

DOMÄNENÜBERGREIFENDE GESTALTUNG VON KOOPERATIVEN SYSTEMEN

G. Temme, J. Kelsch und F. Köster, Deutsches Zentrum für Luft und Raumfahrt e.V. (DLR)
Institut für Verkehrssystemtechnik (ITS), Lilienthalplatz 7
38108 Braunschweig, Deutschland

Abstract

In diesem Beitrag werden zwei beim DLR im Rahmen von D3CoS¹ entstandene MTTs² vorgestellt. Es handelt sich dabei um die Multi-Systemnutzer Simulationsplattform MoSAIC³ und um Design Pattern für kooperative Systeme. MoSAIC ermöglicht einen Systemtest unter der gleichzeitigen Einbeziehung mehrerer Systemnutzer. Design Pattern für kooperative Systeme stellen eine Möglichkeit bereit, eine Systemgestaltung strukturiert durchzuführen und die gewonnenen Erkenntnisse zwischen unterschiedlichen Domänen auszutauschen. Die vorgestellten MTTs werden anhand eines Beispiels aus der Automotivdomäne erläutert. Dabei handelt es sich um einen beim DLR entwickelten kooperativen Fahrstreifenwechselassistenten⁴. Dieser lässt auf der Autobahn zwei Fahrer miteinander kommunizieren und kooperieren, um einen sicheren Fahrstreifenwechsel auf einen stark befahrenen Fahrstreifen zu unterstützen.

1. EINLEITUNG

Fortbewegung mit dem Auto, dem Flugzeug oder dem Schiff ist ein wichtiger Schlüsselfaktor für unsere moderne Gesellschaft. In den letzten Jahrzehnten kam es zu einer rasanten Zunahme sowohl der Verkehrsnachfrage als auch der Verkehrsdichte in allen Domänen. Als Konsequenz aus dieser Zunahme sinkt die Verkehrseffizienz bei gleichzeitiger Steigerung der Belastung des Autofahrers, des Piloten, des Fluglotsen oder eines nautischen Offiziers. Um der Belastung des Menschen entgegenzuwirken, werden immer mehr Tätigkeiten durch Maschinen unterstützt oder sogar übernommen. Damit wird die Technik zu unserem täglichen Begleiter und Teamplayer [1]. Dies führt zu mehreren Herausforderungen. Zum einen muss zur Vermeidung von Stress und Fehlern die Mensch-Maschine-Schnittstelle (HMI) entsprechend sorgfältig designet werden, so dass eine zuverlässige und verständliche Interaktion und Kommunikation zwischen dem Systemnutzer und der Maschine ermöglicht wird. Zum anderen stellen die aktuellen Begrenzungen der Sensoren zum Erfassen der Umgebung eine weitere Herausforderung dar. Sie limitiert die Möglichkeiten der Maschine, den Menschen zu unterstützen.

Eine Option, dieses Problem zu reduzieren, ist das Miteinbeziehen von Sensordaten anderer Verkehrsteilnehmer oder der Infrastruktur. Daraus resultiert ein verteiltes System, innerhalb dessen Daten kommuniziert und von der jeweiligen Maschine verwendet werden. In der Automotivdomäne zum Beispiel mittels Car-to-Car Kommunikation [2]. Dies ermöglicht eine Weiterentwicklung der Systeme von dem traditionellen, nur innerhalb eines Fahrzeuges agierenden Mensch-Maschine System hin zu kooperativen, über die eigene Fahrzeuggrenze hinaus interagierenden Multi Agenten Systemen [3],[4]. Für eine effektive Gestaltung solcher verteilter kooperativer Systeme wurden im Rahmen des europäischen Projektes D3CoS [P1] mit 21 Partnern (BILD 1) aus den Bereichen Schifffahrt, Automobil sowie bemannter und unbemannter Luftfahrt Me-

thoden Techniken und Werkzeuge (MTTs) [5] entwickelt. Diese unterstützen und beschleunigen das Gestalten, das Aufbauen und das Evaluieren kooperativer Systeme über die Domänengrenzen hinweg.



BILD 1 Logo des D3CoS Projektes und der Partner [P1]

Erprobt wurden die entwickelten MTTs im Rahmen des Projektes bei der Gestaltung, Entwicklung und Evaluierung von Demonstratoren in allen D3CoS Domänen [6]:

- AstA; *Approach Stabilisation Advisory*, Bemannte Luftfahrt ([6] Seite 23 ff.)
- SAA; *Scenario Accident Assistance System*, Unbemannte Luftfahrt ([6] Seite 83 ff.)
- IVI; *In Vehicle Infotainment System*, Automobil ([6] Seite 152 ff.)
- C-LCA; *Cooperative Lane Change Assist*, Automobil ([6] Seite 156 ff.)
- A-VTMIS; *Active Vessel Traffic Management and Information System*, Schifffahrt ([6] Seite 247 ff.)

¹ Designing Dynamic Distributed Cooperative Human-Machine Systems (Projektlaufzeit: 03.2011 – 02.2014)

² Methods, Techniques & Tools

³ Modular and Scalable Application Platform for Testing and Evaluating of ITS Components

⁴ Cooperative lane change assist (C-LCA) auch als "Gap Assist" bezeichnet

2. D3CoS METHODIK

Ein Verteiltes Kooperatives System (Distributed Cooperative System - DCoS) ist zusammengefasst als eine Gruppe von menschlichen oder maschinellen Agenten welche miteinander interagieren und kommunizieren definiert. Dabei existieren die Gruppen von Agenten in einem kooperativen System mindestens einmal, in der Regel im Sinne eines Multi Agenten Systems (MAS) aber mehrfach. Dies ermöglicht das Entstehen eines kooperativen Netzwerkes. Innerhalb des Netzwerkes können den Agenten Aufgaben zugewiesen werden, welche diese dann mittels ihrer eigenen oder mittels mit anderen geteilter Ressourcen lösen. Zudem können sich Agenten zur Laufzeit dem Netzwerk anschließen oder es wieder verlassen. Auch die zu bewältigenden Aufgaben oder die zur Verfügung stehenden Ressourcen können sich zur Laufzeit verändern. All diese Faktoren zusammen müssen in einer generischen DCoS Architektur abgebildet werden (siehe BILD 2).

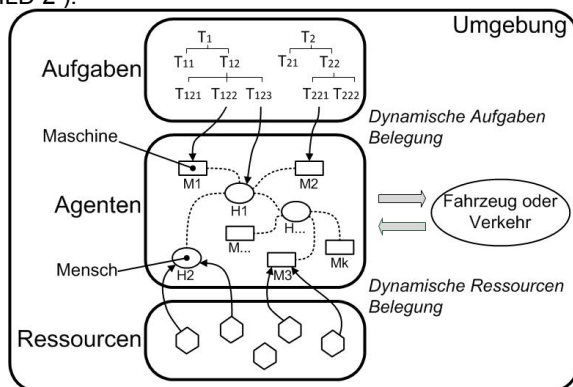


BILD 2 Generische DCoS Architektur für Multi Agenten Systeme [4]

In D3CoS wurde die Grundstruktur von Hoc [7] als Einstiegspunkt verwendet, um die Kooperation eines DCoS zu beschreiben. Dabei konnte gezeigt werden das Hoc's Grundstruktur leicht in die MAS Domäne, mit mehrfach vorhandenen, miteinander interagierenden menschlichen und maschinellen Agenten, überführt werden kann [4].

Daraus ergibt sich, dass fortwährend Agenten ihnen zugewiesene Aufgaben unter Verwendung der zur Verfügung stehenden Ressourcen versuchen zu erfüllen. Dies ermöglicht die Definition von Task-Allocation-and-Resource-Functions (TARF). Diese definieren welcher Agent welche Aufgabe mit welcher Ressource für ein gewähltes Szenario innerhalb eines DCoS bearbeitet.

2.1. Entwicklung von DCoS in D3CoS

Jedes System, so auch DCoS müssen untersucht (DCoS Composition), gestaltet (DCoS Interaction), implementiert (DCoS Interfaces) und begleitend getestet (DCoS Evaluation) werden [5]. Diese Abfolge wird in D3CoS als wiederkehrender Zyklus bis zur abschließenden Fertigstellung des Systems mehrfach wiederholt. BILD 3 beschreibt einen D3CoS Entwicklungszyklus im Detail:

DCoS Composition: Zu Beginn jedes Zyklus werden das DCoS und dessen Ziele untersucht. Hierfür werden die durch das DCoS zu bewältigen Aufgaben, die verfügbaren Agenten und Ressourcen sowie das Szenario und die TARF ermittelt, untersucht, neu aufgestellt oder verbessert.

DCoS Interaction: Als nächster Schritt wird das Zusammenspiel der Agenten gestaltet sowie deren Kommunikationspfade festgelegt. In diesem Schritt wird sowohl das HMI als auch die Funktionsweise von Arbitern [8] (siehe auch Abschnitt 3) definiert und ausgestaltet.

DCoS Interfaces: Im letzten Schritt jedes Zyklus werden die zuvor untersuchten, festgelegten und gestalteten Elemente realisiert. Am Ende dieses Schrittes ist eine Version des DCoS verfügbar.

Evaluation of: Begleitend zu allen Schritten jedes Zyklus wird eine Evaluierung der Ergebnisse vorgenommen. Diese umfasst das Definieren von Metriken, das Bewerten der Interaktion sowie das Untersuchen des resultierenden DCoS in jedem Zyklus.

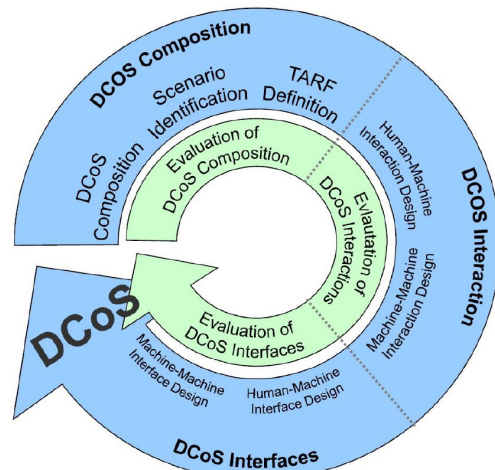


BILD 3 Generische domänenübergreifende Designschritte zum Entwickeln von DCoS in D3CoS [5]

2.2. Methoden, Techniken & Werkzeuge zur Entwicklung von DCoS

Zur Unterstützung der DCoS Designschritte wurden in D3CoS eine Vielzahl von Methoden, Techniken und Werkzeugen (MTTs) neu entwickelt oder weiterentwickelt. Ziel der MTTs ist es Ideen und Techniken unter einem Dach zu versammeln, die die Grundlage zur domänenübergreifenden Modellierung und Gestaltung von MAS und DCoS bieten. Innerhalb von D3CoS sind MTTs definiert als [9]:

- **Methoden:** Regeln, Verfahren oder Schritte notwendig für das Gestalten eines Systems oder das Ermitteln einer Problemlösung (z.B. Design Pattern)
- **Techniken:** Instanziierungen einer Methode (z.B. in Form von Algorithmen oder Protokollen)
- **Werkzeuge:** Konkrete Hardware oder Softwarelösungen die bei der Entwicklung von DCoS unterstützen (z.B. Simulationsplattformen)

In D3CoS wurden rund 30 MTTs entwickelt, weiterentwickelt und verwendet. Eine Übersicht über alle D3CoS MTTs, zugeordnet zu ihren Anwendungsbereich für die D3CoS Designschritten, ist über die Projektwebseite verfügbar [P1]. Nachfolgend werden einige der beim DLR Institut für Verkehrssystemtechnik (TS) entwickelten MTTs genauer beschrieben: Design Pattern zur Gestaltung von Kooperation sowie die Softwarearchitektur Dominion und die Simulationsplattform MoSAIC zum verteilten Implementieren und Testen von DCoS.

3. DESIGN PATTERN FÜR DIE GESTALTUNG VON KOOPERATION

Innerhalb eines MAS kann die Kooperation als ein emergenter Prozess verstanden werden, der durch eine Wechselwirkung der Agenten entstehen kann [10], [11]. Kooperation kann also auf der Systemebene mit Eigenschaften generiert werden, die keinem der Agenten eigen sind. Das Gegenteil einer Kooperation wäre z.B. die Konkurrenz, die ebenfalls in der Wechselwirkung der Agenten entstehen kann. Damit es aber nicht zu einer Konkurrenz, sondern zu dem optimaleren Systemverhalten, der Kooperation, kommt, bedarf genau dieser Aspekt der Wechselwirkung während einer DCoS-Gestaltung einer expliziten Betrachtung.

Die Wechselwirkung zwischen den Agenten kann vom Systemdesigner beeinflusst werden, indem er dem gestalteten System zusätzliche Agenten (Arbiter [8]) hinzufügt. Diese Arbiter nehmen dann Teil im Wechselwirkungsprozess. Durch die explizite Gestaltung des Verhaltens der Arbiter mittels der Anwendung der Methode der Arbitrierung [8], können Konflikte zwischen den Agenten vermieden oder gelöst werden [12], so dass keine Konkurrenz entsteht. Durch die Gestaltung der Arbiter nach bestimmten Regeln kann zusätzlich eine Kooperation zwischen den Agenten initiiert, beibehalten und nach Bedarf anschließend beendet werden.

Diese Theorie lässt sich im DCoS-Design verwenden. Die o.g. „bestimmten Regeln“ einer Kooperation können in einer vorstrukturierten Form als so genannte Design Pattern (DP) zunächst entwickelt und später im Designprozess verwendet werden. „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“ [13].

3.1. Design Pattern in D3CoS

Der Entwicklungsprozess und die Verwendung der Design Pattern für Kooperation in D3CoS sind in BILD 4 dargestellt.

Wenn ein bestimmtes DCoS gestaltet werden soll, dann hat es eine ursprüngliche statische und eine dynamische Konfiguration, die in einem Szenario oder auch Use Case zusammengefasst werden können.

Beispiel: Ein hochautomatisiertes Fahrzeugsystem besteht ursprünglich aus einem Fahrer, einer Automation, einem Fahrzeug und der nahen Umwelt (vgl. auch BILD 10). Diese stehen in einer bestimmten Beziehung (Kopplung [14]) zueinander. Der Fahrer und das Fahrzeug sind über die Stellteile gekoppelt, die Automation gibt dem Fahrer multimodales Feedback über die Situation im System. Dies ist die ursprüngliche Systemstatik. Der Fahrer führt die Aufgabe der Fahrzeugsteuerung aus, das Fahrzeug folgt den Befehlen des Fahrers und die Automation informiert den Fahrer über den Systemzustand. Diese Aufgabenverteilung bildet die ursprüngliche Systemdynamik. Zusammen beschreibt es ein Szenario, welches ein Teil eines Szenario Katalogs sein kann.

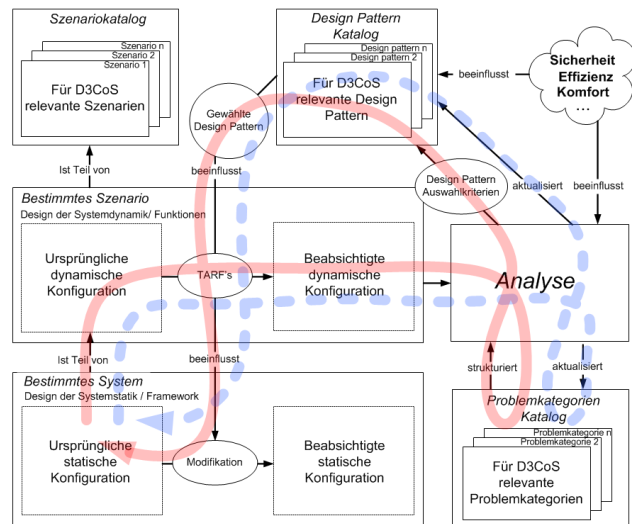


BILD 4 Konzeptionelles und strukturelles Zusammenspiel der DP und des Systemgestaltungsprozesses in D3CoS

Im Designprozess kann durch eine Strukturmodifikation die Systemstatik und durch die TARF's die Systemdynamik verändert werden.

Beispiel: Durch das Hinzufügen zusätzlicher Agenten, z.B. dem Arbiter, kann die Systemstatik modifiziert werden. Die Arbiter können Konflikte in der Fahrer - Automation - Fahrzeug - Umwelt Wechselwirkung beeinflussen und diese mittels der TARF's in Richtung der Kooperation verschieben. Dies kann mit einer Aufgaben-Umverteilung erreicht werden. Wenn der Arbiter einen starken Konflikt feststellt, kann er die Aufgaben im System umverteilen. Er kann z.B. die Kontrolle über das Fahrzeug dem Fahrer durch eine Entkopplung [10] entziehen und sie an die Automation delegieren.

Die Systemstatik und -dynamik kann analysiert werden und in Form eines Problemerkatalogs nach vorher festgelegten Problemerkategorien dokumentiert werden. Die allgemeinen gültigen Lösungen für ermittelte Probleme können den einzelnen Problemen zugeordnet werden. Dadurch entsteht Schritt für Schritt ein Design Pattern Katalog (rote durchgezogene Linie BILD 4). Während des eigentlichen späteren Systemdesigns (blau gestrichelte Linie BILD 4) kann sowohl der Problemerkatalog als auch der Design Pattern Katalog bei Bedarf weiter ausgearbeitet werden.

Informations-verarbeitungsschritt	Design Pattern Name
Initialisierung	PROBLEM AWARENESS
Initialisierung	ROLE ALLOCATION
Initialisierung Beibehaltung	EXPLICIT ADDRESSING
Initialisierung Beibehaltung Beendigung	ALLOCATION TO COOPERATIVE POPULATION
Initialisierung Beibehaltung	TASK AND GOAL ALLOCATION
Beibehaltung	AWARENESS OF OWN TASK AND GOAL
Beibehaltung	AWARENESS OF PARTNERS' TASK AND GOAL
Beibehaltung	CERTAINTY OF TASK AND GOAL
Initialisierung Beendigung	REWARDING

TAB 1 Übersicht über die neun DPs für Kooperation [9] als Teil des DP-Katalogs in D3CoS

Nach dem vorgestellten Konzept wurden in D3CoS die in TAB 1 vorgestellten Design Pattern für Kooperation bestimmt und dokumentiert. Sie wurden dabei nach ihrem möglichen Auftreten im Informationsverarbeitungsprozess (Initialisierung, Beibehaltung und Beendigung der Kooperation eines DCoS strukturiert.

3.2. Design Pattern Dokumentationsschema

Die entwickelten Design Pattern für Kooperation wurden, wie alle anderen in D3CoS entwickelten Pattern nach dem folgenden Schema dokumentiert:

- Einführungsteil:
 - Name: Sinnvoller aussagekräftiger Namen für das Design Pattern
 - Ranking: Sicherheit des Entwicklers über die Priorität des Patterns für das Design
 - Bild: Schematische Darstellung des Patterns
 - Kontext: Hintergrundinformationen zum Pattern
- Hauptteil:
 - Problemformulierung: Benennung des Problems, das mit dem Pattern adressiert werden kann
 - Problembeschreibung: Ausführlichere Beschreibung des zu adressierenden Problems
 - Lösung: Beschreibung des Lösungsansatzes
 - Diagramm: Diagramm des Lösungsansatzes
- Abschlussteil:
 - Referenzen: Verknüpfungen zu anderen bereits existierenden Pattern, Literaturliste etc.

3.3. Design Pattern Beispiel

Das im Abschnitt 5 verwendete Design Pattern „ALLOCATION TO COOPERATIVE POPULATION“ mit seiner konkreten Ausprägung der Kooperationsdyade [8] kann in dem angegebenen Schema folgendermaßen beschrieben werden:

Name: ALLOCATION TO COOPERATIVE POPULATION

Ranking: Dieses Design Pattern ist ein generisches Pattern, welches auf unterschiedliche Weisen implementiert werden kann. Die Agenten sollten wissen, wann und mit wem sie kooperieren, wenn der Kooperationsprozess beginnt, ob die Kooperation noch beibehalten wird oder ob sie schon beendet ist. Die Autoren sind sich sehr sicher, dass dieses Pattern sehr wichtig für die Initialisierung, die Beibehaltung und die Beendigung einer Kooperation ist.

Bild:

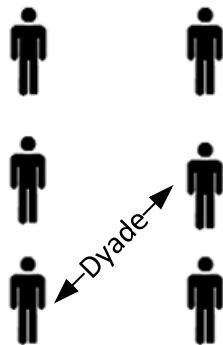


BILD 5 Schematische Darstellung des DP: Allocation to Cooperative Population

Kontext: Dieses Pattern hilft bei der Implementierung der abstrakteren Design Pattern, die sich als Schema des Kooperationsprozesses formulieren lassen: „Initialisierung der Kooperation“, „Beibehaltung der Kooperation“ und „Beendigung der Kooperation“

Problemformulierung: Oft wissen Agenten nicht, dass und mit wem sie kooperieren oder kooperieren müssen.

Problembeschreibung: Am Anfang und während eines Kooperationsprozesses kann das Bewusstsein der Agenten über die möglichen Kooperationsbeziehung und die Kooperationspartner fehlen. Am Ende einer Kooperation kann das Wissen darüber fehlen, ob die Kooperation noch beibehalten wird oder nicht.

Lösung: Das beschriebene Problem kann gelöst werden, indem alle an der Kooperation beteiligten Agenten zu einer kooperativen Population zugeordnet werden, z.B. mittels einer Dyade, Triade etc. [10]

Diagramm:

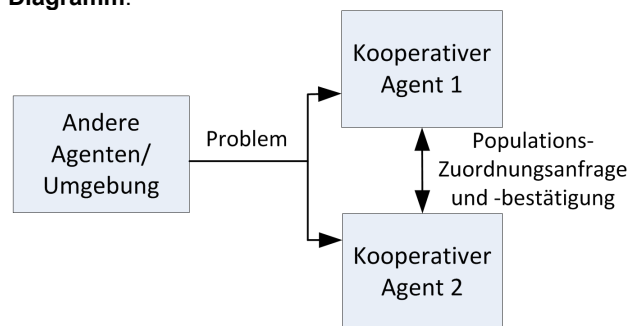


BILD 6 Diagramm mit Lösungsansatz des DP: Allocation to Cooperative Population

Referenzen: Verschiedene konkretere Design Patterns sind vorstellbar, um die Zuordnung der Agenten zu einer kooperativen Population zu kommunizieren. Z.B. können in der HMI-Ressource „Head-Up Display“ die kooperierenden Agenten mittels einer Linie bzw. eines augmentierten Bogens verbunden werden. Die Nutzung zueinander ähnlicher Symbole, die Ähnlichkeiten der Agenten aufzeigen, ist auch vorstellbar.

Die anderen acht Design Pattern für Kooperation samt einer Beschreibung des Einsatz in Prototypen zur hochautomatisierter Fahrzeugführung können unter [9] nachgeschlagen werden.

4. DOMINION UND MOSAIC FÜR VERTEILTES IMPLEMENTIEREN UND TESTEN

Zwei der in D3CoS verwendeten MTTs zur Beschleunigung der Gestaltung von kooperativen MAS in der Automobil Domäne sind die Softwarearchitektur Dominion und die Simulationsplattform MoSAIC. Beide MTTs unterstützen beim Gestalten und Testen von kooperativen Systemen und werden in den genannten Entwicklungsschritten „D3CoS Interfaces“, „D3CoS Evaluation of Interaction“ sowie „D3CoS Evaluation of Interfaces“ angewandt.

4.1. Dominion

Dominion ist eine am DLR-TS entwickelte Softwarearchitektur zum Entwickeln und Testen von fortschrittlichen Fahrerassistenzsystemen (ADAS). Dominion folgt dabei dem Konzept der serviceorientierten Architektur [15]. Als Basis stellt Dominion eine Reihe von Standardservices zur Verwaltung der Architektur wie z.B. eine GUI zum Steuern der Services oder Werkzeuge zum Beobachten, Manipulieren, Aufzeichnen und erneuten Abspielen von Datenströmen zur Verfügung. Mittels Codegenerierung und einer teilformalen Beschreibung der verwendeten Daten und Services erleichtert Dominion auch das Erstellen und Entwickeln neuer Services. Die Architektur wird in allen DLR-TS Simulatoren und Versuchsfahrzeugen der Automobil-Domäne eingesetzt [16]. Dies ermöglicht einen sehr flexiblen, modularen und effizienten Austausch von ganzen Prototypen zwischen den DLR-TS Forschungsanlagen. Die erste Version von Dominion wurde bereits 2007 entwickelt. Seitdem wurde Dominion erfolgreich in nationalen Projekten wie z.B. FAMOS [P2] oder IMoST [P3] und in internationalen Projekten wie z.B. HAVEit [P4] InteractiVe [P5] oder D3CoS [P1] als Softwarearchitektur zum Gestalten und Untersuchen von ADAS eingesetzt.

Im Rahmen des D3CoS Projektes wurde die Dominion Architektur als MTT zur Gestaltung und Erprobung kooperativer MAS für die Automobil-Domäne eingebracht. Hierfür wurde Dominion um die Fähigkeit des Verwaltens von mehreren identischen Services erweitert [17].

BILD 7 zeigt die resultierende modifizierte Dominion Architektur, die die Verknüpfung mehrerer MAS über die Verwendung einer Master - Slave Architektur ermöglicht. Mehrere an Dominion angeschlossene Services können dabei zusammen die Funktionalität eines Agenten im D3CoS Sinn bilden. In BILD 8 z.B. bilden im Abschnitt „Applikationen“ mehrere Services zusammen die Grundlage für den kooperativen Agenten *Automation*. Dieser wiederum wirkt zusammen mit dem kooperativen Agent *Fahrer*, welcher in den Simulatoren sitzt, auf den nichtkooperativen Agenten *Fahrzeug* ein. Das Fahrzeug ist wiederum ein Teil der Simulationsumgebung. Das Wirken der beiden kooperativen Agenten, *Automation* und *Fahrer* auf das Fahrzeug wird dabei über den Agenten „Interaktion“ (als Arbitr [8]) verhandelt und realisiert. Globaler betrachtet können die Agenten jedoch mehrfach, als Teil jedes intelligenten (mit z.B. einem ADAS ausgestatteten) Fahrzeuges, innerhalb der Simulation existieren. Mit im Rahmen des Projektes D3CoS entwickelten Modifikationen ist Dominion in der Lage die Datenströme auch auf der globalen Verkehrsebene zu verwalten.

Das MTT Dominion wurde im Projekt D3CoS ausschließlich für die Entwicklung der Simulationsplattform MoSAIC und damit in der Automobil-Domäne eingesetzt. Ein Einsatz von Dominion in anderen Domänen ist aber grundsätzlich möglich und wurde bereits erfolgreich realisiert. Als Beispiel aus der Luftfahrt-Domäne kann das Obstacle-City Szenario genannt werden. Das Szenario wurde vom DLR Institut für Flugsystemtechnik entwickelt und stellt in der Simulation Piloten im Fall einer Landung vor anspruchsvollen Situationen. Der Straßenverkehr sowie die Ampelsteuerung im Szenario wird dabei über Dominion Services realisiert [18] (Seite 71 ff.).

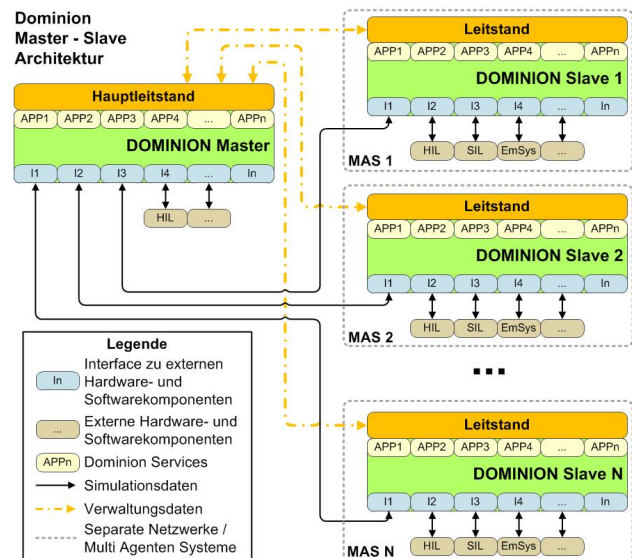


BILD 7 Dominion, Master - Slave Architektur zur Verwaltung mehrerer MAS

4.2. MoSAIC

Mit der Softwarearchitektur Dominion und einer Reihe von standardisierten Forschungsanlagen [16] steht dem DLR eine flexible, modulare und skalierbare Infrastruktur zur Verfügung. Der bisherige Fokus dieser Infrastruktur lag auf dem Entwickeln und Untersuchen von ADAS, Mensch-Maschine-Kommunikation sowie dem Analysieren und Verstehen des Menschen als Fahrer eines Fahrzeuges. Zur Bearbeitung dieser Themenfelder reichte ein durch eine Testperson gesteuertes Fahrzeug mit oder ohne ADAS Funktionen aus. Dieses Fahrzeug ist in der Regel innerhalb der Simulation umgeben von simulierten Fahrzeugen, welche begrenzt auf den menschlichen Fahrer reagieren können.

Mit dem Projekt D3CoS wurde der Infrastrukturfokus um das Untersuchen, Gestalten und Entwickeln von kooperativen ADAS, mit mehreren menschlichen Fahrern innerhalb einer Simulation, erweitert. Im Zuge dieser Erweiterung wurden ebenfalls neue Themenfelder wie das Untersuchen von sozialpsychologischen Phänomenen im Straßenverkehr, (z.B. Verantwortungsdiffusion) möglich. Leider steigt bei mehreren menschlichen Fahrern auch der Koordinierungsaufwand (für Szenarien, Software, Infrastruktur, ...) zur Untersuchung der Fragestellungen merklich an.

Um dem Ansteigen des Koordinierungsaufwandes entgegenzuwirken wurde die Simulationsplattform MoSAIC innerhalb des D3CoS Projektes als MTT zur Gestaltung kooperativer Systeme zusammengestellt. Der Begriff MoSAIC, steht dabei für: „Modular und Scalable Application Platform for Testing and Evaluating of ITS Components“. BILD 8 verdeutlicht die wesentlichen Elemente welche in der Simulation Plattform MoSAIC zusammengefasst werden.

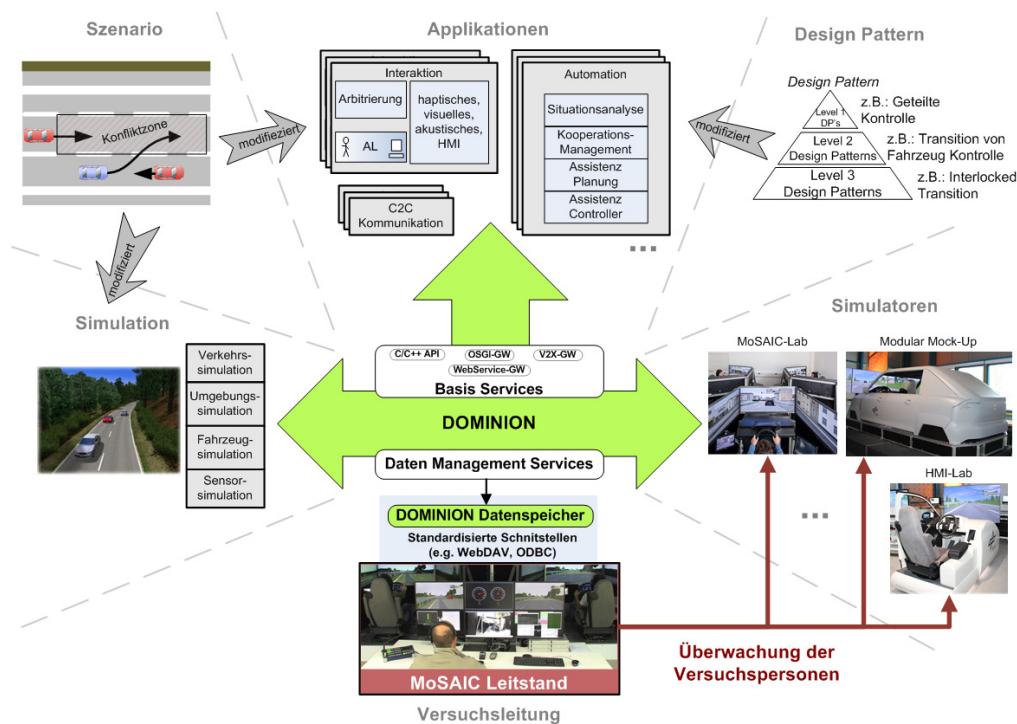


BILD 8 Elemente des MTT - Simulationsplattform MoSAIC unter Verwendung des MTT Dominion als zentrale Softwarearchitektur [7]

Dominion: Im Zentrum des MTT MoSAIC steht die Softwarearchitektur Dominion. Sie dient als Bindeglied zwischen allen weiteren Elementen der Simulationsplattform und ermöglicht sowohl das Verwalten, als auch das Debuggen und Testen der angeschlossenen Services und ihrer Datenströme.

Applikationen: Kern jedes ADAS sind eine Vielzahl von Services, welche gruppiert in Form von Applikationen oder Agenten die geplante Funktionalität der Assistenz realisieren.

Simulation: Zum Erproben der entwickelten Funktionalität steht mittels MoSAIC ein modularer Baukasten von Modulen zur Verfügung, um bedarfsgerecht die Umgebung, den Verkehr, die Fahrzeuge und die Sensoren zu simulieren.

Simulatoren: Als Schnittstelle zum Menschen sind verschiedene Simulatoren der Plattform MoSAIC zum Einbinden eines Fahrers in die Simulation verfügbar. Dabei ist es möglich die Simulatoren modular und skalierbar in allen möglichen Kombinationen zu verwenden. Dies erlaubt einen optimierten und effizienten Einsatz der Simulatoren in der Entwicklung und Evaluierung von ADAS innerhalb der Simulationsplattform MoSAIC.

Design Pattern: DPs spielen eine wesentliche Rolle in der Gestaltungsphase (DCoS Interaction) neuer Assistenzfunktionen und wirken bei Anwendung direkt auf relevante Services ein. Sie sind Bestandteil der MoSAIC Plattform, um einen kontinuierlichen Wissenstransfer zwischen Projekten und Prototypen zu realisieren.

Szenario: Ebenso wie die DPs wirkt sich auch das gewählte Szenario auf die Services aus. Darüber hinaus wirkt das Szenario zusätzlich auf die Simulation ein. Insbesondere das Gestalten von Szenarien mit mehreren

menschlichen Fahrern birgt aufgrund der gesteigerten Komplexität und dem Umstand des freien Willens der Fahrer einen erhöhten Abstimmungs- und Testaufwand. Dieser ist notwendig um sicherzustellen, dass während der Simulation des Szenarios auch die Situationen auftreten oder herbeigeführt werden, die ein Untersuchen der relevanten Fragestellungen ermöglichen. MoSAIC unterstützt die Gestaltung von Szenarien über Tools [19][20] und das Sicherstellen von Wissensaustausch zwischen den Projekten als zentrale Schnittstelle.

Versuchsleitung: Abschließend verfügt das MTT MoSAIC über einen Leitstand der es ermöglicht, das Verwalten von Versuchen, dessen Szenarien und das Aufzeichnen von Daten zentral zu koordinieren. Die Hardwareinfrastruktur des Leitstandes ist dafür geeignet mit wenigen Versuchsleitern mehrere Versuchspersonen effizient anzuleiten und zu beobachten.

Das MTT MoSAIC wurde im Projekt D3CoS zusammen mit dem MTT Dominion erstmals zur Entwicklung eines kooperativen Fahrstreifenwechselassistenten eingesetzt. MoSAIC konnte dabei erfolgreich den Entwurf, die Implementierung und den Test des neuen kooperativen ADAS in Form von drei Experimenten unterstützen und beschleunigen. Über D3CoS hinaus konnte die Simulationsplattform bereits erfolgreich in dem nationalen Projekt UR:BAN [P6] für die Durchführung eines Versuches zur Analyse der Interaktion mehrerer menschlicher Fahrer in einem Ampelszenario eingesetzt werden.

Der Fokus der aktuellen MoSAIC Simulationsplattform liegt auf der Automobil-Domäne. Durch den modularen skalierbaren Aufbau der Plattform ist ein Einsatz in anderen Domänen aber ebenfalls möglich.

5. ANWENDUNG FÜR KOOPERATIVEN FAHRSTREIFENWECHSELASSISTENT

Zur Erprobung und Evaluierung der D3CoS Methodik und der D3CoS MTTs wurden im Rahmen des Projektes DCoS Prototypen in den Domänen Automobil, bemannte und unbemannte Luftfahrt sowie Seefahrt gestaltet und entwickelt. Das folgende Beispiel beschreibt die Entwicklung eines kooperativen Fahrstreifenwechsellassistenten (C-LCA) in der Automobil-Domäne. Der abschließende C-LCA Prototyp wurde innerhalb von drei vollen Entwicklungszyklen entwickelt. Am Ende jedes Zyklus wurde ein Feedback an die MTT Entwickler gegeben, um ein kontinuierliches verbessern der MTTs im Projekt zu ermöglichen. Im Folgenden werden die D3CoS Designschritte sowie deren Ergebnisse für den C-LCA Prototypen zusammengefasst.

5.1. DCoS Composition

Die detaillierte DCoS Definition des C-LCA wurde über drei Entwicklungszyklen hinweg aufgestellt. Zu Beginn des ersten Zyklus stand die Idee Car-to-Car Kommunikation [2] als Kommunikationsmedium zu verwenden, um über den Austausch von Daten zwischen Verkehrsteilnehmern zur besseren Abstimmung in schwierigen Situationen ein kooperatives Netzwerk zu bilden.

Als Szenario (siehe BILD 9) wurde hierfür der Wunsch eines Verkehrsteilnehmers (V1) zum Wechseln auf einen gut gefüllten Fahrstreifen gewählt. Um diesen Wechsel möglichst effizient und sicher durchführen zu können, sollte V1 eine Kooperation mit einem Fahrzeug (z.B. V2) auf dem Zielfahrstreifen anstreben. Das Fahrzeug V2, welches auf die Kooperation eingeht, sollte dann vor sich eine Lücke öffnen, die für V1 groß genug ist, um den Fahrstreifenwechsel sicher durchzuführen.

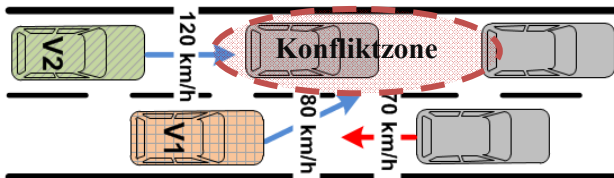


BILD 9 Vereinfachte Problemdarstellung der Ausgangssituation des Szenarios für den C-LCA

Um die Kooperation von V1 mit V2 effizient zu gestalten, wurde vorgesehen, dass beide an der Kooperation beteiligte Fahrzeuge über den C-LCA verfügen. Der C-LCA übernimmt dabei die Aufgaben einen geeigneten Kooperationspartner zu ermitteln, diesen dem eigenen Fahrer zu kommunizieren und auf Wunsch des Fahrers mit der Bitte um Kooperation in Kontakt zu treten. Kommt eine Kooperation zwischen V1 und V2 zustande, unterstützt der C-LCA bei der Erkennung ob eine ausreichend große Lücke geöffnet wurde. Zur Vorbereitung der Lücke unterstützt der C-LCA den Fahrer von V2 je nach eingestelltem Automationsgrad über Hinweise zum Lückenstatus oder öffnet die benötigte Lücke mittels automatischer Geschwindigkeitsregulierung selbstständig. Je nach Automationsgrad kann danach der Fahrer von V1 den Fahrstreifenwechsel manuell durchführen oder an den C-LCA abgeben, der den Fahrstreifenwechsel dann automatisch durchführt.

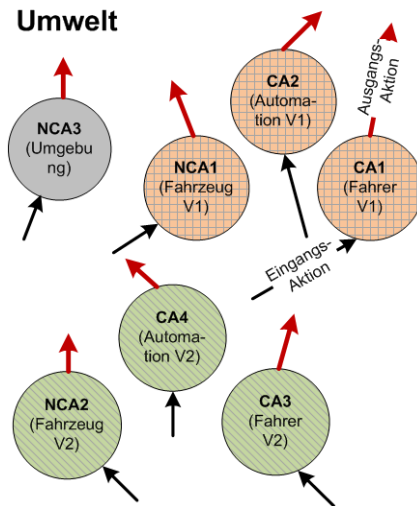


BILD 10 Zerlegung des Szenarios auf Agentenebene ohne Arbitr

Betrachtet man das beschriebene Szenario (siehe BILD 9) lässt es sich zerlegen in Agenten (siehe BILD 10), deren Ressourcen (z.B. Sensoren) und in zu bewältigende Aufgaben (z.B. sicherer Fahrstreifenwechsel durchführen). Auf Basis dieser Zerlegung lassen sich anschließend die TARFs definieren, welche erlauben das Szenario bzw. die Ausgangsproblemstellung zu lösen. Vereinfacht dargestellt sind folgende TARFs notwendig um das beschriebene Szenario zu lösen:

- Kooperationsbedarf erkennen
- Kooperation initiieren
- Kooperation beibehalten
- Kooperation beenden

5.2. DCoS Interaction

Basierend auf der Definition des C-LCA wurde über mehrere Zyklen hinweg die Interaktion des DCoS gestaltet. Besonderes Augenmerk wurde dabei auf die Interaktion zwischen Mensch und Maschine gelegt. Aber auch die Interaktion zwischen Maschine und Maschine spielt bei der Kommunikation zwischen den Fahrzeugen eine Rolle. Zusammengefasst läuft die Kommunikation in dem beschriebenen Szenario wie folgt ab:

- 1) Maschine (V1) informiert Menschen (V1) darüber, dass der Bedarf für eine Kooperation erkannt wurde.
- 2) Mensch (V1) teilt Maschine (V1) mit, dass er eine Kooperation wünscht.
- 3) Maschine (V2) wird von Maschine (V1) angefragt und teilt Mensch (V2) die Bitte um Kooperation mit.
- 4) Mensch (V2) bestätigt die Kooperation der Maschine (V2), was sie Maschine (V1) mitteilt.
- 5) Maschinen informieren Menschen darüber, dass eine Kooperation initiiert wurde.
- 6) Maschine (V2) hilft Mensch (V2) die Lücke zu öffnen und teilt mit, wenn Lücke ausreichend groß für V1 ist.
- 7) Maschine (V1) erkennt, dass Lücke groß genug ist und teilt dies Mensch (V1) mit.
- 8) Mensch (V1) vollführt den Fahrstreifenwechsel oder übergibt diese Aufgabe an Maschine (V1).
- 9) Maschine (V1) beendet Kooperation mit Maschine (V2).
- 10) Beide Menschen werden über das Ende der Kooperation durch die Maschinen informiert.

Jeder der zehn Interaktionspunkte zwischen den Agenten muss dabei im Detail gestaltet werden. Bei der Gestaltung kann das MTT DP-Katalog unterstützen indem es bei der Auswahl von geeigneten DPs und damit bei der Wahl eines Problemlösungsverfahrens unterstützt.

Beispiel: In Interaktionspunkt 4 bis 9 befinden sich beide menschlichen Fahrer (Agenten) in einer Kooperation miteinander (TARF: Kooperation beibehalten). Im Verlauf dieser Kooperation muss sichergestellt werden, dass den beteiligten Agenten bewusst wird und bleibt, dass sie Teil einer Kooperation sind. Außerdem sollten die Agenten darüber informiert werden, wer ihr Kooperationspartner ist. Das DP *Allocation to Cooperative Population* liefert dazu Lösungsansätze. Im konkreten Beispiel des C-LCA wurde das Bestehen einer Kooperation über Funksymbole im Head Up Display des Fahrzeuges angezeigt. Die Funksymbole pulsieren während einer Kooperationsanfrage und werden im Anschluss für die Dauer der Kooperation statisch dargestellt. Zusätzlich vermitteln die Funksymbole einen groben Eindruck über den Kooperationspartner (siehe BILD 11). In dem in BILD 11 gezeigten Beispiel ist der Kooperationspartner ein Fahrzeug, das sich auf dem linken Nachbarfahrstreifen hinter V1 befindet. Zeitgleich informiert die rote Fläche vor dem Kooperationspartner darüber, dass noch keine ausreichend große Lücke zum sicheren Fahrstreifenwechsel vorhanden ist. In der Mitte der roten Fläche beschreibt weiter ein animiertes Symbol (rotierender Kreis), dass der Kooperationspartner die Lücke aktuell öffnet. Die Gesamtkonstellation der Anordnung wird als 9er Schema [6] (Seite 192 ff.) bezeichnet. In diesem Schema können in acht Sektoren um das eigene im Zentrum befindliche Fahrzeug herum, grobe, räumlich aufgelöste ADAS relevante Informationen dem Fahrer zur Verfügung gestellt werden.

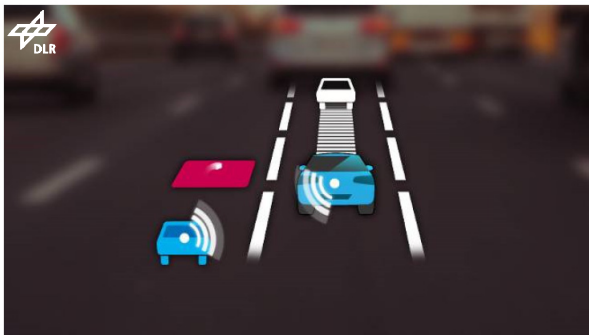


BILD 11 Entwurf des C-LCA HMI unter Berücksichtigung des DP „Allocation to Cooperative Population“ aus Sicht von V1

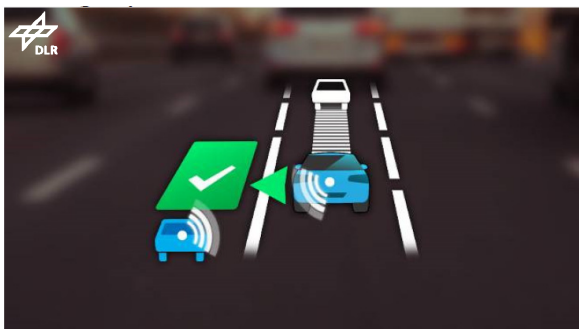


BILD 12 C-LCA HMI Entwurf, der beschreibt, dass eine Lücke zum sicheren Fahrstreifenwechsel vorhanden ist für Fahrzeug V1 im Zentrum der Anzeige

BILD 12 informiert den Fahrer von V1, dass eine geeignete Lücke für einen sicheren Fahrstreifenwechsel vom Kooperationspartner geöffnet wurde. Gleichzeitig wird über ein blinkendes grünes Dreieck dem Fahrer V1 eine Handlungsempfehlung zum Wechseln in die geöffnete Lücke (siehe Interaktionspunkt 7) gegeben.



BILD 13 C-LCA HMI aus Sicht von Fahrzeug V2 bei Beendigung der Kooperation inklusive einer Danksagung von V1 für den erfolgreich durchgeführten kooperativen Fahrstreifenwechsel

Ein weiteres Beispiel für den Einsatz eines DP: *Rewarding* findet sich in BILD 13. In diesem wird der kooperierende Fahrer (V2) für seine Kooperationsbereitschaft in Form eines „Danke“ durch die Maschine (V1) entlohnt. Gleichzeitig ist mit der Entlohnung ein klarer Abschluss (Interaktionspunkt 10) der Kooperation gegeben (siehe auch TARF: Kooperation beenden).

5.3. DCoS Interfaces

Das während der drei DCoS Interaction Zyklen entstandene HMI, sowie die definierte Kommunikation zwischen den Maschinen wurde in den DCoS Interfaces Zyklen implementiert. Als Simulationsplattform und Architektur für die Realisierung wurden die MTTs MoSAIC und Dominion verwendet. BILD 14 zeigt den resultierenden Prototypen am Ende des D3CoS Projektes. In dem Bild befindet sich der Fahrer von V2 im Vordergrund. Der Fahrer von V2 hat soeben die Lücke erfolgreich für das Fahrzeug V1 (im Hintergrund) geöffnet und bekommt dies von seiner Maschine im Head Up Display mitgeteilt. Der Fahrer von V1 hat diese Information ebenfalls bereits erhalten und bereitet aktuell den Fahrstreifenwechsel in die entstandene Lücke vor.



BILD 14 Finale Version des Fahrstreifenwechselsassistenten mit Head Up Display unter Verwendung einer transportablen Konfiguration der Simulationsplattform MoSAIC (vorne Fahrzeug V2, hinten Fahrzeug V1, demonstriert auf der D3CoS Abschlussveranstaltung).

5.4. DCoS Evaluation

Die Entwicklung des DCoS ADAS C-LCA Prototyp wurde in den durchlaufenden Entwicklungszyklen mittels mehrerer Experimente gestützt und evaluiert [6] (Seite 180 ff.).

Dabei wurde in einem ersten Experiment die grundsätzliche Kooperationsbereitschaft menschlicher Fahrer ohne Unterstützung durch ADAS, sowie deren Lösungsstrategien untersucht. Hierfür wurde das MTT MoSAIC mit zwei menschlichen Fahrern gleichzeitig eingesetzt.

In einem zweiten Experiment wurden einige HMI Elemente des späteren C-LCA HMIs untersucht. Ferner wurde das sozialpsychologische Phänomen der Verantwortungsdiffusion genauer beleuchtet und nach Ansätzen zur Vermeidung dieses Phänomens gesucht. Während des Experiments kam ein erster, vereinfachter C-LCA zum Einsatz, welcher innerhalb der Simulationsplattform MoSAIC mit drei menschlichen Fahrern gleichzeitig Anwendung fand.

Im dritten, finalen Experiment, wurde das fertige HMI des C-LCA in Form einer Usability Studie untersucht. Diese wurde in MoSAIC mit einem menschlichen Fahrer, abwechselnd als V1 und V2, (siehe BILD 9) sowie einem computergesteuerten Kooperationspartner realisiert.

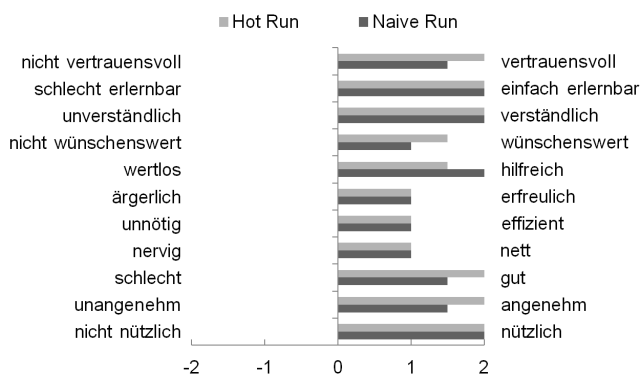


BILD 15 Bewertung des finalen HMI Designs des C-LCA aus dem dritten C-LCA Experiments [6]

BILD 15 zeigt die Bewertung des untersuchten C-LCA durch die Probanden im dritten Experiment. Insgesamt nahmen 8 Probanden an dem dritten Experiment teil. Die Beurteilung des C-LCA wurde nach einem naiven Versuchsdurchlauf (Naive Run), ohne vorherige Erklärung des HMIs und nach einem Hot Run erhoben. Vor dem Hot Run wurde das HMI des C-LCA allen Probanden im Detail vorgestellt, um als Ausgangsbasis für den zweiten Lauf das gleiche Verständnis des C-LCA HMIs bei allen Probanden sicherzustellen. Die Probanden haben jeweils nach den Versuchsdurchläufen den C-LCA ADAS in den im BILD 15 dargestellten Kategorien bewertet.

Das Ergebnis dieser Bewertung zeigt (BILD 15), dass der finale D3CoS C-LCA ADAS Prototyp sowohl im Naive als auch im Hot Run mit gut bis sehr gut durch die Probanden bewertet wurde. Damit ist der C-LCA Prototyp ein erfolgreiches Entwicklungsbeispiel für ein MAS DCoS unter Verwendung der D3CoS MTTs und Methodik.

6. ZUSAMMENFASSUNG UND AUSBLICK

In diesem Beitrag wurde das domänenübergreifende Projekt D3CoS vorgestellt und beschrieben, wie verteilte, kooperative Multi Agenten Systeme gestaltet werden können:

- 1) Bestimmung des Szenarios und Zerlegung in Agenten, Ressourcen und Aufgaben.
- 2) Bestimmung der Task Allocation and Resource Functions (TARF).
- 3) Gestaltung der Interaktion unter Verwendung von Design Pattern für die Mensch-Maschine und Maschine-Maschine Kommunikation.
- 4) Realisierung und Test der entwickelten Interaktion in Form von Interfaces für eine Multi Agenten taugliche Simulationsplattform und Softwarearchitektur.
- 5) Evaluierung des Ergebnisses und zyklische Weiterentwicklung bis zur Fertigstellung des Systems.

Zur Unterstützung der genannten Punkte wurden in dem Projekt eine Vielzahl von Methoden, Techniken und Werkzeugen (MTT) entwickelt, weiterentwickelt und verwendet. Drei dieser MTTs wurden im Rahmen dieses Beitrags genauer beleuchtet. Dies sind im Einzelnen: Design Pattern zur Gestaltung von kooperativen Systems, die Softwarearchitektur Dominion und die Simulationsplattform MoSAIC zum Entwickeln, Testen und Evaluieren von verteilten, kooperativen Multi Agenten Systemen.

Exemplarisch wurde abschließend die Anwendung der D3CoS Methodik und MTTs für die erfolgreiche Entwicklung eines kooperativen Fahrstreifenwechselassistenten beschrieben.

Ausblick: Die im D3CoS Projekt eingesetzten MTTs sowie die D3CoS Methodik werden durch das DLR-TS auch in zukünftigen Projekten verwendet werden.

Forschungsbedarf besteht insbesondere hinsichtlich der Verfeinerung der Methodik zur optimalen Gestaltung von Szenarien für Experimente in einer verteilten Simulation. Diese Optimierung ist notwendig um das Untersuchen von sozialpsychologischen Effekten wie der Verantwortungsdiffusion in Multi-Systemnutzer Szenarien besser unterstützen zu können.

Weiterer Forschungsbedarf besteht auch im Themenfeld der Metriken zum Messen und Analysieren von kooperativen Systemen, wie z.B. der Bestimmung ob eine Kooperation erfolgreich und effizient mit dem System durchgeführt werden konnte.

Auch der C-LCA bietet noch Raum für weitere Arbeiten. Zum einen bieten die Funktionen zur Auswahl eines geeigneten Kooperationspartners Optimierungspotential, da sie im Rahmen der bisherigen Tests nur eine untergeordnete Rolle gespielt haben. Zum anderen sollte in einem nächsten Schritt die Integration des C-LCA in ein umfassendes Assistenzsystem erfolgen, welches dann den Fahrer in einem größeren Szenarienumfang als nur dem Fahrstreifenwechselszenario unterstützt.

Eine fortführende Evaluierung und Validierung der aufgestellten DPs und das hinzufügen zusätzlicher DPs zum DP-Katalog bieten ebenfalls noch Raum für weitere Forschung.

7. REFERENZEN

- [1] Christoffersen, K.; Woods, D. D.: *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
- [2] CAR 2 CAR Communication Consortium: *C2C-CC Manifesto*: public report, 2007
- [3] Weiss G. (Hrsg.): *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. 2. Druck. MIT-Press, Cambridge MA, 2000
- [4] Tango F.; Baumann M.; Moravek Z.; Mengerlinghausen T. et al.: *D3-07 Generic DCoS Framework – Final Version*: D3CoS public Deliverable, 2011
- [5] Torschmied A.; Heers R.; Tango F.; Neujahr H. et al.: *D3-12 DCoS Development Methodology – Final Version*: D3CoS public Deliverable, 2014
- [6] D3CoS Consortium: *D2-02 Demonstrators – Final Version*: D3CoS public Deliverable, 2014, S. 156-197
- [7] D3CoS Consortium: *D3-08 Platform for Modeling and Simulation*: D3CoS public Deliverable, 2014, S. 36-41
- [8] Kelsch, J.; Temme, G.; Schindler, J.: *Arbitration based framework for design of holistic multimodal human-machine interaction*. AAET, Braunschweig, 6.-7. Feb. 2013
- [9] D3CoS Consortium: *D3-09 & D3-10 – Reference Designs and Design Patterns for Cooperation, DCoS State Inference and Adaptation & Multimodal Human-Machine Interfaces – Final Version*: D3CoS public Deliverable, 2014
- [10] 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; 03/2014
- [11] 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
- [12] 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, Braunschweig, 8.-9. Feb. 2012
- [13] Alexander, C.; Ishikawa, S.; Silverstein, M.; Jacobson, M.; Fiksdahl-King, I; Angel: *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press, 1977
- [14] Borchers, J: *A Pattern Approach to Interaction Design*. Wiley, New York, 2001, ISBN: 978-0471498285
- [15] Gacnik, J.; Häger, O.; Hannibal, M.: *A Service-Oriented System Architecture For The Human Centered Design Of Intelligent Transportation System*: Lyon, France, 2008
- [16] Fischer M., Richter A., Schindler J., Plättner J., Temme G., Kelsch J., Assmann D. and Köster K.: *Modular and Scalable driving Simulation Hardware and Software for the Development of Future Driver Assistance and Automation Systems*: Driving Simulation Conference, Paris 2014
- [17] D3CoS Consortium: *D3-11 Reference Architecture for DCoS Operations – Final Version*: D3CoS public Deliverable, 2014, S. 18-22
- [18] Büchel M.: *Erstellung einer dynamischen Sichtssimulation für maritime Hubschraubereinsätze*: Institutsbericht IB 111-2014/06, DLR Institut für Flugsystemtechnik, Braunschweig, 2014
- [19] Schindler, J., Hesse, T.; *Coping with Complex Driving Scenarios: Exploratory Scenario Design*: Driving Simulation Conference, Paris, 2014
- [20] Schindler, J., Kelsch, J., Heesen, M., Dziennus, M., Temme, G., Baumann, M.: *A Collaborative Approach for the Preparation of Cooperative Multi-User Driving Scenarios*: In: 10. Berliner Werkstatt Mensch-Maschine-Systeme, Berlin, Germany, 2013

Referenzierte Projekte:

- [P1] Link zur Webseite des Projektes D3CoS: <http://www.d3cos.eu/index.php/home>: besucht am 12.8.2014
- [P2] Link zur Webseite des Projektes Famos: <http://www.famos-project.eu/>: besucht am 12.8.2014
- [P3] Link zur Webseite des Projektes IMoST: <http://imost.informatik.uni-oldenburg.de/>: besucht am 12.8.2014
- [P4] Link zur Webseite des Projektes HAVEit: <http://www.haveit-eu.org>: besucht am 12.8.2014
- [P5] Link zur Webseite des Projektes Interactive: <http://www.interactive-ip.eu/>: besucht am 12.8.2014
- [P6] Link zur Webseite des Projektes UR:BAN: <http://urban-online.org/de/urban.html>: besucht am 12.8.2014