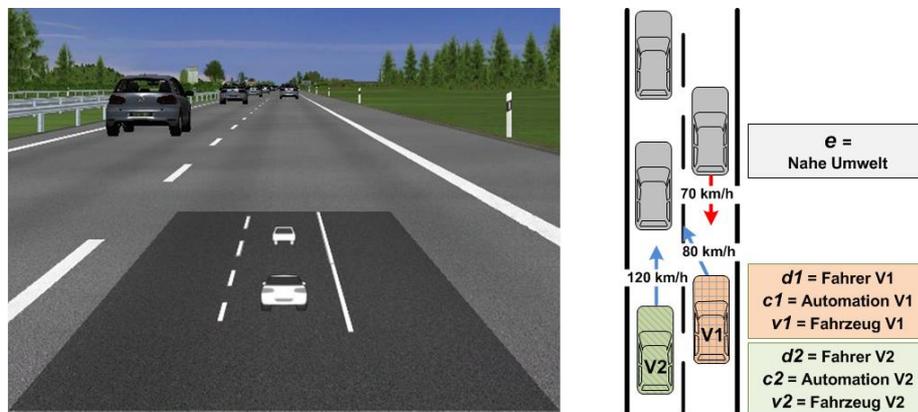


Abbildung 6-8: Simulationsansicht und die Dekomposition des Szenarios für Beispielsystem 2

6.2.3 Konfliktanalyse und Semantisches Mapping

Für dieses System wurde angenommen, dass die Automation konfliktfrei funktioniert. Ein Ausschnitt der fahrer- und der umgebungsbezogener Konfliktanalyse des gegebenen Anfangssystemzustandes ist in der Tabelle 6-3 dargestellt.

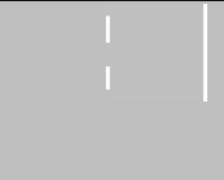
Tabelle 6-3: Auszug aus Konfliktanalysewerkzeug für das Beispielsystem 2

ANFANGS-SYSTEM-ZUSTAND		KONFLIKTBESCHREIBUNG					LÖSUNG
		Agentenbezogene Konflikte				Umgebungs-konflikte	
		Bewusstse in	Intention Entscheidung	Koordination	Handlung		
Fahrer V1	Fahre, meide Kollisionen, kooperiere	Abgelenkt und/oder reagiert nicht	Eigene Handlungs-alternativen unbekannt	Erkennt nicht, dass die Lücke offen ist	Nutzt die Lücke zu früh/zu spät	Ein langsames Fahrzeug voraus, Lücke links blockiert	Konfliktvermeidung: Ein Ressourcenkonflikt um die Lücke unvermeidbar Konfliktlösung: Kooperation zwischen zwei Fahrzeugen rechtzeitig aufbauen, beibehalten und beenden mit dem Ziel, eine Lücke aufzumachen und die Fahreraktionen zu lenken. Dies muss deutlich an die Fahrer und zwischen den Fahrern kommuniziert werden
Automa-tion V1	Koordiniere Spurwechsel						
Fahrer V2	Fahre, meide Kollisionen, kooperiere	Abgelenkt und/oder reagiert nicht	Eigene Handlungs-alternativen unbekannt	Erkennt nicht, dass die Lücke offen ist	Macht die Lücke falsch auf	Ein Fahrzeug versucht, vorne einzuscheren, Lücke vorne zu klein	
Automa-tion V2	Koordiniere Spurwechsel						

Bei der Konfliktanalyse ist aufgefallen, dass viele mögliche Konflikte entweder mit der aufzumachenden Lücke oder mit der Unsicherheit der Fahrer bei der Handlungswahl in Bezug auf die Art und den Zeitpunkt der Handlung zu tun haben können. Deswegen wurde entschieden, sich beim Gestalten der Arbitrierungsstrategie auf die Lücke und auf die rechtzeitige Darstellung der adäquaten nächsten Handlung zu fokussieren.

Semantisches Mapping für visuelle Arbitrierungssignale wurde in mehreren kurzen Vorstudien und mittels Analyse bereits vorliegender Prototypen durchgeführt. Das Ergebnis des Semantischen Mapping ist in der Tabelle 6-4 dargestellt^{[Klm13][KeD15]}.

Tabelle 6-4: Auszug aus dem Werkzeug „Semantisches Mapping“ visueller Signale für die Gestaltung hochautomatisierter Fahrzeugführung. Gezeigt ist der Symbolsatz für das Beispielsystem 2

Verlauf	Element	Variationsmöglichkeiten	Generische Bedeutung
Diskret	Fahrzeug		Farbe, Größe, Position auf dem Display Eigenfahrzeug (blau), Fremdfahrzeug, besonderes Fremdfahrzeug (rot)
Diskret	Spurmarkierung		Farbe, Position auf dem Display, Linienart (durchgezogen, gestrichelt, doppelt etc.) Aktueller Zustand der Spurmarkierungen um das Eigenfahrzeug herum (weiß), Spurrerlassenswarnung (rot)
Diskret	Lückenfüller		Farbe, Größe, Position auf dem Display Geschlossene (rot)/offene (grün) Lücke, angefragte Lücke (rot), "Radarkeule"
Diskret	Aktionen		Farbe, Position auf dem Display, Blinkverhalten Intention (rot), auszuführende/ausgeführte Handlung (grün)
Diskret	Fertig!		keine Lücke ist fertig!
Diskret	Nachricht		Position auf dem Display, Blinkverhalten Nachricht von Automation an Fahrer
Kontinuierlich	Kommunikation		Position auf dem Display, Richtung Kommunikation wird aufgebaut in die Richtung...
Kontinuierlich	Warten		keine Bitte warten...
Zusammengesetzt	Dyade		Position auf dem Display, Richtung Kooperationsdyade
Zusammengesetzt	9er-Schema		9 Felder mit beliebigen einfachen Elementen Führen einer Kooperation

6.2.4 Lösungsexploration

Für das Beispielsystem 2 aus Abbildung 6-8 kann ein Modell aufgestellt werden, das anschließend auf die Entscheidungskonvergenz untersucht werden kann. In der Abbildung 6-9 ist der entsprechende Zustandsübergangsgraph gezeigt.

Die Abbildung zeigt ein System aus Agenten (Fahrer, Arbitrer) der beiden kooperierenden und manuell gefahrenen Fahrzeuge (V1, V2) des Beispielsystems 2, die über die beiden Arbitrer miteinander gekoppelt sind. Die Konfiguration eines solchen Systems kann wie folgt aussehen:

hen: Die Fahrer ($d1, d2$) und die Arbitr ($a1, a2$) sind CAs mit Eigenmanipulationen ($k_n^{d1d1} = k_n^{d2d2} = k_n^{a1a1} = k_n^{a2a2} = 1$) (vgl. Abschnitt 4.2.3). Die beiden Fahrer können sich gegenseitig nicht direkt manipulieren ($k_n^{d1d2} = k_n^{d2d1} = 0$), sondern nur indirekt über die Arbitr ($k_n^{d1a1} = k_n^{a1a2} = k_n^{d2a2} = k_n^{a2a1} = 1$), z.B. Beantworten von Kooperationsanfragen. Sie empfangen dafür ein Feedback vom Arbitr ($k_n^{a1d1} = k_n^{a2d2} = 1$), z.B. bekommen Kooperationsanfragen.

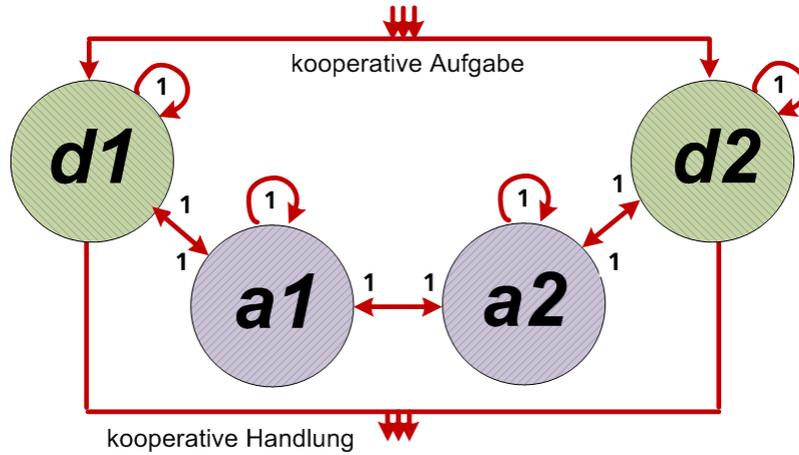


Abbildung 6-9: Zustandsübergangsgraphen des Szenarios für das Beispielsystem 2

Mit dazugehöriger Kopplungsmatrix nach (5) ergibt sich bei allen Annahmen (21).

$$K_n = \begin{pmatrix} k_n^{d1d1} & k_n^{a1d1} & k_n^{d2d1} & k_n^{a2d1} \\ k_n^{d1a1} & k_n^{a1a1} & k_n^{d2a1} & k_n^{a2a1} \\ k_n^{d1d2} & k_n^{a1d2} & k_n^{d2d2} & k_n^{a2d2} \\ k_n^{d1a2} & k_n^{a1a2} & k_n^{d2a2} & k_n^{a2a2} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}; \quad \vec{a}_n = \begin{pmatrix} a_n^{d1} \\ a_n^{a1} \\ a_n^{d2} \\ a_n^{a2} \end{pmatrix} \quad (21)$$

Mit (11) lassen sich Zeitreihen für das aufgestellte Modell entwickeln. Damit kann man z.B. zeigen, wie der Arbitr des V1 den Kooperationsbedarf im System weiter geben kann oder was passiert, wenn der Fahrer von V2 nicht kooperieren möchte.

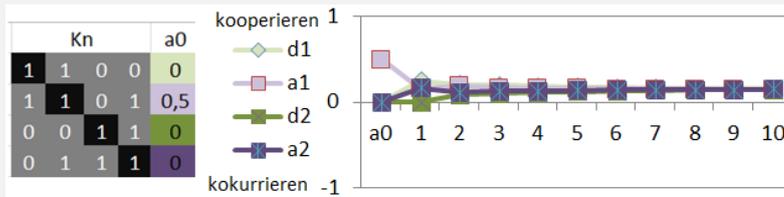
Die Tabelle 6-5 zeigt Beschreibungen einiger relevanten Strukturvarianten (oben links), die vom Autor prospektiv erwarteten Entscheidungskonvergenzen innerhalb des Modells (oben rechts), Anfangswerte für K_n und \vec{a}_n (unten links), die entsprechenden Zeitreihen mit der MP-Aktivierung (Entscheidungskonvergenz) der Agenten bis zum 10. Berechnungsschritt (unten Mitte) und ihre Interpretationen (unten rechts).

Die auf diese Weise explorierten Arbitrierungsstrategien und Schwellwerte (+0,5 für den Kooperationsbedarf, -1 für die Ablehnung der Kooperation) für Manipulationspotenziale können noch weiter exploriert werden und im weiteren Schritt präzisiert und operationalisiert werden.

Tabelle 6-5: Restrukturierungsvarianten des Modells des Beispielsystems 2, erwartete Entscheidungskonvergenzen, entsprechende Anfangsparameter, Zeitreihenentwicklungen und ihre Interpretationen

<p>Struktur 1 (Weitergabe des Kooperationsbedarfs): Fahrer der beiden Fahrzeuge (d1, d2) und der Arbitr a2 kooperieren nicht ($a_n^{d1} = a_n^{d2} = a_n^{a2} = 0$). Der Arbitr a1 hat gerade einen Kooperationsbedarf festgestellt ($a_n^{a1} = 0,5$)</p>	<p>Erwartete Entscheidungskonvergenz: Beide Fahrer werden über ihre Arbitr zur Kooperation angeregt.</p>
---	---

Anfangsparameter und Zeitreihenentwicklung

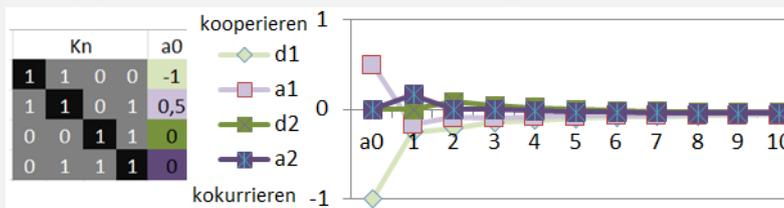


Interpretation: CS-Entscheidung konvergiert gegen „kooperieren“ und folgt damit der Empfehlung des Arbiters a1. Der Fahrer d1 reagiert darauf mit Kooperation indem er z.B. diese über einen Knopfdruck bestätigt. Fahrer d2 folgt dem und kooperiert auch.

Struktur 2 (Fahrer d1 lehnt Kooperation ab): Fahrer d2 und Arbitrer a2 kooperieren (noch) nicht ($a_n^{d2} = a_n^{a2} = 0$). Der Arbitrer a1 hat gerade einen Kooperationsbedarf festgestellt ($a_n^{a1} = 0,5$), Fahrer d1 ist aber negativ gegenüber einer Kooperation eingestellt ($a_n^{d1} = -1$).

Erwartete Entscheidungskonvergenz: CS-Entscheidung sollte gegen „konkurrieren“ konvergieren.

Anfangsparameter und Zeitreihenentwicklung

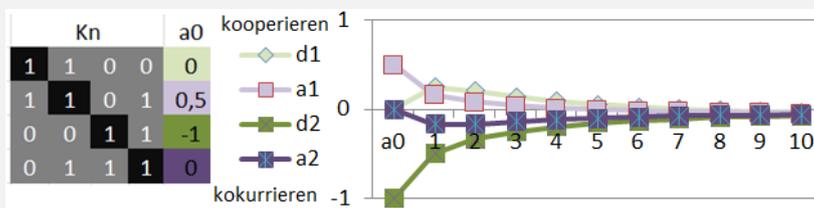


Interpretation: CS-Entscheidung konvergiert gegen „konkurrieren“. Auch wenn der Fahrer d2 sich anfangs kooperativ zeigt, kommt es im System zu keiner Kooperation, da der Fahrer d1 die Kooperation, z.B. über einen Knopfdruck, ablehnt.

Struktur 2 (Fahrer d2 lehnt Kooperation ab): Fahrer d1 und Arbitrer a2 kooperieren (noch) nicht ($a_n^{d1} = a_n^{a2} = 0$). Der Arbitrer a1 hat gerade einen Kooperationsbedarf festgestellt ($a_n^{a1} = 0,5$), Fahrer d2 ist aber negativ gegenüber einer Kooperation eingestellt ($a_n^{d2} = -1$).

Erwartete Entscheidungskonvergenz: CS-Entscheidung sollte gegen „konkurrieren“ konvergieren.

Anfangsparameter und Zeitreihenentwicklung



Interpretation: CS-Entscheidung konvergiert gegen „konkurrieren“. Auch wenn der Fahrer d1 sich anfangs kooperativ zeigt, kommt es im System zu keiner Kooperation, da der Fahrer d2 die Kooperation, z.B. über eine Nichtreaktion, ablehnt.

6.2.5 Operationalisierung der Aktionstension

Für die Operationalisierung der Aktionstension wurde zunächst ein deduktiver Ansatz mithilfe einer Literaturrecherche und Systemanalyse nach relevanten Daten und Konzepten verwendet. Anschließend wurde das konzeptionelle Ergebnis der Recherche in einem induktiven Ansatz mit Probanden innerhalb einer Usability Studie überprüft.

Während der funktionalen CS-Dekomposition wurden nach CWA (vgl. Abschnitt 2.3.1) Domänenzweck und -funktion definiert: „Beförderung von Personen im Straßenverkehr“, „der Verkehr soll sich energieeffizient verhalten“ und „die Gefährlichkeit der Verkehrssituation sollte minimiert werden“. Um die Effizienz des Verkehrs und die Sicherheit in dem vorgeschlagenen Szenario (vgl. Abbildung 6-8) zu verbessern, ist es sinnvoll, einen Prozess der Kooperation zwischen V1 und V2 zu initiieren. Das Fahrzeug V2 könnte langsam und somit energieeffizient verzögern, um einen Spurwechsel für das Fahrzeug V1 zu ermöglichen. Danach könnte V1 die Fahrspur fließend und ebenfalls effizient wechseln. Als Leitkonzept für die Lösung festgestellter Konflikte (vgl. Tabelle 6-3) wurden also die Initiierung, Beibehaltung und Beendigung einer Kooperation zwischen den Fahrzeugen V1 und V2 gewählt (Abbildung

6-10). Durch die positive Wechselwirkung in Form einer Kooperation zwischen allen Agenten sollte die Sicherheit, Effizienz und Komfort des gesamten CS erhöht werden.

Mittels Literaturrecherche wurde entschieden, den Kehrwert der TTC für die Beschreibung des kontinuierlichen Wertes der Aktionstension zu verwenden. Je kleiner die TTC ist, desto größer sind der Kehrwert der TTC und die Notwendigkeit zu kooperieren. Auf dem Verlauf des Kehrwertes der TTC wurden Bereiche festgelegt, die die kooperative Aktionstension in MPs aufteilen. Grundsätzlich gibt es zwei dichotome Bereiche: Kooperation ist nicht nötig und Kooperation ist nötig. Der Bereich „Kooperation ist nicht nötig“ ist das entspannte MP des CS. Der Bereich „Kooperation ist nötig“ ist in vier MPs aufgeteilt: Kooperationsanfrage an V1, Kooperationsanfrage an V2, Lückenproduktion und Lückenkonsum. Die zu den MPs gehörende Manipulationen sind: Bestimme die Kooperationsnotwendigkeit, bestimme die Kooperationsakzeptanz von V1, bestimme die Kooperationsakzeptanz von V2, führe die Produktion der Lücke, führe den Konsum der Lücke und beende die Kooperation (Abbildung 6-10 unten).

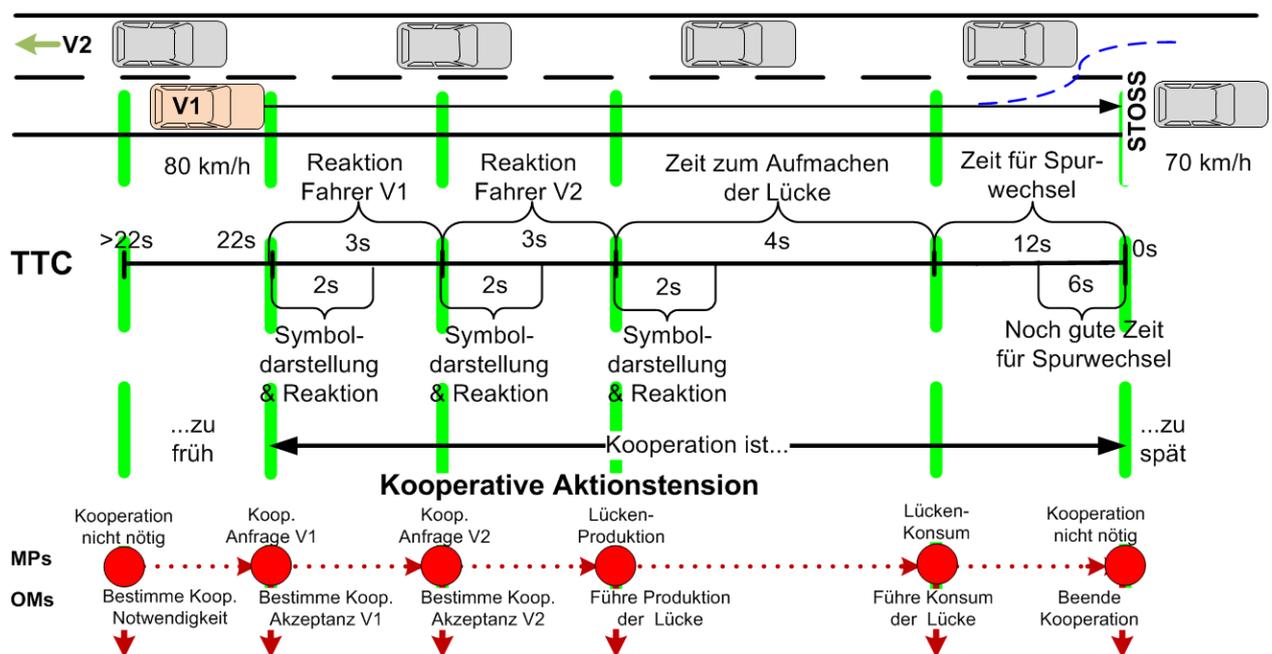


Abbildung 6-10: Operationalisierung der Aktionstension für das Beispielsystem 2

Für die Operationalisierung der MPs, dazugehöriger Manipulationen und Trigger-Ereignisse wurde das vorliegende Szenario in Bezug auf Differenzgeschwindigkeiten, Reaktionszeiten der Fahrer bei Symboldarstellungen und der Automation bei Abbremsvorgängen und Spurwechseln analysiert. Als Ereignis für den Spurwechsel im Falle einer erfolgreichen Kooperation wurde der Zeitpunkt gewählt, wenn die meisten Fahrer sich für einen Spurwechsel bei einer Annäherung an ein langsames Fahrzeug entscheiden würden^[FaH01].

Die deduktiv erarbeitete Definition der kooperativen Aktionstension wurde mit Probanden in einem induktiven Prozess evaluiert. Eines der Ziele der Evaluation war, herauszufinden, ob die theoretisch identifizierten Tensionsbereiche dem subjektiv wahrgenommenen Kooperationsbedarf der Probanden mit anderen Verkehrsteilnehmern entsprechen (Abbildung 6-11).

Nach dem zweckorientierten Usability Paradigma (vgl. Abschnitt 2.3.1) wurden acht Probanden (4 Männer, 4 Frauen, 19-58 Jahre alt) mit Situationen konfrontiert, die als prototypisch für erarbeitete MPs angesetzt wurden. Die Probanden fuhren ein simuliertes Fahrzeug im

ichten Verkehr zuerst auf der rechten (V1) und dann auf der linken (V2) Fahrspur einer geraden zweispurigen Autobahn. Nach einer randomisierten Zeit wurde ein Kooperationsbedarf erzeugt und die Arbitrierungsstrategie aus Abschnitt 6.2.6 durchgespielt. Die Probanden erlebten unterschiedliche MPs der kooperativen Aktionstension, indem sie zum kooperativen Spurwechsel angeregt wurden. Die Probanden mussten reagieren und danach konnten sie die eben erlebte Situation auf einem semantischen Differential bewerten.

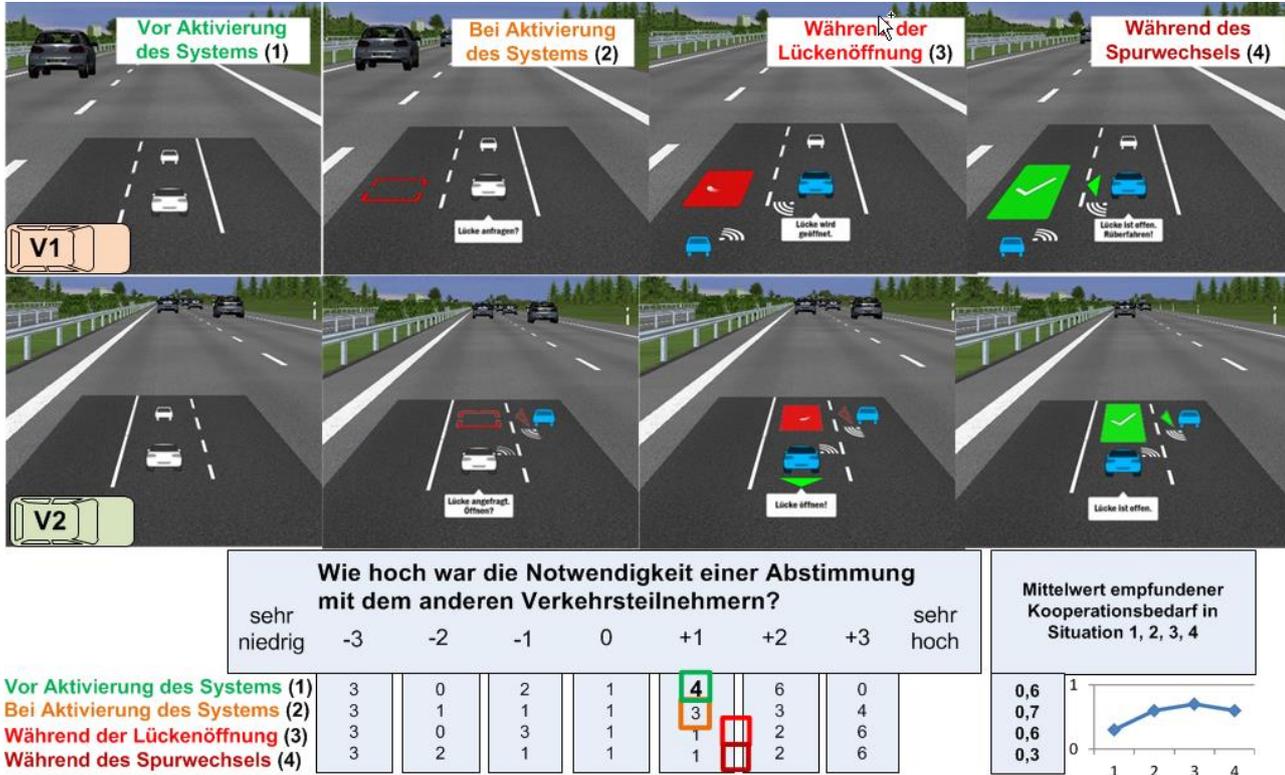


Abbildung 6-11: Auszug der Auswertungsergebnisse für deduktiv operationalisierte kooperative Aktionstension

Ein kurzer Ausschnitt der Ergebnisse ist in der Abbildung 6-11 gezeigt. Die Zahlen unter der Skala zeigen die Anzahl der jeweiligen Antworten der Probanden. Die farbigen Quadrate markieren die entsprechenden Medianwerte. Rechts unten in der Abbildung sind die Mittelwerte des empfundenen Kooperationsbedarfs berechnet und grafisch dargestellt. Im Allgemeinen wurde beobachtet, dass der empfundene Kooperationsbedarf stetig ansteigt und am Ende wieder fällt, wenn die Situation sich durch den Spurwechsel entspannt. Dies entspricht der deduktiv hergeleiteten Definition der MPs der kooperativen Aktionstension.

6.2.6 Arbitrierungsstrategie

Mithilfe der operationalisierten und evaluierten kooperativen Aktionstension wurden eine integrierte Arbitrierungsstrategie und ein dazugehöriges Interaktionsdesign entwickelt. Das Syntaktische Storyboard dafür ist in der Abbildung 6-12 gezeigt. Das Storyboard kann von links nach rechts gelesen werden. Wenn ein neues MP erreicht ist, kann er von links nach rechts, von rechts nach links entlang der gestrichelten Pfeile oder durch den Abbruch der Kooperation über die Trigger-Ereignisse gewechselt werden. Für die Erklärung des Syntaktischen Storyboards wird der Durchlauf vom linken zum rechten Ende verwendet.

In der Situation des Hauptszenarios für Kooperation bei Spurwechseln (vgl. Abbildung 6-8) teilen die Automationen der beiden Fahrzeuge (V1 und V2) ihren Fahrern über die leuchten-

den Lenkradknöpfe mit, dass sie gerade auf der Suche nach einer Kooperationsnotwendigkeit sind. Am ersten Trigger zwischen den MPs, wenn eine Kooperation notwendig wird, fragt die Automation den Fahrer von V1 über das Display und die Lenkradknöpfe, ob er kooperieren möchte. Der Fahrer hat die Möglichkeit, Kooperation über Knöpfe am Lenkrad abzulehnen oder zu bestätigen. Wenn er die Kooperation ablehnt, kehrt der Ablauf des Syntaktischen Storyboards in die Ausgangslage zurück (gestrichelter Pfeil).

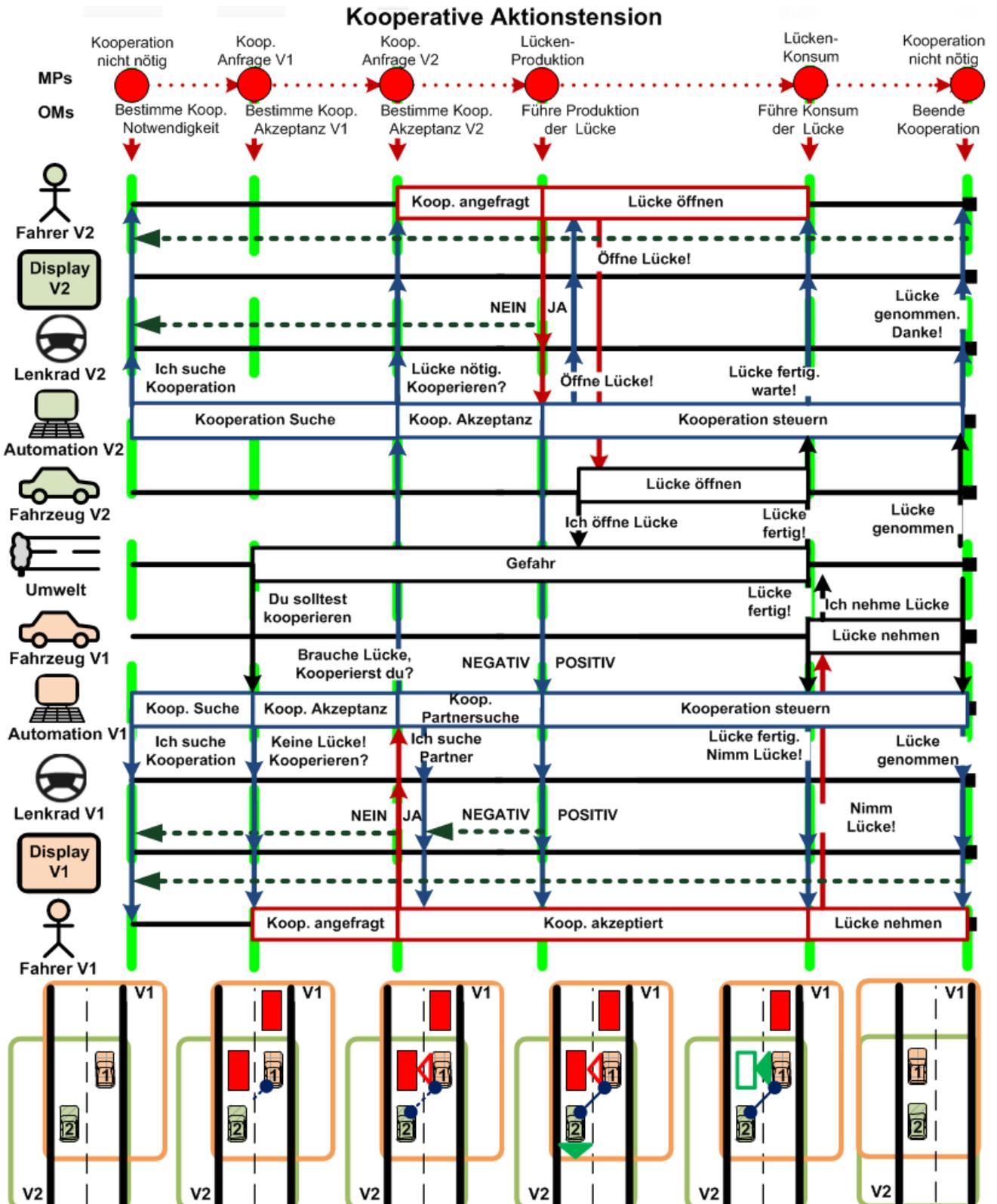


Abbildung 6-12: Arbitrierungsstrategie für das Beispielsystem 2 im Syntaktischen Storyboard

Wenn der Fahrer von V1 eine Absicht zu kooperieren über den Lenkradknopf zeigt, setzt sich seine Automation mit der aus dem Fahrzeug V2 in Verbindung. Mit einer ähnlichen Routine, wie schon im V1, wird festgestellt, ob der Fahrer von V2 ebenfalls kooperieren möchte. Wenn er dies nicht möchte, wird in beiden Fahrzeugen die Kooperation mit entsprechenden Rückmeldungen an die Fahrer abgebrochen. Wenn er kooperieren möchte, bekommt er von seiner Automation eine Unterstützung über das Display, dass und wie er eine Lücke aufmachen sollte. Das V1 wartet unterdessen auf die Lücke. Wenn die Lücke vorbereitet ist, wird dies an die Fahrer beider Fahrzeuge über die Displays mitgeteilt. Daraufhin kann das V1 in die Lücke fahren, womit der Kooperationsvorgang mit einer Bedankung bei V2 abgeschlossen wird.

Die schematische Darstellung dessen, was in den beiden Fahrzeugen auf den Displays angezeigt wird, ist in der Abbildung 6-12 unten gezeigt. Man erkennt, dass der grüne und der orangene Displayrahmen minimal unterschiedliche Ausschnitte derselben Situation anzeigen. Das Eigenfahrzeug ist dabei immer in der Mitte des Displays angeordnet. Die vollen roten Rechtecke kennzeichnen die belegten relevanten Fahrspurabschnitte. Offenes grünes Rechteck steht für eine offene Lücke. Leere rote Dreiecke kennzeichnen die Absichten und die vollen grünen die Handlungen der Fahrer bzw. die Handlungsanweisungen an die Fahrer von ihren Automationen. Die blaue Linie mit zwei Kreisen, die an den Fahrzeugsymbolen befestigt sind, steht für eine „Kooperationsdyade“, die eine gerade entstehende (gestrichelt) oder bereits entstandene Kooperation kennzeichnet. Alle genannten Elemente können je nach Bedarf und Situation im so genannten „9er-Schema“ (wegen $3 \times 3 = 9$ Felder) angeordnet werden. Um das Eigenfahrzeug herum können Spurmarkierungen, andere relevante Fahrzeuge, Kommunikationselemente, freie und belegte Straßenabschnitte, Fahrerabsichten, Handlungen und Anweisungen angezeigt werden. Die tatsächlich gestalteten Bilder aus den umgesetzten Displays sind außerdem in der Abbildung 6-11 im oberen Teil und Tabelle 6-4 bereits gezeigt.

Mithilfe der Dyaden sollte die Verantwortungsdiffusion^[Wak64] bei den Fahrern unterbunden und die Kooperation aufrechterhalten werden. Die Absichts- und die Lückenverfügbarkeitsrückmeldung sollten das Situationsbewusstsein der Fahrer verbessern. Die Handlungsanweisungen sollten die kooperativen Handlungen der Fahrer initiieren und begleiten. Insgesamt hat die gestaltete Arbitrierungsstrategie zum Ziel, einen Kooperationsprozess zwischen den Fahrzeugen V1 und V2 zu initiieren, ihn zu erhalten und entlang der kooperativen Aktions-tension zum erfolgreichen Abschluss zu führen.

6.2.7 Umsetzung als Prototyp

Das entwickelte Interaktionsdesign wurde im Interaktionskontroller (Abschnitt 5.7) implementiert. Dafür wurde das in der Abbildung 6-12 gezeigte Storyboard in das Arbitrierungswerkzeug übertragen (Abschnitt 5.6). In der Abbildung 6-12 unten gezeigter schematischer Ablauf der Displayinformation im 9er-Schema wurde mittels der in der Tabelle 6-4 erarbeiteten Semantik als Head-Up Display im Simulator des IDEELab umgesetzt (Abbildung 6-13 links). Der Simulator enthält eine statische Fahrsitzkiste und eine 180°-Projektion der Fahrsimulation.

Des Weiteren wurde ein Lenkrad entwickelt, das mehrere Knopfreihe enthält (Abbildung 6-13 rechts). Diese Knöpfe können für taktile Eingaben vom Fahrer zur Automation und für visuelle Rückmeldungen von der Automation zum Fahrer frei verwendet werden. In dem umgesetzten Prototyp wurde ein roter Knopf für das Ablehnen und ein blauer für das Bestätigen

der Kooperation implementiert. Der rote Knopf ist der zweite von oben links und der blaue ist der zweite von oben rechts. Bei passivem Verhalten der Automation in der Arbitrierungsstrategie, wenn die Kooperation nicht notwendig ist, der Fahrer sie abgelehnt oder ignoriert hat, schimmern die Knöpfe in entsprechenden Farben. Bei Fragen in der Arbitrierungsstrategie blinkt der blaue Knopf und bei eingegangener Kooperation leuchtet er im kräftigen Blau.

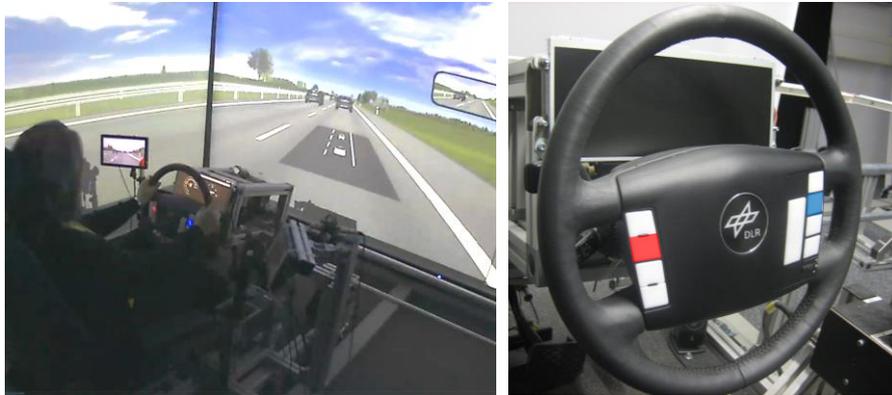


Abbildung 6-13: Versuchsanordnung für die Usability Studie für das Beispielsystem 2 (links) und die rot/blau leuchtenden Knöpfe am Lenkrad, die für Versuche mit visueller Modalität entwickelt wurden (rechts)

6.2.8 Usability Studie

Eines der Ziele der Usability Studie war die Beurteilung, ob das mit osGekoS entwickelte Interaktionsdesign von potenziellen Systemnutzern gut verstanden und akzeptiert wird. Die einleitende Erklärung zur Notwendigkeit und Nützlichkeit dieser Studie ist dieselbe, wie die für die Usability Studie für das Beispielsystem 1 (vgl. ersten Absatz in Abschnitt 6.1.8).

Die Probanden befuhren mit dem entwickelten Head-Up Display und dem speziellen Lenkrad mit leuchtenden Knöpfen eine zweispurige Autobahn im dichten Verkehr (vgl. Abbildung 6-8 und Abbildung 6-13). Der Startpunkt war auf der rechten Fahrspur. Nach dem Erleben des kooperativen Spurwechsels aus der Perspektive von Fahrzeug V1 auf die linke Fahrspur haben die Probanden einen weiteren kooperativen Spurwechsel aus der Perspektive von V2 erlebt. Danach wurden sie gebeten, wieder auf die rechte Fahrspur zu wechseln und erlebten die beiden Spurwechsel erneut. Diese dynamische Abfolge war als eine Versuchsfahrt definiert.

Nach dem zweckorientierten Usability Paradigma (Abschnitt 2.3.1) nahmen acht Probanden (4 Männer, 4 Frauen, 19-58 Jahre alt) mit unterschiedlicher Fahr- und Fahrerassistenzenerfahrung an der Studie teil. Nach zehnminütigem Training trafen die Probanden bei der ersten Versuchsfahrt ohne Systemwissen auf den Prototyp (Naiv Run). Sie fuhren vier kooperative Spurwechsel und beurteilten das System mittels semantischer Differentiale. Danach wurde das Systemwissen vermittelt und mit Checkfragen überprüft, ob der Prototyp verstanden wurde.

Die Probanden durften den Prototyp zu Trainingszwecken mehrmals ausprobieren. Danach folgte die zweite Versuchsfahrt (Hot Run) mit anschließendem Fragebogen. Gemessene Variablen waren die empfundenen Gefallen, Nützlichkeit, Nachvollziehbarkeit etc. Es wurde zwischen den eigenen Spurwechseln (Perspektive V1) und Spurwechseln anderer Verkehrsteilnehmer (Perspektive V2) unterschieden. Ein Auszug der Auswertung ist in der Abbildung 6-14 dargestellt. Es handelt sich um Medianwerte der aufgenommenen Probandenantworten, die auf eine Likert-Skala von einer alternierenden Skala (-3...3) übersetzt wurden.

Insgesamt zeigte sich, dass Probanden das System in der Gesamtbeurteilung sowohl nach Erstkontakt als auch nach einem Training konstant positiv bezüglich Gefallen, Nützlichkeit und Nachvollziehbarkeit bewerteten. Bereits ohne Systemwissen nach der ersten Versuchsfahrt konnten alle Probanden die Funktionalität des Prototyps richtig erklären. Als verbesserbar stellte sich heraus, dass die Probanden gerne gewusst hätten, mit welchem konkreten Fahrzeug sie kooperieren (3 von 8). 2 von 8 Probanden gaben einen wichtigen Hinweis darauf, dass der grüne Anweisungspfeil nach hinten als Bremsen interpretiert werden kann, wobei zum Aufmachen einer Lücke man nur vom Gas gehen bräuchte. „Ohne Lücken-anfrage hätte ich keine Lücke gemacht“ - sagte einer der Probanden, was auf eine erfolgreiche Aktivierung der Kooperation im gestalteten CS durch den Prototyp hindeutet.

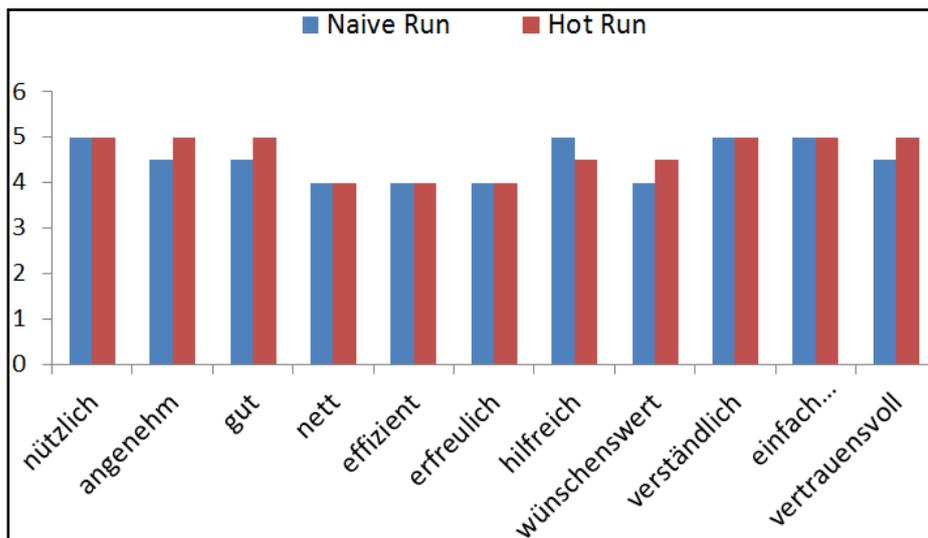


Abbildung 6-14: Befragungsergebnisse nach der ersten und der zweiten Fahrt des Prototyps für das Beispielsystem 2

6.2.9 Zusammenfassung

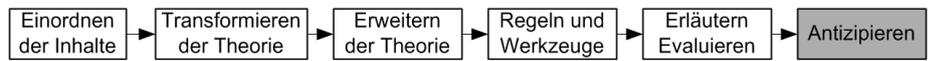
Das Assistenzsystem für Kooperation bei Spurwechseln wurde mittels der osGekoS-Konzepte, Regeln und Werkzeuge gestaltet. Während der Usability Studie wurde im Allgemeinen beobachtet, dass das gestaltete CS von den Probanden gut verstanden und mit gutem bis sehr gutem subjektiven Feedback angenommen wird. Dies wurde durch anschließende Datenauswertung bestätigt. Daraus lässt sich schließen, dass mit osGekoS akzeptable und gebrauchstaugliche CS im Bereich hochautomatisierter Fahrzeugführung gestaltet werden können.

6.3 Zusammenfassung der Evaluation und Erläuterung

In diesem Abschnitt wurde osGekoS mittels Anwendungsbeispiele aus dem Bereich hochautomatisierter Fahrzeugführung erläutert. Die in Kapitel 5 eingeführten Gestaltungsregeln und Werkzeuge wurden in der vorgeschlagenen Reihenfolge der Prozessschritte für die Gestaltung von zwei Fahrerassistenzsystemen angewendet. Die erläuterten und evaluierten Beispielasstanzsysteme decken exemplarisch den Raum primärer Fahraufgaben bei geteilter Fahrzeugkontrolle, wie die Längs- und Querführung des Fahrzeuges und die Koordination mehrerer Fahrzeuge untereinander. Dies deutet darauf hin, dass beliebige Fahrerassistenzsysteme mit osGekoS erfolgreich entwickelt werden können. Zusammen mit den integrierenden Konzepten der MPS, der Aktionstension und der Arbitrierung wird die Stärke von osGekoS auch au-

Berhalb des Automotivbereichs deutlich. Damit lassen sich komplexe kognitive Systeme vom abstrakten bis zu einem sehr konkreten, operativen Level in einer integrierten Form gestalten.

Die gute bis sehr gute Bewertung der gestalteten CS lassen vermuten, dass der verwendete Gestaltungsrahmen entsprechend gute Qualität besitzen kann. Das Ziel der Evaluation von osGekoS in dieser Arbeit war seine prinzipielle Anwendbarkeit für die operative Gestaltung gebrauchstauglicher Systeme mit hoher Akzeptanz der Nutzer im Bereich hochautomatisierter Fahrzeugsysteme. Bei vorliegender Datenlage und durch Erfahrungen in der Anwendung von osGekoS lässt sich behaupten, dass dieses spezielle Evaluationsziel erreicht wurde.



7 Zusammenfassung und Ausblick

Diese Arbeit begann mit der Forschungsfrage: *Wie kann die Struktur eines kognitiven Systems verstanden und beschrieben werden?* Mit gewonnenen Erkenntnissen kann als eine der möglichen Antworten auf diese Frage behauptet werden, dass mittels des MPS-Konzepts sich jedes CS strukturell verstehen und als Modell beschreiben lässt. CS können als eine Menge von kognitiven und nichtkognitiven Agenten (MPs) zerlegt werden. Die Agenten stehen miteinander in einer komplexen Wechselwirkung und können auf der Systemebene Konkurrenzen und Kooperationen generieren. Mit dem Konzept der Aktionstensionen lassen sich die internen Motivationsstrukturen der Agenten auf die CS-Ebene externalisieren. Prozesse der Konkurrenz und Kooperation können an einem solchen Modell untersucht werden. Operationalisierte Aktionstensionen, die die Organisation des CS beschreiben, können außerdem ein CoFoR und eine allgemeine Datenrepräsentation in Soft- und Hardware implementierter CS darstellen.

Als Forschungsbedarf in diesem Zusammenhang wird die Untersuchung generativer Eigenschaften noch komplexerer Systeme gesehen. Es ist vorstellbar, die mithilfe von osGekoS-Methoden erstellten System- und Wechselwirkungsmodelle für größere Populationen der Agenten zu simulieren. Damit könnte man eine Grundlage haben, um sowohl „lokale“ Effekte der Konkurrenz, Kooperation zwischen einzelnen Agenten als auch „globale“ Effekte auf der Ebene von Agentenpopulationen zu untersuchen. Für den Bereich der hochautomatisierten Fahrzeuge würde es bedeuten, dass mit diesen Grundlagen Konkurrenzen und Kooperationen innerhalb eines hochautomatisierten Fahrzeuges untersucht werden können, wie die Wechselwirkung zwischen dem Fahrer und der Automation bei geteilter Fahrzeugführung. Umfassendere Effekte könnten darauf folgend beschrieben und untersucht werden, wie die Wechselwirkung mehrerer kommunizierender Fahrzeuge, wie diese Fahrzeuge sich zu Verkehrsflüssen organisieren und wie die Verkehrsflüsse untereinander wechselwirken.

Die zweite Forschungsfrage war: *Wie kann eine operative Gestaltung an die Struktur eines kognitiven Systems angebunden werden?* Wenn die Strukturen des CS verstanden, zerlegt und modelliert sind, ist es möglich, diese Systeme mittels der gestaltungskompatiblen Architektur der CS und mittels der osGekoS-Methoden gestaltungstechnisch zu beeinflussen. Durch die auf sprachwissenschaftlichen Grundlagen basierende Gestaltungsschnittstelle der Architektur und das Konzept der Arbitrierung lassen sich die CS operativ beeinflussen, verändern oder neu gestalten. Für die CS können generische Arbitrierungsmodelle entwickelt werden, die nahtlos in verschiedene Simulationsumgebungen integriert werden können. Damit kann der operative Gestaltungsprozess formalisiert, generalisiert und dadurch insgesamt sehr effektiv und zielführend für die Entwicklung gebrauchstauglicher kognitiver Systeme sein.

Als Forschungsbedarf an dieser Stelle ist die weitere Evaluation von osGekoS zu sehen. Obwohl in dieser Arbeit bereits gezeigt wurde, dass mittels osGekoS umfassende und gut bewertete Systeme entwickelt werden können, sollte der Gestaltungsrahmen weiter evaluiert werden. So wurde bereits im Rahmen von EU-Projekt AdaptIVe während eines Systementwicklungsworkshops eine Expertenstudie durchgeführt, wo über 30 Experten von verschiedenen Forschungs- und Entwicklungsabteilungen der Automobilhersteller und ihrer Zulieferer zwei Tage lang mit einer etwas angepassten Version des osGekoS-Werkzeuges „Syntaktisches

Storyboard“ gearbeitet haben. Anschließende Bewertung des Werkzeugs mittels semantischer Differentiale gibt Hinweise darauf, dass das Werkzeug von den Experten gut verstanden, akzeptiert (vgl. Anhang D) und in ihren Arbeitsfluss übernommen wurde. Auch die Werkzeuge „Konfliktanalyse“ und „Arbitrierungsmatrix“ wurden in Projekten, wie EU-InteractIVe und EU-D3CoS, mit positiver Expertenrückmeldung eingesetzt. „Interaktionskontroller“ ist inzwischen zum Standardwerkzeug der Interaktionsentwicklung im Automotivbereich des Instituts für Verkehrssystemtechnik des DLR geworden. Nach und nach sollte jedes Werkzeug, Regel und Methode des osGekoS für sich allein und im Verbund weiterhin evaluiert werden und mit anderen Methoden und Werkzeugen des Instituts integriert werden.

Im Verlauf dieser Arbeit wurden Beispielasistenzsysteme entwickelt. Diese decken bereits einen umfassenden Teil der hochautomatisierten Fahrzeugführung ab, in dem sie sowohl die geteilte Längs- und Querführung als auch die Koordination hochautomatisierter Fahrzeuge abdecken. Gute Bewertungen von potenziellen Systemnutzern haben dazu beigetragen, dass diese Prototypen den Kern der hochautomatisierten Interaktionsprototypen am Institut für Verkehrssystemtechnik bilden. Sie werden in den Nachfolgeprojekten bereits intensiv weiter genutzt. Auch die entwickelte spezielle Hardware der Prototypen, wie die aktiven Pedale oder das spezielle Lenkrad mit leuchtenden Knöpfen, wurden und werden vielfach reproduziert und als Standardhardware des Instituts für Verkehrssystemtechnik eingesetzt.

Als Forschungsbedarf an dieser Stelle ist die weitere integrative Arbeit zu nennen. Die entwickelten und hier vorgestellten Interaktionsprototypen sollen im nächsten Schritt miteinander und mit einer Reihe anderer Prototypen zu größeren Arbitrierungsmodellen und Interaktionsprototypen zusammengeführt und in Usability Studien evaluiert werden. Die Entwicklung und die Erforschung weiterer neuartiger und integrativer Hardware sind ebenfalls geplant. Im Projekt AdaptIVe^[SPR14] beispielsweise, werden die hier vorgestellten Interaktionsprototypen auf die Hardware eines den Fahrer umgebenden visuellen akustischen und haptischen Displays angepasst (multimodal ambient display). Dabei ist es systemergonomisch anvisiert, in schnellere Reaktionszeiten der Fahrer und intuitivere Bedienkonzepte hochautomatisierter Fahrzeuge vorzudringen. Durch die Nutzung neuartiger Interaktionskonzepte, die Reflexe, Affekte und Emotionen der Fahrer mitnutzen, können noch intuitivere und weiterhin gebrauchstaugliche Interaktionsprototypen erforscht und entwickelt werden. Die auf MPS, Aktionstensionen und Arbitrierung bauende osGekoS-Theorie ist bestens dafür vorbereitet.

Insgesamt löst der vorgestellte Gestaltungsrahmen osGekoS nicht alle Probleme der Gestaltung von kognitiven Systemen. Dennoch stellt er insgesamt ein sehr strukturiertes auf der einen Seite und ein offenes „Ökosystem“ auf der anderen Seite zur Verfügung. Durch die Strukturierung kann der Systemingenieur bereits jetzt viele Gestaltungsfehler, wie „sich in der Komplexität der Gestaltung verlieren“ oder „Willkür der menschlichen Entscheidung aus den Augen verlieren“ meiden. Durch die Offenheit macht osGekoS möglich, in dieser Arbeit vorgestellte theoretische Konzepte, Regeln und Werkzeuge weiter zu entwickeln. Die semiformale Semantik von osGekoS kann weiter formalisiert werden, so dass im nächsten Schritt die Entwicklung und der Einsatz von automatisierten semantischen Interpretern möglich wäre. Die Regeln und Werkzeuge können ebenfalls weiter formalisiert und integriert werden.

8 Anhang

8.1 Anhang A: Wissensrepräsentation mit Concept-Maps

Es ist eine Herausforderung, komplexes Wissen effizient begreifen, reduzieren, weiter entwickeln und vermitteln zu können. Dies ist nur mit einer geeigneten Methode der Wissensrepräsentation möglich. Zielführend kann eine Wissensrepräsentation sein, die intuitiv verständlich ist und eine Komplexitätsreduktion ohne einen Verlust der Realitätsanbindung erlaubt.

Beim Beantworten komplexer Forschungsfragen gibt es die Herausforderung der Ressourcenbegrenztheit. Dabei ist keine alles umfassende Behandlung der Komplexität der realen Welt möglich. Diesem Umstand kann mit dem Modellieren und Konzeptualisieren begegnet werden, wobei die reale Welt auf ein für die Beantwortung der jeweiligen Frage ausreichendes Maß reduziert wird. Konzepte werden im nächsten Schritt in Theorien eingebettet. Eine gute Wissensrepräsentation sollte also ermöglichen, auf der Ebene der Konzepte zu arbeiten.

Beim Modellieren und Theoretisieren verliert man einen Teil der Information, die insbesondere für eine ganzheitliche Betrachtung der realen Welt essenziell sein kann. Extremfall davon ist wenn ein Konzept oder ein Modell nach der Reduktion der Komplexität unbrauchbar ist. Um auch diesem Umstand zu begegnen, muss eine adäquate Wissensrepräsentation das reduzierte (modellierte) und das komplexe (realitätsnahe) Wissen möglichst ausbalancieren.

Wenn man alle genannten Anforderungen an die Wahl der Wissensrepräsentation zusammenführt, so liegt die Entscheidung nahe, *Concept-Maps* als geeignete Wissensdarstellung zu wählen^[NoC06]. Sie ist ein Derivat der semantischen Netze, deren Diskussion bis ins Jahr 1906 zurückgeht. Schon damals hatte Charles S. Peirce eine grafische Notation von Knoten und Kanten vorgeschlagen, die er "existential graphs" und "the logic of the future" nannte^[Ros73].

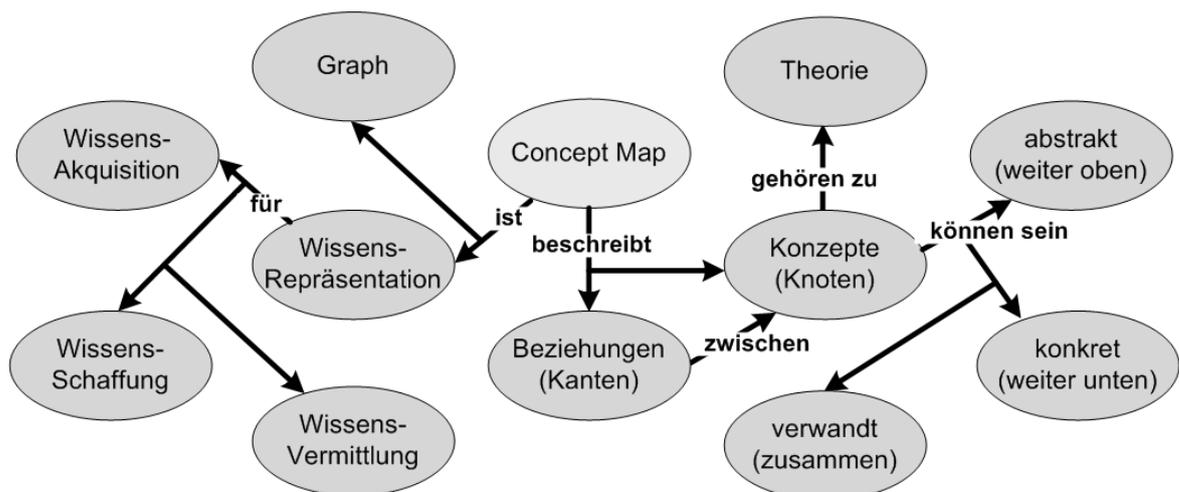


Abbildung 8-1: Concept-Map mit der Repräsentation des Wissens aus diesem Anhang A

Eine Concept-Map (Abbildung 8-1) ist ein Graph, dessen Knoten die jeweiligen Konzepte (bzw. Modelle, Theorien) beschreiben. Zwischen den Knoten befinden sich semantische Verknüpfungen, die die Konzepte in Beziehung zueinander setzen. Sinngemäß näher beieinander liegende Konzepte liegen im Diagramm ebenfalls näher beieinander. Hierarchisch übergeord-

nete oder abstraktere Konzepte werden weiter oben im Graph platziert. Konkretere Konzepte finden weiter unten Platz. Eine Concept-Map enthält mehrere Wissensstränge, wo nahezu beliebige Pfade durch Knoten und Kanten gelegt werden können. Durch eine geschickte Filterung und Anordnung der Knoten und Kanten bei der Diagrammerstellung wird daraus eine gleichzeitige Abbildung des gesamten relevanten Wissens.

Mit ihrer Idee erfüllen die Concept-Maps sehr gut den Nutzen, die Komplexität des akquirierten Wissens zu reduzieren und unter Kontrolle zu halten, das Wissen für die Informationsvermittlung zugänglich darzustellen und neues Wissen zu generieren. Durch die Objektivierung der Konzepte als Knoten hat man einen Zugang zur Wissensdarstellung auf einer abstrakten, reduzierten Konzeptebene. Gleichzeitig wirkt eine Concept-Map der Wissensreduktion durch die semantische Verknüpfung der Konzepte entgegen. Des Weiteren gibt es von den Concept-Maps aus einen Weg in Richtung der Formalisierung der Wissensrepräsentation, z.B. mithilfe von formalen Ontologien und OWL^[W313a].

In dieser Arbeit stellen die Concept-Maps das Mittel der Wahl für die Darstellung des relevanten komplexen Wissens. Die im Original vorgeschlagenen Knoten und Kanten werden genauso beibehalten. Allerdings werden die wichtigsten Knoten (Konzepte) direkt in der Concept-Map ausführlicher gezeichnet. Dies beginnt mit der Wahl verschiedener Farben für gleiche oder ähnliche Konzepte und geht bis hin zu den ausführlich gezeichneten Inhalten der Konzepte innerhalb deren Knoten. Dies gestattet die Stärke der Concept-Maps und die Stärke der visuellen Veranschaulichung der Konzepte selbst miteinander zu verbinden und das Wissen noch effektiver darzustellen und zu vermitteln.

8.2 Anhang B: Beschreibungsmittel für kognitive Systeme

Im Kontext der Gestaltung kognitiver Systeme muss man sich mit einem geeigneten Beschreibungsmittel dafür beschäftigen. Die Wahl des Beschreibungsmittels hängt davon ab, um welche Art der Systeme und um welche Art der Gestaltung es im Einzelnen geht. Basierend auf Vorerfahrungen des Autors im Bereich der CS-Gestaltung kamen für die Formalisierung von kognitiven Systemen zwei miteinander verwandte Arten der graphentheoretischen Beschreibungsmittel in die engere Wahl: Petri-Netze^[Scr92] und synchrone Automaten.

Die subjektive Bewertung der Eignung der beiden Beschreibungsmittel auch für die Ziele der vorliegenden Arbeit ist in der Tabelle 8-1 zusammengefasst. Als Bewertungskriterien wurden die Metriken Mächtigkeit, Fokus, Zugänglichkeit, Vorerfahrung und Probeeinsatz definiert. Nach subjektiver Wertung mittels dieser Metriken auf der Skala (+, o, -) für (geeignet, neutral, ungeeignet) wurde eine Entscheidung zugunsten der synchronen Automaten getroffen.

Tabelle 8-1: Subjektiver Vergleich der Konzepte Petri-Netze und Automaten auf ihre Eignung für die Formalisierung der kognitiven Systeme im Rahmen dieser Arbeit

Konzept/ Metrik	Petri-Netze	Synchrone Automaten
Mächtigkeit	...sind sehr mächtig. Durch explizit modellierbare Transition sind die Modellabläufe und Nebenläufigkeiten gut beobachtbar und beschreibbar. (+)	...sind weniger mächtig. Implizit modellierbare Transitionen „verstecken“ die Modellabläufe und Nebenläufigkeiten (-)
Fokus	...haben eher einen prozess- und ablauforientierten Fokus, was weniger der Strukturfokussierung dieser Arbeit entspricht (o)	...haben eher einen struktur-orientierten, generativen Fokus, was den Hauptideen dieser Arbeit entspricht (+)
Zugänglichkeit	...sind mittelschwer im Einsatz. Konzentration auf physikalisch anmutende Entitäten (Marken) macht den Einsatz weniger Zugänglich für „schnelle Modellierung“ struktureller Zusammenhänge (o)	...sind einfach im Einsatz. Konzentration auf einfache algebraische Umformungen machen den Einsatz Zugänglich für „schnelle Modellierung“ struktureller Zusammenhänge (+)
Vorerfahrung	Vorerfahrungen des Autors waren vorhanden (o)	Vorerfahrungen des Autors waren vorhanden (o)
Probeeinsatz	...haben nach zwei Tagen Probeeinsatz zu keinem zufriedenstellenden Ergebnis geführt (-)	...haben nach einem Tag Probeeinsatz zum zufriedenstellenden Ergebnis geführt (+)
Ergebnis	o	++

Wie bereits erwähnt, ist die Entscheidung des Autors zugunsten der synchronen Automaten als Beschreibungsmittel für CS einer eher subjektiven Natur. Zudem liegt das Konzept der Automaten und der Petri-Netze nah beieinander, so dass es für den interessierten Systemingenieur möglich sein sollte, zwischen diesen beiden oder gar zu einem ganz anderen mathematischen Beschreibungsmittel zu wechseln.

8.3 Anhang C: Zur grafischen Nomenklatur in Abbildungen

Bei der Darstellung komplexer Inhalte, wie sie in dieser Arbeit vorliegen, muss man sich mit der geeigneten Art der Wissensrepräsentation (vgl. Anhang A) und mit der Auswahl eines geeigneten Beschreibungsmittels (vgl. Anhang B) beschäftigen. In die gleiche Kategorie der Überlegungen gehört auch die Auswahl bzw. Schaffung einer passenden grafischen Nomenklatur für Abbildungen. Drei Hauptkriterien wurden dabei als wichtig erachtet: Einfachheit der Formgebung, Farbgebung und Erweiterbarkeit.

Zur Einfachheit der Formgebung: eine einfache Formgebung ist leicht zu verstehen und anzuwenden, was für Abbildungen im komplexen Wissensumfeld einen großen Vorteil darstellt. Die aus der Literatur bekannten grafischen Nomenklaturen, wie UML, wären bereits auf Einfachheit der Formgebung optimiert. Dennoch wurde vom Autor entschieden nur einen geringen Teil davon zu verwenden: kreisförmige Knoten, gerichtete Pfeile als Kanten und sehr kurzen Text als Kennzeichnungen (Abbildung 8-2). Dies liegt darin begründet, dass die Inhalte der vorliegenden Arbeit mit den genannten grafischen Elementen bereits ausreichend gut vermittelt werden können und keine höhere Komplexität der Formgebung nötig ist.

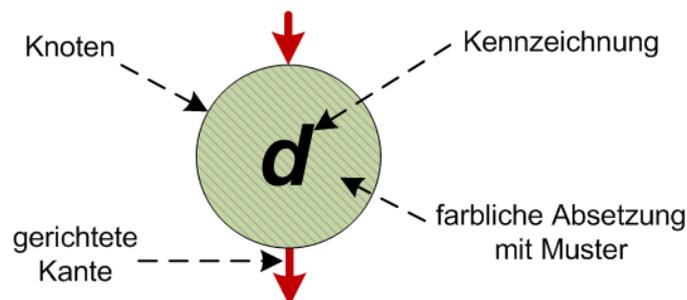


Abbildung 8-2: Beispiel der Form- und Farbgebung der grafischen Nomenklatur der vorliegenden Arbeit

Zur Farbgebung: Entgegen der in der Fachliteratur üblichen monochromen Art der grafischen Darstellung wurde entschieden, mit einer geringen Auswahl an Farben zu arbeiten. Der Hauptvorteil einer farblosen Darstellung läge darin, dass die Grafiken auf einem Schwarzweißdruck gut lesbar wären. Diese Anforderung wird in der vorliegenden Arbeit durch Mustergebung erreicht. Der wichtige Vorteil der sonst verwendeten Farbe ist die Möglichkeit, unterschiedliche zusammenhängende Inhalte innerhalb einer Abbildung voneinander abzusetzen, so dass komplexe Abbildungen durch Nutzung der Farbe insgesamt besser lesbar sind.

Zur Erweiterbarkeit: Die gewählte und bereits beschriebene Nomenklatur ist für den Bedarf der Inhalte der vorliegenden Arbeit gut erweiterbar. Die kreisrunden Knoten können z.B. Teile anderer Knoten sein oder selbst Knoten beinhalten. Angesichts der inhaltlich angestrebten Darstellung der Selbstständigkeit der durch solche Knoten dargestellter Entitäten (vgl. Abschnitt 3.4) erscheint die runde Form dafür sehr geeignet.

Insgesamt lässt sich sagen, dass die Wahl der grafischen Nomenklatur ein längerer Prozess war. Die Nomenklatur hat sich mit der Erarbeitung und dem Fortschritt der Inhalte der Arbeit ebenfalls gewandelt, um am Ende dieses Prozesses die beschriebene Form anzunehmen.

8.4 Anhang D: Evaluation des Syntaktischen Storyboards

Im Rahmen eines Workshops für das EU-Projekt AdaptVe^[SPR14] wurde unter anderem die Evaluation der Methode und einer noch ablaufbasierten (statt tensionsbasierten) Vorstufe des Werkzeuges Syntaktisches Storyboard (vgl. Abschnitt 5.5) durchgeführt. Die Beschreibung und die Ergebnisse dieser Evaluation sind im Folgenden dargestellt.

An dem o.g. Workshop haben ca. 30 HMI- und Fahrzeugfunktionsexperten aus verschiedenen europäischen Ländern teilgenommen. Das Ziel des Workshops war, im Kontext des hochautomatisierten Fahrens kombinierte Fahrzeugverhaltens- und HMI-Abläufe für bereits vordefinierte Szenarien zu entwickeln. Des Weiteren sollten für die definierten Hauptszenarien eine agentenbasierte Dekomposition durchgeführt und alternative Szenarien überlegt werden. Sowohl für Hauptszenarien als auch für entwickelte alternative Szenarien sollten mögliche Fahrzeugverhaltens- und HMI-Abläufe als Syntaktische Storyboards dokumentiert werden.

Die Methode und das Werkzeug „Syntaktisches Storyboard“ wurden in einer ca. zehnminütigen Präsentation eingeführt und alle Workshop-Teilnehmer wurden danach in Gruppen eingeteilt (ca. 5-6 Personen pro Gruppe). Allen Gruppen lagen Muster mit Syntaktischen Storyboards (vgl. Abbildung 8-3 links) in Papierform und teilweise als Projektion auf Whiteboards vor (vgl. Abbildung 8-3 rechts).

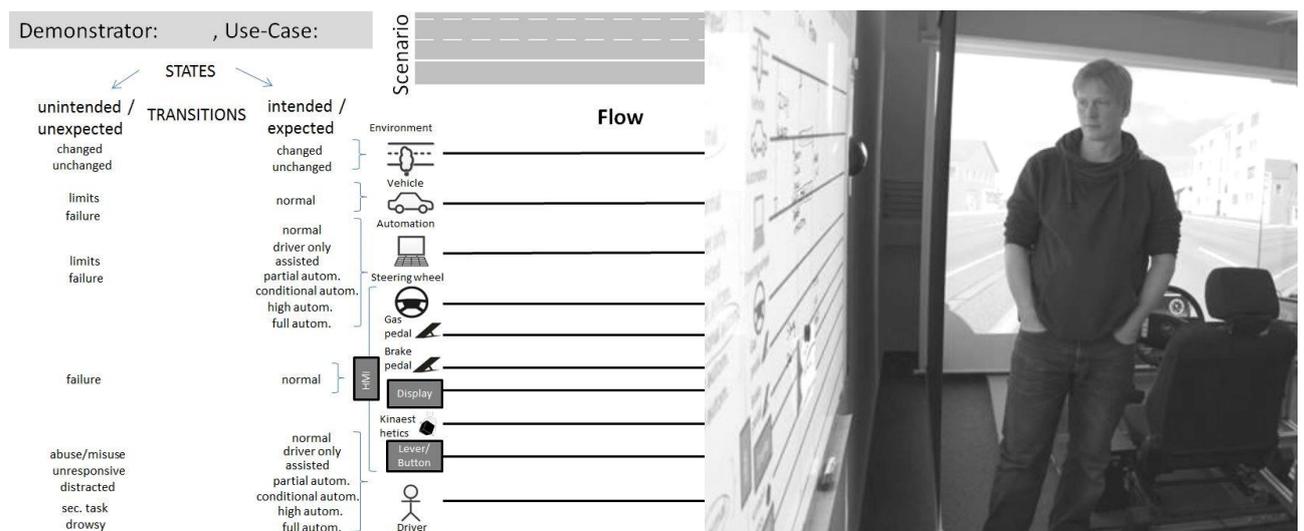


Abbildung 8-3: Muster eines Syntaktischen Storyboards (links) und ein Workshop-Teilnehmer bei der Arbeit (rechts)

Im ersten Schritt haben die Teilnehmer die auf dem Muster vorgeschlagene Systemdekomposition an die jeweiligen Szenarien und Assistenzfunktionen angepasst. Wenn es z.B. um ein haptisches Fahrerassistenzsystem ging, dann wurde die haptische Dimension in mehrere Agenten aufgeteilt. Mit dem Ergebnis wurden die Muster der Syntaktischen Storyboards angepasst. Im zweiten Schritt wurden die vorliegenden Szenarien vervielfältigt, so dass am Ende pro Hauptszenario ein „Standardablauf“, in dem es keine Fehler seitens der Automation und keine Fehlbedienung seitens des Fahrers gibt, und mehrere fehlerbehaftete „Alternativabläufe“ vorlagen. Für jedes Haupt- und Alternativszenario wurden im dritten Schritt Syntaktische Storyboards entwickelt und entweder in Papierform oder als Fotografie dokumentiert. Zum Ende des Workshops wurden die Ergebnisse im vierten Schritt gruppenintern zusammengefasst und dann gruppenübergreifend präsentiert, diskutiert und homogenisiert.

Nach dem Workshop wurde an die Teilnehmer ein Fragebogen verteilt, der die Bewertung der Gebrauchstauglichkeit der im Workshop benutzten und oben beschriebenen Methode sowie des Werkzeuges Syntaktisches Storyboard zum Ziel hatte (vgl. Abbildung 8-4).

neutral

useful	<input type="radio"/>	useless							
comfortable	<input type="radio"/>	uncomfortable							
bad	<input type="radio"/>	good							
nice	<input type="radio"/>	annoying							
efficient	<input type="radio"/>	needless							
angrily	<input type="radio"/>	enjoyable							
valuable	<input type="radio"/>	valueless							
undesirable	<input type="radio"/>	desirable							
understandable	<input type="radio"/>	not understandable							

Abbildung 8-4: Fragebogen zur Bewertung der Gebrauchstauglichkeit des Werkzeuges Syntaktisches Storyboard

Die Auswertung der Ergebnisse ist in der Tabelle 8-2 zusammengefasst.

Tabelle 8-2: Auswertung der Ergebnisse der Methoden- und Werkzeugbewertung Syntaktisches Storyboard

Methode	N	Min	Max	Mittelwert	STD	Werkzeug	N	Min	Max	Mittelwert	STD
nützlich	24	3	7	6,00	1,103	nützlich	24	3	7	6,04	1,083
komfortabel	24	3	7	5,58	1,176	komfortabel	24	3	7	5,63	1,056
gut	22	4	7	5,95	0,950	gut	22	3	7	5,95	1,090
nett	23	3	7	5,61	1,076	nett	23	3	7	5,78	1,166
effizient	23	4	7	5,70	0,876	effizient	24	3	7	5,87	1,116
spaßig	22	3	7	5,64	1,432	spaßig	22	3	7	5,86	1,125
wertvoll	23	3	7	5,83	1,072	wertvoll	23	3	7	5,91	1,041
wünschenswert	22	3	7	5,73	1,032	wünschenswert	22	3	7	5,73	1,202
verständlich	24	3	7	5,75	1,327	verständlich	24	3	7	6,21	1,062

Aus den vorliegenden Ergebnissen und aus der persönlichen Beobachtung während des Workshops lässt sich zusammenfassen, dass die überwiegende Mehrheit der Teilnehmer gut bis sehr gut sowohl die Methode als auch das Werkzeug Syntaktisches Storyboard angenommen hatte. Im Mittel hielten die Teilnehmer sowohl die Methode als auch das Werkzeug für gut (5,95/7; 5,95/7) und nützlich (6,00/7; 6,04/7). Insbesondere das Werkzeug wurde als sehr verständlich bewertet (6,21/7). Weil diese Methode und das Werkzeug wichtige Bestandteile von osGekoS sind, lässt sich zum in der Arbeit erbrachten indirekten Nachweis der Gebrauchstauglichkeit von osGekoS durch die gute Bewertung damit entworfener Systeme nun auch ein direkter Nachweis der Gebrauchstauglichkeit eines Teils von osGekoS hinzufügen.

Abbildungsverzeichnis

Abbildung 1-1: Komplexe Beziehung innerhalb eines (kognitiven) Fahrer/ACC/Fahrzeug/Verkehr-Systems	2
Abbildung 1-2: Concept-Map mit grundsätzlichen Komponenten einer operativen Systemgestaltung .	4
Abbildung 1-3: osGekoS ist ein Gestaltungsrahmen für kognitive Systeme	6
Abbildung 1-4: Ergebnisse der vorliegenden Arbeit zeigen den Weg von der Theorie zur Praxis von osGekoS	8
Abbildung 2-1: Einordnung und relative Positionierung des operativen strukturfokussierten Gestaltungsrahmens für kognitive Systeme (osGekoS) aus dieser Arbeit zu den existierenden Gestaltungsrahmen	9
Abbildung 2-2: V-Diagramm: Systemgestaltungsablauf in Systems Engineering (adaptiert nach [USD07])	12
Abbildung 2-3: Sequenzdiagramm (links) und Klassendiagramm (rechts) nach UML	15
Abbildung 3-1: Beispiel eines Entscheidungsvorgangs in einem generischen kognitiven System	31
Abbildung 3-2: Modell eines einfachen kognitiven Multiagentensystems bestehend aus einer Population zweier interagierender kognitiven Agenten (d und c). Interaktion ist dabei ein Austausch von Aktionen	34
Abbildung 3-3: Dekomposition eines hochautomatisierten Fahrzeuges als kognitives MAS	35
Abbildung 3-4: Kanalisierung der semantisch aufgetrennten Information zwischen den CAs	37
Abbildung 3-5: Äußere Struktur eines Agenten d als MP mit IMs, OMs sowie Ereignissen, die diese Teilstrukturen verbinden. Abfolgen von IMs und OMs auf der CS-Ebene sind jeweils Aufgaben und Handlungen	39
Abbildung 3-6: IMs und EMs des d generieren ein aktives MP, das seine OM nach außen sendet.....	40
Abbildung 3-7: Struktur eines einfachen CS: d und c führen eine gemeinsame Aufgabe aus und interagieren, was sich als gegenseitige Manipulation zwischen den aktiven MPs der beiden CAs verstehen und beschreiben lässt	41
Abbildung 3-8: Struktur eines CS bestehend aus zwei CAs (d und c): IMs, OMs und EMs der Agenten können insgesamt zur positiven (+), unbestimmten (o) oder negativen (-) Wechselwirkung auf der CS-Ebene führen. Diese Wechselwirkung bestimmt das CS-Verhalten und lässt sich durch MP-Transitionen modifizieren	42
Abbildung 3-9: Dekomposition eines hochautomatisierten Fahrzeuges als Zustandsübergangsgraph.	45
Abbildung 4-1: Dekomposition eines hochautomatisierten Fahrzeuges als Zustandsübergangsgraph.	54
Abbildung 4-2: Zeithorizont, Zeitkonstanten und Effekte der Beeinflussung des CA-Verhaltens können durch eine Gruppierung der Begriffe der Konzepte der kognitiven Informationsverarbeitung ermittelt werden	55
Abbildung 4-3: Graph einer Fahrer/Automation/Fahrzeug -Systemkonfiguration bei geteilter Kontrolle	57

Abbildung 4-4: Concept-Map zum Zusammenhang zwischen der psychologischen Kraft, der Aufforderung, der Aktionstension und der Manipulation eines „langsamen Agenten“. Blaue ovale sind agentenintern, orangen sind extern	58
Abbildung 4-5: Beispiel eines auf einer Aktionstension basierenden Geschwindigkeitsassistenten	59
Abbildung 4-6: Erweiterung des MPS-Konzeptes durch die Aktionstension. Aktionstension kann die MP-Generierung bzw. -Aktivierung und damit die OMs aller Agenten, sowie ihre Interaktion zeitlich untereinander abstimmen.....	60
Abbildung 4-7: Zustandsübergangsgraph des Fahrer/Automation/Fahrzeug/Umwelt-Systems modifiziert nach der Anwendung des Aktionstensionskonzeptes	65
Abbildung 4-8: Erweiterung des MPS-Konzeptes durch die Arbitrierung. Der Arbitrer nutzt die Aktionstension und kann auf die CAs wirken, damit sie die ihre OMs für die positive Wechselwirkung zeitlich und qualitativ abstimmen.....	68
Abbildung 4-9: Zwei grundlegende Arbitrierungsstrategien: Konfliktlösung und Konfliktprävention ^[GrK12]	70
Abbildung 4-10: Zustandsübergangsgraph eines Fahrer/Automation/Fahrzeug-Systems modifiziert nach der Anwendung des Aktionstensions- und Arbitrierungskonzeptes	76
Abbildung 4-11: Generische gestaltungskompatible Architektur kognitiver Systeme	80
Abbildung 4-12: Informationsfluss zwischen dem Systemingenieur/Arbitrer und dem Systemnutzer ...	83
Abbildung 4-13: Schnittstelle zur gestaltungskompatiblen Architektur der kognitiven Systeme über ein gestaltbares Arbitrierungsmodell (links). Wahrheitstabellen für AND und OR der ternären Logik (rechts)	84
Abbildung 5-1: Werkzeug zur Lösungsexploration, mit dem Entscheidungskonvergenzen innerhalb des CS und damit das CS-Verhalten exploriert werden können. Es ist in MS-Excel umgesetzt.....	89
Abbildung 5-2: Syntaktisches Storyboard für den Konflikt an einer Weggabelung beschreibt den Interaktions- und Verhaltensfluss der Agenten des CS. Das Werkzeug ist in MS-Visio umgesetzt.....	92
Abbildung 5-3: Arbitrierungswerkzeug für den Konflikt an einer Weggabelung implementiert die Arbitrierungsstrategie und -signale. Das Werkzeug ist mittels C++ und FLTK-Bibliothek umgesetzt ...	94
Abbildung 5-4: Aufbau des Interaktionskontrollers bildet die gestaltungskompatible CS-Architektur für den Bereich hochautomatisierter Fahrzeugführung nach. Sie ist als Software-Framework „Interaktionskontroller“ mittels C++ und des Software-Modellierungswerkzeuges BOUML 4.2.3 Patch 6 ^[Pas12] umgesetzt	96
Abbildung 6-1: Simulationsansicht und die Dekomposition des Szenarios für das Beispielsystem 1 ..	100
Abbildung 6-2: Zustandsübergangsgraphen des Szenarios für das Beispielsystem 1	101
Abbildung 6-3: Operationalisierung der Aktionstension für das Beispielsystem 1.....	103
Abbildung 6-4: Auszug der Auswertungsergebnisse für deduktiv operationalisierte longitudinale Aktionstension	104
Abbildung 6-5: Arbitrierungsstrategie für das Beispielsystem 1 im Syntaktischen Storyboard	105

Abbildung 6-6: Versuchsanordnung für die Usability Studie für das Beispielsystem 1 und die aktiven Pedale, die speziell für die Versuche mit haptischer Modalität entwickelt wurden	106
Abbildung 6-7: Ergebnisse der subjektiven Befragung nach der ersten (naiven) und der zweiten (nach Training) Fahrt des Prototyps für das Beispielsystem 1 in zwei unterschiedlichen Automationsgraden	107
Abbildung 6-8: Simulationsansicht und die Dekomposition des Szenarios für Beispielsystem 2	109
Abbildung 6-9: Zustandsübergangsgraphen des Szenarios für das Beispielsystem 2	111
Abbildung 6-10: Operationalisierung der Aktionstension für das Beispielsystem 2	113
Abbildung 6-11: Auszug der Auswertungsergebnisse für deduktiv operationalisierte kooperative Aktionstension	114
Abbildung 6-12: Arbitrierungsstrategie für das Beispielsystem 2 im Syntaktischen Storyboard	115
Abbildung 6-13: Versuchsanordnung für die Usability Studie für das Beispielsystem 2 (links) und die rot/blau leuchtenden Knöpfe am Lenkrad, die für Versuche mit visueller Modalität entwickelt wurden (rechts)	117
Abbildung 6-14: Befragungsergebnisse nach der ersten und der zweiten Fahrt des Prototyps für das Beispielsystem 2	118
Abbildung 8-1: Concept-Map mit der Repräsentation des Wissens aus diesem Anhang A	123
Abbildung 8-2: Beispiel der Form- und Farbgebung der grafischen Nomenklatur der vorliegenden Arbeit	127
Abbildung 8-3: Vorlage eines Syntaktisches Storyboards (links) und ein Workshop-Teilnehmer bei der Arbeit	129
Abbildung 8-4: Fragebogen zur Bewertung der Gebrauchstauglichkeit des Werkzeugs Syntaktisches Storyboard	130

Tabellenverzeichnis

Tabelle 2-1: Bewertung der um osGekoS liegenden Gestaltungsrahmen	11
Tabelle 2-2: Gemeinsamkeiten und Unterschiede zwischen Objekten und Agenten	17
Tabelle 2-3: Nutzer- und menschenzentrierte Gestaltungsansätze innerhalb des Usability Engineering und des CSE	22
Tabelle 2-4: Vergleich zwischen menschlichen und animalischen Ausdrucksformen	28
Tabelle 4-1: Mögliche Probleme und Konflikte innerhalb eines kognitiven Systems	72
Tabelle 4-2: Mögliche Lösungen für Probleme und Konflikte innerhalb eines kognitiven Systems	72
Tabelle 4-3: Restrukturierungsvarianten des Modells eines hochautomatisierten Fahrzeuges, erwartete Entscheidungskonvergenzen, entsprechende Anfangsparameter, Zeitreihenentwicklungen und ihre Interpretationen	77
Tabelle 4-4: Beispiele von Arbitrierungsstrategien und -signalen für die drei Ebenen der Fahrzeugführung	78
Tabelle 5-1: Konfliktanalysewerkzeug des osGekoS mit dem eingetragenen Beispiel des Kontrollkonfliktes an einer Weggabelung beim Durchfahren in einem hochautomatisierten Fahrzeug. Es ist in MS-Excel umgesetzt	88
Tabelle 5-2: Auszug aus dem Werkzeug „Semantisches Mapping“ haptischer Signale für die Gestaltung hochautomatisierter Fahrzeugführung. Das Werkzeug ist in MS-Excel umgesetzt	90
Tabelle 5-3: Arbitrierungsmatrix mit vier Arbitrierungsstrategien für unterschiedlich starke Kontrollkonflikte zwischen Fahrer und Automation beim Durchfahren einer Weggabelung. Das Werkzeug ist in MS-Excel umgesetzt	91
Tabelle 5-4: Zusammenfassung des osGekoS Gestaltungsprozesses, dazugehöriger Werkzeuge und Regeln	98
Tabelle 6-1: Auszug aus Konfliktanalysewerkzeug für das Beispielsystem 1	100
Tabelle 6-2: Restrukturierungsvarianten des Modells für die Querführung des Beispielsystems 1	102
Tabelle 6-3: Auszug aus Konfliktanalysewerkzeug für das Beispielsystem 2	109
Tabelle 6-4: Auszug aus dem Werkzeug „Semantisches Mapping“ visueller Signale für die Gestaltung hochautomatisierter Fahrzeugführung. Gezeigt ist der Symbolsatz für das Beispielsystem 2	110
Tabelle 6-5: Restrukturierungsvarianten des Modells des Beispielsystems 2, erwartete Entscheidungskonvergenzen, entsprechende Anfangsparameter, Zeitreihenentwicklungen und ihre Interpretationen	111
Tabelle 8-1: Subjektiver Vergleich der Konzepte Petri-Netze und Automaten auf ihre Eignung für die Formalisierung der kognitiven Systeme im Rahmen dieser Arbeit	125
Tabelle 8-2: Auswertung der Ergebnisse der Methoden- und Werkzeugbewertung Syntaktisches Storyboard	130

Abkürzungsverzeichnis

ACC	Adaptive Cruise Control
AEBS	Advanced Emergency Braking System
C++	Allgemeine objektorientierte Programmiersprache
CA	Cognitive Agent
CF	Cognitive Function
CFA	Cognitive Function Analysis
CoFoR	Common Frame of Reference
CS	Cognitive System
CSE	Cognitive Systems Engineering
CWA	Cognitive Work Analysis
EEG	Elektroenzephalografie
EM	Eigen-Manipulation
FLTK	Fast Light Toolkit
GOMS	Goals, Operators, Methods and Selection rules
GUI	Graphical User Interface
HMI	Human-Machine Interface
HTA	Hierarchical Task Analysis
IM	Input-Manipulation
LDW	Lane Departure Warning
LKA	Lane Keeping Automation
MAS	Multiagenten System
MP	Manipulations-Potenzial
MPS	Manipulations-Potenzial Systeme
MRT	Multiresourcen Theorie

NA	Nichtkognitiver Agent
OM	Output-Manipulation
OsGekoS	Operativer strukturfokussierter Gestaltungsrahmen für kognitive Systeme
OWL	Web Ontology Language
RECA	Rear End Collision Avoidance
TTC	Time-To-Collision
UAV	Unmanned Aerial Vehicle
UML	Unified Modelling Language
XML	Extensible Markup Language

Literaturverzeichnis

- [ACN13] ACEA - European Automobile Manufacturers' Association (2013): Code of Practice for the Design and Evaluation of ADAS. http://www.acea.be/images/uploads/files/20090831_Code_of_Practice_ADAS.pdf. Zugriff: 26.07.2013
- [ADE13] ADOBE (2013): *Adobe Creative Cloud*. <http://www.adobe.com/de/products/creativecloud.html>. Zugriff: 26.07.2013
- [ALA13] ALTIA Design (2013): *The GUI Software for Embedded Systems*. http://www.altia.com/products_design.php. Zugriff: 27.07.2013
- [All77] Alexander, C.; Ishikawa, S.; Silverstein, M.; Jacobson, M.; Fiksdahl-King, I; Angel, S (1977): *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press.
- [AnM97] Anderson, J.R.; Matessa, M.; Lebiere, C. (1997): *ACT-R: A Theory of Higher Level Cognition and its Relation to Visual Attention*. *Human-Computer Interaction*,12,S.439-462
- [Bad93] Barfield, L. (1993): *The User Interface: Concepts & Design*. Addison-Wesley
- [BaM01] Bauer, B.; Müller, J.P.; Odell, J. (2001): *Agent UML: A Formalism for Specifying Multiagent Software Systems*. *International Journal of Software Engineering and Knowledge Engineering*, Vol. 11, No. 3, S. 1-24
- [Blh12] Bloch, A. (2012): *Meinung Assistenzsysteme: Über hyperaktive Elektronik-Helfer*. *Auto Motor und Sport*. Heft 21/2012. <http://www.auto-motor-und-sport.de/news/meinung-assistenzsysteme-ueber-hyperaktive-elektronik-helfer-5721582.html>. Zugriff: 22.07.2013
- [Boy00] Boy, G (2000): *Using Cognitive Function Analysis to Prevent Controlled Flight Into Terrain*. In *Human Factors and Flight Deck Design Book*, Hashgate Pub.
- [Boy94] Boy, G. (1994): *Interface Agents for handling fuzzy descriptors in information retrieval*. 5th Int. Conference on Processing and Management of Uncertainty in Knowledge-Based Systems, Paris, France, July 4-8, 1998
- [Boy98] Boy, G. (1998): *Cognitive Function Analysis for Human-Centered Automation of Safety-Critical Systems*. *Conference on Human Factors in Computing Systems*, Los Angeles, California, USA, Apr. 18-23, S. 265-272
- [BUZ13] Bundesimmissionsschutzgesetz (2013): *Begriffsbestimmungen*. §3 Abs. 6. http://www.gesetze-im-internet.de/bimschg/_3.html. Zugriff: 26.07.2013
- [Cai98] Castelfranchi, C. (1998). *Modeling Social Action for AI Agents*. *Journal Artificial Intelligence - Special issue: artificial intelligence 40 years later archive Volume 103 Issue*, Aug. 1-2, 1998

- [CaM83] Card, S.; Moran, T.P.; Newell, A. (1983): *The Psychology of Human Computer Interaction*. Lawrence Erlbaum Associates, 1983
- [Cao09] Castro, C. (2009): *Human Factors of Visual and Cognitive Performance in Driving*. CRC Press, Taylor & Francis Group, ISBN: 9781420055306
- [ChW02] Christoffersen, K.; Woods, D.D. (2002): *How to make automated systems team players*. In: Salas, E. (Hrsg.): *Advances in Human Performance and Cognitive Engineering*. Research Bd. 2. Elsevier Science Ltd., 2002, S. 1-12
- [Cor93] Conner, D. R. (1993): *Managing at the speed of change*. Villard Books, New York
- [D3S12] D3CoS Community (2012): *Reference Designs and Design Patterns for Cooperation & DCoS State Inference and Adaptation*. D3-03 Deliverable EU-D3CoS
- [Deo78] DeMarco, T. (1978): *Structured Analysis and System Specification*. New York, NY: Yourdon, 1978
- [Der03] Dekker, S.W.A. (2003): *When human error becomes a crime*. *Human Factors and Aerospace Safety*, 3(1), S. 83-92
- [DOD01] DOD (2001): *Systems Engineering Fundamentals: Supplementary Text*. Defense Acquisition University Press Fort Belvoir, VA 22060-5565, January 2001
- [DoN96] Donges, E.; Naab, K. (1996): *Regelsysteme zur Fahrzeugführung und -stabilisierung in der Automobiltechnik*. *Automatisationstechnik* 5, S. 226-236
- [Dör01] Dörner, D. (2001): *Bauplan für eine Seele*. Rowohlt Verlag, Reinbek bei Hamburg
- [EIG08] Elm, W.C.; Gualtieri, J.W.; McKenna, B.P.; Tittle, J.S.; Peffer, J.E.; Szymczak, S.S.; Grossman, J.B. (2008): *Integrating Cognitive Systems Engineering Throughout the Systems Engineering Process*. *Journal of cognitive Engineering and Decision Making*, Vol. 2, Num. 3, Fall 2008, S. 249-273
- [Eny03] Endsley, M.R. (2003): *Designing for Situation Awareness: An Approach to User-Centered Design*. Taylor & Francis, 2003
- [Eny95] Endsley, M. R. (1995): *Toward a theory of situation awareness in dynamic systems*. *Human Factors* 37, S. 32-64
- [ERO13] ERTICO (2013): *RESPONSE 3*. <http://www.ertico.com/response>. Zugriff 26.07.2013
- [ESL13] ESTEREL (2013): *SCADE Suite*. <http://www.esterel-technologies.com/products/scade-suite/>. Zugriff: 27.07.2013
- [ETS13] ETAS (2013): *ETAS Produkte*. http://www.etas.com/de/products/ascet_software_products.php. Zugriff: 27.07.2013

- [EUN06] European Commission (2006): *COMMISSION RECOMMENDATION of 22 December 2006. On safe and efficient in-vehicle information and communication systems: Update of the European Statement of Principles on human machine interface*. Official Journal of the European Union. 6. Feb. 2007
- [FaH01] Fastenmeier, W.; Hinderer, J.; Lehnig, U.; Gstalter, H. (2001): *Analyse von Spurwechselvorgängen im Verkehr*. Zeitschrift für Arbeitswissenschaft 55, Nr. 1
- [FAT12] Fast Light Toolkit (2012): <http://www.fltk.org>, Zugriff: Dez: 2012
- [Flh00] Flemisch, F. (2000): *The horse metaphor as a guideline for vehicle automation*. Proposal to the National Research Council, München
- [FIH12] Flemisch, F.; Heesen, M.; Hesse, T.; Kelsch, J.; Schieben, A.; Beller, J. (2012): *Towards a dynamic balance between humans and automation: authority, ability, responsibility and control in shared and cooperative control situations*. Cognition, Technology & Work (CTW), 05/2012, 14(1), S. 3-18
- [FIK08] Flemisch F.; Kelsch J.; Löper C.; Schieben A.; Schindler J. (2008): *Automation spectrum, inner/outer compatibility and other potentially useful human factors concepts for assistance and automation*. De Wart, D. (Ed). Human Factors for Assistance and Automation, Shaker Publishing
- [FIK10] Flemisch, F.; Kelsch, J.; Heesen, M.; Löper, C. (2010): *Wie erkunden wir die Lebenswelt kooperativer Technik? Skizze des Gestaltungsraums haptisch-multimodaler Kopplung Mensch, Co-Automation und Regelstrecke als Teil einer kooperativen Bewegungsbeeinflussung*. GfA Frühjahrskonferenz "Neue Arbeits- und Lebenswelten gestalten", Darmstadt
- [FIS08] Flemisch, F.; Schindler, J.; Kelsch, J.; Schieben, A.; Damböck, D. (2008): *Some Bridging Methods towards a Balanced Design of Human-Machine Systems, Applied to Highly Automated Vehicles*. Applied Ergonomics International Conference, Las Vegas, USA
- [GaA12] Gasser, T.M.; Arzt, C.; Ayoubi, M.; Bartels, A.; Bürkle, L; Eier, J.; Flemisch, F.; Häcker, D.; Hesse, T.; Huber, W.; Lotz, C.; Maurer, M.; Ruth-Schumacher, S.; Schwarz, J.; Vogt, W. (2012): *Rechtsfolgen zunehmender Fahrzeugautomatisierung*. Berichte der Bundesanstalt für Straßenwesen. Fahrzeugtechnik, F83, S. 9
- [Gan03] Gasson, S. (2003): *Human-centered vs. user-centered approaches to information systems design*. The Journal of Information Technology Theory and Applications (JITTA), 5(2), 2003, S. 29-46
- [Gin77] Gibson J.J. (1977): *The Theory of Affordances*. In *Perceiving, Acting, and Knowing*. Eds. Robert Shaw and John Bransford, ISBN: 0470990147
- [GNU13] GNU (2013): *About GNU Octave*. <http://www.gnu.org/software/octave/about.html>. Zugriff: 27.07.2013

- [GoL01] Gobet, F.; Lane, P.C.R.; Croker, S.; Cheng, P. C-H.; Jones, G.; Oliver, I.; Pine, J. M. (2001): *Chunking mechanisms in human learning*. TRENDS in Cognitive Sciences, 5, S. 236-243
- [Gon97] Goldstein, E.B. (1997): *Wahrnehmungspsychologie*. Spektrum Akademischer Verlag, Heidelberg.
- [GrG05] Griffiths, P.G.; Gillespie R. B. (2005): *Sharing control between humans and automation using haptic interface: primary and secondary task performance benefits*. Human Factors. 2005 Fall; 47(3), S. 574-590
- [GrK12] Griesche, S.; Kelsch, J.; Heesen, M.; Martirosjan, A. (2012): *Adaptive Automation als ein Mittel der Arbitrierung zwischen Fahrer und Fahrzeugautomation*. AAET - Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel, 8.-9. Feb. 2012, Braunschweig, Deutschland
- [Grn00] Green, M. (2000): *How Long Does It Take to Stop? Methodological Analysis of Driver Perception-Brake Times*. Transportation Human Factors, Vol.2:3, S.195-216
- [GrP00] Griffiths, R.; Pemberton, L.; Borchers, J.; Stork, A. (2000): *Pattern languages for interaction design: Building momentum*. CHI 2000 Extended Abstracts (The Hague, Apr. 1–6, 2000), p. 363. ACM Press, 2000
- [Had72] Hayward, J.Ch. (1972): *Near miss determination through use of a scale of danger*. Report no. TTSC 7115, Pennsylvania State University, Pennsylvania
- [HAE14] HAVEit-Webseite (2014): <http://www.haveit-eu.org/>. Zugriff: 29.04.2014
- [Har86] Hacker, W. (1986): *Arbeitspsychologie*. Huber, Bern
- [HeB11] Heesen, M.; Beller, J.; Flemisch, F. (2011): *Making automation surprises less surprising*. In: Reflexionen und Visionen der Mensch-Maschine-Interaktion - Aus der Vergangenheit lernen, Zukunft gestalten, 33, S. 32-33. VDI Verlag, Düsseldorf. ISBN: 9783183033225
- [HeD14] Heesen, M.; Dziennus, M.; Hesse, T.; Schieben, A.; Brunken, C.; Löper, C.; Kelsch, J.; Baumann, M. (2014): *Interaction design of automatic steering for collision avoidance: challenges and potentials of driver decoupling*. IET Intelligent Transport Systems. Im Druck. Stand 15.04.2014
- [Heg06] Herczeg, M. (2006): *Interaktionsdesign. Gestaltung interaktiver und multimedialer Systeme*. München: Oldenbourg, S. 7
- [HeV02] Herber, H.-J.; Vásárhelyi, É. (2002): *Lewins Feldtheorie als Hintergrundparadigma moderner Motivations- und Willensforschung. (Im Vergleich zu Behaviorismus, Psychoanalyse, Gestalt- und Kognitionspsychologie)*. In: Salzburger Beiträge zur Erziehungswissenschaft 6 (1), S. 37-100

- [Hoc01] Hoc, J.M. (2001): *Towards a cognitive approach to human-machine cooperation in dynamic situations*. International Journal of Human-Computer Studies, 54, S. 509-540
- [Hoc07] Hoc, J.M. (2007): *Human and Automation: A Matter of Cooperation*. Author manuscript, published in HUMAN 07, Timimoun, Algeria
- [Hog88] Hogeweg, P. (1988): *Cellular automata as a paradigm for ecological modelling*. Applied Mathematics and Computation 27 (1). S. 81-100
- [HoM08] Hoffman, R.R.; Militello, L.G. (2008): *Perspectives on cognitive task analysis: Historical origins and modern communities of practice*. Taylor&Francis, New York
- [Hon07] Holzmann, F. (2007): *Adaptive Cooperation between Driver and Assistant System*. Dissertation. Springer
- [HoW82] Hollnagel, E.; Woods, D. D. (1982): *Cognitive Systems Engineering: New wine in new bottles*. International Journal of Human-Computer Studies, Volume 51, Nr. 2, Aug. 1999, S. 339-356
- [HuH88] Hutchins, E.; Hollan, J.; Norman, D. A. (1988): *Direct Manipulation Interfaces*. Human Computer Interaction, 1, S. 311-338, 1988
- [IBM12] IBM (2012): *Rational DOORS DXL: Reference Manual*. Release 9.4, 2012
- [INE14] InteractIVe-Webseite (2014): <http://www.interactive-ip.eu/>. Zugriff: 29.04.2014
- [INY12] InteractIVe Community (2012): *IWI Strategies*. Deliverable D3.2 for EU-Project InteractIVe, 31.07.2012
- [ISO3b] ISO 15008:2003 (2003): *Ergonomische Aspekte von Fahrerinformations- und Assistenzsystemen: Anforderungen und Bewertungsmethoden der visuellen Informationsdarstellung im Fahrzeug*
- [ISO3c] ISO 17287:2003 (2003): *Ergonomische Aspekte von Fahrerinformations- und Assistenzsystemen: Verfahren zur Bewertung der Gebrauchstauglichkeit beim Führen eines Kraftfahrzeuges*
- [IS12a] ISO 15006:2012 (2012): *Straßenfahrzeuge - Ergonomische Aspekte von Verkehrsinformations- und Assistenzsystemen - Anforderungen und Konformitätsverfahren für die Ausgabe auditiver Informationen im Fahrzeug*
- [IS12b] ISO 26262-10:2012(E) (2012): *Road vehicles - Functional safety - Part 10: Guideline on ISO 26262*
- [ISO04] ISO 9241-110:2004 (2004): *Ergonomische Anforderungen der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung*

- [ISO08] ISO 9241-210:2008 (2008): *Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems*
- [ISO09] ISO 9241-920:2009 (2009): *Ergonomics of human-system interaction - Part 920: Guidance on tactile and haptic interactions*
- [ISO98] ISO 9241-11:1998 (1998): *Ergonomische Anforderungen für Bürotätigkeiten und Bildschirmgeräten. Teil 11: Anforderungen an die Gebrauchstauglichkeit.*
- [Ke12a] Kelsch, J. (2012): *Arbitration between Driver and Automation: why Overriding is just the Tip of the Iceberg.* InteractiVe Summer School, 04.-06.06.12, Corfu, Greece
- [KeD15] Kelsch, J.; Dziennus, M.; Köster, F. (2015): *Cooperative Lane Change Assistant: Background, Implementation & Evaluation.* Beitrag zur AAET 2015, Braunschweig, Germany; 02/2015
- [KeF06] Kelsch, J.; Flemisch, F.; Löper, C., Schieben A., & Schindler, J. (2006): *Links oder rechts, schneller oder langsamer? Grundlegende Fragestellungen beim Cognitive Systems Engineering von hochautomatisierter Fahrzeugführung.* Cognitive Systems Engineering in der Fahrzeug- und Prozessführung. DGLR-Bericht 2006-2, S.227-240
- [KeH09] Kelsch, J.; Heesen, M.; Löper, C.; Flemisch, F. (2009): *Balancierte Gestaltung kooperativer multimodaler Bedienkonzepte für Fahrerassistenz und Automation: H-Mode beim Annähern, Notbremsen, Ausweichen.* 8. Berliner Werkstatt MMS. 7.-9. Okt. 2009, Berlin
- [KeH12] Kelsch, J.; Heesen M.; Hesse T.; Baumann M. (2012): *Using human-compatible reference values in design of cooperative dynamic human-machine systems.* EAM 2012, 11-12.09.2012, Braunschweig, Germany
- [Keh14] Kelsch, J. (2014): *Zur Entscheidungskonvergenz in kognitiven Systemen.* 3. Interdisziplinärer Workshop Kognitive Systeme: Mensch, Teams, Systeme und Automaten: Verstehen, Beschreiben und Gestalten Kognitiver (Technischer) Systeme. Magdeburg, 25. - 27. März 2014
- [KeT13] Kelsch, J.; Temme, G.; Schindler, J. (2013): *Arbitration based framework for design of holistic multimodal human-machine interaction.* Contributions to AAET 2013, 6.-7. Feb. 2013, Braunschweig, Germany, ISBN: 9783937655291
- [Kle38] Kleene, S. C. (1938): *On notation for ordinal numbers.* Journal of Symbolic Logic, Vol. 3, No. 4, S. 150-155
- [Klm13] Klemm, R. (2013): *Willst Du mit mir kooperieren? Ja, Nein, Vielleicht!* Masterarbeit, HBK Braunschweig.
- [Kör16] Köster, F. (2016): *Entwurf agentenbasierter Softwaresysteme.* Vorlesungsskript. Department für Informatik. Carl von Ossietzky Universität. Oldenburg.

- [LeL06] Lehman, J. F.; Laird, J.; Rosenbloom, P. (2006): *A Gentle Introduction to Soar*. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.6097&rep=rep1&type=pdf>, Zugriff: 14.12.2015
- [Len38] Lewin, K. (1938): *The Conceptual Representation and the Measurement of Psychological Forces*. Psychological Theory, 4, Duke University Press, Durham, N.C.
- [Ler11] Lemmer, K. (2011): *Handlungsfeld Mobilität: Infrastrukturen sichern. Verkehrseffizienz verbessern*. Exportchancen ergreifen. Springer, 2011, ISBN: 9783642182453
- [Lin07] Lintern, G. (2007): *What is a Cognitive System?* Proceedings of the 14th International Symposium on Aviation Psychology, 18.-21.04.2005, Dayton, S. 398-402
- [Lin13] Lintern, G. (2013): *Tutorial: Work Domain Analysis*. <http://www.lintern.net/Tutorials/Work%20Domain%20Analysis%20Tutorial.pdf>. Zugriff: 03.08.2013
- [LöK08] Löper, C.; Kelsch, J.; Flemisch, F. (2008): *Kooperative, manöverbasierte Automation und Arbitrierung als Bausteine für hochautomatisiertes Fahren*. In: Gesamtzentrum für Verkehr Braunschweig e.V. (Hg.): AAET 2008,. Braunschweig: GZVB, S. 215-237
- [LuM05] Luck, M.; McBurney, P.; Shehory, O.; Willmott, S.; AgentLink Community (2005): *Agent Technology: Computing as Interaction. A Roadmap for Agent Based Computing*. AgentLink III, Sept. 2005, ISBN: 0854328459
- [MaC94] Malone, T.W.; Crowston, K. (1994): *The interdisciplinary study of coordination*. ACN Computing Survey 26 (I), 1994
- [Man02] Martin, J. (2002): *Organizational culture: Mapping the terrain*. Thousand Oaks, CA: SageMartin
- [Mar92] Mandler, J. M. (1992): *How to Build a Baby: II. Conceptual Primitives*. Psychological Review. Vol. 99. No. 4. S. 591
- [Mar99] Mainzer, K. (1999): *Komplexe Systeme und Nichtlineare Dynamik in Natur und Gesellschaft*. Springer, 1999, ISBN: 9783540653295
- [MAS13] MathWorks (2013): *MATLAB: Die Sprache für technische Berechnungen*. <http://www.mathworks.de/products/matlab/>. Zugriff: 27.07.2013
- [Mcd10] McClelland, J. M. (2010): *Emergence in Cognitive Science*. Topics in Cognitive Science 2, Cognitive Science Society, Inc., 2010, S. 751–770
- [MiD09] Militello, L.G.; Dominguez, C.O.; Lintern, G.; Klein, G. (2009): *The Role of Cognitive Systems Engineering in the Systems Engineering Design Process*. Wiley InterScience. DOI 10.1002/sys.20147

- [MIT13] Microsoft (2013): *Microsoft Project 2013: Produktinformationen*. <http://www.microsoft.com/project/de/de/product-information.aspx>. Zugriff: 07.07.2013
- [Miy88] Minsky, M. (1988): *The Society of Mind*. Simon and Schuster, New York. 15. März, 1988. ISBN: 0671657135, S. 308
- [Mos75] Morris, C. W. (1975): *Grundlagen der Zeichentheorie. Ästhetik und Zeichentheorie*. München: Carl Hanser Press
- [MoS98] Mosier, K. L.; Skitka, L. J.; Heers, S.; Burdick, M. (1998): *Automation Bias: Decision Making and Performance in High-Tech Cockpits*. *Automation Bias*. The international Journal of Aviation Psychology, 8(1), 1998, S. 47-63, Lawrence Erlbaum Associates, Inc.
- [NaE95] Nagel, H.; Enkelmann, W.; Struck, G. (1995): *FhG-CoDriver: From Map-Guided Automatic Driving by Machine Vision to a Cooperative Driver Support*. In: *Mathematical and Computer Modelling* 22, 1995, Nr. 4-7, S. 185-212
- [Nel80] Newell, A. (1980): *Physical Symbol Systems*. *Cognitive Science* 4, S. 135-183
- [Nen66] Neumann, J. von (1966): *Theory of self-reproducing automata*, Burks, A.W. (Hrsg.), University of Illinois press.
- [Nin93] Nielsen, J. (1993): *Usability Engineering*. Morgan Kaufmann, San Francisco
- [Nin94] Nielsen, J. (1994): *Heuristic Evaluation*. In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*. New York: John Wiley & Sons. S. 25-62
- [NoC06] Novak, J. D.; Cañas, A. J. (2006): *The Theory Underlying Concept Maps and How To Construct and Use Them*. Florida Institute for Human & Machine Cognition. USA
- [NoD86] Norman, D. A.; Draper, S. (1986): *User centered system design: New perspectives on human-computer interaction*. Lawrence Erlbaum Associates, Mahwah
- [Non90] Norman, D. A. (1990): *The 'problem' with automation: inappropriate feedback and interaction, not 'over-automation'*. *Philosophical Transactions of the Royal Society of London*. B 327, S. 585-593
- [OMG00] OMG (Object Management Group) (2000): *Agent Technology: Chapter 7: The Relationship of Agents and Objects*. 16 März, 2000
- [OMG11] OMG (Object Management Group) (2011): *Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1*, <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/>, Zuletzt aufgerufen am 25.04.2013
- [Onn94] Onken, R. (1994): *DAISY, an Adaptive, Knowledge-based Driver Monitoring and Warning System*. *Proceedings of Intelligent Vehicles '94 Symposium*, S. 544-549

- [PaR97] Parasuraman, R.; Riley, V. (1997): *Humans and automation: Use, misuse, disuse, abuse*. Human Factors 39(2), S. 230-253
- [PaS00] Parasuraman, R.; Sheridan, T. B.; Wickens, C. D. (2000): *A Model for Types and Levels of Human Interaction with Automation*. IEEE Transactions On Systems, Man, And Cybernetics – Part A: Systems And Humans, Vol. 30, No. 3, S. 286-297
- [Pas12] Pagès, B. (2012): *Bouml 2.7.6*. <http://bouml.free.fr/>, Zugriff: 15.12 2012
- [PoR09] Pohl, K.; Rupp, C. (2009): *Basiswissen Requirements Engineering*. dpunkt.verlag, 2009, ISBN 9783898646130
- [Pre06] Price B. D. (2006): *Form follows function? Actually, no*. American Chronicle. <http://amchron.soundenterprises.net/articles/view/13493>. Zugriff: 02.07.2013
- [PuO03] Putzer, H.; Onken, R. (2003): *COSA - A generic cognitive system architecture based on a cognitive model of human behavior*. CTW (2003) 5: S. 140-151
- [Ra86a] Rasmussen, J. (1986): *Information processing and human-machine interaction: An approach to cognitive engineering*. New York, North-Holland, S. 101-115
- [Ra86b] Rasmussen, J.; Leplat, J.; Duncan, K. (Hrsg.) (1986): *New Technology and Human Error*. Chichester, John Wiley & Sons
- [RaG91] Rao, A. S.; Georgeff, M. P. (1991): *Modeling rational agents within a BDI-architecture*. Proceedings of the International Conference on Knowledge, Representation and Reasoning, S. 473-484
- [RaP94] Rasmussen, J.; Pejtersen, A.; Goodstein, L. (1994): *Cognitive Systems Engineering*. New York, Wiley
- [RhW11] Rhede, J., Wäller, C., Oel, P. (2011): *Der FAS Warnbaukasten*. Conference contribution for „Der Fahrer im 21. Jahrhundert“. 08.-09. Nov. 2011, Braunschweig
- [RoC12] Rosenbaum, D.A.; Chapman, K.M.; Weigelt, M.; Weiss, D.J.; van der Wel, R. (2012): *Cognition, action, and object manipulation*. Psychological Bulletin. 2012, Sep; 138(5): 924-46. doi: 10.1037/a0027839
- [Ros73] Roberts, D. D (1973): *The Existential Graphs of Charles S. Peirce*. The Hague: Mouton, 1973
- [Sa95a] Sarter, N. B.; Woods, D. D. (1995): *How in the World did we ever get into that mode? Mode Error and Awareness in Supervisory Control*. Human Factors, 37(1), 1995, S. 5-19
- [Sa95b] Sarter, N.B.; Woods D.D. (1995): *Strong, Silent and Out of the Loop: Properties of Advanced (Cockpit) Automation and their Impact on Human-Automation Interaction*. Cognitive Systems Engineering Laboratory, Ohio State University

- [Sce93] Schmidtke, H. (1993): *Ergonomie*. München: Hanser
- [ScF08] Schieben, A.; Flemisch, F.; (2008): *Who is in control? Exploration of transitions of control between driver and an eLane vehicle automation*. In: VDI-Berichte 2048. VDI Verlag, Wolfsburg, ISBN: 9783180920481, S. 455-469
- [ScK13] Schindler, J.; Kelsch, J.; Heesen, M.; Dziennus, M.; Temme, G.; Baumann, M. (2013): *A Collaborative Approach for the Preparation of Co-operative Multi-User Driving Scenarios*. 10. Berliner Werkstatt MMMS, 10.-12. Oktober 2013, Berlin
- [Scr03] Schweitzer, F. (2003): *Brownian Agents and Active Particles*. Springer, Berlin
- [Scr92] Schnieder, E. (Hrsg.) (1992): *Petrinetze in der Automatisierungstechnik*. Oldenbourg, 1992, ISBN: 9783486220452
- [ShL09] Shoham, Y.; Leyton-Brown, K. (2009): *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press. Revision 1.1, Shoham & Leyton-Brown, 2009, 2010
- [Shn48] Shannon, C. E. (1948): *A Mathematical Theory of Communication*. Bell System Technical Journal, Vol. 27
- [ShV78] Sheridan, T. B.; Verplank, W. (1978): *Human and Computer Control of Undersea Teleoperators*. Cambridge, MA: Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT
- [Sla11] Slotka, F. (2011): *Einführung in die Sprachwissenschaft des Deutschen. Sprachwissenschaft auf den Punkt gebracht: Ein Skript*. Institut für deutsche Sprache und Literatur I. Universität zu Köln
- [SP13a] Springer Gabler Verlag (Hrsg.) (2013): Gabler Wirtschaftslexikon, Stichwort: *System*, <http://wirtschaftslexikon.gabler.de/Archiv/3210/system-v11.html>, Zugriff: 22.07.2013
- [SP13b] Springer Gabler Verlag (Hrsg.) (2013): Gabler Wirtschaftslexikon, Stichwort: *Ereignis*. <http://wirtschaftslexikon.gabler.de/Archiv/57711/ereignis-v10.html>, Zugriff: 06.08.2013
- [SP13c] Springer Gabler Verlag (Hrsg.) (2013): Gabler Wirtschaftslexikon, Stichwort: *Kognition*. <http://wirtschaftslexikon.gabler.de/Archiv/9957/kognition-v8.html>, Zugriff: 23.07.2013
- [SP13d] Springer Gabler Verlag (Hrsg.) (2013): Gabler Wirtschaftslexikon, Stichwort: *Modell*. <http://wirtschaftslexikon.gabler.de/Archiv/495/modell-v11.html>, Zugriff: 24.07.2013
- [SP13e] Springer Gabler Verlag (Hrsg.) (2013): Gabler Wirtschaftslexikon, Stichwort: *Spieltheorie*. <http://wirtschaftslexikon.gabler.de/Archiv/2710/spieltheorie-v8.html>, Zugriff: 08.08.2013

- [SP13f] Springer Gabler Verlag (Hrsg.) (2013): Gabler Wirtschaftslexikon, Stichwort: *Aufgabe*. <http://wirtschaftslexikon.gabler.de/Archiv/2754/aufgabe-v8.html>. Zugriff: 12.08.2013
- [SP13g] Springer Gabler Verlag (Hrsg.) (2013): Gabler Wirtschaftslexikon, Stichwort: *Entscheidung*. <http://http://wirtschaftslexikon.gabler.de/Definition/entscheidung.html>. Zugriff: 21.11.2015
- [SPR14] Springer (2014): *EU-Projekt "Adaptive" erforscht effizientes und sicheres automatisiertes Fahren*. <http://www.springerprofessional.de/eu-projekt-adaptive-erforscht-effizientes-und-sicheres-automatisiertes-fahren/4944622.html>. Zugriff: 15.06.2014
- [Sun03] Sun, R. (2003). *A Tutorial on CLARION 5.0. Technical Report*. Cognitive Science Department, Rensselaer Polytechnic Institute
- [Tay83] Talmy, L. (1983): *How language structures space*. In Pick, H. L., Jr. & Acredolo, L. P. (Hrsg.), *Spatial orientation: Theory, research, and application*. Plenum Press, NY
- [TiM03] Tietjen, T.; Müller D.H. (2003): *FMEA-Praxis. Das Komplettpaket für Training und Anwendung*. 2. Auflage, Carl Hanser Verlag, München, Wien
- [TON13] TortoiseSVN (2013): *TortoiseSVN: the coolest interface to (Sub)version control*. <http://tortoisesvn.net/about.html>. Zugriff: 07.07.2013
- [Trk99] Trask, R. L. (1999): *Language: The Basics*. Taylor & Francis Ltd. ISBN: 9780415340199
- [UNE68] UNECE (1968): Vienna Convention PART I: CONVENTION ON ROAD TRAFFIC (Consolidated Version), Online http://www.unece.org/fileadmin/DAM/trans/conventn/Conv_road_traffic_EN.pdf, Zugriff: 08.11.2015
- [USD07] U.S. Department of Transportation Systems (2007): *Engineering for Intelligent Transportation Systems - An Introduction for Transportation Professionals*, S. 11
- [Vav09] Vakarelov, O. (2009): *The Cognitive Agent*. Department of Philosophy, University of Arizona, Tucson, Arizona. Online: <http://www.u.arizona.edu/~okv/Papers/CogAgent.pdf>. Zugriff: 05.08.2013
- [Vie99] Vicente, K. J. (1999): *Cognitive work analysis: Toward safe, productive, and healthy computer based work*. Lawrence Erlbaum Assoc. Inc.
- [Vyy78] Vygotsky, L. S. (1978): *Mind in Society: The Development of Higher Psychological Processes*. Cole, M.; John-Steiner, V.; Scribner, S; Souberman, E. (Editors). Cambridge, Massachusetts London, England, 1978
- [W313a] W3C (2013): *OWL 2 Web Ontology Language Document Overview*. <http://www.w3.org/TR/owl2-overview/>. Zugriff: 04.06.2013

- [W313b] W3C Recommendation (2013): *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. <http://www.w3.org/TR/REC-xml>, Zugriff: 15.01.2013
- [WaH67] Watzlawick, P.; Helmick Beavin, J.; Jackson, D.D. (1967): *Pragmatics of Human Communication: A Study of Interactional Patterns, Pathologies, and Paradoxes*. Norton, ISBN: 0393010090
- [WaK64] Wallach, M. A.; Kogan, N.; Bem, D. J. (1964). *Diffusion of responsibility and level of risk taking in groups*. *Journal of Abnormal and Social Psychology*, 68
- [WI13a] Wikipedia (2013): *Brainstorming*. <http://de.wikipedia.org/wiki/Brainstorming>. Zugriff: 03.08.2013
- [WI13b] Wikipedia (2013): *Kognition*. <http://de.wikipedia.org/wiki/Kognition>. Zugriff: 23.07.2013
- [WI13c] Wikipedia (2013): *Mind-Map*. <http://de.wikipedia.org/wiki/Mind-Mapping>. Zugriff: 03.08.2013
- [WI13d] Wikipedia (2013): *Modell*. <https://de.wikipedia.org/wiki/Modell>. LZ 04.06.2013
- [WI13e] Wikipedia (2013): *PREVENT*. <http://en.wikipedia.org/wiki/PReVENT>. Zugriff: 26.07.2013
- [WI13f] Wikipedia (2013): *Software-Agent*. <http://de.wikipedia.org/wiki/Software-Agent>. Zugriff: 25.07.2013
- [WI13g] Wikipedia (2013): *System*. <http://de.wikipedia.org/wiki/System>. Zugriff: 04.06.2013
- [WiH09] Winner, H.; Hakuli, S.; Wolf, G. (Hrsg.) (2009): *Handbuch für Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Vieweg+Teubner, 2009, ISBN: 9783834802873, S. 478-520
- [Wis84] Wickens, C.D. (1984): *Processing resources in attention*. In Parasuraman R. & Davies D.R. (Eds.), *Varieties of attention*, S. 63-102. New York. Academic Press
- [WoH87] Woods, D. D. & Hollnagel, E. (1987): *Mapping cognitive demands in complex problem-solving worlds*. *International Journal of Man-Machine Studies*, 26
- [Wom02] Wolfram, S. (2002): *A New Kind of Science*. Wolfram Media.
- [YeD08] Yerkes, R. M. & Dodson, J. D. (1908): *The relation of strength of stimulus to rapidity of habit-formation*. *Journal of Comparative Neurology and Psychology*, 18