

Fakultät für Luft- und Raumfahrttechnik
Institut für Flugsysteme

Agent Supervisory Control als Ansatz zur auftragsbasierten Delegation eines kognitiven Agenten an Bord von UAV

M.Sc. Sebastian Clauß

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

Doktor-Ingenieurs (Dr. Ing.)

eingereichten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Peter Stütz
1. Berichterstatter: Prof. Dr.-Ing. Axel Schulte
2. Berichterstatter: Prof. Dr.-Ing. Florian Holzapfel

Diese Dissertation wurde am 08. Juni 2016 bei der Universität der Bundeswehr München, 85577 Neubiberg, eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 11. Oktober 2016 angenommen.

Tag der mündlichen Prüfung: 03. November 2016

Hohenfelde, den 08. November 2016

Für Charlotte

Danksagung

Die vorliegende Arbeit ist im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Flugsysteme der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München entstanden.

Mein besonderer Dank gilt Prof. Dr.-Ing. Axel Schulte, der mich als Hochschullehrer von meiner Bachelorarbeit bis zur Dissertation begleitet und angeleitet hat. Durch sein besonderes Engagement eröffnete er mir die Möglichkeit direkt im Anschluss an mein Studium als wissenschaftlicher Mitarbeiter an das Institut zurückzukehren. Als mein Doktorvater half er mir dabei meine Vorstellungen für einen eigenen Forschungsbeitrag zu konkretisieren und konsequent zu verfolgen. Über die gesamte Zeit war er mir dabei sowohl fachlich als auch menschlich ein wichtiger Orientierungspunkt, an den ich mich jederzeit mit Fragen richten konnte.

Des Weiteren bedanke ich mich herzlich bei Prof. Dr.-Ing. Florian Holzapfel für das Interesse an meiner Arbeit sowie die Übernahme des Koreferats. Unsere zahlreichen Gespräche vor und während des Promotionsverfahrens waren stets geprägt von offener Freundlichkeit und gaben mir eine wichtige, zusätzliche Perspektive auf meine Arbeit.

Bei meinen Institutskollegen möchte ich mich für die gute Zusammenarbeit und die freundschaftliche Atmosphäre über all die Jahre bedanken. Die gegenseitige Unterstützung und die starke Gemeinschaft in- und außerhalb des Dienstes haben die Zeit am Institut zu einem besonders schönen Lebensabschnitt werden lassen, den ich stets in guter Erinnerung behalten werde. Dies gilt besonders für die gemeinsamen Erlebnisse und Gespräche abseits des Dienstalltags im Rahmen von Konferenzen und bei Besuchen der verschiedenen Münchner Volksfeste.

Den Kollegen des SAGITTA Teams, Nikolaus Theißing, Denis Smirnov und Christian Hellert danke ich für die herzliche Aufnahme in ihren Kreis sowie für die gemeinsamen Bemühungen um die Integration unserer Beiträge, durch die verschiedene Aspekte der vorliegenden technischen Umsetzung erst ermöglicht wurden.

Georg Rudnick sowie Alexander Gebhardt und Peter Pratter danke ich stellvertretend für die zahlreichen Kollegen, mit denen ich insgesamt sieben Jahre lang gemeinsam Realflugversuche durchführen durfte. Die Tage und Nächte in der Bodenkrollstation sowie auf der Flight haben zwar viele Nerven gekostet, jedoch erinnere ich mich immer wieder gerne an sie zurück.

Madeleine Gabler, Melanie Sabella und Isolde Belz danke ich, dass sie mir während meiner Zeit am Institut stets mit Rat und Tat sowie immer mit einem freundlichen Lächeln zur Seite standen, wenn es darum ging die Hürden der Universitätsbürokratie zu bewältigen.

Mein Dank gilt auch meinen Eltern Ursula und Manfred, die mich über all die Jahre des Studiums und der Promotion selbstlos unterstützt haben. Besonders mein Vater stand mir immer beratend zur Seite und entwickelte sich über den Verlauf der Dissertation durch gute Ideen und kritische Fragen zu einer wichtigen und willkommenen Stütze.

Vor allem möchte ich jedoch meiner Frau Conny danken, die mir mit ihrer liebevollen und klugen Art half die Dissertation neben dem beruflichen Alltag fertig zu stellen. Sie verschaffte mir dabei stets eine klare Sicht für die wichtigen Dinge und tut dies bis heute.

Kurzfassung

Gegenstand dieser Arbeit ist die Entwicklung und Evaluierung eines kognitiven Agenten an Bord eines automatisierten UAV mit dem Ziel einen einzelnen menschlichen Operateur bei der UAV-Führung zu unterstützen und den Fähigkeitsumfang des technischen Systems zu steigern. Zu diesem Zweck wird ein Interaktionsverhalten des Agenten abgeleitet, welches sich insbesondere durch Rückführung symbolischer Informationen an den Operateur ausprägt. Das Interaktionsverhalten wird hinsichtlich seines Einflusses auf die Wahrnehmung des Agenten durch den Operateur und die Delegation von Aufgaben durch diesen an den Agenten evaluiert.

Als Grundlage dieser Dissertation werden zunächst die Grenzen heutiger (konventioneller) Automationsansätze beschrieben und anhand des Ansatzes intelligenter KI-Agenten sowie des Konzepts der kognitiven Automation Perspektiven aufgezeigt, wie diese Grenzen überwunden werden können. In diesem Rahmen werden verschiedene kognitive Systemarchitekturen und ihre Anwendung im Bereich der UAV-Automatisierung vorgestellt sowie architekturbedingte Einschränkungen hinsichtlich der Interaktion mit dem menschlichen Operateur diskutiert.

Die zentralen Aspekte dieser Arbeit lauten:

1. Die systematische Entwicklung eines Konzepts zur Beschreibung und Auslegung der Rollen eines kognitiven Agenten als Leiter eines automatisierten UAV-Systems, der durch den menschlichen Operateur geführt wird. Diese Form der Interaktion zwischen Mensch und Agent wird als *Agent Supervisory Control* bezeichnet. Hierbei wird ein bestehender Ansatz der auftragsbasierten UAV-Führung um eine auftrags- und situationsabhängige Rückführung symbolischer Informationen an den Operateur erweitert, die auf dem Fähigkeitsbewusstsein des Agenten beruht. Neben der Delegation von Aufträgen sieht das Konzept auch den manuellen Durchgriff des Operateurs auf nachgeordnete Automationsfunktionen vor, der als Ergänzung oder Ersatz UAV-eigener Fähigkeiten verarbeitet wird.
2. Die Implementierung eines prototypischen Agentensystems über eine hierarchisch gegliederte kognitive Agentenarchitektur, welche das vorgestellte Verhaltens- und Interaktionskonzept umsetzt und plattformunabhängig in bestehende UAV-Systeme integriert werden kann. Die dargestellte Wissensmodellierung in Form modularer Wissenskomponenten ermöglicht dem Agenten die zielgerichtete Verarbeitung seiner Umwelt und die Berücksichtigung dynamisch verfügbarer UAV-Automation.
3. Der Aufbau, die Durchführung und die Evaluierung eines Mensch-Maschine Experiments mit 14 Versuchsteilnehmern, die in einer simulierten, militärischen Aufklärungsmission als UAV-Operateure fungieren. Hierbei wird nachgewiesen, dass die gezielte Rückführung von Agenteninformationen den Operateur bei der UAV-Führung unterstützt und die Akzeptanz und das Vertrauen des Operateurs in den Agenten positiv beeinflusst.
4. Die Demonstration des technischen Reifegrades sowie der Plattformunabhängigkeit der Implementierung im Flugversuch durch die Integration des Agentensystems auf zwei unterschiedlichen UAV-Plattformen, die über eine einheitliche Bodenkontrollstation durch den Operateur geführt werden. Der Agent setzt dabei Aufträge jeweils plattformspezifisch um, indem symbolische Handlungspläne entsprechend der UAV-Fähigkeiten abgeleitet und die parametrische Umsetzung von Handlungen individuell angepasst werden.

Inhaltsverzeichnis

1	EINLEITUNG	1
1.1	ÜBERBLICK	3
2	GRUNDLAGEN	5
2.1	AUTOMATISIERUNG UNBEMANNTER LUFTFAHRZEUGE.....	5
2.1.1	Automationsmöglichkeiten in UAV.....	5
2.1.2	Menschliche UAV-Führung.....	9
2.1.3	Aufgabenverteilung in Mensch-Maschine-Systemen.....	15
2.1.4	Automatisierung operationeller UAV-Systeme	21
2.1.5	Grenzen konventioneller Automatisierung von UAV-Systemen.....	23
2.2	PERSPEKTIVEN DURCH INTELLIGENTE KI-AGENTEN	26
2.2.1	Agentenstruktur	28
2.2.2	Anwendung von Agenten in UAV	29
2.3	KOGNITIVE AUTOMATION.....	34
2.3.1	Delegation von Aufgaben an einen intelligenten Agenten	37
2.3.2	Modelle menschlicher Kognition	44
2.3.3	Kognitive Architekturen	48
2.3.4	Entwicklungslinie kognitiver Agenten zur UAV-Führung am IFS.....	56
2.4	PROBLEMDIDENTIFIKATION.....	59
3	KONZEPT.....	62
3.1	AUFTRAGSBASIERTE FÜHRUNG VON UAV	62
3.2	AGENT SUPERVISORY CONTROL	63
3.2.1	Fähigkeitsbewusstsein des Agenten	65
3.2.2	Auftragserteilung und Auftragsverarbeitung	67
3.2.3	Mensch-Agenten Interaktion.....	71
3.2.4	Fähigkeitssubstitution und Fähigkeitsaddition.....	73
3.3	KONZEPTUMSETZUNG DURCH ERWEITERUNG EINER BESTEHENDEN KOGNITIVEN SYSTEMARCHITEKTUR ...	74
3.3.1	Erweiterung der COSA ² Architektur	75
3.3.2	Resultierende Agentenarchitektur	76
4	IMPLEMENTIERUNG	78
4.1	DER KOGNITIVE AGENT	78
4.1.1	Die kognitive Systemarchitektur COSA ²	78
4.1.2	Reaktive Erweiterung für COSA ² - ReX.....	81
4.1.3	Resultierendes Agentendesign	86
4.2	WISSENSMODELLIERUNG IN COSA ²	87
4.2.1	Darstellung von situativem Wissen	88

4.2.2	Concepts.....	88
4.2.3	Motivational Contexts	93
4.2.4	Task Options	93
4.2.5	Procedures.....	96
4.2.6	I/O Interface	96
4.2.7	Function Server	97
4.3	AUTOMATIONSKOMPONENTEN DES UAV	97
4.3.1	<i>Automation Abstraction Layer (AAL)</i>	98
4.3.2	<i>Routenplaner Management (RPM)</i>	100
4.3.3	<i>Datenlink Management (DLM)</i>	103
4.3.4	<i>Sensor- und Perzeptionsmanagement System (SPMS)</i>	104
4.4	MENSCH-MASCHINE SCHNITTSTELLE.....	106
4.4.1	Missionsinterface.....	107
4.4.2	Sensorinterface	108
4.5	PLATTFORMUNABHÄNGIGKEIT DER TECHNISCHEN LÖSUNG	109
4.5.1	Interprozesskommunikation.....	110
5	EVALUIERUNG.....	113
5.1	EVALUIERUNG IM MENSCH-MASCHINE VERSUCH	113
5.1.1	Agentenkonfigurationen	113
5.1.2	Hypothesen.....	114
5.1.3	Methode	117
5.1.4	Ergebnisse	127
5.1.5	Diskussion.....	137
5.2	DEMONSTRATION IM FLUGVERSUCH	139
5.2.1	UAV–Demonstrator – Explorer	141
5.2.2	UAV–Demonstrator – OktoXL.....	143
5.2.3	Mobile Bodenkrollstation	144
5.2.4	Demonstrationsdesign.....	145
5.2.5	Ergebnisse	147
6	ZUSAMMENFASSUNG UND AUSBLICK	155
A	LITERATURVERZEICHNIS	157
B	ABKÜRZUNGSVERZEICHNIS.....	178
C	DOKUMENTE ZUR DATENERHEBUNG IM MENSCH-MASCHINE EXPERIMENT	181

1 Einleitung

Das Ziel dieser Arbeit ist es, den Einsatz von unbemannten Luftfahrzeugen (engl. *Unmanned Aerial Vehicle* - UAV) effektiver, flexibler und über Plattformgrenzen hinweg universeller zu gestalten. Zu diesem Zweck sollen die bereits bestehenden Fähigkeiten von UAV erweitert und einem einzelnen menschlichen Operateur in einer Form verfügbar gemacht werden, die ihm einen umfänglichen und allein-verantwortlichen Einsatz des Luftfahrzeugs ermöglicht. Über den in dieser Arbeit beschriebenen Ansatz der leitenden Kontrolle von UAV durch einen kognitiven Agenten (engl. *Agent Supervisory Control* - ASC) lassen sich diese Forderungen plattformvariabel erfüllen.

Motiviert von dem Anspruch, einen einzelnen menschlichen Operateur zur Umsetzung komplexer Missionsaufgaben mittels eines unbemannten Luftfahrzeugs zu befähigen, soll der Fähigkeitsumfang von UAV erweitert und der Mensch bei ihrer Führung geeignet unterstützt werden. Diese Fähigkeitssteigerung orientiert sich an der wachsenden Forderung nach einer Erhöhung des Autonomiegrades unbemannter Luftfahrzeuge, insbesondere im Bereich militärischer Einsatzszenarien von UAV, bei denen in den vergangenen Jahren ein bedeutsamer Anstieg von Komplexität verzeichnet werden konnte [Office of the Secretary of Defense 2005].

„The UA[S] must improve to higher levels of autonomy and the human to higher levels of management. This would migrate operational responsibility for tasks from the ground station to the aircraft, the aircraft gaining greater autonomy and authority, the humans moving from operators to supervisors, increasing their span of control while decreasing the manpower requirements to operate the UA[S].“ [Office of the Secretary of Defense 2005]

Operationelle UAV-Systeme werden heutzutage über verschiedene Flugphasen hinweg durch mehrere Operateure gleichzeitig geführt, die anhand fester Rollen ihre Führungsaufgaben arbeitsteilig wahrnehmen. Neben den erheblichen finanziellen und logistischen Herausforderungen dieses Multi-Operateur Betriebs lassen sich Grenzen der aktuellen Systemkonfigurationen sowie der Aufgabenverteilung zwischen den Operateuren identifizieren. Der im Folgenden verfolgte Ansatz zur Reduktion der Anzahl der UAV-Bediener auf einen einzelnen Operateur entspricht Anstrengungen zur Invertierung der Führungsspanne von UAV [Dixon et al. 2004], wonach zukünftig ein einzelner Operateur in die Lage versetzt werden soll, mehrere UAV gleichzeitig zu führen. Die zahlenmäßige Reduktion ändert dabei nichts an der zentralen Rolle des Menschen im UAV-System, weswegen eine rein technisch-autonome Lösung, also der gänzliche Verzicht auf einen menschlichen Operateur, weder zielführend wäre [Onken & Schulte 2010] noch derzeit von militärischen Nutzern gefordert wird.

„But without any doubt a human operator will remain at the center of the system and an all-unmanned air war, even in the long run, is not likely to become reality.“ [Stieglitz 2007]

Gemäß der Vorstellungen der Luftwaffe sollen UAV zukünftig zur Abbildung verschiedener Rollen und Fähigkeiten in Form von *Führung, Aufklärung, Wirkung und Unterstützung* genutzt werden [Reichelt 2013]. Auch andere Streitkräfte formulieren den Anspruch UAV in Zukunft in

einer breiten Menge komplexer Szenarien einsetzen zu können [Office of the Secretary of Defense 2005] [Lamonte 2009]. Die steigende Komplexität der Szenarien verlangt dabei Planungs- und Koordinierungsfähigkeiten in dynamisch veränderlichen Umgebungen sowie ein hohes Maß an Anpassungs- und Reaktionsfähigkeit in unvorhersehbaren Aufgabensituationen. In heutigen UAV-Systemen werden diese Fähigkeiten allein durch den Menschen bereitgestellt, der hierbei keine Unterstützung durch Automation erfährt. Vielmehr kontrolliert und koordiniert der Mensch alleine unterschiedliche Automationssysteme, von der Flugsteuerung bis zum Sensor- bzw. Effektoreinsatz.

Die Erhöhung der Missionskomplexität bei einer gleichzeitigen Reduzierung der Anzahl der UAV Operateure stellt zunächst einen Widerspruch dar, da dem System Fähigkeiten und Kapazitäten entzogen werden, es gleichzeitig aber ein breites Spektrum an Aufgaben erfüllen soll. Zur Kompensation dieses Verlusts an Unterstützung durch den Operateur und zur Ergänzung der technischen Fähigkeiten des Systems dient als klassischer Ansatz das Einbringen zusätzlicher Automationskomponenten, die vom Operateur zur Zielumsetzung genutzt werden können. Abschnitt 2.1.1 gibt hierzu eine Übersicht von Automationsmöglichkeiten heutiger UAV. Neben dem beabsichtigten Nutzen können sich mit der steigenden Automatisierung jedoch verschiedene Mensch-Maschine Probleme ausprägen oder verstärkt werden [Miller & Parasuraman 2007]. Abschnitt 2.1.5 untersucht hierzu die Grenzen der Automatisierung von UAV hinsichtlich ihrer Führung durch den Menschen. Andererseits wird die Kompensation bestimmter menschlicher Fähigkeiten (z.B. der Fähigkeiten zum kreativen Problemlösen) bisher mangelhaft adressiert, sodass deren Verlust in der Vergangenheit bereits in bis zu 15% der Flugunfälle die Hauptunfallursache darstellte [Schaefer 2003].

Ansätze zur Autonomiesteigerung von UAV und zur Unterstützung des Operateurs bei der Führung von Automationskomponenten durch technische Systeme existieren bereits in Forschung und Anwendung in Form intelligenter Agenten (vgl. Abschnitt 2.2). Zur Abbildung der bis heute rein menschlichen kognitiven Fähigkeiten bietet sich hierbei insbesondere kognitive Automation an [Onken & Schulte 2010], für die zahlreiche Architekturen bestehen und die in verschiedenen Anwendungen bereits erfolgreich in der UAV-Automatisierung eingesetzt wurde (vgl. Abschnitt 2.3). Hohe Autonomiegrade von UAV-Systemen sollen dabei nicht zur Abkehr von menschlicher Kontrolle führen, sondern verlagern die Aufgaben des Operateurs von der Steuerung automatisierter Einzelkomponenten des Luftfahrzeugs hin zur Systemleitung. Der Mensch bekleidet damit weiterhin die zentrale Rolle im UAV-System, wird jedoch durch technische Systeme entlastet. Bei der Auslegung von UAV-Systemen ist daher keine isolierte Betrachtung der technischen Komponenten und Wirkabläufe möglich. Vielmehr ist es notwendig, das Zusammenwirken von Mensch und System holistisch zu bewerten und eine geeignete Aufgabenverteilung und Interaktionsform umzusetzen.

„Instead of viewing autonomy as an intrinsic property of an unmanned aerial vehicle in isolation, the design and operation of autonomous systems needs to be considered in terms of human-system collaboration.“ [Defense Science Board 2012]

Bezüglich der Interaktion zwischen Mensch und Agenten existieren für den Bereich der UAV-Führung wenig Erfahrungswerte und keine etablierten Konzepte. Es wird jedoch angenommen, dass durch die wahrgenommene Intelligenz von Agenten Aspekte der menschlichen Interaktion,

wie z.B. das dem Agenten entgegengebrachte Vertrauen sowie das Erkennen und Verstehen von Handlungszielen, an Bedeutung gewinnen werden [Lee & See 2004]. Diese Interaktion wird über die reine Systemschnittstelle hinaus durch die Art der darunter liegenden Algorithmen zur Abbildung des Agenten bestimmt.

„As the human interfaces with the UA[S] at higher levels, the human must trust the UA[S] to do more. To develop and keep that trust, the human must be able to determine the intent of the UA[S]. Displays that show intent, as well as the algorithms which develop the intent, must be matured.“ [Office of the Secretary of Defense 2005]

Wird ein kognitiver Agent an Bord eines UAV eingesetzt, so ist dessen Anbindung und Interaktion mit bestehenden Automationsfunktionen zu klären, um die Umsetzung eigener Pläne des Agenten über die Nutzung existierender Systemkomponenten zu ermöglichen. Die Zahl der zivilen und militärischen UAV-Plattformen und Konfigurationen wird über die aktuell bereits existierende Anzahl hinaus noch weiter steigen [Blyenburgh 2014] [Department of Defense 2011]. Eine Plattformunabhängigkeit, die den Einsatz eines kognitiven Agenten auf unterschiedlichen UAV-Plattformen erlaubt, scheint daher erstrebenswert. Hierzu ist es notwendig, die Austauschbarkeit von Systemkomponenten sowie ihre dynamische Verfügbarkeit (z.B. bei Ausfällen durch technischen Defekt oder gezielte Deaktivierung) zu berücksichtigen.

Motiviert durch die oben formulierten Zielsetzungen für den zukünftigen Einsatz von UAV und basierend auf den bestehenden Defiziten moderner Systeme im Hinblick auf die Erfüllung dieser Zielsetzungen ergibt sich die Kernfrage dieser Arbeit wie folgt:

Wie kann ein einzelner Operateur ein hochautomatisiertes UAV in einer komplexen militärischen Aufklärungsmission im Sinne eines direkten Unterstellungsverhältnisses alleinverantwortlich führen?

1.1 Überblick

Kapitel 2 präsentiert zunächst Grundlagen, um aus der Kernfrage die Forschungsfragestellungen abzuleiten, die im Folgenden wissenschaftlich beantwortet werden. Hierzu werden die Automatisierung moderner UAV und die menschliche UAV-Führung beschrieben. Anschließend folgt die Delegation eines intelligenten KI-Agenten zur Überwindung automationsbedingter Mensch-Maschine Probleme. Die Abbildung kognitiver Fähigkeiten in Agenten gelingt über das Konzept der kognitiven Automation sowie über kognitive Architekturen, welche zur technischen Nachbildung menschlicher Kognition geeignet sind. Hierzu wird eine Auswahl gängiger Architekturen vorgestellt und eine Auswahl für die Implementierung der vorliegenden Arbeit getroffen.

Kapitel 3 beschreibt das Konzept zur Beantwortung der zuvor formulierten Forschungsfragestellungen. Unter dem Begriff der *Agent Supervisory Control* (ASC) wird ein kognitiver Agent als Untergebener des Operateurs und Leiter der nachgeordneten UAV-Systems eingebracht. Die Berücksichtigung von Automationsfunktionen geschieht dabei in Form von Fähigkeiten, welche vom Agenten abgeleitet und entsprechend ihrer Verfügbarkeit dynamisch

eingesetzt werden. Im Anschluss an die Auftragserteilung wird die Auftragsverarbeitung und -umsetzung durch den Agenten vor dem Hintergrund begrenzt-verfügbarer Verarbeitungsressourcen beschrieben. Die Betrachtung wird durch einen Ansatz zur temporären Ergänzung von UAV-Fähigkeiten im Rahmen der Auftragserfüllung durch den manuellen Eingriff des menschlichen Operateurs komplettiert. Das Kapitel schließt mit der Beschreibung der Konzeptumsetzung über eine Erweiterung der kognitiven Systemarchitektur COSA² zu einer mehrschichtigen Agentenarchitektur.

Die Implementierung eines Agenten anhand der Ansätze der ASC sowie der Komponenten der UAV-Automatisierung und der Mensch-Maschine Schnittstelle erfolgt in Kapitel 4. Basierend auf der kognitiven Systemarchitektur *COSA*² wird ein kognitiver Agent implementiert, welcher sowohl wissensbasierte Planungs- und Verarbeitungsfähigkeiten als auch reaktives Verhalten abbildet. Hierzu wird die Modellierung der a priori Wissenskomponenten, die im Folgenden die Grundlage des Verhaltens des Agenten darstellen, beschrieben. Zur Integration des Agenten in ein automatisiertes UAV-System werden anschließend Automationskomponenten und ihre Funktionen implementiert und über die aus ihnen abgeleiteten Fähigkeiten für den Zugriff des kognitiven Agenten verfügbar gemacht. Abschließend erfolgt die Vorstellung der Mensch-Maschine Schnittstelle zur Realisierung einer geteilten Missions- und Sensorführung sowie eines Ansatzes für die Plattformunabhängigkeit der technischen Lösung anhand hierarchischer Abstraktionsebenen.

Die Evaluierung der Implementierung des kognitiven Agenten und dessen Interaktionsverhalten in der Simulation sowie im realen Flugversuch erfolgt in Kapitel 5. In einer experimentellen Versuchsreihe wird der Einfluss des Agentenfeedbacks auf das Führungsergebnis menschlicher Operateure untersucht. Zu diesem Zweck wird ein Mensch-Maschine Experiment mit unterschiedlichen Ausprägungen des Agentenverhaltens in einer Simulationsumgebung durchgeführt. Der Nachweis im praktischen Flugversuch erfolgt anhand zweier mini-UAV Flugdemonstratoren, eines Starrflüglers sowie eines Oktokopters, auf denen ein Agent mit gleicher Wissensbasis eingesetzt wird und jeweils angepasstes Verhalten zur Auftragsumsetzung zeigt.

Kapitel 6 fasst abschließend die Ergebnisse dieser Arbeit zusammen und formuliert einen Ausblick auf mögliche Forschungsansätze im Rahmen der Unterstützung von Operateuren und der Fähigkeitssteigerung von UAV.

2 Grundlagen

2.1 Automatisierung unbemannter Luftfahrzeuge

In diesem Kapitel wird der Einsatz technischer Systeme zur Automatisierung unbemannter Luftfahrzeuge und deren Einfluss auf die menschliche UAV-Führung beschrieben. Hierzu werden zunächst Aufgabengebiete vorgestellt, in denen moderne UAV eingesetzt werden sowie die Automation, welche hierfür zur Verfügung steht (Abschnitt 2.1.1). Es folgt die Beschreibung unterschiedlicher Formen der Führung automatisierter UAV-Systeme (Abschnitt 2.1.2), wobei im Schwerpunkt das Konzept der *Human Supervisory Control* betrachtet wird. Anschließend werden Möglichkeiten zur Aufgabenverteilung zwischen Mensch und Automation dargestellt (Abschnitt 2.1.3). Abschnitt 2.1.4 betrachtet dazu exemplarisch drei operationelle UAV-Systeme im Hinblick auf ihre Automationskomponenten sowie ihre jeweiligen Führungskonzepte. Anhand der Beschreibung von durch Automation induzierten Mensch-Maschine Problemen wird abschließend eine Betrachtung der Grenzen konventioneller Automation in UAV durchgeführt (Abschnitt 2.1.5).

Für den Begriff der *Automation* existieren in der Literatur unterschiedliche Definitionen. Im Rahmen dieser Arbeit wird Automation gemäß [Parasuraman & Riley 1997] als Maschine (typischerweise als Computer) aufgefasst, die eine Funktion wahrnimmt, welche vormals vom Menschen erfüllt wurde. Diese Definition stimmt mit denen von [Hacker & Sachse 2013] und [Parsons 1995] überein. Zur besseren Differenzierung wird in Abschnitt 2.2 der Begriff der konventionellen Automation im Sinne von Funktionen mit vollständig beschreibbaren Zustandsräumen eingeführt und in Abschnitt 2.3 gegenüber der sog. kognitiven Automation abgegrenzt.

„The execution by a machine agent (usually a computer) of a function that was previously carried out by a human. What is considered automation will therefore change with time. (...) Today's automation could well be tomorrow's machine.“ [Parasuraman & Riley 1997]

Als Motiv für die fortschreitende Automatisierung technischer Systeme gelten der evolutionäre Druck durch die Forderung nach der Verfügbarkeit moderner, leistungsfähigerer Automation und eine Reaktion auf bestehende Fehler, die durch Automation abgemindert oder verhindert werden sollen [Timpe et al. 2002]. Das Ziel ist die Maximierung der Leistungsfähigkeit (durch die Vermeidung menschlicher Fehler und eine Fähigkeitssteigerung des technischen Systems), die Minimierung von Risiken sowie die Reduktion von Personal und Kosten für den Betrieb des Systems.

2.1.1 Automationsmöglichkeiten in UAV

Die Anzahl der Aufgabengebiete, in denen bemannte Luftfahrzeuge durch unbemannte Systeme ergänzt oder ersetzt werden, steigt im militärischen als auch im zivilen Umfeld, seit ihren Anfängen in den 1930er Jahren, stetig an [Petermann & Grünwald 2011]. Ausgehend von ferngesteuerten Luftfahrzeugen zur Zielerfassung wurden UAV zu Aufklärungsplattformen und darüber hinaus zu komplexen und vielseitigen Einsatzmitteln weiterentwickelt. Vor allem seit dem ersten Golfkrieg *Desert Shield / Desert Storm* wurde die technische Entwicklung vor dem Hintergrund des verstärkten Einsatzes unbemannter Plattformen im Rahmen einer asymmetrischen

Kriegsführung stark vorangetrieben [Petermann & Grünwald 2011]. Heutzutage existieren auch zahlreiche UAV-Plattformen, die genutzt werden, um Wirkmittel zu tragen und einzusetzen. Bis 2010 sammelte die US Armee alleine mit den beiden UAV-Typen RQ-1 und MQ-1 Predator mehr als eine Million Flugstunden [Springer 2013].

[U.S. Department of Transportation 2013] gliedert die zivile Nutzung von UAV in öffentliche Aufgaben durch staatliche Stellen, kommerzielle Nutzung und private bzw. Freizeitnutzung. Hierbei werden folgende zivile Aufgabenfelder beschrieben, die bereits heute durch UAV abgedeckt werden oder zukünftig durch sie abgedeckt werden können:

- **Intelligence, Surveillance and Reconnaissance:** Jegliche Form von Aufklärung oder Überwachung von Zielobjekten mittels an Bord von UAV betriebener Sensorik sowie die Sensordatenauswertung. Die Spanne reicht von der Gewinnung von Einzelbildern bis zu Videodaten, von Echtzeitübertragungen bis zur lokalen Aufzeichnung der Daten und umfasst unbewegliche und bewegliche Zielobjekte.
- **Payload Delivery:** Der Transport von kleinen und großen Gegenständen und Waren sowie zukünftig Personentransport mittels UAV. Möglich ist der Anflug fester Zielkoordinaten oder die Verknüpfung von automatisierter Zielsuche und Zielflug.
- **Environmental:** Jegliche Vermessung, Überwachung (z.B. meteorologische Messungen) oder Beeinflussung der Natur aus der Luft.
- **Search and Rescue:** Rettungsaktionen in gefährlichen oder schwer erreichbaren Gebieten, unter möglicherweise menschenfeindlichen Bedingungen.
- **Communications Relay:** Permanente oder ad-hoc Kommunikationsrelays zur Datenübertragung mittels UAV können als Verbreitungsmedium für z.B. Radio und Fernsehen eingesetzt werden.
- **Robotics:** Der Einsatz von UAV als fliegende Roboterplattformen ermöglicht die ferngesteuerte bzw. automatisierte Manipulation von Systemen oder Gegenständen. Als Manipulatoren können hierbei, unter Berücksichtigung der Stabilität und Tragkraft der UAV, vielfältige Systeme Einsatz finden.
- **Recreation:** RC-Flugmodelle decken bereits heute eine Vielzahl von Funktionen von Full-Size UAV ab. Solche Modelle werden zur Freizeitgestaltung z.B. aber auch im Rahmen von Forschungsprojekten an Universitäten und anderen Forschungseinrichtungen als Versuchsträger und Technologieplattformen eingesetzt.
- **Unique Specialized Mission:** Es sind weitere Applikationen denkbar, die erst durch den Einsatz von UAV bedacht oder ermöglicht werden. Diese Kategorie deckt alle sonstigen Einsatzmöglichkeiten von UAV ab.

Unbemannte Luftfahrzeuge (UAV) werden nicht isoliert, sondern im Rahmen von UAV-Systemen (engl. *Unmanned Aircraft Systems* – UAS) etabliert, welche nach [Deutsche Gesellschaft für Luft- und Raumfahrt 2008] neben der Flugplattform u.a. auch eine Bodenkontrollstation und ein Datenübertragungssystem zwischen Boden- und Luftsegment umfasst. Das breite Spektrum an Aufgaben wird durch eine große Anzahl verschiedener UAS abgedeckt, die eine definierte Menge an Funktionen entsprechend der jeweiligen Einsatzart bereitstellen. Die Bundeswehr setzt derzeit insgesamt fünf verschiedene UAS zu Aufklärungszwecken ein, die sich hinsichtlich ihrer Größe und Reichweite unterscheiden [Petermann & Grünwald 2011]. Die US Armee verfügt momentan über neun Systemtypen, die neben der Aufklärung auch zum Wirkmitteleinsatz genutzt werden

[Petermann & Grünwald 2011]. [Blyenburgh 2014] listet für das Jahr 2013 insgesamt eine Stückzahl von 1708 produzierten UAS (im Jahr 2012 1581 Stück) für zivile und militärische Einsatzzwecke. Zur sprachlichen Abgrenzung kommerziell eingesetzter, ziviler UAV-Systeme wurde die Bezeichnung *Remotely Piloted Aircraft System* (RPAS) etabliert, welche den Anspruch regulativer Behörden hervorhebt, solche Systeme rechtlich als vollwertige Luftfahrzeuge zu betrachten [Blyenburgh 2014].

Analog zur Aufgabenvielfalt der UAS ist die Gestaltung ihrer technischen Umsetzungen und der Möglichkeiten zur Steuerung durch den Menschen innerhalb der Bodenkontrollstationen uneinheitlich. Durch die vergleichsweise junge Technik fehlen etablierte Konzepte bezüglich Funktion und Steuerbarkeit von UAV. Anzeigen und Steuerelemente moderner Bedienerarbeitsplätze, gerade von großen UAV, ähneln daher weitestgehend Cockpits bemannter Luftfahrzeuge (vgl. Abschnitt 2.1.4).

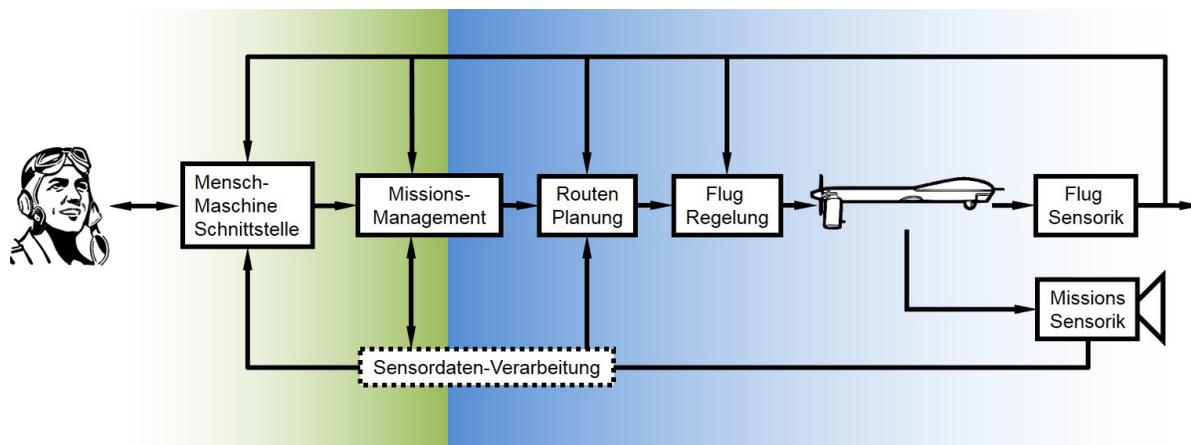


Abbildung 2-1: Schematische Darstellung gängiger Arten von Automation in einem UAV-System

Zur Bereitstellung technischer Fähigkeiten existiert in modernen UAS eine Vielzahl verschiedener Automationselemente. Abbildung 2-1 zeigt die schematische Darstellung eines UAS zu Aufklärungszwecken mit den dafür wesentlichen Automationskomponenten.

- **Flugplattform:** Die Plattform des UAV umfasst das Airframe und die Grundsysteme des Luftfahrzeugs. Die Charakteristika der Plattform (Aerodynamische Konfiguration, Größe, Flugleistung, etc.) definieren u.a. die Flugzeit, die Traglast und die Fähigkeiten des UAV [Austin 2010][Nui 2011].
- **Flugsensorik:** Die Bestimmung des Flugzustands des UAV erfolgt über Inertialsensorik, positionsgebende Sensoren sowie Sensorik in den Grundsystemen des UAV. Zusätzlich dient die Verarbeitung der Signale elektrooptischer Sensoren und von Abstandsmesssystemen, in Form von RADAR oder LIDAR, zur relativen Positionsbestimmung im Raum. Die absolute Positionsbestimmung erfolgt vor allem über Satellitennavigation. Äußere Sensoren, wie z.B. zur Messung von Luftdruck und Temperatur, liefern Informationen über den Umweltzustand [Austin 2010][Kim et al. 2006][Pastor et al. 2007].
- **Flugsteuerung:** Die Flugsteuerung umfasst Systeme zur Flugregelung sowie zum Flugmanagement, welche in der kommerziellen Luftfahrt typischerweise über das *Flight*

Control System (FCS) sowie das *Flight Management System* (FMS) realisiert werden. Die Flugregelung umfasst dabei die automatische Stabilisierung, Bahnführung und die Navigation anhand von Sensordaten. Das Flugmanagement kann alternativ Eingaben des Operateurs oder Ausgaben weiterer Automation (z.B. eines Routenplaners) in Form von Wegpunktlisten, Trajektorien oder Autopilotenkommandos verarbeiten und umsetzen [Brockhaus et al. 2011] [Gupta et al. 2013]. Hierzu realisiert die Flugregelung einen Regelkreis um die Flugsteuerungselemente der Flugplattform [Brockhaus et al. 2011].

- **Routenplanung:** Routenplaner erlauben die automatisierte Erstellung von Flugpfaden, die über FMS bzw. FCS von UAV abgeflogen werden. Die Ausgabeform des Flugpfades orientiert sich an der jeweiligen Schnittstelle des eingesetzten FMS (z.B. in Form von Wegpunkten oder als Trajektorie). Die Routenerstellung berücksichtigt dabei Randbedingungen, welche aus aktuellen Lagedaten und Daten der Flugsensorik abgeleitet werden [Giesbrecht 2004]. Die Routenplanung kann vor dem Flugstart für die gesamte Mission oder während des Fluges für Teilabschnitte bzw. zur Umplanung erfolgen. Eine Routenoptimierung erlaubt die Bestimmung eines bezüglich einer definierten Kostenfunktion günstigsten Flugpfades [LaValle 2006]. Im Bereich der robotischen Anwendungen kann die Routenplanung auch die Bewegungsplanung von Manipulatoren berücksichtigen [Kuffner Jr. & LaValle 2000] [Hwang & Ahuja 1992].
- **Missionssensorik:** Niedrig und hoch-auflösende Sensorsysteme (u.a. in Form elektrooptischer, und Infrarotsensoren sowie RADAR und LIDAR) zur Erzeugung von Livedaten oder zur Aufzeichnung. Außerdem Nutzlast für konkrete Einsatzzwecke des UAV (z.B. chemisch-biologische Sensoren oder meteorologische Vermessungsgeräte) und Kameras zum Zweck der UAV-Steuerung durch den Operateur. Missionssensorik wird entweder direkt im Airframe verbaut oder in speziellen Bauteilen (z.B. Kameratürmen) integriert. In operationellen UAV hat die Missionssensorik einen typischen Gewichtsanteil von 10% bis 20% des Luftfahrzeugs [Austin 2010] [Eisenbeiss 2004].
- **Sensordatenverarbeitung:** Neben der Bereitstellung reiner Sensordaten können moderne UAV diese auswerten und aufbereiten, um die Ergebnisse dem Operateur oder technischen Systemen zu übermitteln. Die Verarbeitung der Sensordaten kann dabei an Bord des UAV oder am Boden, online oder offline erfolgen. Eine typische Anwendung ist hierbei die Objektdetektion anhand eigener Sensordaten, die im Weiteren z.B. zum Anflug und zur Verfolgung von Objekten dienen kann.
- **Missionsmanagement:** Moderne UAV implementieren derzeit zwei Formen von Systemen, die zum missionsintegrierten Einsatz genutzt werden. Zum einen existieren Systeme zur Missionsplanung vor dem Abflug, bzw. zur Anpassung von Missionen während ihrer Durchführung, die (teilweise) automatisiert durch das Luftfahrzeug (LFZ) umgesetzt werden. Hierzu werden Funktionen der Routenplanung, des FMS und der Sensorsteuerung genutzt. Zum anderen wird eine Steigerung der Automatisierung von Fähigkeiten des UAV, u.a. durch Funktionen zum automatisierten Starten und Landen (engl. *Automatic Take-Off and Landing* – ATOL) vollzogen. Gleiches gilt für die automatisierte Behandlung von Fehlerzuständen des UAV, wie es z.B. beim Verlust der Datenkommunikation in Form des Abflugs vordefinierter Ausweichrouten (engl. *Automated Recovery*) geschieht.

- **Mensch-Maschine Schnittstelle:** Die Mensch-Maschine Schnittstelle (engl. *Human-Machine Interface* – HMI) kann untergliedert werden in die Darstellung von Informationen des UAV für den Nutzer und die Eingabe von Führungskommandos des Operateurs an das System. [Austin 2010] nennt eine feiner gegliederte Liste von HMI-Subsystemen, die angepasst an das System und die Mission kombiniert werden können:
 - UAV-Flugsteuerung: Anzeigen und Eingabeelemente für die manuelle Steuerung des UAV im Flug oder für die Programmierung von Vorgaben für die automatisierte Flugführung.
 - Sensor Kontrolleinheit: Anzeigen zur Darstellung des Sensorstatus und möglicher Aufzeichnungen sowie Kontrollelemente zur Steuerung der Sensorik.
 - Datenlinksteuerung: Kontrolle und Einstellung der Kommunikationsschnittstelle durch den Operateur.
 - Navigationsanzeigen / Kartendarstellung: Anzeigen zur Überwachung der Position des Luftfahrzeugs auf einer Karte. Möglichkeit zur Missionsplanung und zur Durchführung der dafür notwendigen Berechnungen.

- **Kommunikationsschnittstelle:** Die Kommunikationsschnittstelle von UAS besteht im Wesentlichen aus einem Uplink für Kommandodaten aus der Bodenkontrollstation an das LFZ, einem Downlink für Status- und Sensordaten des UAV sowie für Daten der Missionssensorik [Austin 2010]. Die räumliche Dislozierung von Mensch und Luftfahrzeug macht die Kommunikationsfähigkeit zu einem zentralen Element des UAS. Als Kommunikationsmedium wird zu großen Teilen Funk verwendet, wobei prinzipiell auch Laser oder in niedrigen Flughöhen auch Glasfaserkabel zur Datenübertragung möglich sind [Clot 1999]. Bei großen Distanzen werden außerdem Satelliten zur Kommunikation eingesetzt. Die wesentlichen Kriterien der Kommunikationsschnittstelle von UAV sind ihre Flexibilität, Ausfallsicherheit, Bandbreite und Reichweite [Nonami et al. 2010]. Die Latenz, die Verfügbarkeit und die Integrität der Schnittstelle sind ebenfalls wesentlich.

2.1.2 Menschliche UAV-Führung

Unabhängig von Art und Umfang der eingesetzten Automation werden UAV stets durch mindestens einen menschlichen Operateur geführt. Die Form des Systemeingriffs ergibt sich dabei aus den Fähigkeiten und dem Umfang der Automation, der Gestaltung der Mensch-Maschine Schnittstelle sowie aus dem jeweiligen Einsatzzweck des UAV. Für die Umsetzung von UAS ist es daher notwendig, über die Technologie hinaus die Interaktion und die Aufgabenteilung zwischen Mensch und Automation zu berücksichtigen [Sheridan 2002].

„It should be made clear that all autonomous systems are supervised by human operators at some level, and autonomous systems’ software embodies the designed limits on the actions and decisions delegated to the computer.“ [Defense Science Board 2012]

Im Vergleich zur Führung bemannter Luftfahrzeuge zeichnet sich die UAV-Führung durch die räumliche Separation des menschlichen Operateurs und des Luftfahrzeugs aus. Der Operateur führt das UAV über eine räumlich dislozierte Mensch-Maschine Schnittstelle, wodurch nach [McCarley

& Wickens 2004] Einschränkungen für die Eingaben und vor allem die Wahrnehmung des Operateurs resultieren.

An die Stelle der unmittelbaren Umweltwahrnehmung durch einen Piloten, mittels derer Instrumentenanzeigen im Flug kontrolliert und ergänzt werden können, tritt bei UAV die gefilterte Darstellung ausschließlich solcher Sensorinformationen, die mittels Datenverbindung zur Bodenkontrollstation übertragen und dort zur Anzeige gebracht werden. Hierdurch können vor allem visuelle Umgebungseindrücke, Geräusche sowie kinästhetische und vestibuläre Sinneseindrücke durch den Menschen nicht wahrgenommen werden [McCarley & Wickens 2004].

„As compared to the pilot of a manned aircraft, thus, a UAV operator can be said perform in relative ‘sensory isolation’ from the vehicle under his/her control.“ [McCarley & Wickens 2004]

Die funkbasierte Datenübertragung führt durch ihre Bandbreite und Laufzeit zu einer Einschränkung der übertragbaren Informationen sowie zu Latenzen in einer Größenordnung, die eine kontinuierliche, manuelle Steuerung des Luftfahrzeugs nicht zulässt [Van Erp & Van Breda 1999]. Bei der Datenübertragung, z.B. über geostationäre Satelliten, liegt die Rundlaufzeit für Daten distanzbedingt bei mindestens 500ms. Bereits bei Verzögerungen von mehr als 100ms kann jedoch bei manuellen Regelungsaufgaben ein starker Rückgang der menschlichen Leistung beobachtet werden [Miall & Jackson 2006].

Die Latenz der Anzeigen und Systemeingaben in Abhängigkeit von der eingesetzten Kommunikationsschnittstelle sind ein wesentlicher Treiber für die Anpassung des Grads der Automatisierung bei der Flugführung heutiger UAV. Der Systemeingriff reicht hierbei von der kontinuierlichen, manuellen Steuerung mittels numerischer Zustandsanzeigen oder übermittelten live-Videodaten des UAV, über die Vorgaberegulung und die Selektion von Flugparametern durch den Operateur bis zur Programmierung ganzer Flugabschnitte, die selbstständig durch das Luftfahrzeug abgeflogen werden [McCarley & Wickens 2004]. Verfahren für die Flugphase des UAV unterscheiden sich zudem von denen der Start- und Landephase. Die Eingabegeräte zur UAV-Führung werden dabei der jeweiligen Eingriffsebene des Operateurs angepasst.

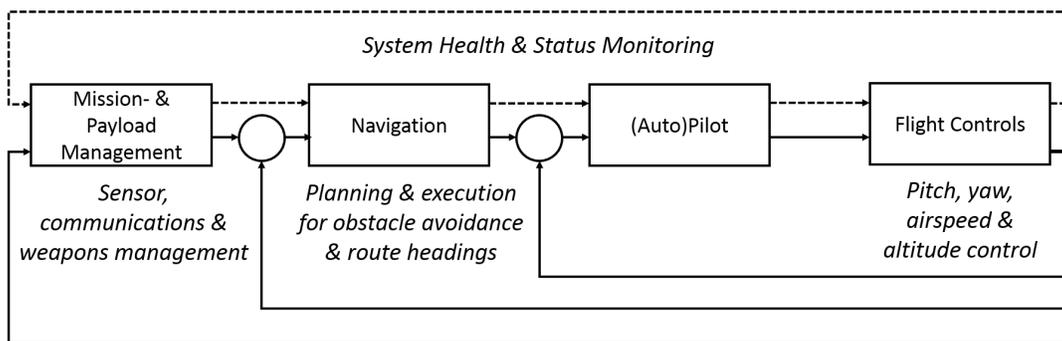


Abbildung 2-2: Hierarchische Führung eines UAV und seiner Subsysteme [Cummings et al. 2009]

Eine abstraktere Betrachtung der Führung eines hochautomatisierten UAV ist durch [Cummings et al. 2009] gegeben. Abbildung 2-2 zeigt die hierarchische Gliederung der Flugführung von UAV mit Hilfe von Komponenten der Flugregelung (Flight Controls), des Autopiloten, der Navigation sowie des Missions- und Payload Managements, welche um eine kontinuierliche Überwachung des

Systemstatus durch den Menschen erweitert wird. Der menschliche Systemeingriff zur UAV-Führung ist nicht explizit dargestellt, soll jedoch in den drei äußeren Führungsschleifen möglich sein. Die manuelle Flugführung durch den Menschen wird in dem Schema nicht betrachtet.

Die innere Führungsschleife beschreibt die kurzzeitige, lokale Steuerung des Luftfahrzeugs unter äußeren Einflüssen, wie z.B. den aerodynamischen Charakteristika des UAV anhand der Vorgaben an ein Autopilotensystem. *Navigation* beschreibt darüber hinaus die Wegpunkt- und Routenplanung durch den Menschen bzw. durch Automation, unter Berücksichtigung von Randbedingungen. Das *Mission- & Payload Management* geht über die reine Flugführung hinaus und beschreibt den Entscheidungsprozess zum missionsgerechten Einsatz der Flugzeugressourcen, wie z.B. der Sensoren und Wirkmittel sowie die Kommunikation mit anderen Luftfahrzeugen und der übergeordneten Führung bzw. äußeren Akteuren. *System Health- & Status Monitoring* beschreibt schließlich die kontinuierliche Überwachung des Systemzustands im Hinblick auf Fehlerzustände, die durch äußere Gegebenheiten oder durch Automationskomponenten entstehen und eine erfolgreiche Umsetzung der Zielvorgaben verhindern würden.

2.1.2.1 Leitende Kontrolle (Human Supervisory Control)

Die fortschreitende Automatisierung von UAV führt zur Verlagerung des menschlichen Eingriffs auf die äußeren Schleifen der hierarchischen Führungsstruktur [Landry 2012]. Gemäß der klassischen Einteilung von Pilotenaufgaben in *Aviation, Navigation, Communication* und *Systems Administration* [Jonsson & Ricks 1995] bedeutet dies die Konzentration menschlichen Handelns auf die letzten zwei Teilaufgaben. Der Mensch definiert Ziel- und Randbedingungen von Automationselementen, die Funktionen zur Kontrolle genau spezifizierter Prozesse bereitstellen, Rückgabewerte der Prozesse verarbeiten und dem Menschen zur Überwachung verfügbar machen. Der Mensch greift damit nicht direkt in den ausführenden Prozess ein, sondern steuert ihn vermittelt über die Automation, die für ihn als Schnittstelle dient. Die Analogie zwischen dieser Kommandierung von Automation und der Delegation von Aufgaben durch einen Vorgesetzten an einen menschlichen Untergebenen führte zu ihrer Bezeichnung als *Human Supervisory Control* (HSC) oder kurz *Supervisory Control* [Sheridan 1992]. Im deutschsprachigen Raum hat [Johannsen 1993][Johannsen 1994] für dieses Interaktionskonzept den Begriff der *Leitenden Kontrolle* geprägt. Unter dem Begriff der HSC werden im Bereich der Anthropotechnik Untersuchungen zu Mensch-Maschine Effekten bei der Führung automatisierter Systeme vollzogen, die sich von einer technisch-zentrierten Betrachtung (wie z.B. der Bestimmung von Stabilitätskriterien im Rahmen der Flugregelung) abheben.

„Die leitende Kontrolle kann als Interaktionskonzept für die Behandlung von Mensch-Maschine-Systemen aufgefasst werden. Danach führt der Mensch einen technischen Prozess grundsätzlich im interaktiven Durchgriff über mehrere meistens dezentral hierarchisch angeordnete Rechnerebenen.“ [Johannsen 1994]

In [Sheridan 1992] werden zwei verschiedene Definitionen der HSC formuliert. Im strengen Sinn handelt es sich um HSC, wenn ein oder mehrere Operateure Zielvorgaben für Automation definieren und ihre Ausgaben überwachen, welche ihrerseits einen inneren Regelkreis um einen zu steuernden Prozess schließt. Im weiteren Sinn kann HSC auch das Wirken eines Operateurs durch Automation bedeuten, welche lediglich Kontrollanweisungen und Messgrößen zwischen dem

Menschen und dem maschinellen Prozess vermittelt, wobei jedoch noch keine Regelkreise geschlossen werden.

Abbildung 2-3 vergleicht die leitende Kontrolle (im strengen Sinn) mit der manuellen Kontrolle (engl. *Manual Control*) und der voll-automatisierten Kontrolle (engl. *Full Automatic Control*) eines Prozesses. Bei der manuellen Kontrolle (Abbildung 2-3a) erfährt der Operateur keine Unterstützung durch Automation und interagiert direkt mit dem Prozess. Abbildung 2-3b repräsentiert die leitende Kontrolle im weiteren Sinn. Der Operateur kommandiert Vorgaben für die Automation, welche diese über den Prozess umsetzt. Ausgaben des Prozesses haben dabei jedoch, abgesehen vom Einschreiten des Operateurs, keinen Einfluss auf das Handeln der Automation. Abbildung 2-3c und 2-3d beschreiben beide die Leitende Kontrolle im engeren Sinn. Ausgabewerte des kontrollierten Prozesses werden durch die Automation auf die Größen der Prozessführung zurückgeführt. Der Unterschied zwischen den beiden Abbildungen ist was den Umfang der an die Automation übertragenen Aufgaben angeht lediglich quantitativer Natur. Während in Abbildung 2-3c nur kleinschrittige Aufgaben durch die Automation durchgeführt werden, sind diese in Abbildung 2-4d umfangreicher, wodurch selteneres Eingreifen durch den Menschen verlangt wird. Abbildung 2-3e zeigt schließlich die voll-automatische Kontrolle eines Prozesses, bei dem der Mensch keine Möglichkeit zum Eingreifen mehr besitzt, sondern lediglich die Ausgaben der Automation überwachen kann.

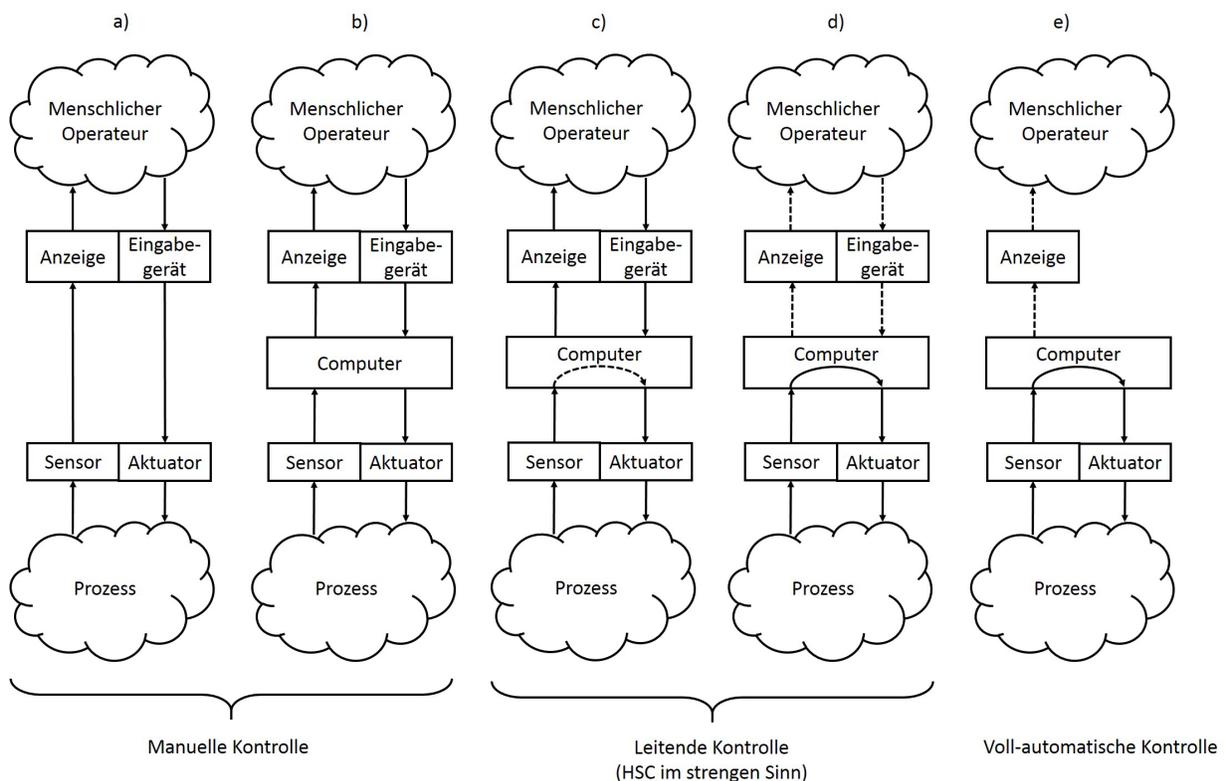


Abbildung 2-3: Leitende Kontrolle (im strengen Sinn), manuelle Kontrolle und voll-automatische Kontrolle eines technischen Prozesses [Sheridan 1992]

Die in Abbildung 2-3 dargestellte Beziehung zwischen Operateur und Automation lässt sich auf die Kommandierung und Überwachung beliebig vieler Automations-elemente erweitern. Hierbei

nimmt der Operateur die Rolle des menschlichen Supervisors ein, der die einzelnen Elemente koordiniert und gemeinsam zum Erreichen seiner eigenen Ziele einsetzt [Sheridan 2012]. In seiner Rolle als menschlicher Leiter erfüllt der Operateur nach [Sheridan 1992] im Allgemeinen die fünf Leitungsfunktionen (engl. *Supervisory Functions*): **Planen** (engl. *plan*), **Instruieren** (engl. *teach*), **Überwachen** (engl. *monitor*), **Eingreifen** (engl. *intervene*) und **Lernen** (engl. *learn*) [Herczeg 2014]. Die Funktionen werden dabei vom Operateur während der Leitenden Kontrolle in drei geschachtelten Kontrollschleifen durchlaufen (vgl. Abbildung 2-4).

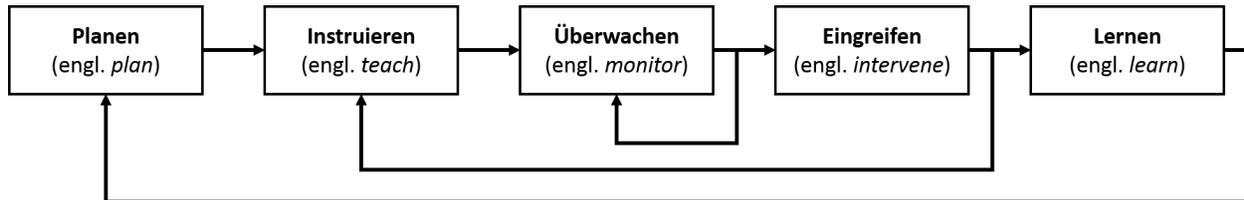


Abbildung 2-4: Leitungsfunktionen eines Operateurs als geschachtelte Kontrollschleifen
(aus [Sheridan 1992] übersetzt von [Herczeg 2014])

Beim **Planen** wählt bzw. verarbeitet der Operateur seine aktuelle Aufgabe und entwickelt Strategien zur Zielerreichung unter Berücksichtigung der zur Verfügung stehenden Automation. Hierfür nutzt der Operateur ein mentales Modell des physikalischen Prozesses und des Verhaltens der Automation.

Das **Instruieren** setzt entwickelte Strategien über Anweisungen an die Automation um. Hierfür werden einzelne Kommandos gewählt und in Kontrollhandlungen übersetzt. Für die Kommandierung benötigt der Operateur Wissen über die Kommandosprache der Automation, die angesprochen wird.

Überwachen beschreibt die Beobachtung der Prozessinformationen und das Verstehen des Systemzustands über eine Aufmerksamkeitslenkung. Der Abgleich von Ausgabewerten mit dem mentalen Modell des Operateurs führt zur Detektion von etwaigen Verhaltensabweichungen und kann zum **Eingreifen** führen. Eine Voraussetzung für die Detektion von Fehlern ist ein korrektes Situationsbewusstsein des Menschen.

Eingreifen meint die Anpassung einer vorangegangenen Kommandierung durch den Operateur als Reaktion auf detektiertes Fehlverhalten. Ein solches Fehlverhalten stellt sich für den Operateur als Abweichung zum erwarteten Systemverhalten dar, woraus das Nichterreichen der Zielvorgaben innerhalb existierender Randbedingungen abgeleitet wird. Abweichungen können zum einen aus fehlerhaften Eingaben des Operateurs resultieren, die ihrerseits sowohl auf unbemerkte Fehler bei der Eingabe bzw. auf mangelndes Verständnis der Eingabeschnittstelle oder des hinterlegten technischen Prozesses zurück zu führen sind. Alternativ resultieren Abweichungen aus zur Zielerreichung ungeeignetem Automationsverhalten. Dieses kann wiederum unvollständiges oder unpassendes Automationsdesign als auch technische Fehlfunktionen zur Ursache haben.

Systemerfahrung kann beim Operateur zum **Lernen** führen, welches ermöglicht, mentale Modelle der physikalischen Umgebung oder des Automationsverhaltens zu verfeinern. Erfahrung führt somit schließlich zu einer Anpassung des Operateurverhaltens. Lernen durch externe Wissensvermittlung, wie z.B. das Studium von Systembeschreibungen, ist in diesem Zusammenhang nicht Teil des Prozesses der Leitenden Kontrolle.

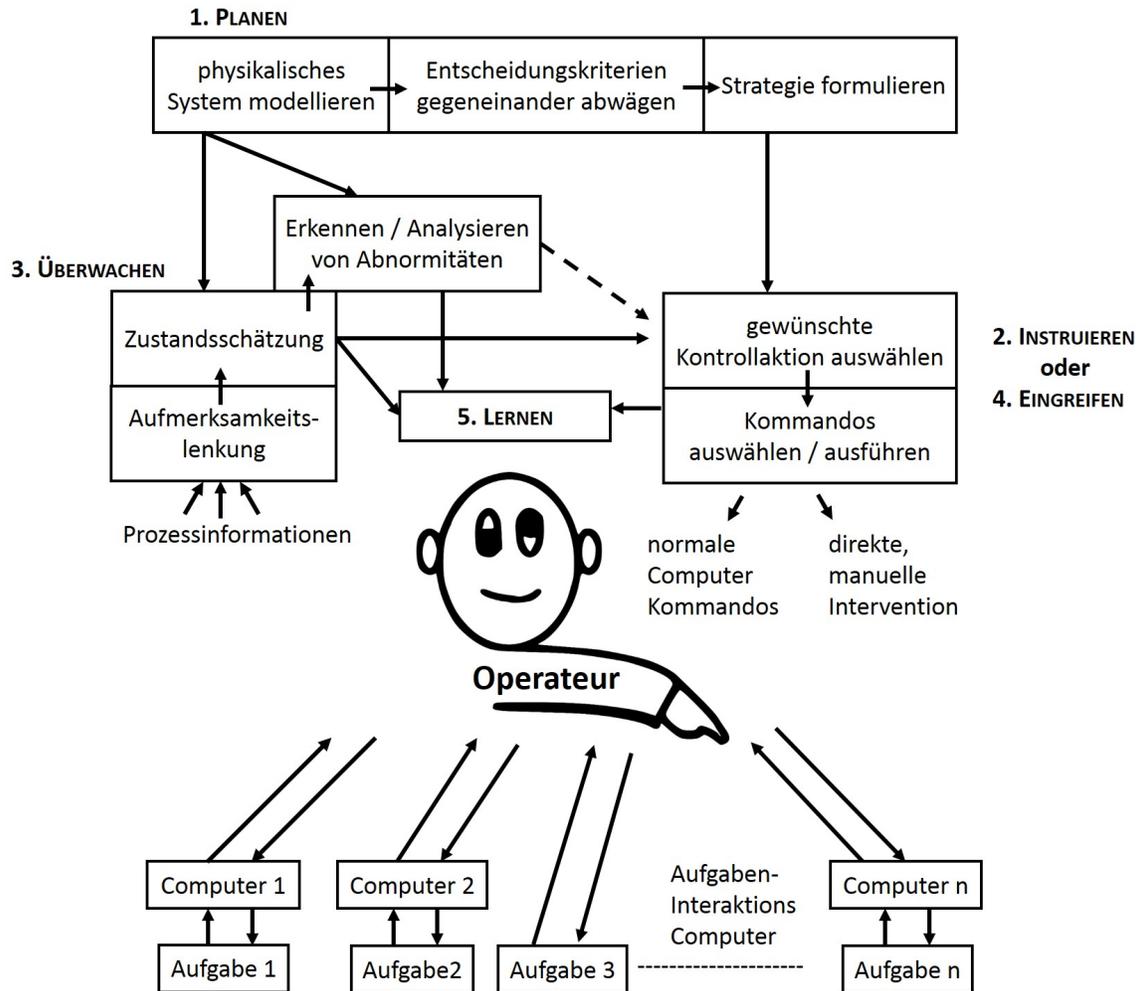


Abbildung 2-5: Flussdiagramm der Leitungsfunktionen des Operateurs (aus dem Englischen nach [Sheridan 1992])

Die Unterteilung in Kontrollschleifen gemäß Abbildung 2-4 lässt sich auch anhand der unterschiedlichen Zeitintervalle ihres Durchlaufs erklären [Sheridan 2012]. Die Überwachung geschieht meist fortwährend, wobei feine Korrekturen (**Eingreifen**) in größeren Abständen gemacht werden. Plananpassungen erfolgen im Allgemeinen in sogar noch größeren Abständen.

Abbildung 2-5 integriert die Leitungsfunktionen aus Abbildung 2-4 in ein Flussdiagramm, in dem ein Operateur gleichzeitig mehrere Prozesse über Automationselemente leitet. Beim **Planen** formuliert der Operateur hierbei eine Strategie, die über geeignete Anweisungen an die Prozesse kommandiert werden. Dies kann gemeinsam oder einzeln erfolgen. Anschließend überwacht der Operateur die Informationen, die von den Prozessen zurückgeführt werden. Der innere Kontrollkreis wird über die Überwachungsfunktion (bzw. deren Subfunktionen *Aufmerksamkeitslenkung* und *Zustandsschätzung*) geschlossen [Sheridan 1992]. Erkennt der Operateur Abweichungen des Systemzustands, kann er entweder über korrigierende Automationskommandos oder direkt, manuell intervenieren. Das **Lernen** wird dabei parallel zu den anderen Handlungen des Operateurs durchlaufen und basiert auf den Beobachtungen des Systemzustands, insbesondere auf Abweichungen des erwarteten vom tatsächlichen Systemverhalten sowie der menschlichen Systeminteraktion.

2.1.3 Aufgabenverteilung in Mensch-Maschine-Systemen

Das Konzept der Leitenden Kontrolle beschreibt die prinzipielle Interaktionsart zwischen einem Operateur und einem automatisierten System. Die Aufgabenverteilung zwischen Mensch und Maschine sowie die Interaktionsart des Menschen ist durch die fortschreitende Automatisierung technischer Systeme und ihrer Funktionen bleibender Forschungsgegenstand [Dekker & Woods 2002][Miller & Parasuraman 2003][Miller & Parasuraman 2007][Parasuraman et al. 2000] [Parasuraman 2000]. In diesem Abschnitt sollen bereits bestehende Ansätze zur Interaktion vorgestellt und im Hinblick auf ihren Nutzen für die Auslegung und die Beschreibung moderner automatisierter Systeme untersucht werden.

2.1.3.1 Automationsphilosophien

Die Entscheidung über die Automatisierung vormals manuell geführter Tätigkeiten ist im Hinblick auf den hieraus resultierenden Nutzen nicht trivial, wie [Sheridan 2000] für die Abhängigkeit der Ausführungszeit von der Aufgabenkomplexität darstellt. Demnach sei die Automatisierung von Funktionen nur im mittleren Komplexitätsbereich sinnvoll, da hier die Ausführungszeit durch Automation im Vergleich zur menschlichen Ausführung verringert werden kann. Seit Ende der 1940er Jahre wird bereits über die korrekte Aufgabenverteilung in automatisierten Systemen bzw. die passende Automatisierungsstrategie für Mensch-Maschine Systeme diskutiert. Hierbei wurden über die Jahre verschiedene Prinzipien abgeleitet, die jeweils auf unterschiedlichen Ansichten über menschliche Fähigkeiten und menschliches Verhalten beruhen. Teilweise werden diese Prinzipien deswegen als Automationsphilosophien bezeichnet [Hollnagel 2004].

Das einfachste Prinzip stellt hierbei die Automatisierung aller technisch realisierbaren Funktionen dar. Solche Funktionen eines Systems, die durch Automation abgebildet werden können, werden automatisiert – alle übrigen Funktionen bleiben dem Operateur zur manuellen Kontrolle. Dieses sog. **Left-Over** Prinzip beruht auf der Annahme, dass Automation prinzipiell effizienter eingesetzt werden kann als der Mensch [Hollnagel 2004]. Dieser rein technologisch motivierte Ansatz vernachlässigt dabei vollständig die Eigenschaften des menschlichen Eingriffs bzw. dessen Fähigkeiten bei der Erfüllung von Aufgaben mit sehr geringer oder hoher Komplexität. [Dekker & Woods 2002] sehen in der Substitution menschlichen Handelns durch Automation die Ursache für zu erwartende und unerwartete Mensch-Maschine Probleme, vor allem im Bereich der Luftfahrt.

„(...) the designer who tries to eliminate the operator still leaves the operator to do the tasks which the designer cannot think how to automate.“ [Bainbridge 1983]

Eine weitere Philosophie beschreibt die statische Gegenüberstellung der Fähigkeiten und Limitationen von Mensch und Automation, auf deren Grundlage die Funktionszuweisung für Aufgaben optimal geschehen könne. Dieses Prinzip wird als Kompensationsprinzip (engl. *Compensatory Principle*) bzw. mit *Men Are Better At – Machines Are Better At* (MABA-MABA) bezeichnet. Vorgaben für die Funktionsallokation stammen originär von [Fitts 1951] in Form der sog. *Fitts' List*, die im Rahmen der Entwicklung von Cockpits und Radarsystem für die US Army Air Force während und unmittelbar nach dem Zweiten Weltkrieg entstand. Seitdem sind verschiedene Listen mit spezifischen Fähigkeiten von Mensch und Maschine entstanden (u.a. [Lanc 1975][Sheridan 1988][Hoyos 1990]). Als wesentliche Stärke der Maschinen werden repetitive Aufgaben mit hohen Anforderungen an die Reaktionszeit oder die Nebenläufigkeit von

Handlungen beschrieben. Bei Menschen werden besonders kognitive Fähigkeiten der Antizipation, der Planung und der Problemlösung in komplexen Situationen als Stärke bewertet (vgl. u.a. [Hoyos 1990]).

„A more serious irony is that the automatic control system has been put in because it can do the job better than the operator, but yet the operator is being asked to monitor that it is working effectively.“ [Bainbridge 1983]

MABA-MABA geht gemäß [Dekker & Woods 2002] von der fälschlichen Annahme aus, dass eine menschliche Schwäche durch eine Stärke der Automation ersetzt und kompensiert werden könne. Vielmehr würden hierdurch neue menschliche Stärken und Schwächen hervorgebracht, wie z.B. durch [Bainbridge 1983] für unerwartete Effekte bei industriellen Prozessen dargestellt wird. Die statische Zuweisung von Stärken und Schwächen betrachtet Mensch und Automation als separate Entitäten, denen Funktionen beliebig filigran zugeordnet werden können. Hieraus würden jedoch zusätzliche Aufgaben für den Menschen resultieren, die bei MABA-MABA nicht betrachtet werden [Dekker & Woods 2002]. [Hoffman et al. 2002] argumentieren demgegenüber, dass es nicht primäres Ziel sein sollte menschliche Limitationen und Variabilität zu kompensieren, sondern Automation so einzusetzen, dass die Intelligenz, das Können und die Kreativität des Menschen in einer komplexen Umgebung bestmöglich genutzt werden können. Hierzu veröffentlichen sie als Gegenentwurf die sog. *Un-Fitts List*, die die menschliche Leistungsfähigkeit und den möglichen Nutzen von Automation formalisiert.

„Do not devalue the human in order to justify the machine. Do not criticize the machine in order to rationalize the human. Advocate the human-machine system in order to amplify both.“ [Hoffman et al. 2002]

Ein weiteres Prinzip stellt *Price's Decision Matrix* nach [Price 1985] dar, welches berücksichtigt, dass sich Stärken und Schwächen von Mensch und Maschine nicht gegenseitig ausschließen, sondern bestimmte Aspekte gleichermaßen gut oder schlecht von beiden ausgeführt werden können. [Price 1985] beschreibt einen zweidimensionalen Raum mit der menschlichen Leistung auf der X-Achse und der korrespondierenden Leistung von Automation auf der Y-Achse, welche jeweils von nicht-akzeptabel bis exzellent reichen. Die Leistung einer Funktion wird als Punkt im Raum notiert, welcher in einer von sechs möglichen Zonen liegt, die jeweils qualitativ unterschiedliche Implikationen für die Funktionszuweisung angeben. Durch die fehlende Beschreibung der expliziten Bewertung von Leistungen und der ungenauen Beschreibung der Zoneneinteilung findet *Price's Decision Matrix* in der Praxis jedoch wenig Anwendung.

Als dritte Philosophie lässt sich das **Komplementaritätsprinzip** (engl. *Complementary Principle*) definieren, bei dem Mensch und Maschine als miteinander arbeitende „kognitive Einheit“ kooperativ zusammenarbeiten. Das Ziel ist die Orientierung der Aufgabenteilung am Grad der Aufgabenerfüllung vor dem Hintergrund der Definition menschengerechter Aufgaben und situationsangepasster Flexibilität der Interaktion [Grote et al. 1999].

Die statische Bestimmung der komplementären Aufgabenteilung ist dabei im Allgemeinen nicht optimal möglich, da sich die Anforderungen der Aufgaben dynamisch verändern. Moderne, kognitive Ansätze untersuchen in diesem Zusammenhang die Automation unter Berücksichtigung

der kognitiven Fähigkeiten des Menschen für eine dynamische Funktionsallokation während der Aufgabenerfüllung.

Seit den 1980er Jahren wurden unter den Begriffen **Cognitive Engineering** [Norman 1986] oder **Cognitive Systems Engineering** [Rasmussen et al. 1994] Anstrengungen auf dem Gebiet solcher komplexer sozio-technischer Systeme zusammengefasst. [Hoffman et al. 2002] stellen in diesem Zusammenhang eine Vielzahl von Begriffen und Akronymen zusammen, die seither zur Benennung von Ansätzen auf diesem Feld verwendet werden und beschreiben die diesen zugrunde liegenden Konzepte und Forschungsfelder. [Billings 1997] zeigt mittels *Human-Centered Automation* Probleme bei der Nutzung von Automation in kommerziellen Luftfahrzeugen auf und beschreibt Ansätze zum verbesserten Informationsaustausch im Cockpit. [Hancock & Scallen 1996] schlagen *Adaptive Function Allocation* als Konzept zur dynamischen Zuweisung der Kontrolle von Aufgaben zwischen Operateur und Automation vor, um damit Beschränkungen statischer Zuweisung entgegen zu wirken. *Coactive Design* nach [Johnson et al. 2011b] basiert auf der gegenseitigen Abhängigkeit von Operateur und Automation in Form intelligenter Agenten. Handlungen von Mensch und Agent dienen einem gemeinsamen Ziel und beeinflussen sich gegenseitig. Der Fokus von *Coactive Design* liegt dabei auf dem gemeinsamen Handeln beider Akteure. [Onken & Schulte 2010] beschreiben schließlich das Konzept der *Cognitive Cooperation*, welches die Verarbeitung gemeinsamer Informationen und die selbstständige Rollenzuweisung und Zusammenarbeit zwischen einem menschlichen Operateur und kognitiven Systemkomponenten (vgl. Abschnitt 2.3) thematisiert. Der Schwerpunkt liegt hierbei auf der Verteilung der kognitiven Aufgabenlast zwischen Mensch und Automation.

2.1.3.2 Automationsgrade

Die komplementären Eigenschaften von Mensch und Automation können in Mensch-Maschine Systemen zu einer komplexen Verteilung von Aufgaben führen [Herczeg 2014]. Die Arbeitsteilung zwischen Mensch und Automation erfolgt dabei nicht streng entlang von Aufgaben- oder Funktionsgrenzen. Automatisierung ist keine binäre Operation, bei der Aufgaben nur vollständig entweder vom Menschen oder durch Automation vollzogen werden, sondern kann feiner granular sein. [Sheridan 2012] beschreibt hierbei einen Zusammenhang zwischen der Aufgabenentropie (als Begriff für die Beschreibbarkeit und Vorhersehbarkeit von zu bewältigenden Aufgaben) und einem sinnvollen Grad an automatisierten Funktionen. Mit abnehmender Aufgabenentropie sei ein höherer Grad an Automation erreichbar, da es besser möglich sei passende Funktionen zu implementieren. Bei Abweichungen von diesem Zusammenhang komme es hingegen entweder zu einer suboptimalen Nutzung technischer Möglichkeiten bzw. dem Einsatz von Automation, der für die Komplexität der Aufgaben unzureichend sei.

Automationsgrade (engl. *Levels of Automation* – LoA) werden verwendet, um die automatisierten Fähigkeiten eines Systems und die Interaktion des menschlichen Operateurs zu beschreiben. Ihr Sinn ist die Gestaltung und Beschreibung von Mensch-Maschine Systemen anhand von Modellen.

„[Der Automationsgrad] ist der Anteil der vom Automatisierungssystem übernommenen Funktionen und Einwirkungen auf den technischen Prozess.“ [Johannsen 1993]

In der Literatur finden sich verschiedene Skalen zur Beschreibung von Automationsgraden [Sheridan & Verplank 1978][Sheridan 1992][Billings 1997][Endsley & Kaber 1999][Sheridan

2002][Cummings et al. 2009]. Tabelle 2-1 zeigt eine Beschreibung von Automationsgraden anhand der menschlichen Interaktionen mit einem Computer im Rahmen eines einzelnen Entscheidungsschrittes für die Steuerung eines Unterwasser-Roboters [Sheridan & Verplank 1978]. Ausgehend von der rein menschlichen Entscheidungsfindung wird der Grad der Automation in zehn Schritten erhöht, wobei der Computer schließlich die vollautomatische Kontrolle übernimmt. Die Skala wird in [Sheridan 1992] und in [Sheridan 2002] weiterentwickelt und vereinfacht, wobei der Mensch in höheren Automationsgraden immer mehr aus der Führungsentscheidung ausgeschlossen wird.

Tabelle 2-1: Automationsgrade für einen Entscheidungsschritt in einem Mensch-Computer System nach [Sheridan & Verplank 1978]

-
1. Human does the whole job up to the point of turning it over to the computer to implement.
 2. Computer helps by determining the options.
 3. Computer helps determine options and suggests one, which human need not to follow.
 4. Computer selects action and human may or may not do it.
 5. Computer selects action and implements it if human approves.
 6. Computer selects action, informs human in plenty of time to stop it.
 7. Computer does whole job and necessarily tells human what it did.
 8. Computer does whole job and tells human what it did only if human explicitly asks.
 9. Computer does whole job and tells human what it did and it, the computer, decides he should be told.
 10. Computer does whole job if it decides it should be done, and if so tells human, if it decides he should be told.
-

[Billings 1997] stellt für den Anwendungsfall der Flugführung bemannter Luftfahrzeuge ein Modell mit sieben Modi vor, die Automationsgraden entsprechen. Hierbei wird dem Umfang der Automationsfunktionen das Eingriffsniveau des Operators gegenübergestellt. Mit steigender Automation hat der Operator weniger Einfluss auf deren Handlungen. Die Modi *Management by Consent* (LoA 5) und *Management by Exception* (LoA 6) dienen in modernen Luftfahrzeugen als gängiger Ansatz zur Auslegung von Automation [Herczeg 2014].

[Parasuraman et al. 2000] wenden das Konzept der Automationsgrade auf ein vierstufiges Modell menschlicher Informationsverarbeitung an. Hierbei werden die Verarbeitungsschritte *Information Acquisition*, *Information Analysis*, *Decision Selection* und *Action Implementation* jeweils durch einen Automationsgrad beschrieben und in einer zweidimensionalen Skala abgebildet.

In klassischen Systemen wird der Automationsgrad, bzw. die Mensch-Maschine Interaktion im Rahmen des Systementwurfs starr festgelegt. Passt ein System automatisch seinen Automationsgrad den mentalen Ressourcen des Bedieners an, so wird es als adaptive Automation bezeichnet [Rouse 1988]. [Parasuraman et al. 1992] identifizieren hierzu fünf Möglichkeiten der

Implementierung über die Bestimmung kritischer Ereignisse, der Leistung des Operateurs, des physiologischen Zustands des Menschen, dessen Modellierung sowie einer Kombination der genannten Methoden. Die besondere Schwierigkeit der Bestimmung der subjektiven Beanspruchung des Menschen wird mit Blick auf sog. selbstadaptive Strategien von multi-UAV-Bedienern betrachtet [Donath 2011].

Die automatisierte Entscheidung über die Nutzung von Automation kann zu Akzeptanzproblemen des Operateurs führen, der die Vorgänge und Zustände ggf. nicht unmittelbar nachvollziehen kann [Sheridan & Parasuraman 2005]. Kann der Operateur hingegen selbst den Automationsgrad vor oder während des laufenden Prozesses anpassen, so spricht man von adaptierbarer Automation [Oppermann 1994][Scerbo 2001]. In beiden Fällen passt sich das kombinierte Mensch-Maschine System an den Aufgabenkontext an; die Unterschiede ergeben sich aus der Zuordnung der Autorität. In diesem Sinne stellt das in Abbildung 2-1 dargestellte kaskadierte Führungssystem eines UAV ein adaptierbares System dar, da der Operateur seine Eingriffsebene und somit den eingesetzten Automationsumfang selbst wählen kann und somit die *Allocation Authority* gemäß [Sheridan 2011] darstellt, die Aufgaben und Funktionen im Mensch-Maschine System zuweist.

2.1.3.3 Autonomiegrade

Die Erhöhung des Automationsgrads führt zu einer gesteigerten Unabhängigkeit eines Systems von menschlichen Eingriffen. Der Zustand der vollen Unabhängigkeit eines Systems von Vorgaben des Operateurs wird in Anlehnung an das menschliche Wesen häufig als autonom (engl. *autonomous*) (u.a. [Sheridan 1992][Billings 1997]) bzw. voll-autonom (engl. *fully autonomous*) bezeichnet (u.a. [Clough 2002a][Huang 2004]). Zur Skalierung der Fähigkeit von Automation, unabhängig vom Menschen zu handeln, werden dementsprechend sogenannte Autonomiegrade (engl. *Levels of Autonomy*) definiert.

Eine einheitliche Definition der (vollen) Autonomie von Automation existiert dabei nicht. Teilweise wird Autonomie über die Wahrnehmungs- und Planungsfähigkeiten von Automation definiert [Huang 2004], teilweise über die Unabhängigkeit von menschlichem Eingriff [Steels 1995] oder über das Abstraktionslevel menschlicher Vorgaben, die von Automation umgesetzt werden können [El Mafkouk et al. 2006]. Manche Definitionen bezeichnen autonome Automation als vollständig selbstständig handelnde Systeme mit einem „freien Willen“ [Clough 2002a].

„Automatic means that a system will do exactly as programmed, it has no choice. Autonomous means that a system has a choice to make free of outside influence, i.e., an autonomous system has free will.“ [Clough 2002a]

Allgemein dient der Begriff der Autonomie als höherwertige Eigenschaft zur Abgrenzung gegenüber einer Automatisierung, bei der ein starr definiertes Verhalten implementiert wird, welches abgerufen werden kann. [Onken & Schulte 2010] argumentieren in diesem Zusammenhang, dass ein vollständig autonomes System nicht erstrebenswert sein könne, da es weder dem Menschen diene (Pragmatismus) noch durch menschliche Werte in seinen Handlungen gebunden sei (Ethik).

Entsprechend der unterschiedlichen Definitionen von Autonomie existieren verschiedene Darstellungen von Autonomiegraden. [Clough 2002b][Huang 2004][Department of Defense 2011] definieren jeweils vierstufige Skalen für Autonomiegrade unbemannter Systeme, wobei die

unterste Ebene die menschliche Fernsteuerung (engl. *Remote Control*) und die oberste Ebene die volle Autonomie des Systems darstellt. Volle Autonomie wird hierbei als freie Erfüllung abstrakter Vorgaben des Operateurs beschrieben. [Clough 2002a] entwickelt eine zehnstufige Skala, bei der einzelne Autonomiegrade den vier Funktionen innerhalb einer *Observe, Orient, Decide, Act* (OODA) Schleife zugewiesen werden. [Hasslacher & Tilden 1995] beschreiben schließlich eine Skala bezüglich der Überlebensfähigkeit autonomer Systeme, in der die Fähigkeiten auf den Gebieten Fortbewegung, Energieversorgung und Selbstschutz bewertet werden. Als obere Grenze der Bewertungsskala dienen hierbei jeweils die menschlichen Fähigkeiten.

Besonders für UAV wird volle Autonomie vielfach als wichtige Systemeigenschaft und als wesentliches Forschungs- und Entwicklungsziel betrachtet [Valavanis 2007]. Neben der technischen Machbarkeit [Gupta et al. 2013] ist es das Ziel, die Fähigkeiten von Luftfahrzeugbesatzungen zu automatisieren, um der Leistung bemannter Luftfahrzeuge zu entsprechen oder durch Kombination mit technisch-taktischen Vorteilen der unbemannten Luftfahrt, zu überflügeln (z.B. [Office of the Secretary of Defense 2005] für militärische UAV). Autonomie wird hierbei ebenfalls als Gegenmaßnahme zu technischen Limitationen, wie z.B. Latenzen bei der Datenübertragung oder menschlichen Fehlern verstanden. Der Ansatz ist hierbei die Substitution abgegrenzter menschlicher Fähigkeiten durch Automation ohne Betrachtung der gegenseitigen Abhängigkeit von Mensch und Maschine [Johnson et al. 2011a].

[Huang et al. 2005] nutzen *Autonomy Levels For Unmanned Systems* (ALFUS) zur Beschreibung der Fähigkeiten unbemannter Systeme und zur Spezifizierung von Anforderungen an Systeme zur Erfüllung bestimmter Aufgaben. ALFUS bietet hierzu eine dreidimensionale Darstellung mit den Dimensionen Missionskomplexität (engl. *mission complexity*), Umgebungseinflüsse (engl. *environmental difficulty*) und Unabhängigkeit vom Menschen (engl. *human independence*).

„[Levels of autonomy] are counter-productive because they focus too much attention on the computer rather than on the collaboration between the computer and its operator/supervisor to achieve the desired capabilities and effects.“ [Defense Science Board 2012]

Während Autonomiegrade gemäß Definition eng mit dem menschlichen Systemeingriff verknüpft sind, vernachlässigen ihre Skalen weitestgehend Aspekte menschlicher Fähigkeiten und Limitationen. Die technische Machbarkeit der Abbildung von Fähigkeiten dient als Treiber, während der erzielbare Nutzen im Mensch-Maschine System in den Hintergrund rückt. Auch die besonderen Anforderungen an den Menschen bei der Führung von Systemen mit steigender Autonomie bleiben unberücksichtigt. [Defense Science Board 2012] z.B. schlägt deswegen vor, anstelle von Autonomiegraden ein *Autonomous System Reference Framework* zu verwenden, welches die Allokation kognitiver Fähigkeiten auf hierarchischen Ebenen (engl. Cognitive Echelons) vorsieht. [Johnson et al. 2011a] diskutieren Autonomiegrade als simplifizierte Form der Selbstständigkeit von Automation sowie der Aufgabenverteilung zwischen Mensch und System. Autonomiegrade würden fälschlicherweise als Richtlinien zum Automationsdesign verstanden, wobei sie hierfür jedoch zu unspezifisch und aus dem Kontext gelöst dargestellt würden.

2.1.4 Automatisierung operationeller UAV-Systeme

Die Automatisierung operationeller UAS orientiert sich weitestgehend an den Erfordernissen ihres jeweiligen Einsatzgebietes und der technischen Machbarkeit [Blyenburgh 2014]. Der Einsatz von UAV zu kommerziellen Zwecken steht durch die noch ausstehende Integration unbemannter Luftfahrzeuge in den nationalen Luftraum weitestgehend aus. Für den deutschen Luftraum beschränken [LuftVO 2012] [LuftVG 2013] die UAV-Nutzung im Regelfall auf Zwecke des Sports und der Freizeitgestaltung. Hierbei ist der Betrieb von Luftfahrtgeräten außerhalb der Sichtweite des Operators und oberhalb eines Systemgewichts des Luftfahrzeugs von 25 Kilogramm grundsätzlich verboten.

Der Betrieb ziviler UAV erfolgt meist durch einen einzelnen Operateur, der das Fluggerät in Sichtreichweite fernsteuert. Zur Steuerung dienen handelsübliche Sender, die neben der Flugsteuerung auch mit Funktionen der Nutzlastkontrolle belegt werden können. Autopiloten unterstützen den Operateur in modernen Systemen bei der Flugführung und stellen Fähigkeiten zur Planung und zum Abflug von Flugrouten bereit [HiSystems GmbH 2015]. Nutzlastdaten (z.B. Kamerabilder und -videos) werden an Bord gespeichert oder unverarbeitet am Boden zur Anzeige gebracht. [Blyenburgh 2014] gibt einen detaillierten Überblick über den aktuellen Stand und die Entwicklungen im Bereich ziviler UAV.

Die fehlende Beschränkung des Flugbetriebs militärischer UAV im Einsatzgebiet und die vergleichsweise einfache Separierung des dortigen Luftraums ermöglichen dagegen das stetig wachsende Einsatzspektrum militärischer UAV und die Erhöhung der Komplexität ihrer Einsätze. Bereits heute existiert eine große Bandbreite an operationellen UAV für den militärischen Einsatz, deren Automatisierung von der Fernsteuerung durch einen Piloten bis zur Überwachung automatisierter Funktionen durch einen Operateur reicht [Austin 2010].



Abbildung 2-6: Tragbare Bodenkontrollstation mit Bediengerät für die Systeme PUMA AE, RAVEN und WASP AE [AeroVironment 2014]

Für die unmittelbare Aufklärung auf dem Gefechtsfeld werden vornehmlich kleine und taktische UAV eingesetzt, die aus der Hand oder vom Boden ferngesteuert starten und Sensordaten live per Datenfunk an eine tragbare Bodenkontrollstation am Boden übermitteln. Das US Militär verfügt hierzu u.a. über die Starrflügler-UAV RQ-20A Puma, Wasp AE und RQ-11B Raven, deren

Flugführung über eine einheitliche Bodenkontrollstation per Joystick oder über vorgeplante Wegpunkte realisiert wird [AeroVironment 2014]. Ein einzelner Operateur führt hierbei das Luftfahrzeug und wertet die Sensordaten aus. Abbildung 2-6 zeigt die Bedienelemente und das Anzeigefeld der allen Systemen gemeinsamen Bodenkontrollstation. Die Anzeige dient zur Darstellung von Sensordaten sowie zur Steuerung des UAV (Frontsicht). Die Bundeswehr nutzt zur Abbildung der Fähigkeit der Nächsbereichsaufklärung die UAV EMT ALADIN [EMT Ingenieurgesellschaft 2014] und AirRobot AR100-B (Mikado) [AirRobot GmbH & Co. KG 2014].

Operationelle *Medium Altitude Long Endurance* (MALE) UAV decken größere Einsatzgebiete, längere Flugzeiten und komplexere Missionen zu Aufklärungszwecken und zum Wirkmitteleinsatz ab. Die UAV-Führung geschieht dabei aus stationären oder per Container verladbaren Bodenkontrollstationen, die üblicherweise mit zwei oder drei Operateuren besetzt sind. Ein Operateur agiert als Navigator zur Flugsteuerung, ein zweiter als Sensor-Operateur und zur Sensordatenauswertung und ein dritter (optional) als Kommandant zur Führung des Systems und zur Kommunikation mit eigenen Kräften [Austin 2010].

Die Flugsteuerung der MALE UAV variiert gemäß Flugphase und Einsatzzweck. Start und Landung erfolgen manuell durch den Operateur über Joystick und Pedale [Williams 2004], während weite Phasen des Reisefluges über Flugrouten vorgeplant und automatisiert abgeflogen werden. Der Operateur kann hierbei jedoch stets manuell Systemeingriffe vornehmen. Die Anbindung von Bodenkontrollstation (BKS) und Luftfahrzeug erfolgt weitestgehend mittels satellitenbasierter Funkkommunikation. Sensordaten werden direkt in der BKS dargestellt und durch einen Operateur ausgewertet. Die Sensorsteuerung erfolgt dabei ebenfalls manuell über Joystick-Eingaben. Einige MALE UAV verfügen außerdem über Funktionen zur Kamerastabilisierung und zur Fixierung von Positionen und Objekten am Boden [Austin 2010].



Abbildung 2-7: Bodenkontrollstation des Predator B – Reaper als Arbeitsplatz für 2 Operateure
[Defense Industry Daily 2011]

Die U.S. Armee setzt als MALE UAV u.a. die beiden Predator Derivate Gray Eagle und Predator-B (MQ-9 Reaper) [General Atomics 2014] zur Aufklärung und zur Wirkmittelunterstützung von Bodentruppen ein. Abbildung 2-7 zeigt die gemeinsame Bodenkontrollstation der UAV, in der zwei

Operateure eingesetzt sind. Die Bundeswehr nutzt im Heer die katapultgestartete EMT LUNA zur Aufklärung in einer Tiefe von bis zu 100km und die raketengestartete Rheinmetall KZO [Wohlers & Blohm 2007] zur Zielortung für Artilleriekräfte. Die Luftwaffe verfügt seit 2010 über IAI HERON [Israel Aerospace Industries 2014], die sie ausschließlich zu Aufklärungszwecken, hauptsächlich zur unterstützenden Gebietsaufklärung einsetzt.

Am Beispiel des Northrop Grumman RQ-4 Global Hawk [Northrop Grumman Corp. 2014] lässt sich der operationelle Einsatz von HALE UAV beschreiben, welcher sich durch hohe Flughöhen und lange Stehzeiten auszeichnet. Der Global Hawk wird aus einer mobilen Bodenkontrollstation manuell gestartet und nach erfolgreichem Start an eine weitere BKS mit zwei Operateuren zur Missionsführung übergeben. Die Flugführung erfolgt anschließend automatisch anhand einer vorgeplanten Route, bei der der Operateur über weite Strecken nicht aktiv eingreift, sondern überwachend tätig ist [Austin 2010]. Die Sensordatenauswertung geschieht dabei manuell durch einen Operateur und angeschlossene Bereiche. Der Luftfahrzeugführer überwacht die Systemdaten und die Flugführung des Global Hawk und greift im Fehlerfall korrigierend ein.

2.1.5 Grenzen konventioneller Automatisierung von UAV-Systemen

Es zeigt sich, dass menschliche Fehler durch den Einsatz von Automation in UAV nicht verhindert werden. Vielmehr führt die fortschreitende Automatisierung zu einer Erhöhung der Systemkomplexität und damit zu einer Erhöhung der Komplexität der Interaktion des Operateurs mit dem System [Hollnagel 1993].

„It is a version of the „a little more technology will be enough, this time“ expectation that has been shown to result in negative consequences – such as unanticipated increases in manpower – to deal with the added complexity.“ [Winograd & Woods 1997]

Trotz der gesteigerten Systemkomplexität können die menschlichen Limitationen nicht egalisiert werden. Vielmehr führt sie zu weiteren, automationsspezifischen Fehlern, wie u.a. [Bainbridge 1983] für den Bereich der zivilen, bemannten Luftfahrt beschreibt. [Bainbridge 1983] bezeichnet in diesem Zusammenhang zwei Punkte als Ironien der Automatisierung (engl. *Ironies of Automation*). Zum einen sei Automation im Allgemeinen nicht fehlerfrei und bestehende Fehler würden ggf. nicht vollständig dokumentiert. Menschliche Fehler würden damit von der manuellen Steuerung auf die Implementierung der Automation verlagert. Zum anderen könnten durch Automation nicht alle Aufgaben abgebildet werden. Vom Operateur würde damit verlangt, all jene Aufgaben zu bewältigen, die nicht in Automation implementiert werden könnten. Dies führe zu einer Verminderung der menschlichen Belastung in Phasen niedriger Aufgabenintensität und zu einer möglichen Erhöhung in Phasen hoher Aufgabenintensität, in denen er wenig oder keine Unterstützung durch Automation erfahre. Dieser Zusammenhang wird von [Wiener 1988] als *Clumsy Automation* bezeichnet.

„When human operators make errors, the usual response is to look at the user interface for the cause. However, the causes of error are often in the logical operation of the system, instead. The user interface may be perfectly adequate given the underlying functionality, but the logical operation of the system is a poor match to the user’s task requirements.“ [Riley 1996]

Die Relevanz von Mensch-Maschine Aspekten für die Auslegung von UAV zeigt sich in Statistiken von Flugunfalluntersuchungen, in denen menschliches Versagen die wesentliche Ursache darstellt. [Williams 2004] benennt für den Zeitraum zwischen 1993 und 2003 den Anteil menschlichen Versagens für unterschiedliche UAV-Muster der U.S.-Streitkräfte auf 21% - 67% der Gesamtunfälle. [Giese et al. 2013] bestätigen dieses Ergebnis für Flugunfälle der UAV Predator und Predator-B im Zeitraum zwischen 2004 und 2012 mit 33% menschlichem Versagen als Hauptunfallursache. Und [Oncu 2014] identifiziert für insgesamt 65% der UAV-Flugunfälle der U.S. Navy in den Jahren von 2011 bis 2014 menschliches Versagen als Hauptursache. [Joslin 2015] führt bei insgesamt 319 UAV-Flugunfällen im zivilen U.S.-Luftraum 27% der Unfälle auf menschliches Versagen zurück.

Bei militärischen UAV wird für die UAV Predator und Predator-B mit 67% menschlichem Versagen die höchste Quote aller untersuchten Systeme identifiziert, wobei hiervon wiederum 75% auf prozedurale Fehler zurückzuführen sind [Williams 2004]. Als Beispiel dafür kann die inkorrekte Übergabe eines Predator UAV an eine andere GCS im Flug dienen, bei der Schritte im Übergabeprotokoll vertauscht wurden und als Resultat der Motor und die Flugstabilisierung deaktiviert wurden, sodass das LFZ unkontrolliert abstürzte.

[Giese et al. 2013][Tvaryanas 2004] beschreiben für 44% aller Predator Flugunfälle einen Zusammenhang mit dessen Mensch-Maschine Schnittstelle, welche das Head-Up Display (HUD), das Head-Down Display (HDD), Warnungen und Alarme sowie insbesondere die Interaktion mit dem Autopiloten umfasst.

Für das HDD der Predator UAV werden dabei vier spezifische Probleme identifiziert:

1. Zur Anzeige von Informationen muss der Operateur durch eine zu große Anzahl von Menüebenen navigieren.
2. Informationen werden nicht intuitiv dargestellt.
3. Es fehlen Hinweise oder Schutzfunktionen für flugkritische Eingaben wie z.B. das Löschen des internen Speichers des UAV.
4. Die Wertebereiche von Anzeigen zeigen Inkonsistenzen zu realen Werten.

[Ouma et al. 2011] identifizieren in diesem Zusammenhang die mangelnde Hardwareergonomie der Bodenkontrollstation (v.a. die Bestuhlung sowie die Temperaturkontrolle) sowie die ineffiziente Kommandierung des UAV als wesentliche Ursachen für das verstärkte Auftreten von Burnout Symptomen bei Operateuren von Predator und Predator-B UAV in der U.S. Air Force.

„On the other hand, a common theme across many of the mishaps reported involved a problem with the command interface to the system. The issues reported in this paper regarding alerts and alarms, display design, and procedural problems are mostly associated with providing information to the pilot/operator about the commanded status of the aircraft.“ [Williams 2004]

Menschliches Versagen wird häufig auf Probleme der Mensch-Maschine zurückgeführt (u.a. [Giese et al. 2013][Oncu 2014]). Hierbei wird zumeist mangelndes Feedback wichtiger Informationen an den Operateur, bzw. deren unverständliche Darstellung als Hauptursache identifiziert [Williams 2006]. Gleiches gilt für die Möglichkeit zum kontrollierten Eingriff in das System als Reaktion auf angezeigte Informationen. Die Reduktion des Problems auf HMI-Fehler

vernachlässigt die Notwendigkeit zur koordinierten Anpassung der zugrunde liegenden Automation, welche die Funktionalität und Informationen bereitstellt und eine geeignete Interaktionsform für den Menschen erlauben muss. Hierbei ist nicht die Betrachtung der Zuverlässigkeit einzelner Komponenten, sondern der Rolle der Automation bei der Führung durch den Menschen von Bedeutung [Joslin 2015].

„In all of these examples, we see evidence that the developers of the automation were unable to predict all possible contingencies. This led to situations in which the automation performed as designed but not as anticipated.“ [Williams 2006]

Als Beispiel für die Notwendigkeit der Betrachtung von Automation dient das Global Hawk UAV, dessen operationeller Einsatz weitestgehend automatisiert erfolgt. Menschliche Fehler sind nach [Williams 2004] vor allem auf den komplizierten Planungsprozess des UAV-Einsatzes und den hohen Automationsgrad zurückzuführen. Der Prozess der Missionsplanung dauert bis zu 270 Tage und umfasst Mitarbeiter verschiedener Organisationen, wobei alleine für die Planvalidierung zehn Tage angesetzt sind. Die hohe Komplexität erschwert Operateuren die Bewertung von Situationen und führte bei einem UAV nach erfolgter Landung zu einer schweren Beschädigung, da durch einen Fehler der Planungssoftware eine zu hohe Taxi-Geschwindigkeit ($v=155\text{kn}$) spezifiziert wurde, ohne dass dies der Operateur-Crew bekannt war [Williams 2004].

[Williams 2006] beschreibt weitere automationsbedingte Probleme für die UAV *Shadow* und *Fire Scout*. Nach einem Unfall bei der Landung eines *Shadow* UAV sendete die Crew ein Kommando zum Ausschalten der Motoren, welches jedoch auf Grund einer Beschädigung der Funkantenne des UAV nicht korrekt übermittelt werden konnte, wodurch die BKS Software keine Empfangsbestätigung erhielt. Nach der anschließenden Übernahme eines anderen UAV durch die Crew wurde das Kommando daher erneut übersendet, diesmal jedoch an das falsche Luftfahrzeug, welches dadurch am Ende nicht mehr gerettet werden konnte. Ein *Fire Scout* UAV stürzte ab, weil ein defekter Höhenmesser eine Höhe $h=0\text{ft AGL}$ angab, während die tatsächliche Höhe $h=498\text{ft AGL}$ betrug. Da die interne Logik des *Fire Scout* daraufhin die Motoren des UAV abschaltete, kam es zum Absturz und zum Verlust des UAV.

Die vorgestellten Beispiele verdeutlichen die Notwendigkeit einer systemergonomischen Betrachtung der automatisierten UAV Führung über die bloße Schnittstellenbetrachtung hinaus. Hierbei lassen sich mit Blick auf die von [Billings 1997] für die bemannte Luftfahrt beschriebenen *Costs of Automation* Probleme identifizieren. Die eingesetzte Automation führte die ihr erteilten Kommandos exakt in dem Sinne aus, in dem sie implementiert wurden, ohne kreative Lösungssuche mittels eigenem Wissen oder externer Informationen durchzuführen (*Literalism*). Darüber hinaus war den Operateuren nicht immer klar, welche Handlungen die Automation zum gegebenen Zeitpunkt ausführte und wie es zu diesem Verhalten kam (*Opacity*). Dies resultiert aus den komplexen, für den Operateur verborgenen Vorgängen der Automation (*Complexity*), die zu einem unvollständigen bzw. fehlerhaften mentalen Modell des Operateurs und damit letztlich zu Fehlern führt. Schließlich zeigen die Beispiele außerdem die geringe Fehlertoleranz der Automationskomponenten, die zu unerwünschtem Verhalten außerhalb der spezifizierten Bedingungen führt (*Brittleness*).

„Wir müssen hier darauf achten, dass wir keine ‚Feuerwehrlaute‘ entwickeln, die einen ‚Brand‘ verschlafen oder irgendwann sogar selbst ‚Brände‘ legen, um endlich wieder einmal im Einsatz zu sein.“ [Herczeg 2014]

Diesen hier aufgezeigten Problemen wird in den folgenden Ausführungen durch den Einsatz kognitiver Automation in Form eines intelligenten Agenten an Bord des UAV begegnet, der intelligentes, situations- und aufgabengerechtes Verhalten im Sinne des Operators zeigen soll (vgl. Abschnitt 2.3). Durch ein geeignetes Interaktionskonzept soll der Operator hierbei befähigt werden, gezielt Vorgaben zu formulieren und deren Umsetzung über die Wahrnehmung und Bewertung wesentlicher Informationen zu überwachen (vgl. Abschnitt 3.2).

2.2 Perspektiven durch intelligente KI-Agenten

Die Integration eines *intelligenten Agenten* in ein UAV-System kann zur Erhöhung der Unabhängigkeit des UAV von hochfrequenten, detaillierten Vorgaben des Operators und damit zur Senkung von dessen Belastung bei der UAV-Führung führen [Meitinger 2008], während sie jedoch die Komplexität des technischen Systems weiter steigert und den Operator weiter von der Umsetzung seiner Aufgaben entfernt. Damit werden die zuvor beschriebenen, automationsbedingten Probleme in einem ersten Schritt zunächst weiter verstärkt. Intelligente Agenten besitzen hierbei jedoch das Potential durch geeignete Ansätze der Verarbeitung und Interaktion die wahrgenommene Komplexität des Systems zu verringern bzw. für den Menschen begreifbar zu machen. Im Folgenden ist nun zu klären, wie ein solch intelligenter Agent für die UAV-Missionsführung ausgestattet sein muss und in welchem Interaktionsverhältnis zum Menschen er eingesetzt wird.

Das Konzept eines Agenten als technisches System stammt dabei aus dem Forschungsgebiet der *künstlichen Intelligenz* (KI), welches verschiedene Methoden zum intelligenten Handeln bereitstellt. Ein ausführlicher Überblick über die Teilgebiete und Konzepte der künstlichen Intelligenz liegt in [Russell & Norvig 2012] vor. [Lunze 2010] beschreibt die Anwendung der KI in den Ingenieurwissenschaften und grenzt sie gegenüber konventionellen Automationsansätzen ab, da sie Beiträge zur Konstruktion von Maschinen mit intelligentem Verhalten leisten wolle, die selbstständig ihre Ziele erkennen und diese durch zweckmäßige Handlungen zu erreichen versuchen. Hierbei werden Methoden der KI vornehmlich für ingenieurtechnische Probleme eingesetzt, für die qualitative Modellierung zum Ziel führe. KI hebe sich somit von der Anwendung in quantitativ beschriebenen (z.B. über mathematische Formeln modellierten) Vorgängen ab, die typischerweise durch Methoden der konventionellen Regelungstechnik behandelt werden.

Zum besseren Verständnis wird an dieser Stelle ein Überblick über die KI-Grundlagen eines *intelligenten Agenten* gegeben und verschiedene Einsatzarten von Agenten in UAV-Applikationen beschrieben. Aus den vorgestellten Konzepten wird anschließend abgeleitet, in welcher Form ein solcher Agent zur UAV-Missionsführung nützlich eingesetzt werden kann. In Abschnitt 2.3 folgt dann die Beschreibung der Umsetzung des Agenten mittels kognitiver Automation, in der Modelle der menschlichen Kognition über Methoden der KI abgebildet werden.

Für das Gebiet der künstlichen Intelligenz und ihrer Zielsetzung existieren verschiedene Definitionen, die sich nach [Russell & Norvig 2012] in zwei Kategorien untergliedern lassen. Die erste Kategorie beschreibt KI als Versuch zur Nachbildung menschlichen Denkens und Handelns.

Die zweite Kategorie ersetzt die menschlichen Attribute durch rationale „Denkregeln“ und sieht KI als Versuch zur Abbildung rationalen Schließens und Verhaltens.

Als übliches Maß der Abbildung menschlichen Verhaltens durch einen Computer gilt auch mehr als 60 Jahre nach seiner Beschreibung der Turing-Test [Turing 1950]. Ein Computer besteht den Turing-Test, wenn ein menschlicher Fragesteller die Antworten des Systems nicht von menschlichen Antworten unterscheiden kann. Für die Imitation menschlichen Verhaltens sieht Turing zwei Ansätze: Die Fähigkeit abstrakte Tätigkeiten durchzuführen, wie das Schachspiel und die Ausstattung von Maschinen mit möglichst guten „Sinnesorganen“ zum Verstehen und Sprechen der englischen Sprache [Turing 1967]. Dieser zweite Ansatz findet sich auch unter dem Begriff der *Behavior-Based Artificial Intelligence* [Maes 1993]. Deren Vertreter konzentrieren sich auf die Interaktion von KI Systemen mit der Umgebung.

Menschliches Denken wird über Ansätze der kognitiven Modellierung abgebildet. Hierbei wird die Intelligenz des Systems nicht nur anhand der korrekten Lösung von Problemen, sondern besonders an der Übereinstimmung der Problemverarbeitung mit der des Menschen bemessen. Für den Einsatz von KI bei der UAV-Führung ist der Anspruch der menschenähnlichen Verarbeitung unnötig restriktiv. Für den menschlichen Operateur ist es dagegen wichtig, ein System zu führen, das in entscheidenden Situationen richtig und nachvollziehbar agiert.

„Im Idealfall ergreift ein intelligenter Agent die bestmögliche Aktion in einer Situation.“
[Russell & Norvig 2012]

Intelligenz besteht nicht als Selbstzweck, sondern ist zweckgebunden und wird eingesetzt, um Aufgaben zu bewältigen. Die Bewertung von Intelligenz erfolgt mit Bezug auf die Erfüllung von Aufgaben und das dabei gezeigte Handeln. Der Begriff der Rationalität meint dabei das Denken und Handeln von KI, das anhand objektiver Kriterien als richtig zu bewerten ist und im Weiteren zum Erfolg führt [Russell & Norvig 2012]. Rationale Denkregeln können dabei über Logikoperationen modelliert und maschinell verarbeitet werden. Die Herausforderung in der Praxis besteht hierbei in der Schaffung formaler Zusammenhänge aus formlosem und zum Teil unsicherem Wissen sowie in der großen Komplexität der Lösung logischer Probleme.

Die Abbildung rationalen Verhaltens erfolgt über rationale Agenten, die logisches Schlussfolgern zur Ableitung von Handlungen nutzen. In komplexen Umgebungen erreichen Agenten dabei auf Grund der Limitierung der Computerleistung im Allgemeinen lediglich „begrenzte Rationalität“, bei der ein rational bestes Verhalten aus einem vereinfachten Lösungsraum gewählt wird [Russell & Norvig 2012].

„Ein rationaler Agent ist ein Agent, der sich so verhält, dass er das beste Ergebnis erzielt, oder, falls es Unsicherheiten gibt, das beste erwartete Ergebnis.“ [Russell & Norvig 2012]

Rationale Agenten schließen dabei den Einsatz kognitiver Modellierung für Funktionen nicht aus. Vielmehr können Ergebnisse aus den Kognitionswissenschaften genutzt werden, um erforderliche Fähigkeiten abzubilden, die für die Übernahme vormals menschlicher Aufgaben durch KI notwendig sind. [Lunze 2010] benennt in diesem Zusammenhang das klassische Ziel der KI, die „bewusste Intelligenz“ des Menschen auf technische Systeme abzubilden, bei der nach [Piaget

2000] Wissen durch Symbole repräsentiert und unter Verwendung von Lösungsstrategien umgeformt wird.

„Das Ziel des Fachgebiets Künstliche Intelligenz ist es, menschliche Erkennungs- und Denkprozesse zu formalisieren und einem Rechner zu übertragen.“ [Lunze 2010]

Intelligente KI-Agenten können nach [Russell & Norvig 2012] über Fähigkeiten aus den Teilgebieten *Problemlösen, Wissen, Schließen und Planen, Unsicheres Wissen und Schließen, Lernen* sowie *Kommunizieren, Wahrnehmen und Handeln* verfügen.

2.2.1 Agentenstruktur

Für die weitere Beschreibung intelligenter Agenten ist das obige Konzept der Rationalität zentral. Agenten existieren nicht isoliert, sondern interagieren über Sensoren und Aktoren mit ihrer Umgebung (vgl. Abbildung 2-8). Die Bewertung eines Agenten bei der Lösung einer Aufgabe erfolgt über ein Leistungsmaß, das eine gegebene Sequenz von Umgebungszuständen auswertet [Russell & Norvig 2012].

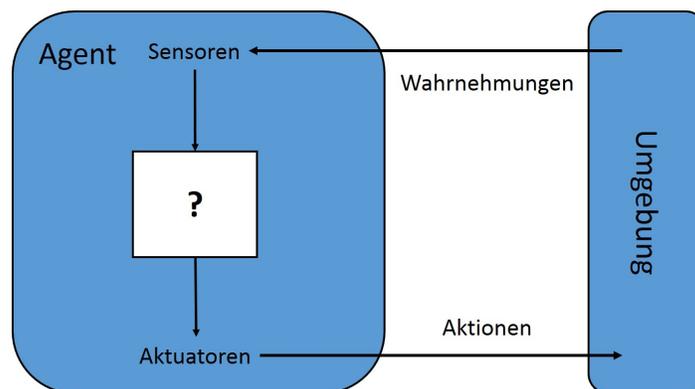


Abbildung 2-8: Grundlegende Interaktion eines rationalen Agenten mit seiner Umgebung [Russell & Norvig 2012]

Intelligente Agenten können u.a. in Form *robotischer Agenten, Artificial-Life Agenten* und als Software-Agenten, als *Softbots*, umgesetzt werden [Franklin & Graesser 1997]. Die Aufgabenumgebungen können nach [Russell & Norvig 2012] über Dimensionen kategorisiert werden, deren Ausprägungen ihrerseits Auswirkungen auf das Verhalten und damit auf die darunterliegende Struktur des Agenten haben.

Die Implementierung von Agenten erfolgt über Agentenprogramme (in Abbildung 2-8 noch als Fragezeichen verallgemeinert dargestellt), die die innere Verarbeitungsstruktur abbilden. Agentenprogramme lassen sich allgemein durch die Wahrnehmung von Eingaben über Sensoren und die Ausgabe von Aktionen über Aktuatoren charakterisieren. Insbesondere kombinieren sie Komponenten verschiedener Grundstrukturen, die sich nach [Russell & Norvig 2012] für rationale Agenten prinzipiell in vier Klassen untergliedern lassen.

- **Einfache Reflexagenten:** Diese wählen auf alleiniger Grundlage der aktuellen Wahrnehmung eine Aktion aus und setzen diese um. Vorangehende Wahrnehmungen werden dabei ignoriert. Die Aktionswahl erfolgt anhand hinterlegter Bedingungen bzw. Regeln im Format:

if Fahrwerk-setzt-auf-Boden-auf then Bremsen-beginnen

Die Vernachlässigung früherer Wahrnehmungen verringert dabei die Menge an Parametern für die Aktionswahl und beschleunigt die Verarbeitungsgeschwindigkeit. Einfaches Reflexverhalten ist auch in komplexen Situationen, z.B. zur verzögerungsarmen Reaktion in gefährlichen Situationen sinnvoll. Voraussetzung ist jedoch die vollständige Beobachtbarkeit der Wahrnehmung.

- **Modellbasierte Reflexagenten:** Diese verwenden ein Modell der sie umgebenden Welt, um auf der Grundlage aktueller und vergangener Wahrnehmungen eine interne Repräsentation der Umgebung zu schaffen. Hierbei umfasst das Modell Sinnzusammenhänge bezüglich der Umgebung und der Auswirkungen eigenen Handelns. Die interne Repräsentation dient als Grundlage für die Auswahl geeigneter Aktionen über hinterlegte Regeln. Ist die Umgebung nicht vollständig beobachtbar, so stellt die interne Repräsentation die „beste Annahme“ des Agenten bezüglich des Umgebungszustandes dar.
- **Zielbasierte Agenten:** Diese besitzen Zielinformationen bezüglich eines wünschenswerten Zustands, auf den der Agent hinwirken soll. Diese Zielinformationen können mit dem Zustandsmodell des Agenten kombiniert werden, um zielgerichtet Aktionen auswählen zu können. Die Auswahl geeigneter Handlungsfolgen zur Zielerreichung ist dabei im Allgemeinen nicht trivial und wird über die KI Teilbereiche des *Suchens* und der *Planung* abgebildet.

Zielbasiertes Verhalten unterscheidet sich vom reflexbasierten Verhalten durch die explizite Berücksichtigung zukünftiger Zustände, die aus eigenen Aktionen und den Zusammenhängen der Umgebung resultieren. Das Verhalten des zielbasierten Agenten ist leicht für verschiedene Ziele anpassbar, im Gegensatz zu reflexbasierten Agenten, bei denen dazu alle Regeln ersetzt werden müssen.

- **Nutzenbasierte Agenten:** Sie ergänzen zielbasierte Agenten durch die nutzenoptimierte Auswahl von Aktionen. Die Optimierung erfolgt dabei über die Maximierung einer Nutzenfunktion, die eine interne Leistungsbewertung darstellt. Nutzenbasiertes Verhalten erlaubt die Priorisierung bei konkurrierenden Zielen. Stimmen interne und externe Leistungsbewertung überein, so verhält sich ein nutzenbasierter Agent rational entsprechend der externen Leistungsbewertung. Partielle Beobachtbarkeit und Abweichungen in der Leistungsbewertung führen zur Optimierung eines „erwarteten Nutzens“ durch den Agenten.

Die Grundstrukturen beschreiben das Leistungselement eines rationalen Agenten [Russell & Norvig 2012]. In lernenden Agenten wird das Leistungselement um ein Lernelement ergänzt, das es dem Agenten erlaubt, kompetenter zu werden und in vormals unbekanntem Situationen rational zu handeln. Das Lernelement nutzt einen Abgleich der Ergebnisse des Agenten mit einem definierten Leistungsstandard. Über einen Problemgenerator können zudem Lernziele in Situationen umgesetzt werden, um lehrreiche Wahrnehmungen für den Agenten zu generieren.

2.2.2 Anwendung von Agenten in UAV

Software-Agenten finden zunehmend in verschiedenen Bereichen der UAV-Automation Anwendung. Agenten werden zur intelligenten Routen- und Trajektorienplanung für einzelne UAV [Goerzen et al. 2010] bzw. UAV-Schwärme [Marsh 2005][Vincent & Rubin 2004]), zum intelligenten Grundsystemmanagement [Ong et al. 2012] und zum intelligenten

Missionsmanagement [Barbier & Chanthery 2004][Sullivan et al. 2004][Valenti et al. 2004] eingesetzt. Die meisten Veröffentlichungen verweisen in diesem Zusammenhang zwar auf zielbasierte Agenten, integrieren jedoch Verfahren zur Steigerung der Optimalität und Senkung der Verarbeitungsdauer.

[Ong et al. 2012] beschreiben die automatische Reaktion auf Systemfehler in einem *More-Electric-Aircraft* (MEA) UAV durch einen intelligenten Agenten. Hierbei werden Systemzustände in ein internes Weltmodell abstrahiert und zur Missionsplanung genutzt, aus der Kommandos zur Steuerung von Energiequellen und -abnehmern resultieren. Missionspläne basieren auf Regeln, die die Auswirkung von Eingriffen auf Systemzustände beschreiben und werden hinsichtlich system- und missionsbedingter Randbedingungen generiert und vom Agenten überwacht.

[Sullivan et al. 2004] nutzen die kognitive Architektur *NASA Advanced Plant Experiments* (APEX) bei der Implementierung eines Agenten zur Missionsführung im Rahmen der luftgestützten Brandsuche mittels UAV. Vordefinierte Prozeduren werden im Flug entsprechend aktueller Sensordaten anhand definierter Randbedingungen selektiert und zu Plänen zusammengeführt, um vom Operateur definierte Ziele umzusetzen. Prozeduren beschreiben hierbei Sets von Regeln zur Kommandierung eines integrierten Autopiloten mittels Wegpunkten. Pläne können im Flug angepasst werden, um auf Umgebungsänderungen oder veränderte Zielvorgaben des Operateurs zu reagieren.

[Valenti et al. 2004] präsentieren einen zielbasierten Agenten zur Führung eines UAV aus einem bemannten Cockpit im Rahmen einer militärischen Aufklärungsmission. Der Operateur wählt aus vordefinierten Kommandos und definiert spezifische Randbedingungen, die anschließend durch den Agenten über einen MILP-basierten Planungsalgorithmus in Routen umgesetzt und an einen integrierten Autopiloten kommandiert werden. Der Agent integriert hierbei einen *Natural Language Parser* zur Entlastung des Operateurs bei der Kommandierung im Flug.

[Gunetti et al. 2013] kombinieren drei SOAR basierte Agenten (vgl. Abschnitt 2.3.3) zu einem intelligenten Missionsmanagement für UAV, welches aus Zielvorgaben des Operateurs Flugpläne ableitet und selbstständig umsetzt (vgl. Abbildung 2-9). Der *Planner Agent* leitet hierzu zunächst eine Handlungskette zur Erreichung der vorgegebenen Ziele ab und erstellt anschließend einen validen (d.h. unter Berücksichtigung des verfügbaren Treibstoffs und einer internen Zielpriorisierung) Flugplan, der durch den *Execution Agent* mittels eines Autopiloten und einer vorhandenen Payloadsteuerung umgesetzt wird. Der *Mission Manager Agent* überwacht währenddessen die Planausführung und reagiert auf Ereignisse, indem ggf. Umplanungen unter angepassten Randbedingungen angestoßen werden. Die Interaktion der Agenten mit dem Operateur ist explizit auf die Kommandierung beschränkt und sieht keinen weiteren Zugriff durch den Menschen vor.

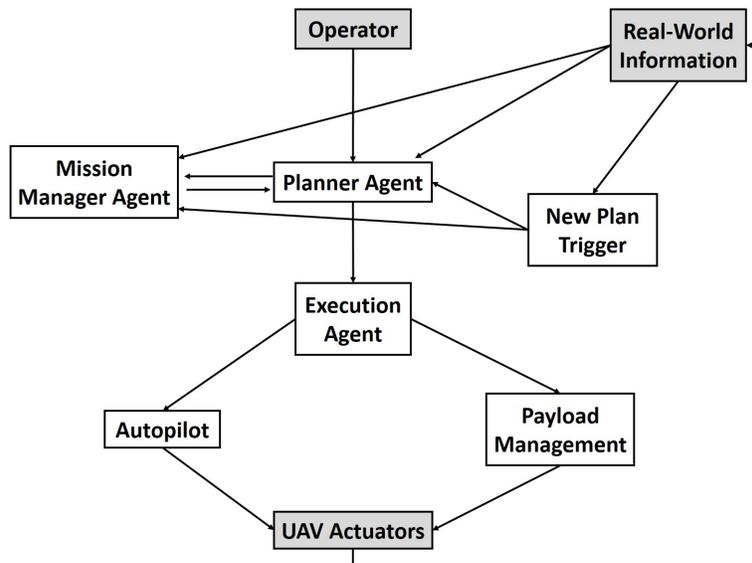


Abbildung 2-9: Informationsfluss zwischen drei Agenten zum Missionsmanagement nach [Gunetti et al. 2013]

An dieser Stelle sei auf *Belief-Desire-Intention* (BDI) Agenten verwiesen, die ausgehend von Anwendungen in Systemen des Flugverkehrsmanagements [Rao & Georgeff 1995] in verschiedenen Projekten zum Missionsmanagement von UAV eingesetzt wurden. [Karim et al. 2004] präsentieren eine hierarchische Struktur zur UAV-Flugführung, an dessen Spitze ein BDI Agent Routen im Flug plant und über Wegpunktkommandos an einen Autopiloten kommandiert. [Tantrairatn & Veres 2015] realisieren eine regelbasierte Konfiguration eines nichtlinearen adaptiven Flugreglers mittels eines BDI Agenten, der Änderungen aerodynamischer Parameter des UAV erkennt und hieraus Maßnahmen zur Kompensation oder Anpassung von Reglerparametern ableitet. BDI Agenten erweitern hierbei den zielbasierten Ansatz um die Berücksichtigung beabsichtigter Handlungen (engl. *Intentions*) bei der Vorhersage über zukünftige Zustände der Umgebung, mit dem Ziel, in einer dynamisch veränderlichen Umgebung zufriedenstellende Ergebnisse zu erzielen.

Die vorgestellten Konzepte beschreiben in erster Linie die Fähigkeiten und die Struktur der jeweiligen intelligenten Agenten und implementieren hierbei zum Teil keine [Ong et al. 2012] oder lediglich rudimentäre Mensch-Maschine Schnittstellen, die im Wesentlichen zur Formulierung und Übermittlung von Vorgaben für das selbstständige Handeln genutzt werden. Die Rückführung von Informationen wird hierbei nicht betrachtet [Gunetti et al. 2013][Tantrairatn & Veres 2015] oder lediglich über proprietäre Anzeigen von Systemkomponenten [Karim et al. 2004] realisiert. Andere nutzen hierzu Kartendisplays zur Anzeige des Flugpfads und numerischer Zustandsdaten [Sullivan et al. 2004][Valenti et al. 2004].

Ein Konzept für die Delegation und Überwachung von Aufgaben in der Domäne der Missionsführung von UAV ist die Playbook Architektur [Miller et al. 2005]. Die Playbook Architektur erlaubt dem Operateur, in Analogie zu amerikanischen Mannschaftssportarten, Spielzüge (engl. *Plays*) aus einem Spielzugbuch (engl. *Playbook*) zu selektieren und zu parametrisieren, um sie dem ausführenden Agenten anzusagen (engl. *call-a-play*).

Spielzüge beschreiben dabei Vorlagen als feste Kombination einzelner Aktionen und ihrer Parameter, wie z.B. Zeit- und Ortsangaben. Während der Laufzeit wählt der Operateur einen gewünschten Spielzug und parametrisiert ihn mit Bezug auf die konkrete Anwendungssituation. Der Nutzen dieses Ansatzes liegt im gemeinsamen Aufgabenmodell von Operateur und Agent, wodurch das Handeln des Agenten für den Operateur vorhersehbar und nachvollziehbar ist. Der Umfang an Aktionen in einzelnen Spielzügen ist dabei flexibel und erlaubt prinzipiell die gleichzeitige Adressierung mehrerer unterstellter Agenten zum Aufruf koordinierten Verhaltens. Zudem ist die Granularität der Spielzüge, die angesagt werden, variabel. So kann eine Spielzugdefinition theoretisch von einem simplen „Starte Motor“ reichen bis zu „Führe eine Aufklärungsmission durch“.

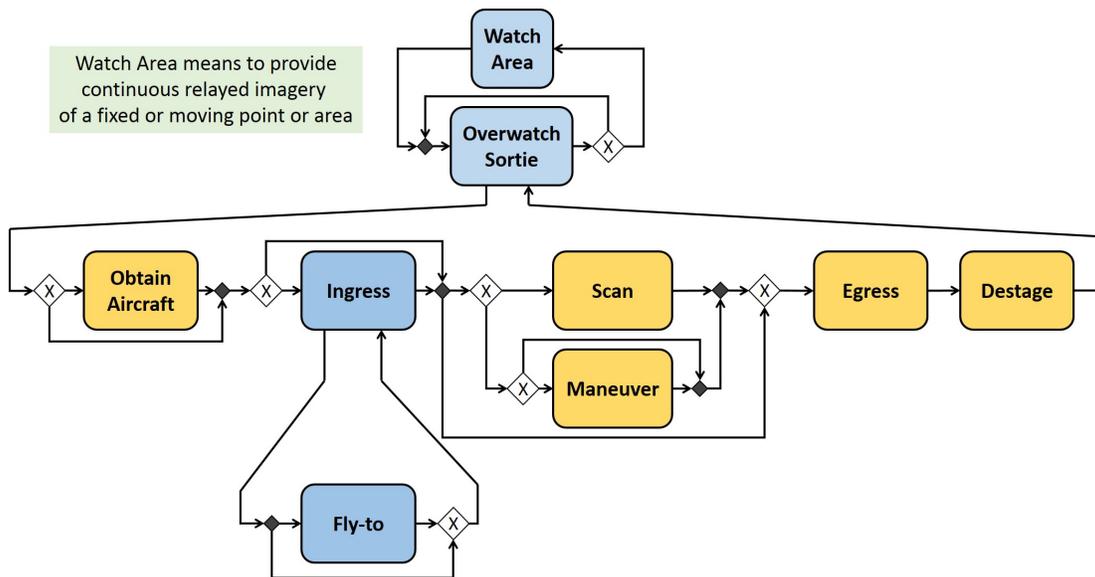


Abbildung 2-10: Vorlage des Playbook-Spielzugs „Watch Area“ [Miller et al. 2005]

Abbildung 2-10 zeigt exemplarisch die Darstellung einer Vorlage für den Playbook Spielzug „Watch Area“, der zur Erzeugung von Aufklärungsdaten eines festen oder beweglichen Punktes oder einer Gegend dient. Ein Spielzug besteht dabei mindestens aus einem Unterspielzug (in diesem Fall „Overwatch Sortie“), welcher seinerseits aus mehreren Einzelaktionen (hier z.B. „Obtain Aircraft“ und „Ingress“) besteht. Diese sind untereinander logisch miteinander verknüpft. Im Beispiel dient der Unterspielzug der korrekten Positionierung des Luftfahrzeugs für die Aufklärung und wird nach Bedarf vom Agenten flexibel umgesetzt.

Das oben gezeigte Beispiel verdeutlicht die Vereinfachung der Delegation, die durch den Aufruf vordefinierter Spielzüge erzielt werden kann. Komplexe Abläufe werden durch ihre Spielzugnamen identifiziert und aufgerufen. Die Überwachung der Aufgabenerfüllung wird hierbei nach [Miller 2005] und [Miller 2014a] durch die Playbook Architektur ebenfalls positiv beeinflusst. Demnach stellt die Delegation eine Absichtserklärung des Operateurs dar, die den Rahmen für die Handlungen des unterstellten Agenten definiert (engl. *Intent Frame*). Durch die Vorgabe von Spielzügen kann die Überwachung für den Operateur von der Fragestellung „Was tut der Agent gerade?“ vereinfacht werden zu „Tut der Agent, was ich ihm angesagt habe?“.

Diese Vereinfachung der Überwachungsaufgabe des Operators führt nach [Miller 2014a] zu einer verminderten Belastung des Operators, einem erhöhten Situationsbewusstsein bezüglich des Zustands und der Handlungen des Agenten und einer beschleunigten Entwicklung von Vertrauen auf Seiten des Operators zum unterstellten Agenten. Der Einfluss von Vertrauen auf die Delegation von Aufgaben an einen Untergebenen wird in Abschnitt 2.3.1.2 genauer untersucht.



Abbildung 2-11: FLEX-IT Karten- und Aktivitätsdarstellung zur adaptierbaren Steuerung von multi-UAV [Miller 2014a]

Die Umsetzung einer Schnittstelle zur adaptierbaren multi-UAV-Führung mittels des Playbook Konzepts ist in Abbildung 2-11 gezeigt. Die FLEX-IT Darstellung ermöglicht eine alternative oder kombinierte Führung von UAV über Benachrichtigungspunkte und Spielzüge. Punkte werden über die Kartendarstellung positioniert und definieren Orte, an denen UAV Meldungen absetzen oder auf weitere Anweisungen warten sollen. Spielzüge können mit Punkten verknüpft und parametrisiert werden. Die Überwachung erfolgt kombiniert über die Karten- und die Aktivitätsdarstellung. Die Kartendarstellung dient dabei der geografischen Einordnung der Entitäten. Die Aktivitätsdarstellung präsentiert textuell aufgabenrelevante Informationen über die UAV und ordnet sie gemäß ihrer aktuellen Tätigkeiten. Die Aktivitätsdarstellung kann dabei zusätzliche Information über den Aufgabenstatus (z.B. zeitlicher Verzug) umfassen.

Die vorgestellten Anwendungen künstlicher Intelligenz in Form von intelligenten Agenten zum Missions- und Systemmanagement von UAV demonstrieren das Potenzial für den Zugewinn von Fähigkeiten und zur Steigerung der Selbstständigkeit unbemannter Systeme. Vergleichsweise wenige und kurze Kommandos des Operators genügen in den gezeigten Beispielen, um Aufklärungs- und Flugziele zu erreichen. Die Komplexität der Aufgaben, die dem Agenten

übertragen werden können, hängt dabei von Art und Umfang der nutzbaren Funktionen sowie der Fähigkeit des Agenten ab, diese zur Aufgabenumsetzung einzusetzen.

Hinsichtlich der in Kapitel 1 formulierten Ausgangsfragestellung lässt sich damit ableiten, dass ein ausreichend gut vernetzter intelligenter Agent an Bord eines hochautomatisierten UAV prinzipiell zum Einsatz in hoch-komplexen Szenarien geeignet ist, wenn es gelingt, die Nutzung der Automationsfunktionen zu planen und umzusetzen. Die Integration eines Agenten in bestehende UAV Automation wird für diese Arbeit in Abschnitt 2.3.1 eingehend beschrieben. Um dem Menschen weiterhin die Kommandierung und Überwachung des UAV im Sinne der HSC zu ermöglichen, muss ein geeignetes Interaktionskonzept zwischen Mensch und Agent realisiert werden, welches die verantwortliche Leitung des Systems ermöglicht.

2.3 Kognitive Automation

Einen alternativen Ansatz zur Überwindung der Grenzen konventioneller Automation stellt künstliche Kognition in Form kognitiver Automation dar, die nach [Onken & Schulte 2010] höhere Aufgabenkomplexität durch menschenähnliche Verarbeitungsansätze bewältigt. Kognitive Automation soll den Menschen bei der Leitung automatisierter Systeme in seinen kognitiven Aufgaben, mit dem Ziel der optimalen Ausnutzung des Automationspotenzials, entlasten und unterstützen.

Kognitive Automation umfasst dabei nach [Onken & Schulte 2010] die künstlichen Fähigkeiten, auch im Falle unvorhergesehener Ereignisse ein Situationsverständnis zu bilden, dieses anhand bekannter Sollzustände zu interpretieren und hieraus konkrete Ziele für eigenes Handeln abzuleiten. Sie umfasst zudem Fähigkeiten zur Entwicklung von Handlungsabfolgen zur optimierten Erreichung dieser Ziele und damit zur Unterscheidung zwischen wichtigen und unwichtigen Informationen sowie mehr oder weniger dringenden Aktionen mit Bezug auf ihre originären Aufgaben. Kognitive Automation umfasst im Weiteren Fähigkeiten, autorisierte Handlungen eigenständig durchzuführen und Kommunikation mit anderen Systementitäten zu initiieren, um Handlungen im Fall von Konflikten und günstigen Gelegenheiten zu koordinieren. Hierbei fungiert kognitive Automation als wissensbasiertes System, dessen Verarbeitung auf der Nutzung von situationsunabhängigem a-priori Wissen sowie Situationswissen beruht.

Die Fähigkeiten kognitiver Automation dienen nach obiger Definition zur Schaffung ziel- oder nutzenbasierter Agenten, die sich von konventionellen Ansätzen durch die Komplexität der Umweltwahrnehmung und der Separierung von Zielbestimmung und Handlungsableitung anhand von a-priori Wissen abheben.

„Kognitive technische Systeme“ sind das Ergebnis des interdisziplinären Feldes der Kognitionswissenschaft (engl. *Cognitive Science*) [Wilson & Keil 2001], die u.a. Erkenntnisse der kognitiven Psychologie und der künstlichen Intelligenz verknüpft. In der Kognitionswissenschaft werden kognitive Prozesse als Informationsverarbeitung abstrahiert, um sowohl menschliche als auch künstliche Kognition abzubilden. Die Modelle der menschlichen Kognition und die Definition einzelner kognitiver Funktionen und Konzepte zur anschließenden Untersuchung stammen dabei aus der kognitiven Psychologie und sollen nach [Anderson 1990] bestenfalls mit einem Computerprogramm vergleichbar sein. Die kognitive Psychologie liefert hierbei u.a. Beiträge zu den Forschungsbereichen der *Aufmerksamkeit*, *Wahrnehmung* und des *Problemlösens*.

„Cognitive Psychology [is] the study of higher mental processes such as attention, language use, memory, perception, problem solving and thinking.“ [American Psychological Association 2014]

Für die technische Umsetzung künstlicher Kognition werden diese Modelle mit Ansätzen der KI zu kognitiven Architekturen kombiniert. Künstlich kognitives Verhalten kann dann über eine Architektur und den in ihr implementierten Inhalt generiert [Lehman et al. 2006] und zur Abbildung des Verhaltens eines kognitiven Agenten verwendet werden.

[Onken & Schulte 2010] schlagen in diesem Zusammenhang zwei unterschiedliche Wege vor, auf denen ein kognitiver Agent mit einem menschlichen Operateur zusammenarbeiten kann. Zum einen könnten Agenten den Menschen unterstützen, indem sie von ihm übertragene Aufgaben selbstständig, unter Verwendung vorhandener Automation umsetzen, zum anderen sei es möglich, Agenten kooperativ mit dem Menschen agieren zu lassen, indem sie als technische Partner fungieren und an gemeinsamen Zielen mit dem Operateur arbeiten. Zur Beschreibung dieser Rollen und zur Identifizierung und Bewertung der Verlagerung kognitiver Fähigkeiten in einem automatisierten System nutzen [Onken & Schulte 2010] eine Systemanalyse anhand des Modells des Arbeitssystems (engl. *work system*). Das Arbeitssystem stellt hierbei die physikalische Repräsentation des entsprechenden Arbeitsprozesses (engl. *work process*) dar, welcher über ein Arbeitsziel (engl. *work objective*) definiert ist und beschreibt, wie (d.h. über welche Handlungsschritte) dieses erreicht wird. Das dazugehörige Arbeitssystem beschreibt die Komponenten, die den Arbeitsprozess und damit das Arbeitsziel umsetzen. Per Definition existiert in jedem Arbeitssystem mindestens ein menschlicher Bediener.

Die allgemeine Grundstruktur eines Arbeitssystems mit seinen Schnittstellen zur Umwelt ist in Abbildung 2-12 dargestellt. Eine übergeordnete Instanz gibt dem Arbeitssystem ein Arbeitsziel vor. Das Arbeitsziel formuliert einen gewünschten Umgebungszustand und fungiert fortan definitorisch für den Arbeitsprozess und damit für das dafür zu entwickelnde Arbeitssystem. Als einzige Komponente des Arbeitssystems besitzt der Mensch die Fähigkeit, das Arbeitsziel selbst zu wählen bzw. selbstständig zu verändern. Auf das Arbeitssystem wirken Umwelteinflüsse (engl. *environmental conditions*), die sich positiv und negativ auf den Arbeitsprozess auswirken können. Zudem nutzt das Arbeitssystem externe Ressourcen und Informationen (engl. *supply & information*), die z.B. aus anderen Arbeitssystemen stammen können. Das Arbeitssystem bewirkt eine Veränderung in seiner Umwelt, welche sich als Arbeitsergebnis (engl. *work system output*) manifestiert und besonders hinsichtlich der Erreichung des Arbeitsziels analysiert wird.

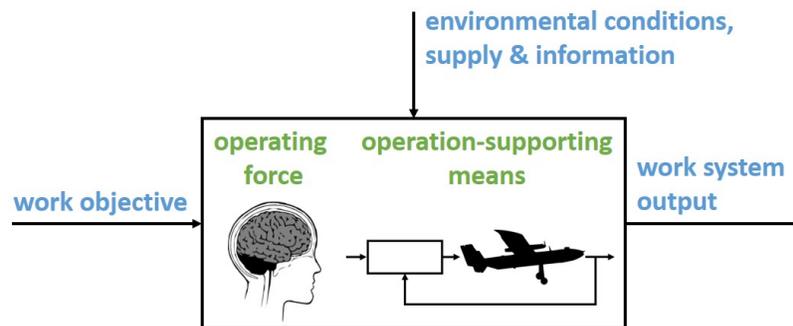


Abbildung 2-12: Allgemeine Grundstruktur eines Arbeitssystems [Onken & Schulte 2010]

Die Hauptbestandteile des Arbeitssystems werden unterschieden in die *Operating Force* und die zur Verfügung stehenden Arbeitsmittel (*operation-supporting means*).

Die *Operating Force* (OF) kennt als einzige Komponente im Arbeitssystem das vollständige Arbeitsziel und leitet den Arbeitsprozess sowie den Einsatz verfügbarer Arbeitsmittel im Sinne der Zielerreichung. Die OF besitzt somit die höchste Autorität im Arbeitssystem. Die OF umfasst mindestens einen Menschen, kann im Prinzip jedoch auch mehrere Menschen sowie künstliche Agenten umfassen. Die menschliche Komponente befähigt das Arbeitssystem prinzipiell zu unabhängigem Handeln, wodurch es per Definition zu einem *autonomen System* wird.

Die *Operation-Supporting Means* (OSM) beschreiben die Menge aller Arbeitsmittel, die der OF zur Nutzung im Arbeitsprozess zur Verfügung stehen. [Onken & Schulte 2010] nennen in diesem Zusammenhang explizit die Arbeitsumgebung (engl. *work site settings*), alle Arten von Werkzeug und Maschinen, zu denen z.B. auch alle in Abschnitt 2.1.1 beschriebenen Systeme zählen. Während des Arbeitsprozesses unterstehen alle OSM der OF.

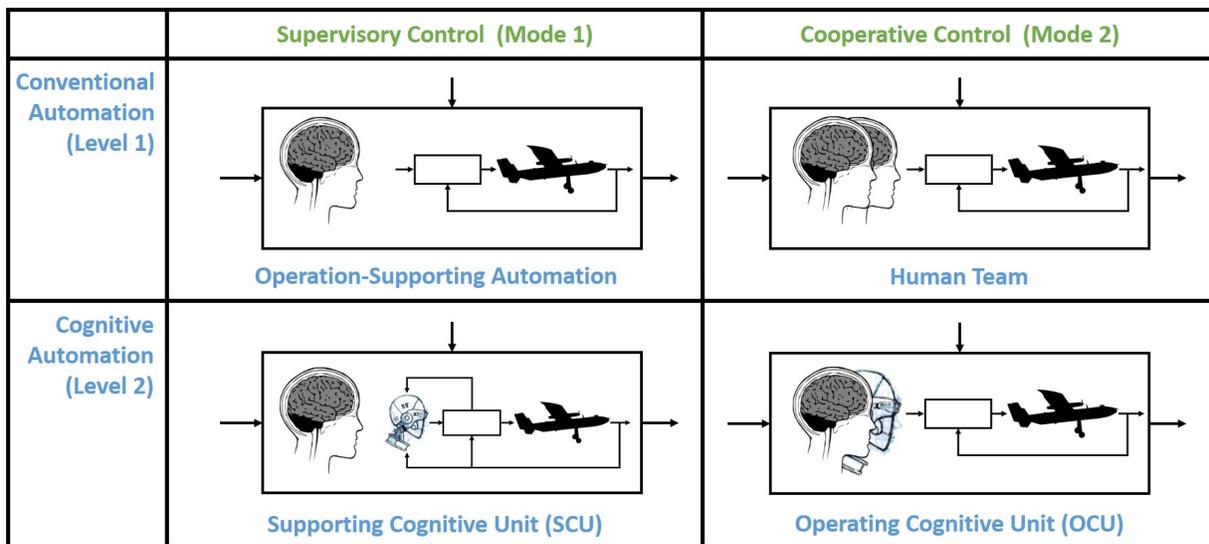


Abbildung 2-13: Varianten der Integration künstlich kognitiver Agenten ins Arbeitssystem [Onken & Schulte 2010]

[Onken & Schulte 2010] nutzen das Arbeitssystem zur Beschreibung möglicher Rollen des kognitiven Agenten über zwei unterschiedliche Ansätze (*Modi*) zur Erweiterung eines konventionell automatisierten Arbeitssystems mittels kognitiver Automation (vgl. Abbildung 2-13), zusammengefasst in dem Konzept der *Dual Mode Cognitive Automation*.

1. Kognitive Automation kann demnach als Teil der OSM in ein Arbeitssystem integriert werden (**Mode 1**). Hierdurch werde ein kognitiver Agent geschaffen, der dem Menschen im Sinne der HSC unterstellt ist und von diesem kommandiert und überwacht wird. [Onken & Schulte 2010] bezeichnen einen solchen kognitiven Agenten als *Supporting Cognitive Unit* (SCU). Die Verlagerung der kognitiven Fähigkeiten auf die Seite der OSM diene der Steigerung des Autonomiegrades des technischen Systems und der teilweisen Vermeidung von Automationsproblemen wie z.B. *Brittleness*, *Literalism* und *Clumsiness* (vgl. Abschnitt 2.1.5). [Meitinger 2008] nennt in diesem Zusammenhang zielorientiertes Verhalten einer SCU, welches die Definition numerischer parametrisierter Systemgrenzen vermeidet und

zweckmäßiges Handeln außerhalb vordefinierter Definitionen ermöglicht. [Uhrmann 2013] argumentiert, dass die Verfolgung explizit und abstrakt formulierter Handlungsziele zusätzlich *Literalism* vorbeugt, da jeweils die aktuelle Situation interpretiert und Handlungsalternativen gezielt gewählt würden.

2. Eine weitere Möglichkeit zur Integration kognitiver Automation besteht für [Onken & Schulte 2010] in Form eines künstlichen Teammitglieds auf Seiten der OF neben dem menschlichen Operateur (**Mode 2**), um ihn bei der Leitung des automatisierten Systems zu unterstützen. Eine solche *Operating Cognitive Unit* (OCU) kenne und verstehe die Ziele des menschlichen Handelns und agiere als Assistenzsystem, das aktiv und zielgerichtet zur Unterstützung des Menschen handle. Durch die Integration kognitiver Automation auf Seiten der OF wandle sich das Führungsparadigma von der Leitenden Kontrolle hin zu einer kooperativen Kontrolle (engl. *Cooperative Control*), bei der Mensch und OCU, die unterstellten OSM gemeinsam als Team leiten. Der menschliche Operateur stellt im Arbeitssystem dabei die höchste Autorität dar.

[Onken & Schulte 2010] bewerten kognitive Automation im Sinne kooperativer Kontrolle als höherwertig, da die Fähigkeiten zur Interaktion und Nutzung konventioneller Automation durch die Beobachtung und die Assistenz des menschlichen Operators erweitert werden. Die technischen Voraussetzungen, und hierbei vor allem die Realisierung der Beobachtung des menschlichen Zustands und der Interpretation seines Verhaltens, gestalten sich allerdings auch wesentlich komplexer.

2.3.1 Delegation von Aufgaben an einen intelligenten Agenten

Integriert man einen kognitiven Agenten im Sinne der *Mode 1 Cognitive Automation* an Bord eines UAV, so ergibt sich eine grundlegende Änderung dessen innerer Struktur und der Interaktion zwischen dem Menschen und dem automatisierten System. Zuvor nutzte der Mensch im Allgemeinen inhomogen gestaltete, separate Ein- und Ausgabeschnittstellen, um diskrete Vorgaben an unterschiedliche Automationsfunktionen zu formulieren und deren Ausgabewerte in einen Gesamtzusammenhang zusammen zu führen und zu bewerten [Sheridan 1992]. Nun erfolgt die Interaktion über einen einzelnen, kognitiven Agenten (vgl. Abbildung 2-13), der nicht nur eine Erweiterung des Fähigkeitsumfangs an Bord des UAV darstellt, sondern selbst leitende Kontrolltätigkeiten wahrnimmt.

2.3.1.1 Mensch-Agenten Interaktion

Der kognitive Agent kann als adaptierbares System betrachtet werden, bei der die Rolle des Operators der eines Vorgesetzten gleicht, welcher Aufgaben an seinen Untergebenen delegiert und ihre Erfüllung überwacht [Sheridan & Parasuraman 2005]. Als zielbasierter Agent erfolgt die Delegation über die Vorgabe von Zielen durch den Menschen und bedeutet die Übertragung der Verantwortlichkeit für die Erfüllung von (Teil)aufgaben an den untergebenen Agenten.

„In short, delegation is a process of assigning specific roles and responsibilities for the subtasks of a parent task for which the delegating agent retains authority (and responsibility).“ [Miller & Parasuraman 2007]

Die Konzeption der Mensch-Agenten Interaktion sowie die Realisierung der technischen Schnittstelle haben Einfluss auf die resultierende Leistung des automatisierten Gesamtsystems und die kognitive Beanspruchung des Operators [Miller & Parasuraman 2007]. Die Art und Form der

Informationen, die vom Agenten an den Operateur zurückgeführt werden, beeinflussen dabei wesentlich dessen Wahrnehmung des Agenten und seine Neigung Aufgaben zu delegieren (vgl. Abschnitt 2.3.1.2).

In Abbildung 2-14 sind die Informationsarten dargestellt, welche zwischen dem Menschen und dem Agenten ausgetauscht werden. Neben Anweisungen des Operateurs an den Agenten, umfasst die Kommunikation Leistungsinformationen einschließlich des Status der Zielumsetzung sowie Informationen über den Status des untergliederten Systems und dessen Automation, die der Agent an den Operateur zurückführt. Der Austausch von Informationen bezüglich Umweltzustand und taktischer Lage zwischen Mensch und Agent soll in einer Form erfolgen, die eine vollständige Transparenz der Agentenebene vermeidet, dem Menschen jedoch trotzdem die Bewertung des UAV-Zustands und seiner Komponenten ermöglicht. Um den manuellen Durchgriff des Operateurs auf Systemressourcen (Automationsfunktionen und Systemkomponenten) zu ermöglichen, die dem Agenten hierarchisch untergliedert sind, ist eine gesonderte Kommunikation zur Koordination von Mensch und Agent erforderlich.

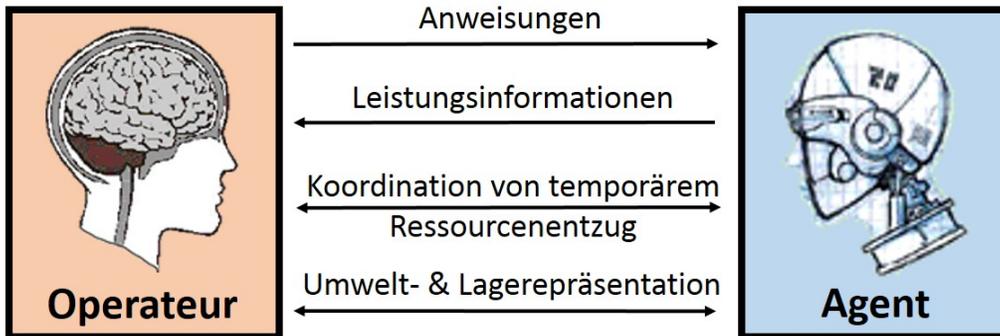


Abbildung 2-14: Informationsarten bei der Kommunikation zwischen Operateur und Agent

Im Sinne des Arbeitssystems leitet der Operateur Vorgaben für den Agenten aus seinem Arbeitsziel ab, die in Form von Anweisungen formuliert und an den Agenten übermittelt werden. Vorgaben dienen dabei der Zielformulierung und definieren einen Rahmen (engl. *Intent Frame*), in dem der Agent selbstständig und nach eigenem Ermessen handeln kann [Miller 2014a]. Dieser Rahmen ist je nach Art und Spezifizierung der Delegation unterschiedlich eng gesteckt und bildet, zusammen mit aufgabenübergreifenden Regeln, sog. „*Backplane Expectations*“ [Miller 2014b], die Randbedingungen, unter denen der Agent die Umsetzung der Ziele ausgestaltet. In Analogie zum menschlichen Unterstellungsverhältnis basieren Anweisungen auf der Bewertung, dass der Untergebene die formulierte Aufgabe versteht, die Mittel bzw. Fähigkeiten zur Umsetzung besitzt (*ability*) und die Aufgabe in Sinne der Führung umsetzen wird (*benevolence & integrity*) [Brower et al. 2000].

Leistungsinformationen beschreiben den Zustand und das Verhalten des Agenten mit Bezug auf dessen nachgeordnete Automation sowie das UAV-Grundsystem. Hierbei ist es nicht das Ziel, vollständige Transparenz bezüglich des agentengeführten Systems einschließlich aller Funktionen, Verhalten und Zustände beim Operateur zu erzeugen. Dies widerspräche dem grundlegenden Ziel der Entlastung des Operateurs, da dem bereits zu überwachenden Bereich lediglich eine zusätzliche Ebene, die des kognitiven Agenten, hinzugefügt würde [Miller 2014a]. Vielmehr zielen die

Leistungsinformationen auf das Verständnis des Operateurs für das aufgabenbezogene Verhalten seines Untergebenen und dessen nachgeordneten Bereich, um zukünftiges Verhalten vorherzusehen und möglicherweise korrigierend eingreifen zu können [Schriesheim et al. 1998]. Nach [Chen et al. 2014] dient dieser Ansatz darüber hinaus der Ausbildung kalibrierten Vertrauens gegenüber dem Agenten und somit schließlich zu einer gesteigerten Systemleistung. Eine eingehende Betrachtung erfolgt hierzu in Abschnitt 2.3.1.2.

Die Delegation von Aufgaben an den kognitiven Agenten schließt den menschlichen Eingriff, d.h. die menschliche Kommandierung nachgeordneter Automationsfunktionen in Form eines manuellen Durchgriffs nicht aus. Ein solcher Durchgriff stellt für den Agenten einen temporären Verlust von Systemressourcen dar, welcher bei mangelnder Koordination Auswirkungen auf die Zielumsetzung des Agenten haben kann. Ein solcher Durchgriff bedarf daher der engen Koordinierung von Abhängigkeiten, Überschneidungen und Berührungspunkten [Smith & Hancock 1995]. Der Mensch kann dabei Automationsfunktionen manuell ersetzen (Fähigkeitssubstitution) oder durch eigene Fähigkeiten ergänzen (Fähigkeitsaddition). Abschnitt 3.2.4 beschreibt die Umsetzung eines solchen manuellen Durchgriffs im Rahmen der auftragsbasierten Führung.

Das Handeln von Mensch und Agent geschieht dabei nicht im kontextleeren Raum, sondern ist eingebettet in ihre Umwelt, mit Lageelementen als Abbildung von Entitäten (z.B. äußere Akteure oder Bedrohungen), mit denen das UAV-System interagieren kann bzw. die ihrerseits während der Missionsführung Einfluss auf das UAV nehmen. Durch den Austausch dieser Informationen zwischen Mensch und Agent wird ein synchronisiertes Lageverständnis [Lee & Seppelt 2009] erreicht, dass die Grundlage für die korrekte Interpretation und Umsetzung von Aufgaben durch den Agenten sowie für eine vorhersehbare und nachvollziehbare Überwachung durch den Operateur bildet. Die jeweilige Informationstiefe der Lagerepräsentationen und ihre Darstellungsform sind dabei im Allgemeinen uneinheitlich und werden entsprechend der Notwendigkeiten der jeweiligen Führungsebene gestaltet.

Interaktionsansätze als Etikette

Untersuchungen menschlicher Unterstellungsverhältnisse nennen neben den Kommunikationsinhalten auch die Interaktionsform, wie z.B. die Interaktionshäufigkeit, die jeweiligen Kommunikationsmuster und die Schlüssigkeit und Struktur von Informationen als wesentlich [Nerdinger et al. 2011] [Graen & Uhl-Bien 1995]. Analog argumentiert [Miller 2002] für technische Systeme, dass Interaktionsansätze als Etikette (*Etiquette*) beschrieben werden können, deren Anwendung potenziell positiven Einfluss auf die Leistung des Gesamtsystems hat. Die generalisierte Etikette beschreibt dabei allgemeine Verhaltensgrundsätze und Normen für die Kommunikation zweier Gesprächspartner (z.B. zum Zweck einer höflichen, angenehmen Interaktion). Die spezifische Etikette umfasst kontextbezogene Verhaltensnormen, die der Vermeidung von Missverständnissen und der Steigerung der Effektivität in bestimmten Anwendungsdomänen und Anwendungen dienen. [Miller 2002] bezieht sich hierbei explizit auf das Beispiel der Luftüberwachung, in der eine eindeutig geregelte Kommunikation vor allem zum Zweck der Sicherheit existiert. [Lee & Seppelt 2009] argumentieren, dass sich eine Etikette positiv auf die Vertrauensbildung beim Operateur auswirkt, auch wenn sich auf Grund abweichender mentaler Modelle (für den Operateur) unerwartetes Verhalten des Agenten ergibt.

Um die Vorteile einer Etikette nutzen zu können, muss die Interaktionsform an den aktuellen Situationskontext sowie die Rolle des kognitiven Agenten angepasst werden. Nach [Parasuraman & Miller 2004] kann dadurch die Effektivität und Sicherheit des Gesamtsystems gesteigert werden. Die Ausprägung des zeitlichen Bezugs und der Häufigkeit der Interaktion zwischen dem Menschen und dem kognitiven Agenten hat großen Einfluss auf den Grad der Unterstützung, die durch den Agenten erzielt werden kann. Alarmierende Informationen und Kommunikation zur Aktionskoordinierung bedürfen hierbei genauer Kalibrierung [Bliss 2003]. Hochfrequente Meldungen, welche Probleme und Abweichungen beschreiben, die ausschließlich in den Verantwortungsbereich des Agenten fallen, belasten den Operateur unnötig. Diese sog. *Unintended Alerts* [Lees & Lee 2007] führen zu verminderter Leistungsfähigkeit und einer erhöhten Reaktionszeit des Menschen bei relevanten Meldungen. Andererseits nimmt der Agent dem Menschen die Möglichkeit zum korrigierenden Eingreifen, wenn Abweichungen mit Auswirkungen auf die Missionsdurchführung nicht übermittelt werden.

2.3.1.2 Delegationsentscheidung des Operateurs

Der Operateur entwickelt auf der Grundlage des Arbeitsziels eine Handlungsstrategie und trifft eine Entscheidung darüber, welche Aufgaben an den Agenten delegiert, und welche manuell durchgeführt werden sollen [Parasuraman & Riley 1997]. Das Delegationsverhalten des Operateurs ist dabei individuell und dynamisch während des Verlaufs seiner Interaktion mit dem Agenten. Für die Identifikation der Kriterien der Delegationsentscheidung des Menschen eignet sich eine Betrachtung bestehender Unterstellungsverhältnisse. Hierzu zählen im Folgenden:

1. Menschliche Unterstellungsverhältnisse in hierarchischen Organisationen sowie
2. die Führung konventionell automatisierter Systeme durch einen menschlichen Operateur.

[Leana 1986] beschreibt Einflüsse auf die Delegationsentscheidung in hierarchischen Strukturen, die aus den Eigenschaften des Vorgesetzten, der Wahrnehmung der Eigenschaften des Untergebenen und der jeweiligen Aufgabensituation resultieren. Demnach delegiere ein Vorgesetzter eher Aufgaben, wenn er seinen Untergebenen die Möglichkeit zur persönlichen Entwicklung bieten will und ein geringes Dominanzverhalten zeigt. Untergebene erhielten im Weiteren mehr Aufgaben, wenn sie als fähig, zuverlässig und vertrauenswürdig wahrgenommen würden. Und schließlich würden eine geringe Bedeutung der jeweiligen Aufgabe sowie eine hohe Belastung des Vorgesetzten dazu führen, dass mehr Autorität delegiert wird. Andere Quellen assoziieren die Delegationsentscheidung mit der Risikobereitschaft von Vorgesetzten [Matthews 1980] und der Zuversicht in die Handlungen der Untergebenen [Hollingsworth & Al-Jafary 1983][Dewhirst et al. 1987].

Ähnliche Kriterien für die Delegationsentscheidung werden von [Parasuraman & Riley 1997] für der Leitung eines automatisierten Systems durch einen menschlichen Operateur beschrieben. Abbildung 2-15 überträgt die Betrachtungen von [Parasuraman & Riley 1997] auf die Delegation von Aufgaben an einen Agenten und beschreibt die Bereitschaft des Operateurs, Aufgaben zu delegieren (engl. *reliance*). Eine hohe Neigung zur Delegation führe dabei zur Überantwortung großer Teile von Aufgaben an den Agenten und zu einer steigenden Konzentration auf Leitungstätigkeiten durch den Operateur. Diese Neigung werde direkt beeinflusst durch das Selbstvertrauen (engl. *confidence*) des Menschen eine Aufgabe manuell (d.h. durch direktes Kommandieren konventioneller Automation) erfüllen zu können, durch sein Vorwissen (engl. *state*

learning), seinen aktuellen Grad an Müdigkeit (engl. *fatigue*) sowie das mit einem Scheitern assoziierte Risiko (engl. *perceived risk*).

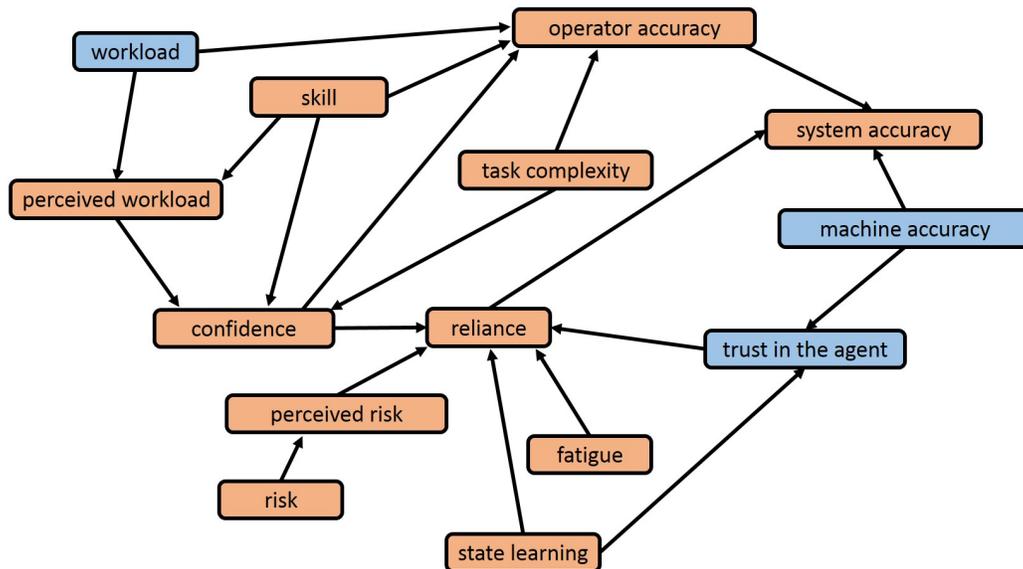


Abbildung 2-15: Kriterien für die Delegationsentscheidung des menschlichen Operators. Kriterien, welche direkt durch die Auslegung des Agentenverhaltens beeinflusst werden, sind blau gefärbt (nach [Parasuraman & Riley 1997])

Die meisten Kriterien in Abbildung 2-15 sind durch das technische System nicht direkt beeinflussbar, sondern werden indirekt geformt über die aktuelle Aufgabenkomplexität (engl. *task complexity*) und den Zustand des Operators. Die blau gefärbten Kriterien werden hingegen direkt durch die Fähigkeiten und das Verhalten des Agenten beeinflusst und sind somit durch seine Auslegung beeinflussbar:

- **Maschinengenauigkeit** (engl. *machine accuracy*)
- **Vertrauen in den Agenten** (engl. *trust in the agent*)
- **Belastung des Operators** (engl. *perceived workload*)

Maschinengenauigkeit beschreibt in diesem Zusammenhang die Leistung des Agenten bei der Leitung der konventionellen Automationsfunktionen des UAV-Systems. Hierzu leitet der Operator das Leitungsverhalten des kognitiven Agenten aus dem Gesamtverhalten des UAV-Systems ab und bewertet es hinsichtlich einer projizierten Eigenleistung bei manueller Aufgabenerfüllung. Die Leistung des kognitiven Agenten wird dabei anhand seiner Lenkbarkeit (engl. *directability*), Vorhersehbarkeit (engl. *predictability*) und Aufgabenerfüllung (engl. *task satisfaction*) bewertet [Klein u.a. 2004]. Eine hohe Lenkbarkeit bedeutet eine Aufgabenerfüllung des Agenten, die möglichst genau den Absichten des Operators entspricht. Die Vorhersehbarkeit des Agentenverhaltens wird vom Operator gemeinsam mit dem Aspekt der Nachvollziehbarkeit der Handlungen bei der Bildung des Vertrauens in den Agenten berücksichtigt. Nach [Klein u.a. 2004] könne insbesondere bei intelligenten Agenten die Vorhersehbarkeit durch situationsangepasstes Verhalten reduziert werden, was ironischerweise zu einer verminderten Bereitschaft zur Delegation führen könne.

Die Bedeutung von **Vertrauen in den Agenten** für die Delegationsentscheidung des Operateurs ist wegen dessen Komplexität als hoch zu bewerten. Der Agent zeigt Verhalten jenseits vordefinierter, prozeduraler Strategien, wodurch dessen vollständiges Verständnis verhindert wird [Lee & See 2004]. Das Verhalten des kognitiven Agenten muss hierbei die Bildung eines kalibrierten Vertrauens (engl. *calibrated trust*) [Lee & See 2004] erlauben. Kalibriertes Vertrauen meint in diesem Zusammenhang einen Grad an Vertrauen, in dem der Mensch die Fähigkeiten des Agenten weder über- noch unterschätzt, und der eine situationsgerechte Entscheidung über die Delegation von Aufgaben erlaubt.

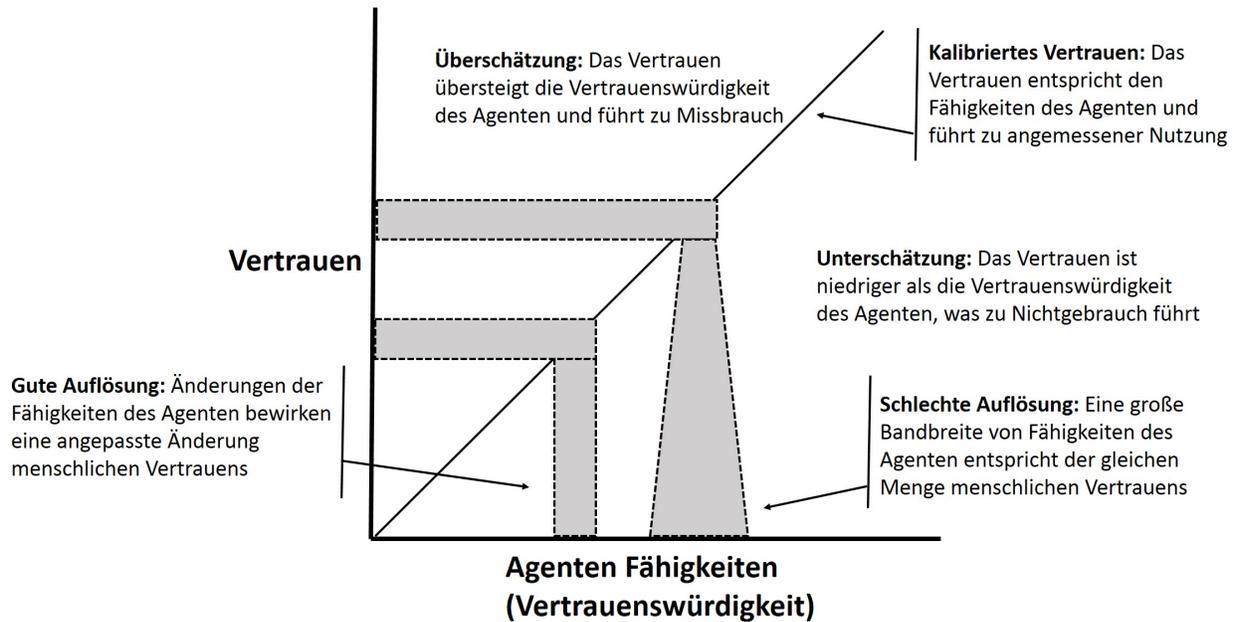


Abbildung 2-16: Zusammenhang zwischen Kalibrierung, Auflösung und Fähigkeiten des Agenten bei der Bestimmung angepassten Vertrauens in den Agenten. Überschätzung führt zu Missbrauch, Unterschätzung zu Nichtgebrauch des Agenten (nach [Lee & See 2004])

Abbildung 2-16 stellt den Zusammenhang zwischen kalibriertem Vertrauen und der Unter- bzw. Überschätzung des Agenten grafisch dar. Bleibt das Vertrauen des Operateurs hinter der Vertrauenswürdigkeit des Agenten zurück, handelt es sich um eine Unterschätzung seiner Fähigkeiten, was dazu führen kann, dass der Agent trotz Eignung nicht verwendet wird [Lee & See 2004]. Wird dem Agenten dagegen übermäßig viel Vertrauen entgegengebracht, so wird dieser überschätzt, was zu Missbrauch führen kann. Während Missbrauch dabei vor allem zu schlechten Ergebnissen der Aufgabenbewältigung führt, hat der Nichtgebrauch negative Auswirkungen auf die Belastung des Operateurs, da die erzielbare Belastungsminderung durch den Agenten nicht genutzt wird.

„[Trust] is the attitude that an agent will help achieve an individual’s goals in a situation of uncertainty and vulnerability.“ [Lee & See 2004]

Die Auflösung von Agentenfähigkeiten bezeichnet die Genauigkeit, mit der der Mensch die Vertrauenswürdigkeit von Agenten wahrnehmen bzw. einschätzen kann. Bei einer schlechten Auflösung wird einer großen Bandbreite von Agentenfähigkeiten das gleiche Maß an menschlichem Vertrauen entgegen gebracht. Unterschiedliche Fähigkeiten erzeugen somit ein

gleiches Maß an Vertrauen [Lee & See 2004]. Eine gute Auflösung bezeichnet eine sachgerechte Beurteilung der einzelnen Agentenfähigkeiten durch den Menschen. Änderungen der Agentenfähigkeiten bewirken angemessene Änderungen menschlichen Vertrauens.

Die Bildung eines kalibrierten Vertrauens in den Agenten ist somit abhängig von dessen (Kommunikations-)Verhalten (siehe Abschnitt 2.3.1.1). Die Art der Informationen und die Form, in der sie kommuniziert werden, haben beide Einfluss auf die Vertrauensbildung beim Menschen und damit letztlich auf seine Delegationsentscheidung [Lee & Moray 1992].

Die **Belastung** des Operators beschreibt ein objektives Maß der von außen auf den Menschen wirkenden Einflussfaktoren und ist damit ein Indikator für die Schwierigkeit einer Aufgabe. Delegiert der Operator Aufgaben an den kognitiven Agenten, verändert er dadurch die Belastung, der er ausgesetzt ist. Der kognitive Agent beeinflusst die Belastung des Operators dabei durch zwei Faktoren: Durch die Übernahme einer spezifischen Aufgabe, die anderenfalls der Mensch übernehmen müsste, sowie der mit dem Akt des Delegierens verbundenen Interaktion des Menschen mit dem Agenten.

Das primäre Ziel der Einführung eines kognitiven Agenten an Bord des UAV ist die Unterstützung des menschlichen Operators und dessen Entlastung bei kognitiven Aufgaben. Insgesamt muss somit die erzielbare Entlastung durch die Delegation von Aufgaben an den Agenten den damit verbundenen zusätzlichen Aufwand (durch die Interaktion) übersteigen. Hierbei gilt, dass die Neigung des Menschen Aufgaben an einen untergebenen Agenten zu delegieren steigt, je weniger Aufwand durch den Akt des Delegierens entsteht [Lewandowsky et al. 2000][Parasuraman & Riley 1997].

Die aus der Delegation von Aufgaben resultierende Belastung kann dabei gemäß den grundsätzlichen Leitungsfunktionen in drei Komponenten untergliedert werden:

1. Die eigentliche Delegationsentscheidung,
2. den Akt des Delegierens (also die Kommandierung des Agenten) sowie
3. die Überwachung des Agenten während der Aufgabenerfüllung.

Die Komplexität der Delegationsentscheidung ist dabei abhängig von den verfügbaren Informationen über die Fähigkeiten des Agenten und deren Auflösung, den Informationen bezüglich der Anforderungen der aktuellen Aufgabe sowie der Kompatibilität der beiden Informationen. Das Zusammenführen unterschiedlicher Informationen steigert dabei den Arbeitsaufwand des Operators und birgt die Gefahr von Fehlern bzw. von Abweichungen bei der Interpretation auf Grund von Unterschieden der mentalen Modelle von Mensch und Agent [Chen et al. 2014]. Eine größtmögliche Unterstützung erfährt der Operator somit, wenn der Agent seine Informationen in einem Format übermittelt, das dem Menschen möglichst wenig Transformationsleistung abfordert. Die Kommandierung und Überwachung des Agenten ist im Wesentlichen abhängig von der Mensch-Agenten Schnittstelle und den Informationen, die hierüber ausgetauscht werden. Abschnitt 3.2 betrachtet in diesem Zusammenhang einen adaptierbaren, auftragsbasierten Führungsansatz, der den Operator bei der Interaktion mit dem agentengeführten UAV-System wesentlich entlastet.

2.3.2 Modelle menschlicher Kognition

Abbildung 2-12 stellt den Menschen als simplifizierten Kopf mit Hirn dar und symbolisiert damit sein, durch höhere kognitive Fähigkeiten geprägtes, menschliches Verhalten. Modelle menschlicher Kognition bilden Aspekte der Verarbeitungsprozesse ab, die ihrerseits menschliches Verhalten bewirken.

Ein Modell menschlicher Kognition umfasst dabei drei Stufen:

1. Der Stimulus wird in eine interne Darstellung übersetzt.
2. Die Darstellung wird von kognitiven Prozessen manipuliert, um neue interne Darstellungen daraus abzuleiten.
3. Diese neuen Darstellungen werden zurück in Aktionen umgesetzt.

Es existieren unterschiedliche Modelle als vereinfachte Darstellungen menschlicher Kognition, die je nach Ursprung und Anwendungszweck ihren Schwerpunkt auf unterschiedliche Aspekte (z.B. Wahrnehmung oder Entscheidungsfindung) legen. Einen Überblick über Modelle menschlicher Kognition geben [McMillan et al. 1991], die Modelle im Hinblick auf den Entwurf automatisierter Systeme bewerten und kategorisieren.

Ein kognitives Modell für menschliches Verhalten in bekannten Situationen ist das *Recognition-Primed Decision* (RPD) Modell nach [Klein 2008], in dem Erfahrungen in Form von Mustern hinterlegt sind. Das RPD Modell nutzt Muster zum Abgleich mit aktuellen Situationen und die daraus resultierende Handlungsauswahl. Das Modell kombiniert dabei Intuition (der Abgleich vorhandener Muster) mit einem bewussten analytischen Vorgang, bei dem der Operateur vorhandene Handlungsmuster anpassen kann, falls die aktuelle Situation gegenüber dem gespeicherten Muster leichte Abweichungen aufweist.

Einen Ansatz zur Beschreibung menschlichen Verhaltens auch in unbekannt Situationen bietet das dreischichtige Modell nach [Rasmussen 1983] (vgl. Abbildung 2-17). Es unterscheidet zwischen den Regulationsebenen des fertigkeitbasierten (engl. *skill-based*), regelbasierten (engl. *rule-based*) und wissensbasierten (engl. *knowledge-based*) Verhaltens, welche abhängig von der jeweiligen Situation und der Erfahrung des Menschen kombiniert durchlaufen werden.

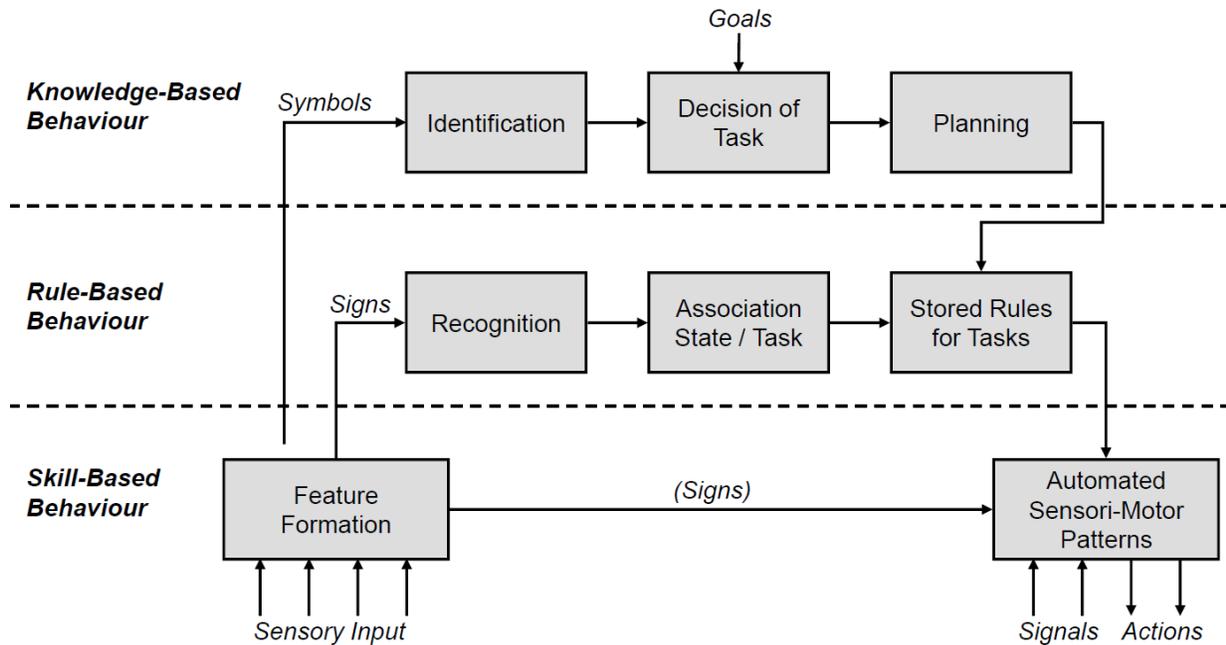


Abbildung 2-17: Vereinfachte Darstellung der drei Regulationsebenen menschlichen Verhaltens nach [Rasmussen 1983]

Auf der untersten Ebene bildet fähigkeitsbasiertes Verhalten unterbewusstes, höchst effizientes, kontinuierliches Handeln durch automatisierte sensomotorische Handlungsmuster ab. Fähigkeitsbasiertes Verhalten äußert sich in regulativem Handeln als direkte Reaktion auf eine kontinuierliche Sollwertabweichung oder als kontinuierliche Steuerung, mit der eigenes Handeln mit dem Zustand der Umwelt synchronisiert wird. Aktionen werden dabei über Signale (engl. *Signs*) aktiviert. Als Beispiel dient die kontinuierliche Steuerung eines LFZ per Steuerknüppel. Sensomotorische Muster können über *Stored Rules for Task* regelbasiert moduliert und an die jeweilige Situation angepasst werden.

Regelbasiertes Verhalten beschreibt das bewusste Abrufen einer Sequenz von Handlungsschritten in bekannten Situationen, für die Verhaltensregeln auf Grund früherer Erfahrung, Ausbildung oder durch Planung (engl. *Planning*) existieren. Das Verhalten wird dabei durch Regeln (engl. *Rules*) gesteuert, die implizit aus der aktuellen Situation oder aus früheren Erfahrungen resultieren. Zur Bestimmung der gewünschten Verhaltenssequenz wird zunächst versucht die aktuelle Situation zu erkennen (engl. *Recognition*) und anschließend eine Aufgabe zugeordnet (engl. *Association State / Task*).

In unbekanntem Situationen, für deren Bewältigung keine Regeln und Handlungsmuster existieren, muss eine höhere, konzeptionelle Ebene durchlaufen werden, deren Leistung zielbasiert und wissensbasiert (engl. *knowledge-based*) ist und ein explizites mentales Modell der Umgebung nutzt. Zunächst wird hierbei die Situation symbolisch identifiziert (engl. *Identification*) und anschließend, in Abstimmung mit den allgemeinen Absichten der Person, ein explizites Ziel zu formuliert (engl. *Decision of Task*). Anschließend können mehrere Pläne erstellt werden, die bezüglich ihres Nutzens für das eigene Ziel verglichen werden, um eine geeignete Auswahl zu

treffen. Tritt zu einem späteren Zeitpunkt eine vergleichbare Situation erneut auf, so kann diese direkt regelbasiert behandelt werden.

Entsprechend ihrer Verwendung werden Informationen aus der Umwelt auf den drei Ebenen unterschiedlich wahrgenommen. Auf der fertigkeitbasierten Ebene sind es Signale (*Signals*), die ohne gesonderte Bedeutung als kontinuierliche Parameter der zeitlich-räumlichen Dynamik dienen. Regelbasiertes Verhalten nutzt Zeichen (*Signs*), welche über Namen identifiziert werden und Zustände einer Situation beschreiben. Für das Verständnis und die Verarbeitung unbekannter Situationen bedarf es Symbole (*Symbols*), die auf der wissensbasierten Verarbeitungsebene verwendet werden. Symbole sind mit abgespeicherten Wissenskonzepten verknüpft und erlauben deren Verarbeitung und damit Schlussfolgerungen. Während Zeichen auf Zustände und Aktionen der Umgebung referenzieren, sind Symbole Teil des inneren mentalen Modells des Menschen und damit ohne direkten Bezug auf die Umwelt.

Rasmussens Modell menschlichen Verhaltens ist ein viel zitiertes Modell in den Kognitionswissenschaften und hat im Laufe der Jahre vielfache Verwendung und Weiterentwicklung (z.B. durch [Rasmussen et al. 1994]) erfahren. [Onken & Schulte 2010] präsentieren im Rahmen der Einführung kognitiver Automation eine Interpretation des Rasmussen Schemas mit Blick auf informationsverarbeitende Aspekte sowie die Implementierung von Wissen und die Wissensverarbeitung. Auf dieses Modell soll im Folgenden kurz eingegangen werden, da es der in dieser Arbeit verwendeten kognitiven Systemarchitektur COSA² [Brüggenwirth 2014] zu Grunde liegt.

Die Darstellung erfolgt über generische Blöcke, die kognitive Subfunktionen (engl. *Cognitive Sub-Functions*) repräsentieren und die das darin verwendete Wissen (engl. *Knowledge*) sowie ihre Ein- und Ausgabewerte (engl. *input / output*) identifizieren (vgl. Abbildung 2-18). Das einer solchen Subfunktion zugeordnete Wissen besteht aus semantisch kodierten expliziten oder impliziten Wissensbaukomponenten, äquivalent zum Inhalt des menschlichen Langzeitspeichers. Es wird folglich als *a-priori Wissen* bezeichnet und ist in der Abbildung blau dargestellt. Das *situative Wissen* (rot dargestellt) wird dagegen während der Laufzeit verarbeitet, bzw. generiert und dient als Ein- und Ausgangsinformation der kognitiven Subfunktion.

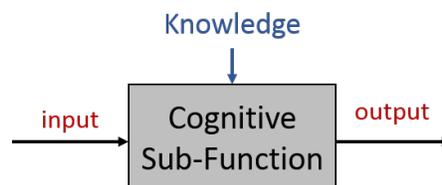


Abbildung 2-18: Formelle Beschreibung einer kognitiven Sub-Funktion und des relevanten Wissens nach [Onken & Schulte 2010]

Das Resultat der Interpretation des Verhaltensmodells nach Rasmussen ist in Abbildung 2-19 dargestellt. Die originären Bezeichnungen der regelbasierten und wissensbasierten Verhaltensebenen wurden durch die Begriffe prozedurbasiert (engl. *procedure-based*) und konzeptbasiert (engl. *concept-based*) ersetzt, um die vorrangige Form des Wissens der jeweiligen Verarbeitungsstufen zu erfassen. Zur Verdeutlichung der Art und Funktion einzelner Subfunktionen wurden ihre Bezeichnungen ebenfalls angepasst.

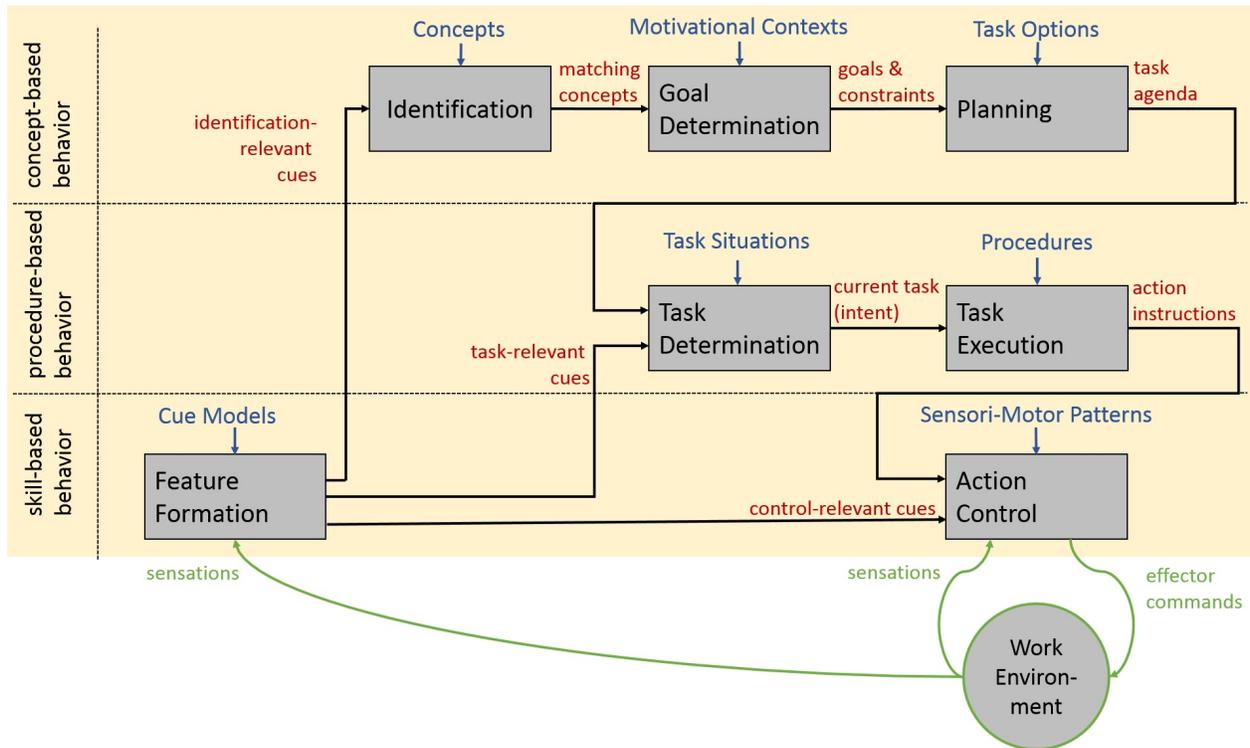


Abbildung 2-19: Interpretation des menschlichen Verhaltensmodells nach Rasmussen mit Blick auf informationsverarbeitende Aspekte und die Wissensrepräsentation nach [Onken & Schulte 2010]

Auf der fähigkeitsbasierten Ebene wird die *Feature Formation* als unterbewusster Wahrnehmungsprozess vollzogen, der Sinneseindrücke in räumlich-zeitliche Situationsmerkmale (engl. *cues*) überführt. Die Art der generierten Situationsmerkmale hängt dabei vorrangig von der anschließenden Verarbeitungsebene ab. Die Aktionskontrolle (engl. *Action Control*) verwendet steuerung- und regelungsrelevante Situationsmerkmale (engl. *control-relevant cues*) zur Auswahl und Parametrisierung sensomotorischer Muster (engl. *sensory-motor patterns*). Diese Merkmale können explizit als auch implizit sein und somit z.B. die Straßenkrümmung bei einer Kurvenfahrt durch einen Autofahrer beschreiben.

Auf der prozedurbasierten Verhaltensebene wird die Aufgabenbestimmung (engl. *Task Determination*) durch den Abgleich aufgabenrelevanter Merkmale (engl. *task-relevant cues*) mit a-priori Wissen über Aufgabensituationen (engl. *task situations*) durchgeführt. Als Beispiel für Aufgabensituationen nennen [Onken & Schulte 2010] dabei das Anhalten eines Fahrzeugs an einer roten Ampel. Die Aufgabenbestimmung ist damit ein bewusst vollzogener Prozess, der auf abgespeichertem Regelwissen beruht. Die Aufgabenausführung (engl. *Task Execution*) wählt anschließend eine passende Prozedur zur identifizierten Aufgabe aus und überführt diese in Handlungsanweisungen (engl. *action instructions*).

Bei prozedurbasiertem Verhalten ist die konzeptbasierte Verhaltensebene nicht inaktiv, sondern überwacht das Verhalten bezüglich ihrer Übereinstimmung mit eigenen Absichten, Zielen und einer aktiven Aufgabenagenda (engl. *task agenda*). Konzeptbasiertes Verhalten bedarf der Identifikation (engl. *Identification*) relevanter Konzepte (engl. *matching concepts*) über den Abgleich von a-priori Wissen mit identifikationsrelevanten Merkmalen. Der daraus abgeleitete

Weltzustand wird mit einem gewünschten Zustand (engl. *motivational contexts*) verglichen. Bei Abweichungen werden in der Zielbestimmung (engl. *Goal Determination*) Ziele und Randbedingungen (engl. *goals and constraints*) bestimmt, um den Zielzustand wiederherzustellen. Die Vorgaben dienen der Planung (engl. *Planning*) einer Handlungsagenda (engl. *task agenda*) zur Überführung des Zustands in den gewünschten Zielzustand. Hierzu wird a-priori Wissen über Handlungsoptionen (engl. *task options*) genutzt, die dem Menschen zur Verfügung stehen. Die Handlungsagenda wird kaskadiert über Prozeduren und sensomotorische Muster umgesetzt.

Das interpretierte Modell menschlichen Verhaltens nach [Onken & Schulte 2010] eignet sich auf Grund seiner modularen Darstellung kognitiver Subfunktionen als Prozesse besonders zur Überführung in eine kognitive Verarbeitungsarchitektur.

2.3.3 Kognitive Architekturen

Nach [Vernon et al. 2007] bemühen sich kognitive Architekturen um die Abbildung einer großen Bandbreite von Aspekten der kognitiven Psychologie, d.h. der psychologischen Vorgänge im Menschen zur Abbildung von Wahrnehmung, Erkenntnis und Wissen in einer einzigen einheitlichen Theorie. Bei kognitiven Architekturen wird zwischen kognitivistischen und emergenten Ansätzen unterschieden, wobei im Weiteren ausschließlich kognitivistische Ansätze behandelt werden, deren Fokus auf zeitlich konstanten und aufgabenunabhängigen Aspekten der Kognition liegt.

Da bis heute keine Architektur die gesamte Bandbreite menschlicher Kognition abbildet, existieren zahlreiche verschiedene kognitive Architekturen mit unterschiedlichen Anwendungszwecken. [Vernon et al. 2007] und [Chong et al. 2007] präsentieren jeweils eingehende Betrachtungen kognitiver Architekturen und identifizieren die im Folgenden beschriebenen Systeme als wesentliche kognitivistische Ansätze.

Die Architekturen **EPIC** [Kieras & Meyer 1997] und **ACT-R** [Anderson et al. 2004] dienen primär der realitätsnahen Modellierung des Menschen und dessen Informationsverarbeitung und nur sekundär der Abbildung kognitiver Fähigkeiten zur Funktionsdarstellung. Sie eignen sich somit lediglich bedingt zur Realisierung kognitiver Agenten.

Mit **EPIC** (*Executive Process Interactive Control*) wird menschliche Leistung simuliert, indem Modelle von Wahrnehmung und Motorik über ein zentrales Produktionssystem verbunden werden. Über die Vorgabe von Wissen in Form von Produktionsregeln und aufgabenabhängigen Modellparametern wird die zeitliche Dauer der Informationsverarbeitung eines Menschen simuliert. Hierzu dienen sog. Prozessoren, die menschliche Sinne und die Motorik modellieren und die sowohl parallel als auch seriell durchlaufen werden. Der typische Anwendungsbereich für EPIC ist der Entwurf und die Bewertung von Mensch-Maschine Schnittstellen an Computer-Arbeitsplätzen [Kieras & Meyer 1997]. [Kieras & Meyer 2000] beschreiben hierzu konkret die Anwendung von EPIC zur Simulation von Operateurverhalten in einer synthetischen Aufgabenstellung, bei der gleichzeitig auf unterschiedlichen Bildschirmen ein Luftfahrzeug per Joystick verfolgt und anhand eines Radarbildes Luftfahrzeuge klassifiziert werden mussten.

Bei **ACT-R** (*Adaptive Control of Thought-Rational*) wird Kognition modular in einem System implementiert, und es wird untersucht, wie diese Module integriert werden müssen, um kohärente Kognition zu schaffen. Module bilden dabei bestimmte Hirnregionen des Menschen in Form

verteilter Informationsverarbeitung ab, die über ein zentrales Produktionssystem verbunden sind [Anderson et al. 2004]. Die Module stellen ihre Informationen dabei über Buffer bereit, in die jeweils nur ein Chunk, d.h. eine einzelne Einheit deklarativen Wissens, hinterlegt werden kann. Über das zentrale Produktionssystem wird deklaratives Wissen von prozeduralem Wissen getrennt. Hier werden einzelne Chunks zyklisch verwendet und dabei symbolisches Wissen und numerische Parameter kombiniert angewandt [Howes & Young 1997]. Bei der Verarbeitung in ACT-R existieren bewusst zwei Flaschenhälse, um die Nachbildung menschlicher Performanzlimitationen zu erzielen. Zum einen ist der Inhalt der Buffer jeweils streng auf einen Chunk limitiert, zum anderen feuert pro Zyklus nur eine einzelne Produktionsregel. Falls mehrere Produktionen feuern können, feuert diejenige, bei der die größte Wahrscheinlichkeit besteht, das aktuelle Ziel zu erreichen. ACT-R kann während der Laufzeit neue Chunks und Produktionsregeln lernen. Neue Chunks werden durch Zielerfüllung oder neue registrierte Events zum deklarativen Wissen hinzugefügt. Neue Produktionsregeln lernt ACT-R durch die Kombination zweier nacheinander gefeuerter Regeln zu einer einzelnen Regel. Hierdurch ist es mit ACT-R möglich „Abkürzungen“ zu lernen [Taatgen et al. 2006]. ACT-R wurde in vielfältigen Projekten zur Simulation menschlichen Verhaltens eingesetzt, z.B. zur Untersuchung und Prädiktion von Hirnaktivitäten [Borst & Anderson 2014] sowie der menschlichen Sprachverarbeitung [Howes et al. 2011]. [Gluck et al. 2003] evaluieren die Vergleichbarkeit des Verhaltens eines mittels ACT-R simulierten UAV-Operators mit menschlichen Probanden bei der Durchführung grundlegender Flugmanöver. [Dimperio & Gunzelmann 2008] untersuchen darauf aufbauend die Vergleichbarkeit von Missionsleistungen im Rahmen einer Predator Aufklärungsmission in Bedrohungsgebieten. [Gluck et al. 2005] nutzen das auf ACT-R basierende *Verbalization Between Operators and Synthetic Entities* (VERBOSE) Modell zur Simulation eines virtuellen Teammitglieds in einem Predator UAV-Team zu Zwecken des Trainings sowie der Verhaltensanalyse und -bewertung.

Die **Subsumption** Architektur [Brooks 1999] stellt einen *bottom-up* Ansatz zur konkreten Einbettung eines Agenten in seine physikalische Umgebung dar, indem Wahrnehmung und Handlung direkt über einzelne einfache, verhaltensbasierte Module verbunden werden. Subsumption hat seinen Ursprung in der Robotik und verfolgt, entgegen der klassischen AI, keine zentrale Zielumsetzung durch ein geschlossenes System, sondern vielmehr eine hierarchisch gegliederte Regelung des Umweltzustands entsprechend des vorgegebenen Ziels. Die Module einer jeden Hierarchieebene können dabei unabhängig voneinander die Umwelt entsprechend von Vorgaben regeln und werden über die jeweils übergeordnete Ebene subsumiert. Während das Verhalten der unteren Ebenen damit eine Reaktion auf das Umweltverhalten darstellt, verfolgen die oberen Ebenen mit ihrem Verhalten das vorgegebene Ziel. Durch die modulare Architektur lassen sich höhere Abstraktionsebenen durch das Hinzufügen übergeordneter Hierarchieebenen einfügen, ohne dass die existierenden Module dafür angepasst werden müssen [Brooks 1999]. Gleichzeitig ermöglichen die unteren Hierarchieebenen die Abbildung von Realzeitverhalten.

Subsumption verzichtet hierbei vollständig auf die Verwendung symbolischen Wissens, mit dem Argument, dass es die Umgebung unvollständig und simplifiziert abbilden könne [Brooks 1990]. Ebenso existiert keine zentrale Wissensrepräsentation zur Abbildung des Umweltzustands, sondern es werden subsymbolische Informationen dezentral in den einzelnen Modulen verarbeitet. Anwendung findet Subsumption in verschiedenen Robotik Anwendungen zur Navigation, der Vermeidung von Hindernissen sowie im Rahmen einer Multi-Agenten Anwendung im

Roboterfußball [Nakashima & Noda 1998]. [Hartley & Pipitone 1991] realisieren mit Subsumption ein einfaches Missionsmanagement für eine Flugsimulation, wobei das Flugverhalten und der Einsatz der Missionspayload in Form von Zustandsautomaten in Subsumption Modulen implementiert wurden. Sie kritisieren hierbei, dass einzelne Module nicht komplett unabhängig voneinander zu implementieren seien und somit hierarchisch übergeordnete Module inkrementell komplexer werden. Zu viele, parallel operierende Module führten zu Ressourcenkonflikten und es sei nicht möglich exklusives Verhalten in Subsumption abzubilden.

Die kognitiven Architekturen **ICARUS** [Langley & Choi 2006], **BDI** [Rao & Georgeff 1995] und **Soar** [Laird et al. 1987] eignen sich ebenfalls für die Umsetzung der Fähigkeiten intelligenter Agenten. Im Gegensatz zur Subsumption nutzen sie jedoch eine geschlossene Infrastruktur mit Formalismen, Repräsentationen und Verfahren zur Verarbeitung expliziten Wissens, welches auf eingehende Informationen während der Laufzeit angewendet wird. Während ICARUS entsprechend seines Ursprungs hauptsächlich zur Schaffung physikalischer Agenten eingesetzt wird, dienen BDI und Soar in verschiedenen Applikationen zur Umsetzung von Software Agenten.

ICARUS stimmt in großen Teilen mit ACT-R und Soar überein, da alle drei Architekturen symbolische Wissensrepräsentationen nutzen, Mustererkennung zur Auswahl relevanter Elemente verwenden und ihr Verhalten in Form eines *Recognize-Act Cycle* organisieren [Langley & Choi 2006]. ICARUS behandelt jedoch vorwiegend die Wahrnehmung und Handlung des Agenten, welche über die Kognition verbunden sind (engl. *Primacy of Perception and Action*). Hierzu wird das symbolische Langzeitwissen in Kategorien, d.h. boolesche Konzepte zur Kategorisierung von Wahrnehmungen und Fähigkeiten mit spezifischen Zielen und Bedingungen unterteilt (engl. *Separation of Categories and Skills*). Die Elemente des Langzeitwissens sind streng hierarchisch strukturiert (engl. *Hierarchical Structure of Memory*) und dienen als Grundlage, aus denen Instanzen im Kurzzeitwissen aufgerufen werden (engl. *Long-Term/Short-Term Correspondence*). Schließlich werden Kosten für Fähigkeiten definiert, deren Gewichtung situationsabhängig variiert und die entlang eines Lösungspfades aufsummiert werden. Bei konkurrierenden Pfaden wird die kostenminimale Lösung zur Durchführung verwendet (engl. *Value-Driven Nature of Behavior*).

Gleichzeitig verfolgt ICARUS jeweils nur ein Ziel, welches sich aus einer internen Priorisierung ergibt. Führt keine Fähigkeit direkt zur Zielerreichung, wird über eine *Means-Ends* Analyse, d.h. durch das Aufteilen des Ziels in Unterziele, ein Lösungspfad in Form einer Verkettung einzelner Fähigkeiten bestimmt [Chong et al. 2007]. [Langley & Choi 2006] beschreiben verschiedene Applikationen für ICARUS. In einer Innenstadt-Fahrsimulation wird ICARUS zur Planung und Durchführung von Steuereingaben (u.a. Geschwindigkeits- und Lenkradeingriff) und Routen an ein Fahrzeug zur Auslieferung von Paketen eingesetzt. Hierbei reagiert das System auf das Verhalten simulierter Fremdfahrzeuge und agiert entsprechend einer vereinfachten Straßenverkehrsordnung. In einer weiteren Anwendung wird ICARUS zur Stabilisierung eines zweidimensionalen, invertierten Stab-Pendels verwendet, in dem der Winkel und die Winkelgeschwindigkeit des Stabes gemessen und der Eingriff über Bewegung am unteren Stabende erfolgt. [Choi et al. 2007] nutzen ICARUS zur Simulation eines menschlichen Computerspielers in einem Ego-Shooter bei der Erkundung der Missionskarte auf der Suche nach einer platzierten Bombe. Hierbei lernt der Agent anhand geglückten Verhaltens für zukünftige Missionen, wodurch Gebiete nur einmal erkundet werden müssen. Die Ein- und

Ausgabeschnittstellen des Agenten wurden dabei softwareseitig synthetisiert, um Wahrnehmung und Handlungen zu vereinfachen.

BDI (*Belief-Desire-Intention*) Architekturen [Rao & Georgeff 1995] implementieren das zugrunde liegende BDI Modell, welches den Ansatz der praktischen Vernunft (engl. *practical reasoning*) nach [Bratman et al., 1988] verfolgt. Der Ansatz der praktischen Vernunft beschreibt deliberative Entscheidungen zur Bestimmung welche Handlungen auf welche Weise vollzogen werden sollen. Die Architektur eignet sich zur Implementierung von Softwareagenten, die als BDI Agenten bezeichnet werden.

„*Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes.*“ [Bratman et al., 1988]

Die BDI Architektur verknüpft drei Arten von Wissen: Ziele (engl. *desires*), Weltwissen (engl. *beliefs*) und Absichten (engl. *intentions*). Eine Datenbank verwaltet das Weltwissen, welches Fakten über die Umgebung, den inneren Zustand sowie Inferenzregeln zur Schlussfolgerung neuer Fakten enthält. Die hinterlegten Ziele sind für den BDI Agenten handlungsbestimmend und werden entsprechend des vorliegenden Weltwissens vom Agenten selbst aktiviert. Absichten beschreiben das vom Agenten gewählte Verhalten zur Erreichung der Ziele. Sie generieren sich über *Means-Ends* Verknüpfung aus in der Wissensdatenbank hinterlegten, hierarchisch organisierten Plänen, welche jeweils aus Teilzielen bestehen, die über Handlungsfolgen vom Agenten umgesetzt werden [Chong et al. 2007]. BDI Agenten wurden ursprünglich als Fehlerdiagnosesystem für das *reaction control system* im NASA Spaceshuttle Programm konzipiert [Georgeff & Ingrand 1989], seitdem jedoch auch zur Werkssteuerung und zum Geschäftsprozessmanagement eingesetzt [Georgeff & Rao 1996].

Die **Soar** Architektur ist ein früher KI Ansatz zur Abbildung allgemein intelligenten Verhaltens insbesondere des Problemlösens [Laird et al. 1987]. Die frühere Bezeichnung *State, Operator and Result* (Soar) beschreibt hierzu den allgemeinen Verarbeitungsansatz: Ein Operator wird auf einen Zustand (engl. *state*) angewendet und führt zu einem Ergebnis (engl. *result*). Der Anspruch der Abbildung „der ganzen Bandbreite“ kognitiven Verhaltens [Laird et al. 1987] verweist auf den Ansatz von Soar zur Verwirklichung einer einheitlichen Kognitionstheorie (engl. *unified cognition theory*) nach [Newell 1990]. Soar basiert als Architektur auf einem Produktionssystem bzw. einem regelbasiertem System [Lunze 2010], bestehend aus einer *Wissensbasis* mit aufgabenunabhängigen Fakten und gültigen Regeln, einem *Arbeitsspeicher*, in dem der Ausgangszustand und der aktuelle Problemzustand abgelegt sind sowie einer *Inferenzmaschine*, die die Verarbeitung des Wissens realisiert. Die Wissensverarbeitung gliedert sich in die Auswahl der anzuwendenden Regeln und Fakten und einen *Regelinterpreter*, der die Anwendung der gewählten Regeln realisiert und den *Arbeitsspeicher* aktualisiert.

Soar nutzt prozedurales Wissen im (Langzeit) Produktionsspeicher (engl. *production memory*) und einen davon getrennten Arbeitsspeicher (engl. *working memory*). Die Verarbeitungsstruktur umfasst einen Produktions- und einen Entscheidungszyklus. Zunächst feuern alle Produktionsregeln, die auf den aktuellen Arbeitsspeicher anwendbar sind, und verändern damit seinen Inhalt. Dieser Zyklus wird solange wiederholt, bis keine Regel mehr feuert. Anschließend

wird ein Entscheidungszyklus (engl. *decision procedure*) durchlaufen, bei dem eine einzelne Aktion (aus allen möglichen Aktionen) ausgewählt und angewandt wird. Auf eine Entscheidung entfallen damit beliebig viele Produktionszyklen [Vernon et al. 2007]. Kann keine Aktion bestimmt werden, so generiert Soar einen neuen Problemraum mit dem konkreten „Unterziel“ der Lösung des Konflikts. Dieser Vorgang wird als *universal subgoalng* bezeichnet. Findet Soar auf diese Weise eine Lösung, verwendet es eine Lernmethode (engl. *chunking mechanism*) zur Ableitung einer entsprechenden Regel, die im Produktionsspeicher für zukünftige Aufgaben hinterlegt wird.

Ein Überblick über Projekte, in denen Soar zur Realisierung von Agenten zur Steuerung unbemannter Fahrzeuge in simulierten Umgebungen eingesetzt wurde, findet sich z.B. bei [Lehman et al. 2006]. *TacAir-Soar* [Jones et al. 1999] und *RWA-Soar* [Hill Jr. et al. 1998] stellen hierbei die derzeit größten Soar Projekte dar (*TacAir-Soar* besitzt mehr als 8000 Produktionsregeln). In *TacAirSoar* wird Soar zur Realisierung von Agenten verwendet, die individuell und gemeinsam in dynamischen, nichtdeterministischen Simulationsumgebungen taktische Luftmissionen für Starr- und Drehflügler durchführen. Die Soar Modelle wurden im Rahmen des Trainings menschlicher Piloten eingesetzt; ihre Kommunikation geschah dabei über standardisierte Militärsprache über simulierte Radiosysteme. Soar MOUTBOT [Wray et al. 2004] setzt die Ansätze von *TacAir-Soar* und *RWA-Soar* für militärische Umgebungen in urbanem Gelände um.

Basierend auf Soar verbindet **ADAPT** (Adaptive Dynamics and Active Perception for Thought) [Benjamin et al. 2004] den robotischen Ansatz der aktiven Wahrnehmung (*active Perception*), d.h. die Betrachtung von Wahrnehmung als zielgerichteter Problemlösungsprozess, mit kognitiven Verarbeitungsmechanismen. ADAPT implementiert hierbei zusätzlich Aspekte von ACT-R (deklarativer Langzeitspeicher mit sensomotorischen Mustern) und EPIC (u.a. werden alle Wahrnehmungsprozesse parallel ausgeführt) [Vernon et al. 2007].

Ergänzend zu den in der Literatur beschriebenen Architekturen wurden am Institut für Flugsysteme (IFS) der UniBwM zwei kognitive Systemarchitekturen zur Schaffung kognitiver Agenten im Sinne von Mode 1 oder Mode 2 *Cognitive Automation* entwickelt. Zunächst stellt [Putzer 2004] die auf Soar basierende *Cognitive System Architecture (COSA)* vor. COSA basiert auf einem „einheitlichen, formalisierten Schema für Denkprozesse zur Erzeugung zielgerichteten Verhaltens“ – dem Kognitiven Prozess (KP) [Putzer 2004]. Das Ziel dabei ist die Abbildung menschenähnlichen Verhaltens gegenüber der Umwelt durch die Anlehnung der internen Verarbeitung von COSA an menschliche Denkprozesse. Eine zentrale Eigenschaft von COSA ist dabei die strikte Trennung von Wissen und Verarbeitungsprozessen. Das uniforme Verarbeitungskonzept des KP verarbeitet statisches a-priori Wissen und Situationswissen, welches während der Laufzeit erzeugt und durch Transformatoren verändert wird. Explizit hinterlegte Wünsche dienen der Zielbestimmung, deren Erreichung über Pläne mittels hinterlegter Vorgehens- und Anweisungsmodelle realisiert wird. Zur Interpretation von Umweltwahrnehmungen dient hierbei ein symbolisches Umweltmodell. [Putzer 2004] beschreibt zusätzlich eine horizontale Teilung des KP in mehrere Schichten, was zu *Paketen* oder *Modulen* führt, die im hohen Maße wiederverwendbar sind und unterschiedliche Anwendungsdomänen bedienen.

COSA wurde am IFS bereits in verschiedenen Forschungsvorhaben u.a. zur Realisierung kognitiver Agenten zum Missionsmanagement einzelner UAV sowie von kooperierenden UAV

und von Assistenzsystemen für UAV-Operateure eingesetzt. (Eine chronologische Betrachtung der für diese Arbeit relevanten Projekte findet sich in Abschnitt 2.3.4). Hierbei konnte gezeigt werden, dass COSA eine geeignete Basis für die Abbildung wissensbasierten Verhaltens von zielebasierten Agenten darstellt. Die in COSA realisierte Planung ermöglicht über *Means-Ends* Verknüpfungen komplexe Pläne mit großem Einfluss des Entwicklers auf das Laufzeitverhalten, mindert damit jedoch die mögliche Planungsflexibilität [Brüggenwirth 2014]. Bei der Ausführung von COSA ergeben sich hohe Anforderungen an die Prozessorleistung und den Arbeitsspeicher des genutzten Computersystems, welche zu überwiegendem Anteil (90%) dem Produktionssystem des Soar Prozessors zuzuordnen sind [Matzner 2010]. Zur Verbesserung der Performanz stellt [Matzner 2010] daher eine algorithmische Weiterentwicklung von COSA vor, welche ein Produktionssystem basierend auf Graphtransformation realisiert und damit die Abbildung erheblich komplexerer Problemstellungen ermöglicht.

Zur Überwindung der genannten Restriktionen beim Einsatz von COSA setzen [Brüggenwirth & Schulte 2012] mit *Cognitive System Architecture with Centralized Ontology and Specific Algorithms (COSA²)*, auf COSA aufbauend das weiterentwickelte Rasmussen Schema nach [Onken & Schulte 2010] in einer Systemarchitektur um. [Brüggenwirth 2014] benennt hierbei die Plattformunabhängigkeit der Architektur bei der Integration einer klassischen Planungskomponente sowie einer robusten Planausführung als wesentliche Komponenten der kognitiven Architektur. Sie besitze dabei „*bold Charakter*“, d.h. sie verfolge langfristige Ziele und führe Umplanungen für existierende Pläne nur dann durch, falls diese nicht mehr durchführbar seien. Die kognitiven Subfunktionen, welche jeweils über spezifische, effiziente Algorithmen implementiert seien, seien in COSA² hierbei unter strikter Trennung vom Anwendungswissen hinterlegt.

Abbildung 2-20 zeigt die algorithmischen Klassen zur Umsetzung der kognitiven Subfunktionen des interpretierten Rasmussen Modells in COSA². Die Situationswahrnehmung wird über inferenzbasierte Algorithmen implementiert, die ein Produktionensystem unter Nutzung von Graphentransformation nach [Matzner 2010] darstellen [Brüggenwirth 2014]. Die zielgerichtete Planung erfolgt anschließend über suchbasierte Verfahren, wobei für COSA² alle zum *Planning Domain Definition Language* (PDDL) Standard kompatiblen Planer als generischer Standard genutzt werden können. Nach der Planung der Handlungsagenda ist es notwendig, diese zu terminieren, d.h. Ausführungszeitpunkte für einzelne Handlungen festzulegen. Hierfür wird aus der Handlungsagenda ein *Constraint Optimization Problem* formuliert, welches in COSA² über die externe IBM CPLEX [IBM Corp. 2015a] Bibliothek gelöst wird. Während der Ausführung ist die zentrale Aufgabe der *Task Determination* Funktion die Weiterschaltung der Handlungsagenda. Das Herunterbrechen von Aufgaben in Handlungsanweisungen erfolgt in der *Task Execution* über prozedurale Algorithmen. Hierfür werden Skripte oder Prozeduren als direkt ausführbare Anweisungen in COSA² hinterlegt und bei Bedarf aufgerufen. Externe Algorithmen stellen in der COSA² Architektur Funktionen konventioneller Automation dar, die COSA² z.B. Informationen in geeigneter Weise zuführen oder dessen Anweisungen umsetzen. Limitierungen der algorithmischen Umsetzung des COSA² Frameworks werden in Abschnitt 3.3 im Hinblick auf die Implementierung eines kognitiven Agenten beschrieben.

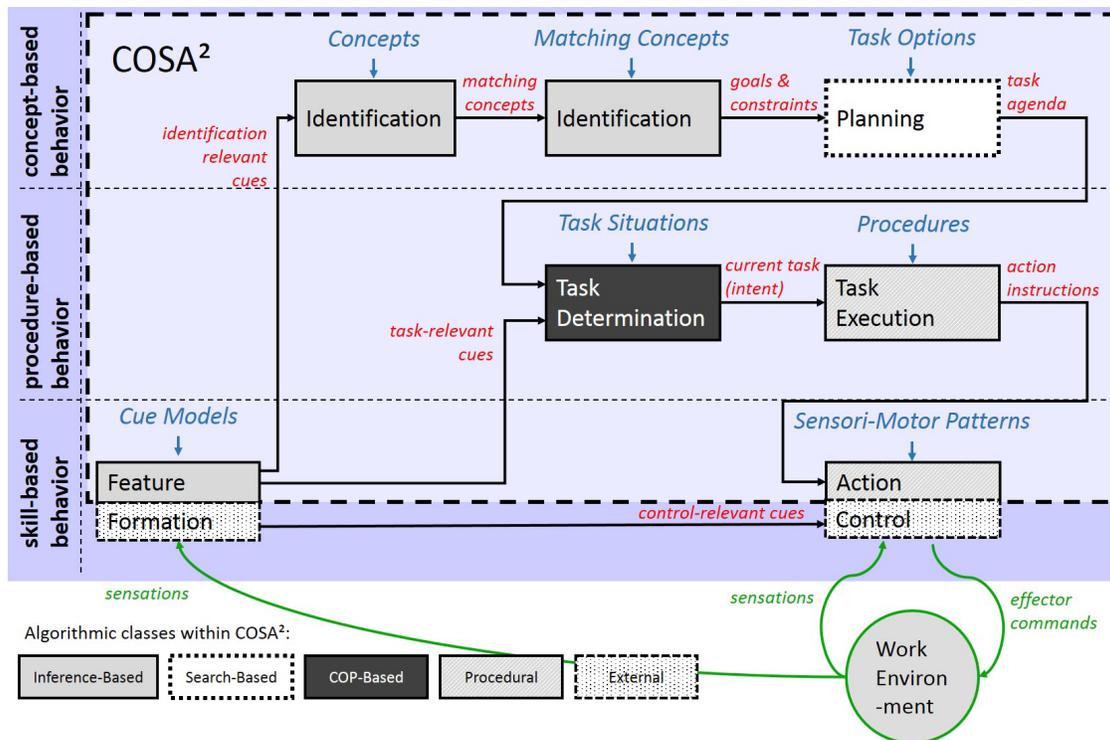


Abbildung 2-20: Algorithmische Klassen in COSA² zur Umsetzung des interpretierten Rasmussen Modells für menschliches Verhalten; COP = Constraint Optimization Problem [Brüggenwirth 2014]

Die inferenzbasierte Situationswahrnehmung (also die kognitiven Subfunktionen *Feature Formation*, *Identification* und *Goal Determination*) wird von [Brüggenwirth & Schulte 2012] mit einem Modell des Situationsbewusstseins nach [Endsley 1995] in einen Zusammenhang gebracht, welches drei Level von Regeln definiert (vgl. Abbildung 2-21). *L1 Regeln* beschreiben Regeln zur Wahrnehmung von Eingangssignalen, die insbesondere zur Wandlung von subsymbolischen Informationen in symbolische Wissensinterpretationen dienen. Hierbei werden neue Wissens Elemente geschaffen oder vorhandene aktualisiert. *L2 Regeln* dienen dem aktuellen Lageverständnis und werden auf das Situationswissen angewendet. Sie dienen der Interpretation und Bewertung von Zuständen, um daraus neues Wissen zu schaffen. *L3 Regeln* umfassen Regeln zur Prädiktion des Verhaltens der Umwelt und deren Größen.

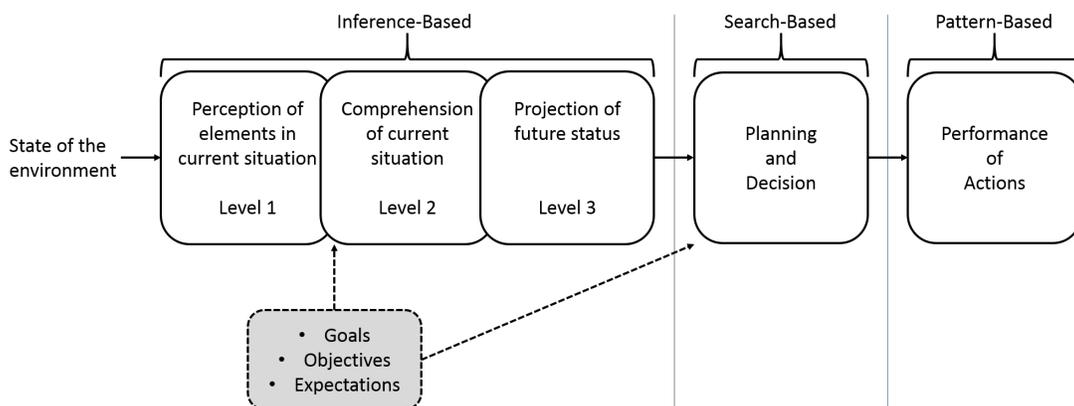


Abbildung 2-21: Interpretiertes Modell zur algorithmischen Umsetzung von Funktionen für die Schaffung von Situationsbewusstsein ([Brüggenwirth 2014] nach [Endsley 1995])

Feature Formation nutzt *L1 Regeln*, um eingehende Rohdaten zunächst mit semantischen Informationen zu verknüpfen und als *identification relevant cues* zur Verfügung zu stellen. *Identification* erstellt hieraus ein höheres Verständnis, indem es mittels *L2 Regeln* die aktuelle Situation analysiert und mittels *L3 Regeln* Vorhersagen über künftige Zustände trifft. *Goal Determination* vollzieht auf dieser Basis einen Abgleich zu gewünschten Zuständen und Kriterien, die als *Motivational Context* modelliert sind [Brüggewirth & Schulte 2012].

Planning nutzt in COSA² einen Projektionsansatz, bei dem Effekte von Handlungsoptionen (*task candidate*) während der Suche über das bereits für die *Identifikation* und *Goal Determination* modellierte Wissen (*L1* und *L2 Regeln* sowie *Motivational Contexts*) virtuell nachvollzogen werden (vgl. Abbildung 2-22). Der projektionsbasierte Planungsansatz nutzt konsequent die zentrale Wissensrepräsentation in COSA², da zuvor verwendetes a-priori Wissen wiederverwendet wird.



Abbildung 2-22: Projektionsbasierter Planungsansatz in COSA² nach [Brüggewirth 2014]

A-priori Wissen wird in COSA² unter Nutzung der *Cognitive Modelling Language* (CML) dargestellt, welche eine Weiterentwicklung der *Cognitive Programming Language* (CPL) darstellt. [Aurich 2011] schlägt für die Wissensmodellierung einen Verfahrensansatz ähnlich der objektorientierten Programmierung in der Softwareentwicklung vor. Demnach soll zunächst eine statische Analyse, ausgehend von Anforderungen, zu einem Klassendiagramm führen, welches in einer anschließenden dynamischen Analyse durch gewünschtes Verhalten erweitert wird und schließlich in ein CML-Modell mündet. Die anwendungsbezogene Erstellung eines CML-Modells in COSA² mit dem entsprechenden Editor wird in Abschnitt 4.1.1 vollzogen.

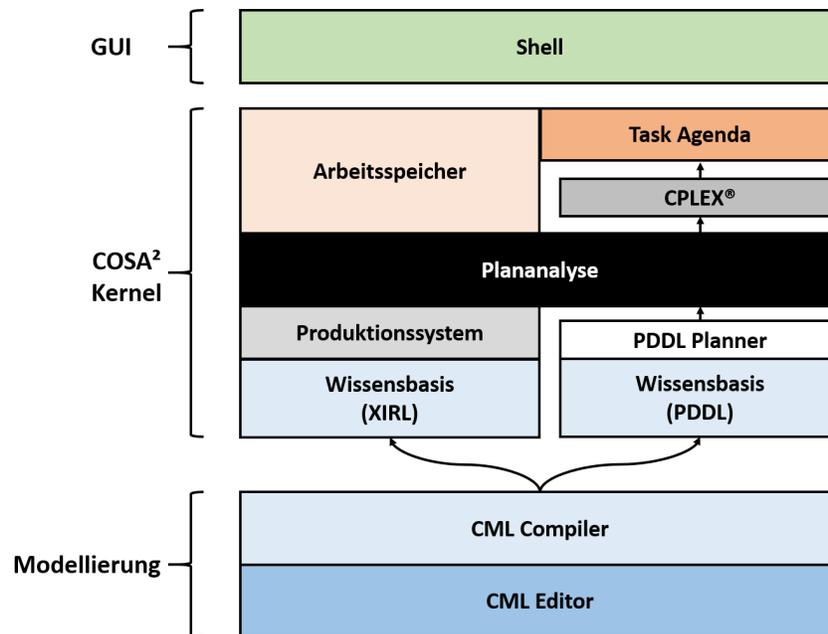


Abbildung 2-23: Komponentenorientierte Darstellung der COSA² Architektur [Brüggenwirth 2014]

Abbildung 2-23 zeigt eine komponentenorientierte Darstellung der COSA² Architektur mit den drei eigenständigen Softwarebausteinen *GUI*, *COSA² Kernel* und *Modellierung* [Brüggenwirth 2014]. Die Modellierung des a-priori Wissens von COSA² erfolgt durch einen Wissensingenieur, der über einen grafischen *CML Editor* das einheitliche CML-Modell erstellt und über den *CML Compiler* in andere Wissensformate (XIRL und PDDL) überführt. Der *COSA² Kernel* umfasst das Produktionssystem für die Inferenz, die Plananalyse und den Arbeitsspeicher. Der modular wählbare *PDDL Planner* sowie die *CPLEX®* Bibliothek zum Terminieren der *Task Agenda* sind an den Kernel angebunden. Zur Überwachung, Steuerung und zu Debugging-Zwecken existiert die *COSA² Shell*, die eine grafische Anzeige des *Arbeitsspeichers* und der *Task Agenda* realisiert (vgl. Abschnitt 4.1.1).

2.3.4 Entwicklungslinie kognitiver Agenten zur UAV-Führung am IFS

Die vorliegende Arbeit reiht sich ein in eine Reihe von Forschungsarbeiten am IFS, die unterschiedliche Aspekte der Schaffung kognitiver Agenten im Kontext der UAV Führung betrachten. Die für diese Arbeit wesentlichen Elemente der Entwicklungslinie werden kurz vorgestellt, um hieraus im Folgenden die Zielsetzung sowie die Alleinstellungsmerkmale der vorliegenden Arbeit abzuleiten.

Die erste Anwendung der COSA Architektur am IFS stellt das *Unmanned Cognitive System for the Flight Domain* (COSY-flight) dar, welches [Putzer 2004] als Zielsystem hauptsächlich zum Zweck der Validierung dient. COSY-flight operiert in einer simulierten, militärischen Aufklärungsmission als zielebasiert agierendes UAV, das dynamisch u.a. auf Umgebungsdaten und andere UAV reagiert. [Frey 2004] erweitert COSA um eine Kontrollschicht zur Überwachung und Kontrolle des kognitiven Systems in UAV-Anwendungen, z.B. bei fehlerhaften Zielspezifikationen durch den Operateur oder unzureichende Vehikelressourcen. Weder [Putzer 2004] noch [Frey 2004] reflektieren in ihren Betrachtungen dabei die Interaktion des kognitiven Agenten mit dem Menschen.

[Meitinger 2008] setzt COSA im Rahmen des Projekts COSY-team zur wissensbasierten, kooperativen Missionserfüllung durch mehrere UAV ein. Die Validierung erfolgt hierzu im Rahmen einer generischen SEAD-/Attack-Missionssimulation, in denen fünf UCAV mit unterschiedlicher Payload im Team ein von *Surface-to-Air-Missile* (SAM) Stellungen geschütztes, militärisches Ziel bekämpfen. [Meitinger 2008] führt dabei erste Experimente zur Untersuchung des Zusammenwirkens eines menschlichen Piloten mit einem COSA Agenten durch. Der Pilot übernimmt dabei die Rolle eines Teammitglieds, welches mit den übrigen agentengeführten UCAV im Team agiert.

In einem *Manned-Unmanned Teaming* (MUM-T) Projekt des IFS werden drei UAV von einem Operateur aus einem bemannten, militärischen Helikoptercockpit in einer dynamischen Missionssimulation geführt. COSA dient dabei der Umsetzung kognitiver Agenten an Bord der UAV [Uhrmann 2013] und als Assistenzsystem zur Unterstützung des multi-UAV-Operators [Rauschert 2013]. Die Führung der onboard Agenten geschieht auftragsbasiert und nutzt das Verständnis der Agenten bezüglich ihrer gegenwärtigen Situation, ihre Fähigkeiten zur reaktiven Auftragsbefreiung und der proaktiven rollenspezifischen Kooperation bei der Arbeit an einem gemeinsamen Auftragsselement. Ein besonderes Merkmal der Agenten ist die Fähigkeit zur proaktiven Erkennung und Nutzung *günstiger Gelegenheiten*, welche die selbstständige Durchführung von Handlungen zur Erreichung eines übergeordneten Ziels ohne Beeinträchtigung der aktuellen Auftragsbefreiung bedeuten [Uhrmann 2013]. Die Interaktion mit dem Agenten erfolgt über eine dynamische Lagekarte, auf der Aufträge für das aktuell selektierte UAV gewählt und kommandiert werden. Rückmeldungen über die UAV erhält der Operateur ebenfalls über die Lagekarte und über eingblendete Textnachrichten. Das Assistenzsystem versucht parallel dazu mögliche Fehler des UAV-Operators bei der auftragsbasierten multi-UAV-Führung zu erkennen und durch entsprechende Konsequenzen entgegenzuwirken. Wissen über Missionselemente, über Auftragsstypen und -status sowie Handlungen des Operators führen dabei zu der Feststellung von Handlungsbedarf und ggf. zu einer Eingriffsentscheidung [Rauschert 2013]. Die Fähigkeiten und die Akzeptanz des Automationsansatzes wurden in einem Mensch-Maschine Experiment in einem komplexen militärischen Szenario in der Simulation evaluiert. [Uhrmann 2013] stellt hierbei fest, dass der auftragsbasierte Führungsansatz die Sicherheit der Mission und die Chance des Missionserfolgs erhöhe und der Operateur befähigt werde in seiner Doppelrolle im Helikoptercockpit zu bestehen.

[Gangl 2015] untersucht an Stelle der Kommandierung einzelner UAV die Gliederung in UAV-Teams, welche sich eigenständig koordinieren und gemeinsam Ziele verfolgen. Die Führung eines solchen UAV-Teams erfolgt dabei aus dem Cockpit eines einsitzigen Kampfflugzeugs. Zur Förderung der Nachvollziehbarkeit für den Operateur besitzen die kognitiven Agenten an Bord der UAV eine Erklärungskomponente, die Informationen über die Gründe ihrer Handlungen liefern [Lettl & Schulte 2013]. Erklärungen werden entweder durch den Operateur angefordert oder auf eigene Initiative der Agenten geliefert. [Gangl 2015] evaluiert in Mensch-Maschine Experimenten den teambasierten Führungsansatz aus dem bemannten Luftfahrzeugcockpit und kann dessen Validität verifizieren. Hierbei wird jedoch in bestimmten Missionsphasen eine deutliche Unterforderung der Piloten festgestellt, die auf den hohen Automationsgrad des Systems zurück zu führen sei.

Parallel dazu beschreibt [Kriegel 2011] das Konzept des *Knowledge Configured Vehicle* (KCV), bei dem das Wissen des Agenten in zwei modulare Schichten, das missionspezifische und das vehikelspezifische Wissen aufgeteilt ist. Beide Schichten sind über die KCV-Schicht miteinander verknüpft, welche den flexiblen Austausch von missions- bzw. vehikelspezifischem Wissen für spezifische Anwendungen erlaubt. Das vehikelspezifische Wissen erlaubt COSA die Kenntnis der Verfügbarkeit und den wissensbasierten Einsatz von Funktionen. Im Rahmen der Validierung des KCV-Konzeptes wurde erfolgreich der erste Realflug eines kognitiven Agenten vollzogen. Die Flugerprobung nach [Kriegel 2011] beschränkt sich auf die Betrachtung eines einzelnen UAV-Demonstrators ohne austauschbare Automationskomponenten und berücksichtigt weder eine Missionsbetrachtung noch die Untersuchung von Aspekten der Mensch-Maschine Interaktion.

Beiträge zur Weiterentwicklung der kognitiven Systemarchitektur COSA werden durch [Matzner 2010] im Bereich des inferenzbasierten Produktionssystems und durch [Brüggenwirth 2014] in Form einer kompletten Neu- und Umgestaltung der Architektur auf Grundlage des interpretierten Modells menschlichen Verhaltens nach Rasmussen geleistet (siehe hierzu Abschnitt 2.3.3). Das hierbei von [Brüggenwirth 2014] vorgestellte COSA² wird im Folgenden für die Entwicklung kognitiver Agenten am IFS genutzt.

[Pecher et al. 2010] setzen einen kognitiven Agenten zum Grundsystemmanagement eines *More Electric Aircraft* (MEA) in der Simulation ein. COSA und später auch ein erster Prototyp von COSA² werden dabei zur Implementierung einer wissensbasierten Überwachung und Kontrolle des Kraftstoff- und des Elektriksystems eines MEA eingesetzt, die gelöst von Missionsbetrachtungen oder menschlichem Eingriff agiert. In dieser ersten COSA² Anwendung werden Modelle der Einzelkomponenten (z.B. der Motoren und der Ventile) und ihrer Funktionen in CPL erstellt. Bei Fehlerzuständen werden anhand des implementierten, allgemeinen funktionalen Wissens und dem situativen Wissen über den konkreten Zustand der Subsysteme Lösungen generiert.

Im Rahmen des Projekts *Zertifizierbarer Autopilot* (ZertAP) verwenden [Wohler & Schulte 2014] COSA² zur Untersuchung eines kognitiven Agenten in der Rolle eines Assistenzsystems (*Mode 2 Cognitive Automation*) an Bord eines UAV im Hinblick auf die Zulassung und die Integration in den zivilen Luftraum. Hierbei werden Szenarien untersucht, bei denen der kognitive Agent eine funktionale Redundanz zum konventionellen Flugsystem einschließlich des Piloten am Boden darstellt. Das Assistenzsystem erkennt selbstständig Handlungsbedarf und unterstützt eskalierend, bis hin zur vollständigen Übernahme von Aufgaben und des Luftfahrzeugs. [Wohler et al. 2014] demonstrieren die Funktionalität des Assistenzsystems am Beispiel einer Kollisionsvermeidung mit anderen UAV im Realflug.

[Aurich 2011] untersucht schließlich im Rahmen des *Cooperative Cognitive Automation through Mathematically optimized Path-Following of UAS* (CoCampus) Projekts die Verknüpfung wissensbasierter UAV-Missionsplanung mit *Optimal Control* Flugpfadplanungs- und -verfolgungsalgorithmen unter Einsatz eines adaptiven L1-Reglers für die Flugregelung [Dobrokhodov et al. 2008]. Hierzu leitet COSA² Spezifikationen der Flugpfadplanung aus der Mission ab und verwendet die *Optimal Control* Komponenten zur Erreichung von Aufklärungszielen unter Berücksichtigung dynamischer Boden-Luft Bedrohungen. [Clauß et al. 2012] führen in diesem Zusammenhang eine erfolgreiche Demonstration der Implementierung im Rahmen einer Flugerprobung durch, die als Vorversuche der vorliegenden Arbeit fungieren.

2.4 Problemidentifikation

Der Einsatz unterschiedlich komplexer, konventioneller Automation (vgl. Abschnitt 2.1) ist für den Einsatz moderner UAV-Systeme unverzichtbar. Bestehende UAV-Führungsparadigmen beziehen sich auf die Interaktion des Menschen mit dieser Automation, welche von der Bedienung einfacher Schalter bis zur Konfiguration und Überwachung komplexer Eingabeoberflächen reicht. Das Feedback der Automation, das der Mensch in seiner Rolle als Operateur dabei überwacht, ist meist kontextlos in dem Sinne, dass es lediglich die jeweils formulierten Kommandos des Operateurs berücksichtigt, da die aktuelle Gesamtaufgabe des UAV bzw. die zugrundeliegende Absicht des Menschen nicht verarbeitet werden.

Zu Beginn dieser Arbeit wurde die Forderung nach der Führung eines hochautomatisierten UAV durch einen einzelnen Operateur formuliert. Das UAV-System soll dabei in der Lage sein, flexibel in komplexen, dynamischen Missionen zu operieren und dem Operateur eine geeignete Interaktion zur Führung zu ermöglichen. Wie in Abschnitt 2.1.5 beschrieben, müssen hierfür die Grenzen konventioneller Automation überschritten werden, indem zusätzlich kognitive Aufgaben zum Zweck der Unterstützung und Entlastung des Operateurs sowie zur Steigerung des Autonomiegrades des UAV-Systems auf die Seite der Automation verlagert werden. Zu diesem Zweck wird dem Operateur im Rahmen dieser Arbeit kognitive Automation in Form eines kognitiven Agenten (*Mode 1 Cognitive Automation*) an die Hand gegeben, der zur Invertierung der Führungsspanne, d.h. zur Reduzierung der Operateure auf einen pro UAV geeignet ist, indem Situationen mit zu hoher Belastung für den Menschen vermieden werden. [Uhrmann 2013] zeigte in diesem Zusammenhang bereits, dass ein kognitiver Agent für diese Aufgabe prinzipiell geeignet ist. Durch die Etablierung einer geeigneten Interaktion zwischen Mensch und Agent wird versucht die Wahrnehmung des Agenten durch den Operateur als auch die menschliche Leistung positiv zu beeinflussen. Der Einsatz kognitiver Automation in Form eines Assistenzsystems (*Mode 2 Cognitive Automation*) wird bereits an anderer Stelle z.B. durch [Wohler & Schulte 2014] untersucht und wird im Folgenden nicht weiter betrachtet.

Nach [Onken & Schulte 2010] führt die Integration eines kognitiven Agenten zu einer Steigerung der Gesamtsystemleistung und kann zudem zumindest eine teilweise Verringerung der Automationsprobleme durch *Brittleness* und *Literalism* bewirken. *Brittleness* wird durch das zielbasierte Verhalten des kognitiven Agenten entgegengewirkt, welches prinzipiell auch in zuvor undefinierten Zuständen intelligente Lösungen ermöglicht indem explizit formuliertes Wissen situationsspezifisch zur Aufgabenerfüllung eingesetzt wird. Ähnliches gilt für das Problem des *Literalism*, da kognitive Automation starr formulierte prozedurale Handlungsabfolgen durch dynamisch generierte Pläne ersetzt, die auf einer wissensbasierten Repräsentation der Lage und der Ziele basieren und somit aufgaben- und situationsspezifisch geeignete Lösungen bieten. Zusätzlich wird der Überbeanspruchung des Operateurs durch *Monitoring Requirements (Data Overload)* entgegengewirkt. Der kognitive Agent verarbeitet die Ausgaben der konventionellen Automation und ermöglicht somit eine abstrahierte Informationsdarstellung für den Operateur.

Das Einbringen einer zusätzlichen deliberativen Verhaltensebene steigert allerdings auch die Komplexität des automatisierten Systems (*Complexity*). Die Implementierung konkreter Aktionen entfernt sich weiter von der menschlichen Eingriffsebene. Dies kann prinzipiell das Problem von *Opacity* bedeuten, wenn für den Operateur auf Grund der Komplexität der Entscheidungsfindung

des Agenten nicht nachvollziehbar ist, wieso dieser auf eine bestimmte Weise handelt. Die kognitiven Fähigkeiten des Agenten erweitern die Möglichkeiten sowie die Neigung des Operateurs, vollständige Teilaufgaben an das automatisierte System zu übertragen (*Fully Autonomous Operations*). Hierbei ist zu klären, inwieweit der Operateur dabei die Autorität über die Handlungen des Systems behält, bzw. auf welche Art und auf Grundlage welcher Informationen ein Eingreifen (und ggf. Übersteuern des Agenten) möglich ist. In diesem Zusammenhang ist auch die korrekte Bewertung der Fähigkeiten des Agenten und des Systems relevant, damit die Delegation von Aufgaben angemessen erfolgt (*Reliance on Automation*).

Ansätze zur Interaktion zwischen menschlichen Operateuren und intelligenten Agenten bei der Delegation von Aufgaben fokussieren zumeist auf die technischen Eigenheiten von KI-Agenten (z.B. [Klein u.a., 2004] und [Milewski & Lewis 1997]) ohne Bezug auf die Aufgabenerfüllung des gesamten Mensch-Maschine Systems. Die Playbook Architektur (vgl. Abschnitt 2.3.3) wiederum generiert die Interaktion mit einem rein prozedural agierenden Agenten, ohne die Möglichkeiten kognitiver Automation auszuschöpfen. [Uhrmann & Schulte 2011] stellen mit dem Ansatz der auftragsbasierten Führung die flexible Delegation eines kognitiven Agenten vor, die zur Entlastung des Operateurs führt. Hierbei wird jedoch die Rückführung von Informationen von der Automation an den Operateur nicht hinreichend betrachtet. Um den oben genannten Problemen komplexer automatisierter Systeme zu begegnen und eine Entlastung des Operateurs durch den kognitiven Agenten zu ermöglichen, ist es nötig, auch das Feedback des Agenten in die Betrachtung einzubeziehen und die Mensch-Agenten Interaktion ganzheitlich zu betrachten.

Vor diesem Hintergrund stellt sich die Frage, inwiefern bestehende Interaktionsformen auf das Interaktionsverhalten eines kognitiven Agenten, insbesondere mit Blick auf die Rückführung von Informationen an den Operateur, angewendet werden können. Das Konzept der auftragsbasierten Führung unbemannter Luftfahrzeuge ist daraufhin zu untersuchen, inwiefern eine Erweiterung um Aspekte des Agentenfeedbacks ein gesamtheitliches Interaktionsverhalten zur Unterstützung des Operateurs ermöglicht. Die forschungsorientierte Fragestellung lässt sich daraus wie folgt ableiten:

Wie muss das Interaktionsverhalten eines kognitiven Agenten, insbesondere die Rückführung von Informationen an einen menschlichen Operateur, ausgeprägt sein, um dessen leitende Kontrolle zu unterstützen?

Das Verhalten eines kognitiven Agenten wird dabei vollständig durch das ihm zur Verfügung stehende Wissen bestimmt [Onken & Schulte 2010]. Das Interaktionsverhalten des Agenten ist damit, ebenso wie die Fähigkeit zur Umsetzung delegierter Aufgaben, abhängig von den Informationen, die der Agent bezüglich der ihm unterstellten konventionellen Automation besitzt und der Form, in welcher er diese verarbeitet. Ferner ist solches Wissen relevant, welches die Übersetzung zwischen Informationen der Delegationsebene (Eingriffsebene des Operateurs) und der Exekutionsebene (Eingriffsebene des Agenten auf die konventionelle Automation) erlaubt. Dies führt zu folgender weiterer Forschungsfragestellung:

Welches Wissen benötigt ein kognitiver Agent über die ihm unterstellten Automationssysteme aber auch über den menschlichen Operateur, um entsprechendes Interaktionsverhalten zu zeigen und Aufgaben flexibel umsetzen zu können?

Zur Beantwortung der oben formulierten Forschungsfragestellungen wird im Folgenden zunächst ein Interaktionskonzept von Mensch und kognitivem Agent hergeleitet, welches anschließend in der Implementierung in Form des kognitiven Agenten und dessen Wissenskomponenten umgesetzt wird. Die Evaluierung des Konzepts hinsichtlich der geforderten Mensch-Maschine Aspekte erfordert anschließend die Durchführung eines Mensch-Maschine Experiments, welches formulierte Hypothesen gezielt untersucht und im Hinblick auf die Forschungsfragestellungen diskutiert (vgl. Abschnitt 5.1). Im Rahmen dieser Arbeit soll das Mensch-Maschine Experiment in einer Simulationsumgebung durchgeführt werden, um eine möglichst gute Kontrollierbarkeit und Wiederholbarkeit der Rahmenbedingungen sowie die Beobachtbarkeit der Probanden zu gewährleisten.

Zur Demonstration der Entwicklungsreife der Implementierung und der Robustheit des Agentensystems gegenüber äußeren Einflüssen eignet sich zusätzlich die Durchführung von Versuchen im Realflug. Ziel ist es hierbei, die Austauschbarkeit der unterstellten Automationskomponenten des kognitiven Agenten und deren Auswirkungen auf die Handlungsoptionen und das Verhalten des Agenten zu demonstrieren. Hierzu sollen zwei unterschiedliche UAV-Plattformen mit spezifischen Automationskomponenten durch einen auf derselben Implementierung beruhenden Agenten geleitet werden (vgl. Abschnitt 5.2). Durch variable und dynamisch veränderliche Missionsszenarien in der Simulation als auch im Realflug kann hierbei die Flexibilität des Agentenverhaltens im Rahmen der Fähigkeiten der jeweiligen UAV-Plattform demonstriert werden.

3 Konzept

Abschnitt 3.1 stellt das bestehende Konzept der auftragsbasierten Führung von UAV zunächst kurz vor und zeigt dessen Beschränkungen hinsichtlich der Beschreibung der Mensch-Agenten Interaktion auf. In Abschnitt 3.2 folgt dann der konzeptionelle Überblick über die Beantwortung der forschungsorientierten Fragestellung aus Abschnitt 2.4. Dieses Konzept erfährt im weiteren Verlauf des Abschnittes dahingehend eine Vertiefung, als dass die Abstraktion und Verarbeitung von Automationsfunktionen als Systemfähigkeiten erläutert werden. Anschließend wird die Form der Auftragserteilung durch den Operateur und das dreischichtige Verarbeitungskonzept des Agenten beschrieben sowie eine auftrags- und situationsgerechte Etikette für die Mensch-Agenten Interaktion vorgestellt. Der Abschnitt schließt mit der Vorstellung einer Möglichkeit zum koordinierten Durchgriff auf Automationsfunktionen an Bord des UAV durch den Operateur. Abschnitt 3.3 betrachtet schließlich die Umsetzung des zuvor entwickelten Konzepts über die Erweiterung einer bestehenden kognitiven Systemarchitektur zu einer mehrschichtigen Agentenarchitektur.

3.1 *Auftragsbasierte Führung von UAV*

Um den Kommandanten eines bemannten Militärhelikopters zu befähigen, neben seinen Aufgaben an Bord mehrere UAV als abgesetzte Sensorplattformen zur Aufklärung im Einsatzgebiet zu führen, präsentieren [Uhrmann & Schulte 2011] das Konzept der auftragsbasierten Führung von UAV. In Anlehnung an die auftragsbasierte Führung in militärischen Strukturen übermittelt der Kommandant hierbei seine Absicht über Zielvorgaben und Randbedingungen an Agenten an Bord der UAV. Er eröffnet ihnen somit Handlungsrahmen, innerhalb derer er eigenständiges Handeln autorisiert. Hierbei werden Listen von Einzelaufträgen (als sog. Auftragsagenda) an die UAV übermittelt, deren Detaillierungsgrad die Autorität der Agenten bei der Ableitung eigener Aufgaben bestimmt. Den Grad dieser Autorität wird durch [Uhrmann 2013] als Stringenz bezeichnet und soll u.a. einen Ausdruck des Vertrauens des Kommandanten in die Fähigkeiten der nachgeordneten Agenten darstellen. Die Auftragsverarbeitung durch die Agenten erfolgt zielorientiert durch die logische Komplettierung der Auftragsagenda des Operateurs und nutzt Agentenwissen im Bereich der jeweils aktuellen Situation sowie das Verständnis von Einsatzgrundlagen und Zielsetzungen.

[Uhrmann 2013] nutzt die kognitive Systemarchitektur COSA zur Implementierung der kognitiven Agenten sowie zur Modellierung des Agentenwissens. Anhand eines komplexen, militärischen Szenarios kann gezeigt werden, dass die Implementierung des auftragsbasierten Führungsansatzes die Beanspruchung des Kommandanten auf einem Niveau halten kann, das es ihm erlaubt, trotz der Doppelbelastung, gleichzeitig drei UAV aus einem bemannten Helikoptercockpit taktisch sinnvoll einzusetzen.

Hinsichtlich der Mensch-Agenten Interaktion beschränkt sich das Konzept nach [Uhrmann 2013] weitestgehend auf die Betrachtung von Agenten als Auftragsempfänger. Es fehlt die Berücksichtigung des Agenten als nachgeordnete Führungsebene, dessen Verhalten und dessen Agentenfeedback den Menschen und seinen Delegationsprozess direkt beeinflussen. Die Rückführung von Informationen durch den Agenten und deren Auswirkungen auf die Wahrnehmung des Agenten durch den Menschen werden nicht berücksichtigt, sodass eine

dynamische Beschreibung der Mensch-Agenten Beziehung, z.B. hinsichtlich einer missionsvariablen Stringenz in der Auftragserteilung, nicht möglich ist.

Im Folgenden soll daher das bestehende Konzept der auftragsbasierten Führung zu einem Konzept der bidirektionalen Mensch-Agenten Interaktion weiterentwickelt werden, das auf der Kommandierung eines kognitiven Agenten im Rahmen der ASC aufbaut.

3.2 Agent Supervisory Control

Durch den Einsatz eines kognitiven Agenten soll ein menschlicher Operateur bei der leitenden Kontrolle eines unbemannten Luftfahrzeugs unterstützt werden. Hierzu wird aufbauend auf dem bestehenden Konzept der auftragsbasierten Führung nach [Uhrmann 2013] ein ganzheitliches Interaktionsverhalten des Agenten abgeleitet, dass die Rückführung von Informationen an den Operateur in Form eines Agentenfeedbacks berücksichtigt.

Ausgehend von den grundlegenden Betrachtungen zur Delegation eines intelligenten Agenten gemäß Abschnitt 2.3.1, wird der Agent als eine dem Menschen nachgeordnete Führungsebene an Bord des UAV betrachtet. Der Agent setzt die Vorgaben des Menschen, vermittelt über diskrete Konfiguration und Koordinierung der Automationsfunktionen, um und überwacht deren Ausgaben (quasi) kontinuierlich. In Anlehnung an den Begriff der *Human Supervisory Control* (HSC) nach Sheridan, lässt sich diese Art der Führung, bei der ein künstlich kognitiver Agent die leitende Kontrolle über ein automatisiertes, technisches System innehat, als *Agent Supervisory Control* (ASC) beschreiben [Clauß & Schulte 2014a]. Die Handlungen des Agenten untergliedern sich dabei gemäß der Leitungsfunktionen (*Supervisory Functions*) nach [Sheridan 1992].

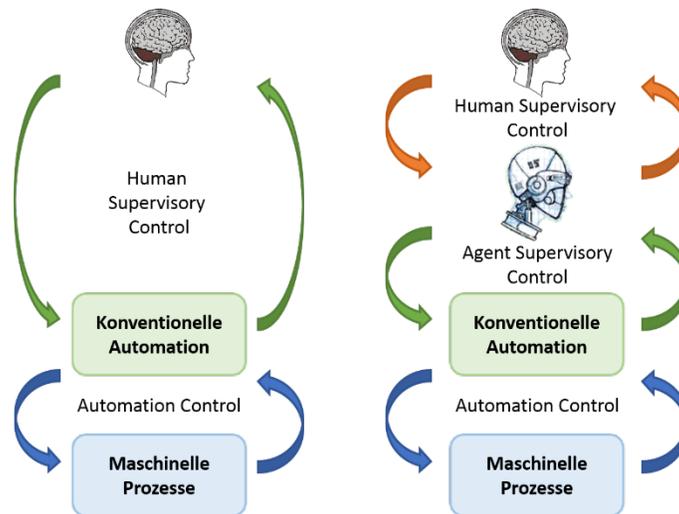


Abbildung 3-1: Agent Supervisory Control (ASC) als zusätzliche Führungsschleife durch Einbringen eines kognitiven Agenten zwischen Mensch und konventioneller Automation [Clauß & Schulte 2014b]

Der Operateur führt das UAV weiterhin im Sinne der leitenden Kontrolle, konzentriert sich jedoch auf die Ebene der Missionsführung, wobei er die Führung des Luftfahrzeugs (wie z.B. die Routenplanung und Flugführung) und das Management der Subsysteme (z.B. den Sensoreinsatz und die Grundsystemüberwachung) an den kognitiven Agenten delegiert. Der Agent fungiert dabei als Abstraktionsebene zwischen dem automatisierten UAV-System und der menschlichen Leitungsebene, welche nun über eine nachgeordnete Führungsebene, den Agenten, das

Gesamtsystem leitet (vgl. Abbildung 3-1). Das Agentenfeedback erfolgt in Form einer *Etikette* (vgl. Abschnitt 2.3.1.1), die den Menschen weiterhin entlasten und darüber hinaus auch eine *kalibrierte Delegationsentscheidung* (vgl. Abschnitt 2.3.1.2) ermöglichen soll.

„[The agent] takes the role of an Agent Supervisor or an artificial pilot, performing such functions that were formerly solely left to the human supervisor.“ [Clauß & Schulte 2014a]

Als Leiter des automatisierten Subsystems gliedert sich das Handeln des kognitiven Agenten im Sinne der ASC in die *Supervisory Functions*. Bei der Betrachtung der auftragsbasierten Führung des Agenten durch den Menschen ist es möglich, die *plan* Funktion in drei konsekutive Subfunktionen zu untergliedern. Zunächst wertet der Agent eingehende Aufträge hinsichtlich *wesentlicher Lageänderungen* aus und überführt sie in ein für die anschließende Planung nutzbares Format (*Evaluate*). Anschließend komplettiert der Agent die im Allgemeinen unvollständig formulierten Aufträge anhand impliziter Zwischenschritte (*Supplement*) und leitet schließlich Einzelaufgaben für die nachgeordneten Automationskomponenten aus den Aufträgen ab (*Decompose*). Die mehrschichtige Auftragsverarbeitung wird im Hinblick auf begrenzt verfügbare Verarbeitungsressourcen an Bord des UAV bei der Konzeptumsetzung durch eine *Layered Agent Architecture* berücksichtigt (vgl. Abschnitt 3.3).

Der Agent abstrahiert dabei die Automationsfunktionen an Bord des UAV in Form *symbolischer Fähigkeiten*, deren dynamische Verfügbarkeit in Abhängigkeit der Systemressourcen und des Flugzustands bestimmt wird. Das *Fähigkeitsbewusstsein* entkoppelt die wissensbasierte Verarbeitung des Agenten von der technischen Umsetzung der nachgeordneten Automationskomponenten und ermöglicht die Anbindung variabler Systemkonfigurationen. Mittels hierarchischer Abstraktion werden UAV-Fähigkeiten für den Einsatz des Luftfahrzeugs im Missionsrahmen formuliert, die als Teil des Agentenfeedbacks an den Operateur zurückgeführt werden. Die zeitlich begrenzte, direkte Führung von Automationsfunktionen durch den Menschen zur manuellen Umsetzung von Einzelaufgaben berücksichtigt der Agent als *Substitution* und *Addition* von UAV-Fähigkeiten.

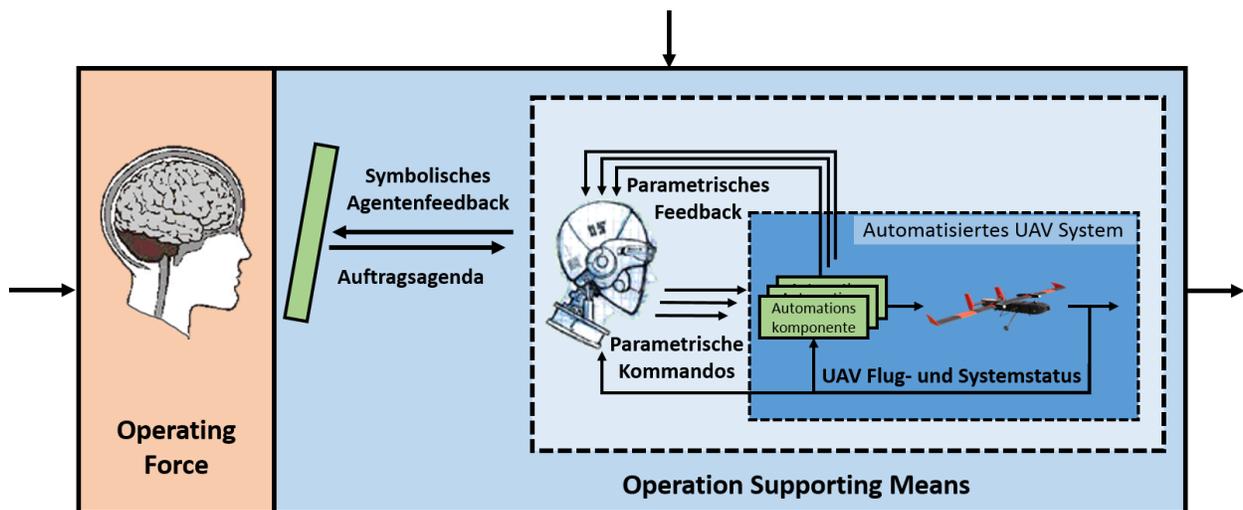


Abbildung 3-2: Hierarchische Abstraktion der Systeminteraktion durch Integration eines auftragsbasierten kognitiven Agenten unter Rückführung symbolischen Agentenfeedbacks

Die Mensch-Agenten Interaktion gliedert sich dabei in *ereignisabhängige* und *–unabhängige Informationen*, die einheitlich als Symbole auf Ebene der Missionsführung kommuniziert werden. Regelmäßig werden Leistungsinformationen durch den Agenten an den Operateur zurückgeführt und Umwelt- und Lageinformationen synchronisiert. Gesonderte Kommunikation initiiert der Agent in Fällen, in denen Ereignisse und Lageänderungen zu Vorgabekonflikten im Agenten führen, die die Autorität des Agenten übersteigen und eine Intervention des Menschen verlangen. Der Agent gibt dazu zum Zweck der Nachvollziehbarkeit und Unterstützung *auftragsbezogen* Auskunft über Ursache und Wirkung des jeweiligen Konflikts. Subsymbolische Informationen sowie Ereignisse ohne Einfluss auf die Auftragserfüllung werden im Sinne einer spezialisierten *Etikette* zur Vermeidung der Desensibilisierung für missionsrelevante Ereignisse nicht an den Operateur zurückgeführt. Abbildung 3-2 zeigt die hierarchische Abstraktion der Systeminteraktion ausgehend vom automatisierten UAV System, über den kognitiven Agenten zum Operateur.

3.2.1 Fähigkeitsbewusstsein des Agenten

Der kognitive Agent handelt durch Konfiguration, Kommandierung und Überwachung von Automation an Bord des UAV. Zur Planung und Bewertung des Einsatzes von Automationsfunktionen nutzt der Agent eine einheitlich kodierte Wissensbasis, die über die parametrische Beschreibung ihrer Ein- und Ausgabeschnittstellen bzw. über Leistungsparameter hinausgeht. Er nutzt dabei symbolisches Wissen bezüglich der Voraussetzungen des Fähigkeitseinsatzes, des Ressourcenbedarfs sowie des erzielbaren Effekts und ist somit geeignet Abhängigkeiten zwischen den Automationsfunktionen zu berücksichtigen.

Abbildung 3-3 zeigt die im Rahmen dieser Arbeit berücksichtigten Automationskomponenten an Bord des UAV und verdeutlicht ihre zentrale Anbindung an den kognitiven Agenten. Der Agent nutzt die Routenplanung zur Generierung kostenoptimierter Flugrouten, die über das Flugführungssystem in Steuerkommandos an das Luftfahrzeug überführt werden. Die Aktivierung und Deaktivierung von Flugrouten sowie der direkte Anflug von Zielkoordinaten erfolgt durch Kommandierung der Flugführung. Funktionen zum Sensoreinsatz nutzt der Agent über die Sensorik, deren Daten über die Sensordatenverarbeitung ausgewertet und dem Agenten bereitgestellt werden. Hierbei kann der Agent die Sensordatenverarbeitung aufgabengerecht parametrisieren und Perzeptionsergebnisse gezielt anfordern. Das Datenlinkmanagement bietet dem Agenten neben Funktionen zur Datenübertragung die Möglichkeit zur Konfiguration der Kommunikationsschnittstelle sowie zur Prädiktion der Datenlinkverfügbarkeit über den Missionsverlauf. Der bodenseitige Datenlink sowie die Mensch-Maschine Schnittstelle sind im Schaubild nicht dargestellt.

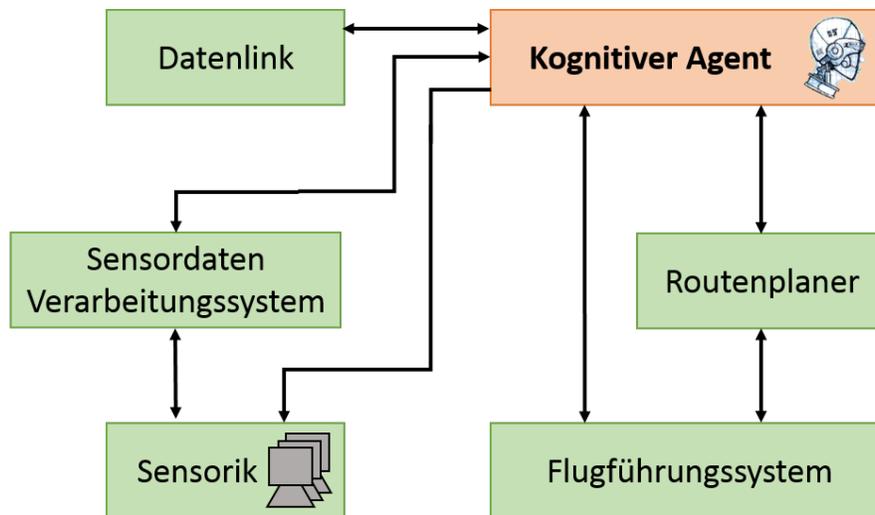


Abbildung 3-3: Systemschaubild der Interaktion des kognitiven Agenten mit den wesentlichen Automationskomponenten an Bord des UAV

Der Agent abstrahiert aus den Automationsfunktionen symbolisch kodierte *Fähigkeiten* zur Umsetzung spezifischer Aufgaben, die bei Bedarf abgerufen werden können. Eine verfügbare Fähigkeit kann im Sinne dieser Arbeit durch den Agenten gewählt und angewendet werden, um unabhängig von der zugrunde liegenden Implementierung der Funktion, Handlungen mit dem UAV zu vollziehen. Hierzu umfassen Fähigkeiten Informationen über die Aufgabe, die durch sie umgesetzt wird, die Vorbedingungen für ihre Nutzung und die benötigten Systemressourcen. Die Menge der Fähigkeiten und ihre Verfügbarkeit ist als Wissen im Agenten hinterlegt und bildet dessen *Fähigkeitsbewusstsein*.

„Fähigkeit: durch bestimmte Anlagen, Eigenschaften geschaffene Möglichkeit, gewisse Funktionen zu erfüllen, gewissen Anforderungen zu genügen, etwas zu leisten.“

[Dudenverlag 2014]

Im Rahmen der Auftragsverarbeitung nutzt der Agent sein Fähigkeitsbewusstsein in Verbindung mit Missionswissen, um Lösungen in Form logischer Verknüpfungen einzelner Fähigkeiten zu generieren. Der Umfang des Agentenwissens bestimmt dabei den Detaillierungsgrad mit dem einzelne Fähigkeiten abgeleitet werden können und die Spezifität, mit der der Agent sie in der Planung berücksichtigt. Mit steigendem Wissensumfang ist eine feiner granulare Definition einzelner Fähigkeiten und ihrer Eigenschaften für spezifische Aufgaben möglich, wodurch eine höhere Flexibilität der Auftragsverarbeitung erzielt wird. Abbildung 3-4 stellt hierzu beispielhaft unterschiedliche Detaillierungsgrade von Fähigkeiten als symbolische Repräsentation von Automationsfunktionen dar.

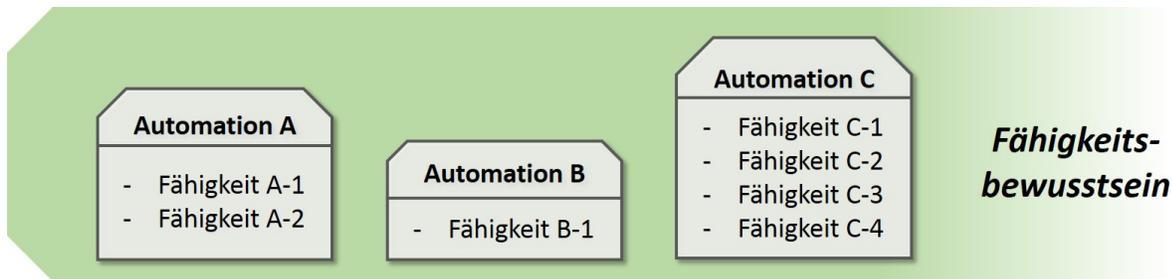


Abbildung 3-4: Die Spezifität einzelner Fähigkeiten im Fähigkeitsbewusstsein bestimmt die Flexibilität der Aufgabenplanung und beruht auf hinterlegtem Wissen zu Automationsfunktionen und -eigenschaften

Die Verfügbarkeit von Fähigkeiten ist dabei nicht starr, sondern dynamisch veränderlich in Abhängigkeit vom Zustand des Systems und der Umgebung. Zur Prädiktion ihrer Verfügbarkeit nutzt der Agent während der Planung das Fähigkeitsbewusstsein, um in jedem Planungsschritt die Vorbedingungen der Fähigkeiten mit den Systemressourcen und Systemzuständen abzugleichen. Während der Plandurchführung bestimmt der Agent die Verfügbarkeit der Fähigkeiten über das parametrische Feedback der Automationsfunktionen. In gleicher Weise nutzt der Agent sein Wissen, um Handlungsfolgen zur Aktivierung auftragsrelevanter Fähigkeiten abzuleiten. Ist z.B. die Fähigkeit zur Datenübertragung aktuell nicht verfügbar, kann der Agent anhand von Wissen bezüglich der Strahlungscharakteristik des Datenlinks ableiten, an welchen Ort das Luftfahrzeug überführt werden muss, um eine Datenübertragung zu ermöglichen.

Die Abstrahierung der Automationsfunktionen in Fähigkeiten beschränkt sich nicht auf die Ebene des kognitiven Agenten. Es erfolgt eine weitere hierarchische Abstraktion in UAV-Fähigkeiten, deren Verfügbarkeit an den Operateur zurückgeführt wird und die als Grundlage für den Einsatz des Luftfahrzeugs in der Missionsplanung dienen. Diese hierarchische Abstraktion von Fähigkeiten schafft Abstraktionsräume (*abstraction spaces*), die nach [Sacerdoti 1974] helfen können die Komplexität der Wissensbasis für die Führungsebene der Operateurs auf ein überschaubares Maß reduzieren.

3.2.2 Auftragserteilung und Auftragsverarbeitung

Das Konzept der auftragsbasierten Führung sieht die Delegation von Aufträgen in Form einer Auftragsagenda vor. Im Folgenden werden die Form der Auftragserteilung und anschließend die Strukturierung der Auftragsverarbeitung durch den Agenten spezifiziert.

Auftragserteilung

Der Operateur formuliert Aufträge als Handlungsanweisungen mit Bezug auf ein Lageelement. Die Handlungsanweisungen umfassen dabei Elemente der Flugführung (z.B. *FlyTo*, *Cross Corridor* und *Land*) sowie der Aufklärung (z.B. *Recon Target*, *Scan for Vehicles*), und setzen jeweils die Verfügbarkeit bestimmter UAV-Fähigkeiten voraus. So muss das UAV z.B. für den Auftrag *Recon Target* über die Fähigkeiten zur Aufklärung und zur Datenübertragung bzw. Datenaufzeichnung verfügen. Nicht alle Handlungsanweisungen können dabei logisch mit allen Lageelementen verknüpft werden. Dies ergibt sich implizit aus der jeweiligen Definition der Handlungsanweisung, falls diese auf eine bestimmte Menge von Elementtypen begrenzt ist. Ein Auftrag zur Landung auf einer feindlichen Luftverteidigungsstellung (SAM) wäre in diesem Sinne z.B. nicht möglich.

Die Auftragserteilung in Form einer Auftragsagenda geschieht als Liste von Einzelaufträgen mit fester Reihenfolge, die für den Agenten bindend ist und dem Operateur ihre zeitliche Priorisierung ermöglicht [Theißing et al. 2012][Theißing & Schulte 2013]. Der Operateur nutzt Randbedingungen, um die Handlungsoptionen des Agenten bei der Auftragsumsetzung einzuschränken oder vorzugeben. Über die Menge der Randbedingungen kann der Operateur den Aufwand der Delegation entsprechend seinem Vertrauen in den kognitiven Agenten und seiner Beanspruchung variieren. Randbedingungen können im Rahmen dieser Arbeit grundsätzlich in drei verschiedene Arten untergliedert werden:

- Randbedingungen, die gewünschtes Verhalten spezifizieren,
- solche, die unerwünschtes Verhalten verbieten und
- solche, die die manuelle Kontrolle von Automation durch den Menschen definieren.

Gewünschtes Verhalten wird über zusätzliche Einzelaufträge spezifiziert, die vom Agenten als verbindliche Zwischenschritte für die Zielumsetzung verarbeitet werden. Bei der Landung kann dies z.B. durch die Vorgabe einer bestimmten Landebahn geschehen. Die Spezifikation gewünschten Verhaltens dient dazu, eine variable Stringenz nach [Uhrmann 2013] herzustellen.

Unerwünschtes Verhalten wird über die Einschränkung der Handlungsoptionen des Agenten für spezifische Lageelemente ausgedrückt. Für das obige Beispiel bedeutet dies das Verbot zur Nutzung einer spezifischen Landebahn bei der Landung. Sollten aus den Einschränkungen durch den Menschen Vorgabekonflikte auf Seiten des Agenten resultieren, erfolgt eine Fehlermeldung an den Operateur als Teil des Agentenfeedbacks (vgl. Abschnitt 3.2.3).

Die Spezifikation der manuellen Kontrolle von Automation durch den Menschen erfolgt jeweils für Einzelaufträge durch Substitution oder Addition von UAV-Fähigkeiten. Hierbei beeinflusst der Operateur den Agenten über dessen Fähigkeitsbewusstsein und zwingt ihn somit zur Auswahl manuell gesteuerter Handlungsoptionen in der Planung. Der Agent handelt dabei weiterhin auftragsbasiert, überträgt dem Operateur während der Ausführung jedoch temporär die manuelle Kontrolle über einzelne Automationsfunktionen (vgl. Abschnitt 3.2.4).

Auftragsverarbeitung

Die Handlungen des Agenten als Leiter des automatisierten UAV-Systems gliedern sich im Rahmen der ASC entsprechend der Supervisory Functions nach [Sheridan 1992]. Da der Agent im Gegensatz zum menschlichen Operateur weder die Fähigkeit noch die Autorität zur Formulierung eigener Absichten und Ziele besitzt, reduziert sich dessen *plan* Schritt auf die Verarbeitung der Auftragsagenda des Operateurs. Hierbei ist eine möglichst hohe Flexibilität erstrebenswert, um komplexe Zusammenhänge spezifisch berücksichtigen zu können [Laux et al. 2012]. Hohe Flexibilität setzt auf Seiten des Agenten einen großen Umfang an a-priori Wissen voraus. Da die wissensbasierte Planung als Suchproblem NP-äquivalent ist (d.h. nicht-deterministisch in Polynominalzeit zu lösen [Russell & Norvig 2012]), würde der Zeit- und Speicheraufwand mit steigender Anzahl der zu durchsuchenden Pfade und Objekte jedoch schnell in makroskopische Dimensionen wachsen [Wegener 2003]. Zur Berücksichtigung der endlichen Verfügbarkeit von Rechenressourcen an Bord des UAV, soll die Planungskomplexität begrenzt werden, ohne den Agenten in der Auftragsumsetzung einzuschränken.

Fall, verwirft der Agent die bestehende Handlungsagenda, beendet ihre Ausführung und stößt die Neuplanung an. Während der Neuplanung hält der Agent dabei das UAV in einem Warteflug um die aktuelle Position herum. Werden keine wesentlichen Differenzen festgestellt, führt der Agent die bestehende Handlungsagenda weiter aus und fährt nach erfolgreichem Abschluss mit der Verarbeitung der neuen Zielvorgaben fort. Dieses Vorgehen ermöglicht dem Operateur die Anpassung zukünftiger Auftrags Elemente, ohne die Auftragsverarbeitung durch den Agenten zu unterbrechen.

3.2.2.2 Supplement

Der Agent ergänzt die Auftragsagenda des Agenten zu einer logisch konsistenten und ressourcenoptimierten Kette von Handlungen des UAV. Die Ergänzung erfolgt als symbolische Planung und nutzt die gleiche Form von Handlungsanweisungen, wie sie auch für die Auftragsformulierung des Operateurs genutzt werden. Zu diesem Zweck bewertet der Agent die verfügbaren Fähigkeiten hinsichtlich der *Mission Constraints* und leitet daraus Handlungsoptionen für Einzelschritte ab. Das Ergebnis der Planung ist eine komplettierte Handlungsfolge, mit der das UAV aus seinem aktuellen Zustand heraus die einzelnen Zielvorgaben umsetzt.

Nach erfolgreicher Planung führt der Agent die komplettierte Handlungsfolge zurück an den Operateur, um ihn über die eigene Absicht zu informieren. Kann der Agent innerhalb der gegebenen Randbedingungen keine Handlungsfolge zur Umsetzung der Zielvorgaben ableiten, wird stattdessen eine Fehlermeldung an den Operateur zurückgeführt, in der die Ursache der Fehlplanung benannt wird. Ursachen können hierbei mangelnde Systemfähigkeiten und -ressourcen sowie die eingeschränkte Erreichbarkeit bzw. Nutzbarkeit von Lageelementen darstellen. Die ursachenbezogene Fehlermeldung erleichtert dem Operateur die Identifizierung und Umsetzung von Anpassungen bei der Auftragsformulierung.

3.2.2.3 Decompose

In einem abschließenden Planungsschritt überführt der Agent die komplettierte Handlungsfolge in Aktionsfolgen, die direkt über die nachgeordneten Automationskomponenten umgesetzt werden können. Dazu werden den Handlungsschritten regelbasiert Prozeduren zugeordnet, die als feste Abfolge von Einzelaktionen im Agenten hinterlegt sind. Zu jeder Handlung bestimmt der Agent zunächst die Menge der möglichen Prozeduren anhand des Handlungstyps und der verfügbaren Fähigkeiten. Über eine starre Priorisierungsmetrik bestimmt der Agent aus dieser Auswahl die jeweils optimale Prozedur. Die Parametrisierung der einzelnen Aktionen erfolgt subsymbolisch entsprechend der situationsspezifischen Eigenschaften und Geometrien der betroffenen Lageelemente sowie der Lage und des Zustands des UAV.

Nach Abschluss des *Decompose* Schrittes verfügt der Agent über eine Liste aus Einzelaktionen, die sequenziell und/oder parallel auszuführen sind, um die aktuelle Zielvorgabe umzusetzen. Eine Rückführung der Aktionsliste an den Operateur erfolgt im Sinne der Vermeidung einer vollständigen Transparenz der Agentenebene nicht. Im Weiteren dient die Aktionsliste dem Agenten als Grundlage zur Parametrisierung von Kommandos an nachgeordnete Automationskomponenten.

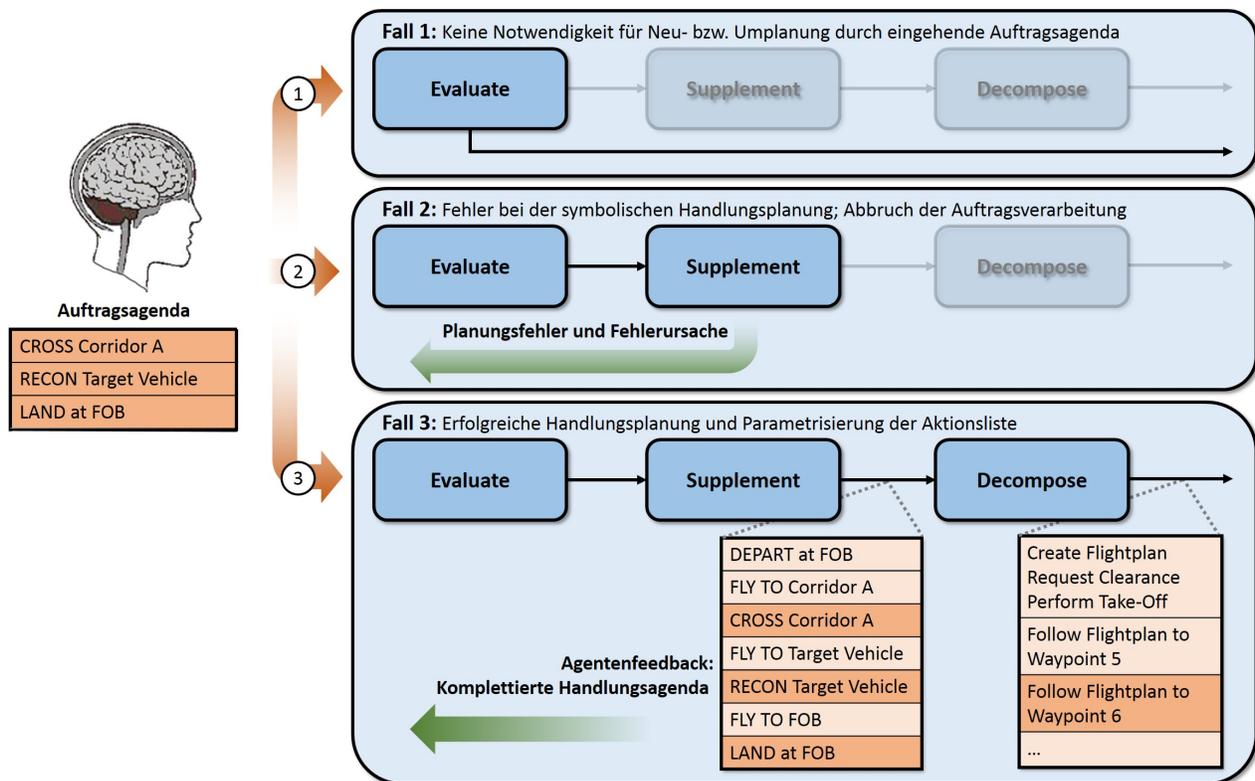


Abbildung 3-6: Drei Anwendungsfälle der Auftragsverarbeitung des Agenten mit der Rückführung von Feedbackinformationen anhand einer beispielhaften Auftragsagenda

Zusammenfassend lassen sich für die Verarbeitung einer Auftragsagenda des Operateurs somit drei verschiedene Fälle identifizieren. Im ersten Fall identifiziert der Agent bei Auftragseingang keine Notwendigkeit zur symbolischen Planung und führt die bisherige Aktionsliste weiter aus. Die Schritte *Supplement* und *Decompose* werden dabei übersprungen. Im zweiten Fall führt die symbolische Handlungsplanung des Agenten zu keinem Ergebnis und wird als Fehlplanung abgebrochen. Der Agent beendet daraufhin die Auftragsverarbeitung und führt eine Fehlermeldung inklusive der Fehlerursache an den Operateur zurück. Im letzten Fall ist die Handlungsplanung erfolgreich und der Agent kann eine detaillierte Aktionsliste ableiten und parametrisieren. Für die Nachvollziehbarkeit des Agentenhandelns wird die komplettierte Handlungsagenda als Teil des Agentenfeedbacks an den Menschen übermittelt. Die Aktionsliste nutzt der Agent anschließend intern zur Kommandierung und Überwachung nachgeordneter Automation. Abbildung 3-6 stellt die drei Verarbeitungsfälle für eine beispielhafte Auftragsagenda grafisch dar.

3.2.3 Mensch-Agenten Interaktion

Die Mensch-Agenten Interaktion erfolgt, entsprechend der Planungsebene des Operateurs, einheitlich auf der Ebene der Missionsführung, um die Beanspruchung des Menschen bei der Verarbeitung von Informationen des Agenten zu reduzieren. Der Informationsaustausch umfasst die Aufträge des Operateurs sowie die Rückführung des Agentenfeedbacks, komplettiert durch den Austausch gemeinsamer Lageinformationen zwischen Mensch und Agent (vgl. Abschnitt 2.3.1.1). Die Interaktion zur Koordinierung der manuellen Kontrolle von Systemkomponenten durch den Operateur im Rahmen der auftragsbasierten Führung wird gesondert im folgenden Abschnitt

betrachtet. Das Agentenfeedback lässt sich in ereignisabhängige und ereignisunabhängige Informationen gliedern, deren Form und Häufigkeit einer spezialisierten Etikette entsprechen.

Die regelmäßig zurückgeführten Leistungsinformationen des Agenten beschreiben den Zustand des automatisierten UAV über die Verfügbarkeit von UAV-Fähigkeiten und abstrahierte Fluglagedaten zur Kartendarstellung für den Operateur. Außerdem beinhalten sie den Agentenstatus einschließlich des Erfüllungsgrads der Auftragsumsetzung. Hierbei wird zunächst der operationelle Status des Agenten (u.a. *Einsatzbereit, Planung*) zurückgeführt. Während der Auftragsumsetzung wird dieser durch den Fortschritt der Handlungsagenda des Agenten über den Erfüllungsgrad der Einzelschritte ersetzt. Die Synchronisation von Umwelt- und Lageinformationen erfolgt ebenfalls ereignisunabhängig im Hintergrund.

Bei auftragsrelevanten Systemvorkommnissen und Lageänderungen sowie bei Meilensteinen der Auftragsumsetzung initiiert der Agent gesonderte Kommunikation in Form auftrags- und ereignisbezogener Meldungen an den Operateur. Durch Vermeidung von Meldungen, die nicht die Führungsebene des Menschen betreffen (sog. *Unintended Alerts* [Lees & Lee 2007]), soll dabei die Belastung des Menschen reduziert und sein Vertrauen in den Agenten erhöht werden. Die Entscheidung bezüglich der Relevanz von Meldungen trifft der Agent auf Grundlage der resultierenden Zielabweichung im projizierten Handlungsverlauf. Zielabweichungen können dabei prinzipiell eine von drei eskalierenden Stufen annehmen:

1. Eine Zielabweichung, die durch Anpassung der subsymbolischen Parametrisierung innerhalb einer existierenden Handlungskette behoben werden kann. Hier z.B. die Umplanung der Flugroute des UAV, um innerhalb einer vorgegebenen Zeit ein Hindernis umfliegen zu können.
2. Eine Zielabweichung, durch die die bestehende Handlungskette invalide wird und die eine symbolische Umplanung erforderlich macht. Wird z.B. ein prädestinierter Flugkorridor durch den Operateur gesperrt, muss der Agent eine neue Handlungskette mit einem Ausweichkorridor bestimmen.
3. Eine Zielabweichung, durch die eine symbolische Umplanung der Handlungskette unter Berücksichtigung der Vorgaben des Operateurs nicht möglich ist. Existiert im obigen Beispiel kein Alternativkorridor, ist es dem Agenten unmöglich, einen den Vorgaben entsprechenden Transitionsflug durchzuführen.

Fall 1. und 2. führen zur Intervention (*Intervene*) durch den kognitiven Agenten. In Fall 3. ist eine selbstständige Fehlerkorrektur durch den Agenten unmöglich. Prinzipiell sind für die Lösung des Vorgabenkonflikts unterschiedliche Ansätze denkbar. [Myers & Morley 2001] beschreiben eine Priorisierung von Vorgaben entsprechend einer internen Bewertungs- und Nutzenfunktion, sodass Vorgaben niedriger Priorität überstimmt werden können. Dieser Ansatz birgt jedoch die Gefahr unerwünschter bzw. unerwarteter Handlungen für den Operateur und wird als mögliche Ursache für *Opacity* im Folgenden nicht betrachtet. Im Rahmen dieser Arbeit wird bei Zielabweichungen nach Fall 3. daher der vorgesetzte Operateur involviert, der durch eine Anpassung der Auftragsagenda oder ihrer Randbedingungen den Agenten neu kommandieren kann. Hierfür führt der Agent eine Fehlermeldung unter Angabe der Ursache für die Zielabweichung an den Operateur zurück (vgl. Abschnitt 3.2.2.2). Die Involvierung des Operateurs bindet zwar dessen Ressourcen, soll aber im Sinne einer *Etikette* zu einer Verbesserung der Nachvollziehbarkeit des Verhaltens des Agenten und damit zu einer Kalibrierung des *Operateurvertrauens in den Agenten* führen.

Auf gleiche Weise können Freigaben für spezifische Handlungen des Agenten durch den Operateur realisiert werden. Vor Beginn der Handlung initiiert der Agent hierzu die Kommunikation mit der Bitte um Freigabe. Bei einer Bestätigung durch den Operateur kann der Agent gemäß Plan fortfahren. Verweigert der Operateur die Freigabe, bricht der Agent die Auftragsumsetzung ab und initiiert eine Umplanung unter Vermeidung der spezifischen Handlungsoption.

3.2.4 Fähigkeitssubstitution und Fähigkeitsaddition

Das auftrags- und situationsspezifische Agentenfeedback dient dem Operateur während des Verlaufs der Mensch-Agenten Interaktion zur Ausprägung eines kalibrierten Vertrauens in die Fähigkeiten des Agenten (vgl. Abschnitt 2.3.1.2). Auf dieser Basis entscheidet der Mensch, welche Aufgaben er an den Agenten delegiert und welche manuell vollzogen werden sollen. Diese aufgabenspezifische Delegationsentscheidung setzt der Operateur in der Auftragserteilung durch eine Spezifikation der manuellen Kontrolle einzelner Automationsfunktionen für bestimmte Einzelaufträge um. Im Rahmen dieser Arbeit erfolgt z.B. eine manuelle Sensorführung bei der Aufklärung eines Straßenabschnitts, damit der Operateur selbstständig wesentliche Objekte fokussieren und bewerten kann. Aus Sicht des Agenten werden hierdurch bestehende UAV-Fähigkeiten durch gleichwertige Fähigkeiten des Menschen ersetzt (*Fähigkeitssubstitution*) oder zusätzliche Fähigkeiten durch den Operateur eingebracht (*Fähigkeitsaddition*).

Der Agent berücksichtigt hierfür menschliche Fähigkeiten als hochpriorisierte, virtuelle UAV-Fähigkeiten, die im Fähigkeitsbewusstsein zunächst inaktiv hinterlegt sind und durch die Spezifikation des Operateurs aktiviert werden. Menschliche Fähigkeiten können dabei hinsichtlich ihrer Effekte redundant zu UAV-Fähigkeiten hinterlegt sein (*Fähigkeitssubstitution*), besitzen jedoch stets individuelle Voraussetzungen zur Nutzung. Generell setzt der Einsatz menschlicher Fähigkeiten die kontinuierliche Verfügbarkeit der Datenverbindung über den Zeitraum des Eingriffs voraus, um die Kommandierung und Überwachung der Automationsfunktionen durch den Operateur gewährleisten zu können. Wird die Datenverbindung während der manuellen Kontrolle unterbrochen, ergreift der Agent über die Nutzung des Datenlinkmanagements selbstständig Maßnahmen, um sie wiederherzustellen.

Der manuelle Eingriff des Menschen führt zu einem temporären Entzug von Fähigkeiten und UAV-Ressourcen für den Agenten. In der symbolischen Planung wird die Ressourcenverfügbarkeit mit den übrigen Handlungen des Agenten koordiniert und als Teil der komplettierten Handlungsagenda an entsprechender Stelle deklariert. Während der Planumsetzung wird der Operateur zu Beginn der manuellen Handlungen durch den Agenten informiert und ihm die Kontrolle über die betroffenen Automationsfunktionen übertragen. Nach Abschluss der Handlungen informiert der Operateur wiederum den Agenten und überträgt ihm erneut die Kontrollautorität.

Die Interaktion des Operateurs mit den manuell geführten Automationsfunktionen geschieht im Gegensatz zur Mensch-Agenten Interaktion im Allgemeinen durch einen Austausch subsymbolischer Parameter, woraus eine Mehrbelastung des Operateurs resultiert. Die Delegationsentscheidung des Menschen ist in diesem Sinne auch eine Abwägung zwischen dem Vertrauen in die Fähigkeiten des Agenten und den eigenen mentalen Ressourcen. Die Mensch-Maschine Schnittstelle muss in jedem Fall geeignete Steuerelemente und Anzeigen für die manuelle Kontrolle von Automationsfunktionen, parallel zur auftragsbasierten Führung des UAV bereitstellen. In Abschnitt 4.4 wird für diese Arbeit das Missions- und Sensorinterface vorgestellt.

Bei Bedarf kann der Agent auch selbstständig die manuelle Bereitstellung von Fähigkeiten durch den Operateur beantragen, um eigene Fähigkeitslücken zu schließen. Sind für eine obligatorische Handlung nur virtuelle UAV-Fähigkeiten im Fähigkeitsbewusstsein des Agenten hinterlegt, initiiert der Agent einen Antrag zur Unterstützung durch den Menschen. Die Unterstützung erfolgt dabei nach obiger Definition generell im Sinne einer *Fähigkeitsaddition*. Verfügt das UAV z.B. über keine ATOL Fähigkeit, kann der Agent bei Verfügbarkeit eines externen Sicherheitspiloten, den manuellen Start bzw. die manuelle Landung des Luftfahrzeugs beantragen. Entspricht der Operateur dem Antrag zur Unterstützung nicht, beendet der Agent die weitere Verarbeitung mit dem Verweis auf die Unausführbarkeit der Auftragsagenda.

3.3 Konzeptumsetzung durch Erweiterung einer bestehenden kognitiven Systemarchitektur

Für die Konzeptumsetzung soll ein Agent auf der Grundlage einer bestehenden kognitiven Systemarchitektur entwickelt werden. Die Auswahl der Architektur erfolgt anhand der in Abschnitt 3.2 formulierten Forderungen an den kognitiven Agenten:

1. Der Agent agiert **nutzenbasiert**, indem Ziele durch die optimierte Auswahl von Handlungsoptionen umgesetzt werden.
2. **Wissensbasierte Planung** dient dem Agenten dabei zur Ableitung logisch konsistenter Handlungsketten zur Umsetzung von Aufträgen des Operateurs.
3. Hierzu nutzt der Agent **symbolisch repräsentiertes Wissen** in Form domänenspezifischen Missionswissens sowie Wissen zur Wahrnehmung und Verarbeitung von Automationsfunktionen als UAV-Fähigkeiten.
4. Der Agent verknüpft äußere Ereignisse mit den Ergebnissen der wissensbasierten Planung, um eine auftrags- und situationsgerechte Mensch-Agenten Interaktion im Sinne einer **spezialisierten Etikette** abzubilden.

Gemäß Abschnitt 2.3.3 dienen die Systemarchitekturen EPIC und ACT-R primär zur realistischen Nachbildung menschlicher Informationsverarbeitung und sind lediglich nachrangig für die Abbildung intelligenten Verhaltens eines kognitiven Agenten geeignet. Subsumption ermöglicht als bottom-up Ansatz kein zielbasiertes Verhalten und verfügt darüber hinaus über keine symbolische Repräsentation von Wissen. ICARUS und ADAPT verknüpfen als robotische Ansätze primär Wahrnehmung und Handlungen eines physikalischen Agenten und sind damit ebenfalls unzureichend zur Implementierung eines Software-Agenten geeignet. Zudem verzichten beide Architekturen explizit auf die Betrachtung der Interaktion mit einem menschlichen Operateur. BDI und Soar ermöglichen beide die zielbasierte Verarbeitung symbolisch repräsentierten Wissens, beschränken sich jedoch auf die Inferenz und bieten keine integrierte Fähigkeit zur wissensbasierten Planung von Handlungsschritten. COSA schließlich wurde bereits durch [Uhrmann 2013] erfolgreich zur Implementierung der auftragsbasierten Führung von UAV genutzt. Hohe Rechenanforderungen der integrierten Soar Inferenzmaschine (vgl. Abschnitt 2.3.3) schränken jedoch die Komplexität und die Flexibilität der Handlungsplanung, besonders auf Grund der Performanzlimitationen an Bord von UAV, deutlich ein. COSA integriert zudem keinen eigenen Planer, sondern greift auf externe Funktionen zurück, wodurch Planungsergebnisse und Planungsfehler nicht unmittelbar im Sinne einer situationsgerechten Mensch-Agenten Interaktion genutzt und kommuniziert werden können.

Im Folgenden wird die Systemarchitektur COSA² für die technische Umsetzung verwendet. Sie zeichnet sich insbesondere durch die integrierte Inferenz, die symbolische Planung und Optimierung sowie die prozedurale Planausführung auf Grundlage einer zentralen Wissensbasis in einer geschlossenen Lösung aus. COSA² kann hierbei über frei konfigurierbare Schnittstellen in ein bestehendes UAV-System integriert werden und bietet die Möglichkeit zur Anbindung verschiedener Automationskomponenten.

3.3.1 Erweiterung der COSA² Architektur

Bestimmte Aspekte des Agentenverhaltens integriert COSA² als Systemarchitektur jedoch nur bedingt, wie [Brüggenwirth 2014] anhand der kognitiven Subfunktionen des modifizierten Modells menschlichen Verhaltens nach [Onken & Schulte 2010] darlegt. Für eine vollständige technische Abbildung des Agentenverhaltens bedarf es demnach einer Erweiterung von COSA². Diese umfasst dabei im Einzelnen folgende Subfunktionen:

- **Task Execution** meint die Überführung der aktuellen Aufgabe (*current task*) in ausführbare Einzelaktionen (*action instructions*) über im a-priori Wissen hinterlegte Prozeduren (*Procedures*). Hierzu wird eine Prozedur, als symbolische Repräsentation einer festen Abfolge von Einzelaktionen, regelbasiert ausgewählt und parametrisiert. COSA² reduziert diesen Vorgang auf eine direkte Verknüpfung von Handlungen mit bestimmten Prozeduren, sodass die eigentliche Prozedurenauswahl nunmehr in der wissensbasierten Aufgabenplanung (*Planning*) mit abgebildet wird.

Zur Begrenzung der Komplexität der symbolischen Planung, integriert die Erweiterung für COSA² ein fähigkeitsbasiertes Entscheidungsmodell zur Prozedurenauswahl. In einer Ergebnismatrix sind die Prozeduren dazu als Alternativen in einer festen Präferenzordnung hinterlegt [Laux et al. 2012]. Die Prozedurenauswahl erfolgt anhand der Verfügbarkeit von Systemfähigkeiten, passend zur aktuellen Aufgabe, die durch COSA² übergeben wird. Die Parametrisierung der gewählten Prozedur erfolgt anschließend auf der Grundlage einer subsymbolischen Lagerepräsentation, die in der Erweiterung für COSA² hinterlegt ist.

„Für Action Control gilt im Wesentlichen das für Feature Formation gesagte – aufgrund der geforderten symbolischen Natur bietet es sich an, subsymbolische Regelungsprozesse als dezidierte Prozesse außerhalb der Architektur einzubinden.“ [Brüggenwirth 2014]

- **Feature Formation** dient dem Menschen zur Wahrnehmung und Vorverarbeitung von Sinneseindrücken, während **Action Control** Teilaufgaben in Handlungen auf der sensomotorischen Ebene umwandelt. Die Implementierung von COSA² vereinfacht diese beim Menschen hoch komplex ausgeprägten Funktionen stark. Die Wahrnehmung wird auf die Schaffung symbolischen Wissens aus Eingangsinformationen reduziert, während *Action Control* lediglich vorgefertigte Handlungssequenzen aufruft. In COSA² ist eine subsymbolische Verarbeitung von Informationen bzw. Parametrisierung von Handlungen im Allgemeinen nicht möglich. Diese Funktionen werden nun in der Erweiterung für COSA² abgebildet. Sie fungiert hierbei als Ein- und Ausgangsschnittstelle des Agenten für die Anbindung subsymbolisch kommunizierender Automationsfunktionen sowie der Mensch-Maschine Schnittstelle. Fähigkeiten werden durch die Erweiterung für COSA² aus Automationsfunktionen abstrahiert und für das Fähigkeitsbewusstsein des Agenten

bereitgestellt. Die Kommandierung und Überwachung von Einzelaktionen innerhalb von Prozeduren erfolgt ebenfalls durch die Erweiterung für COSA². Schließlich ermöglicht sie durch direkte Verknüpfung bestimmter Ereignisse mit Handlungen reaktives Verhalten des Agenten, welches nebenläufig zur Zielumsetzung abgerufen werden kann.

In Anlehnung an die Möglichkeit reaktives Verhalten des Agenten abzubilden, wird die beschriebene Erweiterung für COSA² im Folgenden vereinfacht als *Reaktive Erweiterung für COSA²* (engl. *Reactive Extension for COSA² - ReX*) bezeichnet.

3.3.2 Resultierende Agentenarchitektur

Die kognitive Systemarchitektur COSA² wird durch ReX zu einer mehrschichtigen Agentenarchitektur (engl. *Layered Agent Architecture* [Müller et al. 1995]) ergänzt. Abbildung 3-7 stellt die Zuordnung kognitiver Subfunktionen zwischen COSA² und ReX innerhalb der Agentenarchitektur grafisch dar. COSA² dient dabei zur Realisierung der Subfunktionen *Identification*, *Goal Determination*, *Planning* und *Task Determination* sowie *Task Execution* anteilig. ReX bildet neben der *Feature Formation* und der *Action Control* auch die Subfunktion *Task Execution* gemäß Abschnitt 3.3.1 anteilig ab. Während die konzeptbasierte Verhaltensebene somit weiterhin durch COSA² umgesetzt wird, erfolgt auf der prozedurbasierten Ebene eine Entkopplung der wissensbasierten und der regelbasierten Anteile.

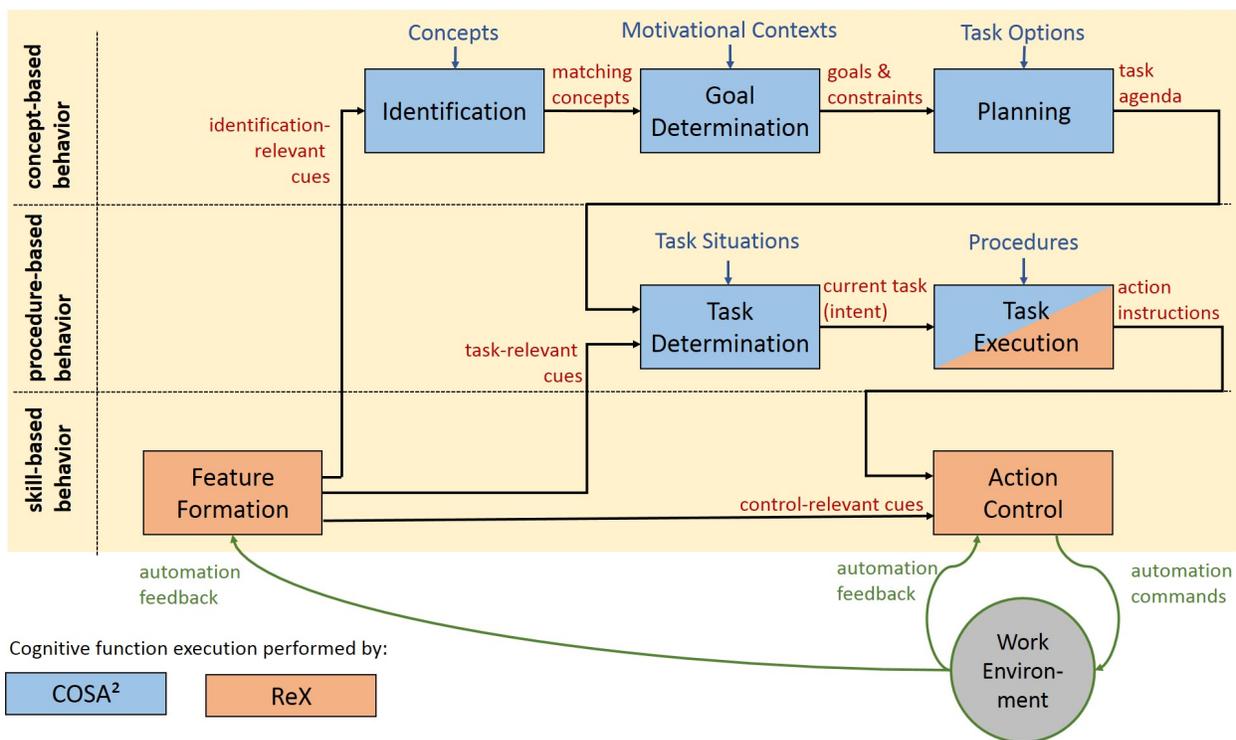


Abbildung 3-7: Realisierung einer zweischichtigen Agentenarchitektur anhand kognitiver Subfunktionen durch den kombinierten Einsatz von COSA² und ReX

COSA² und ReX verwenden dabei keine einheitliche Wissensbasis. Während COSA² für alle Verarbeitungsfunktionen eine symbolische Wissensrepräsentation nutzt, verarbeitet ReX weitestgehend subsymbolische Informationen, um unmittelbar mit den nachgeordneten Automationsfunktionen interagieren zu können. Die Kommunikation innerhalb der

Agentenarchitektur geschieht nachrichtenbasiert durch Austausch symbolischer Informationen, die durch ReX aus subsymbolischen Inhalten generiert bzw. in diese überführt werden. Zu diesem Zweck integriert ReX eine eigene COSA²-Schnittstelle, die auch für die zeitliche Taktung des Nachrichtenflusses sowie für die Prozesssteuerung von COSA² genutzt wird.

Die Auftragsverarbeitung geschieht gestaffelt in COSA² und ReX. Eingehende Auftragsagenden des Operators werden in ReX über eine interne Auftragsverwaltung vorverarbeitet und abschnittsweise verarbeitet, um Mehrfach-Ziele bei der symbolischen Planung in COSA² zu vermeiden. In einem ersten Schritt fasst ReX hierzu Einzelaufträge in Form übergeordneter Zielvorgaben zusammen, die durch COSA² umgesetzt werden können. Die Ziele und Randbedingungen der Einzelaufträge werden hierbei als Vorbedingungen für die übergeordnete Zielvorgabe übernommen. Die übergeordneten Zielvorgaben werden anschließend einzeln an COSA² übertragen, wo die Planung und Zielumsetzung vollzogen werden. Auf diese Weise werden alle Zielvorgaben sukzessive an COSA² übertragen, bis die Auftragsagenda vollständig umgesetzt ist. COSA² plant und überwacht dabei die Auftragsumsetzung wissensbasiert auf der Ebene komplexer UAV-Handlungen, realisiert durch Prozeduren. Die Kontrolle und Überwachung von Einzelaktionen geschieht ausschließlich durch ReX.

Die Mensch-Agenten Interaktion wird dabei vollständig über eine Operator-Schnittstelle in ReX realisiert. Das regelmäßige Agentenfeedback umfasst den jeweils aktuellen UAV-Zustand, abstrahiert über UAV-Fähigkeiten und den COSA²-Prozessstatus bzw. den Erfüllungsgrad der Handlungsagenda in COSA². Ereignisabhängige Interaktion gemäß Abschnitt 3.2.3 initiiert der Agent bei Fehlern in der symbolischen Handlungsplanung sowie für Freigaben und Anträge zur manuellen Auftragsunterstützung. Eine regelbasierte Ursachenanalyse dient ReX, basierend auf der Verfügbarkeit von UAV-Fähigkeiten, zur Bestimmung von Fehlerursachen der internen Lagerepräsentation sowie der Einzelaufträge der Auftragsagenda des Operators.

Automationspezifische Schnittstellen dienen in ReX zur Anbindung der Automationskomponenten an den Agenten. Die Kommandierung der Automation geschieht dabei mittels hinterlegter Kommandoprotokolle, durch Überführung der zuvor parametrisierten Einzelaktionen in Kommandos. Die Überwachung der Ausgabewerte der Automationskomponenten erfolgt subsymbolisch im Hinblick auf die Umsetzung der Aktionen. Im Weiteren werden aus den Automationsfunktionen und ihrem Status Fähigkeiten gebildet und weiterverarbeitet. Reaktives Verhalten wird in ReX unmittelbar durch Initiierung vorgefertigter Einzelaktionen umgesetzt, ohne diese in die symbolische Planung zu integrieren.

Innerhalb der Agentenarchitektur kann eine Zuordnung der Phasen des Verarbeitungskonzepts gemäß Abschnitt 3.2.2 vorgenommen werden. Die Vorverarbeitung und Bewertung der Auftragsagenda des Operators hinsichtlich der Umwelt- und Lagerepräsentation erfolgt in ReX (*Evaluate*). Es folgt die symbolische Planung zur Erstellung einer komplettierten Handlungsfolge auf Grundlage des Fähigkeitsbewusstseins und des Missionswissens des Agenten in COSA² (*Supplement*). Abschließend erfolgt die regelbasierte Prozedurenauswahl und die subsymbolische Parametrisierung ihrer Einzelaktionen wiederum mittels ReX (*Decompose*).

4 Implementierung

Dieses Kapitel beschreibt die technische Umsetzung des kognitiven Agenten sowie der Automationskomponenten gemäß Kapitel 3. Zunächst erfolgt die Darstellung der Agentenarchitektur, insbesondere der modularen Struktur der Reaktiven Erweiterung für COSA² (ReX) in Abschnitt 4.1. Abschnitt 4.2 gibt einen Überblick über die Modellierung der Wissenskomponenten des Agenten in COSA². Es folgen der Entwurf und die Implementierung der Automationskomponenten an Bord des UAV in Abschnitt 4.3 sowie deren Anbindung an den Agenten über eine automationsspezifische Schnittstelle. Die Beschreibung der Mensch-Maschine Schnittstelle zur auftragsbasierten Führung erfolgt dann in Abschnitt 4.4. Abschließend wird in Abschnitt 4.5 die Plattformunabhängigkeit der technischen Umsetzung anhand einer mehrschichtigen Abstraktion von Kommunikations- und Planungsebenen dargestellt.

4.1 Der kognitive Agent

Der kognitive Agent wird gemäß Abschnitt 3.3 als mehrschichtige Agentenarchitektur durch Kombination der Systemarchitektur COSA² mit ReX realisiert. Abschnitt 4.1.1 gibt zunächst einen Überblick über die Entwicklungsumgebung zur grafischen Wissensmodellierung in COSA² sowie die COSA²-Shell Anwendung zur Überwachung des Kernels und zum Debugging des Wissensmodells während der Laufzeit. In Abschnitt 4.1.2 wird dann die modulare Struktur von ReX, die Anbindung externer Automationsprozesse sowie die Kommunikation und Prozesssteuerung für COSA² dargestellt. Abschnitt 4.1.3 zeigt die Verknüpfung von COSA² und ReX in einem Agentendesign sowie die hierarchische Verarbeitungsstruktur dieser Verknüpfung.

4.1.1 Die kognitive Systemarchitektur COSA²

Die Struktur und die Komponenten der kognitiven Systemarchitektur COSA² wurden bereits in Abschnitt 2.3.3 vorgestellt und sollen im Folgenden hinsichtlich der Implementierung des kognitiven Agenten und der Wissensimplementierung betrachtet werden. Die Implementierung von COSA² basiert dabei ganzheitlich auf der .NET Programmiersprache C#. Die drei Systemkomponenten von COSA² (Kernel, Shell und CML-Editor) stellen separate Softwareprogramme dar, die unabhängig voneinander genutzt und dezentral über Interprozesskommunikation miteinander verknüpft werden können.

Die Implementierung des a-priori Wissens in COSA² erfolgt über ein Wissensmodell, abgebildet in CML. Der Wissensingenieur nutzt hierzu eine grafische Oberfläche (den CML-Editor), die neben den Wissenskomponenten (engl. *Knowledge*) die Ein- und Ausgabeschnittstelle (*IO-Interface*) des künftigen Agenten darstellt. Die Wissenskomponenten sind analog zu den a-priori Wissens-elementen des modifizierten Modells menschlichen Verhaltens gegliedert: Konzepte (engl. *concepts*), Ziele (engl. *motivational contexts*), Handlungsoptionen (engl. *task options*) und Prozeduren (engl. *procedures*). Abbildung 4-1 zeigt den CML Editor mit exemplarischen Wissens-elementen sowie einer spezifischen Vorbedingung für die L3-Regel *conductDataTransmission* in COSA².

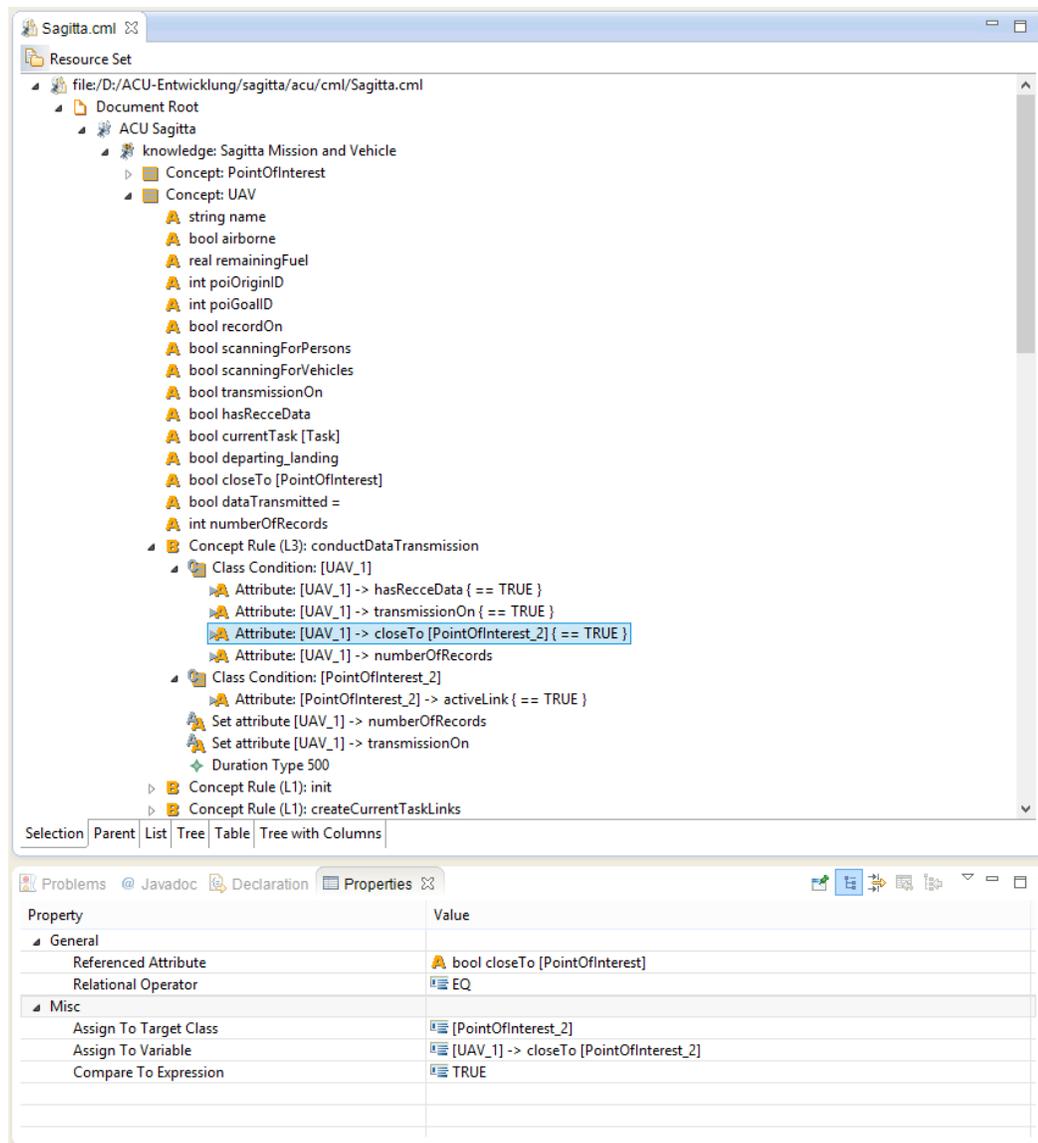


Abbildung 4-1: CML-Editor zur grafischen Wissensmodellierung in COSA²

Konzepte und Ziele werden durch *Attribute* und *Regeln*, ähnlich dem Konzept von Klassen in der objektorientierten Programmierung, beschrieben. Attribute bestimmen dabei die Eigenschaften und die möglichen Zustände der Klassen, während Regeln ihr Verhalten definieren. Attribute können vom Typ *string*, *int*, *bool*, *real* oder *enum* sein, wobei *enums* als Wissenskomponenten vom Entwickler einzeln modelliert werden. Regeln werden über Vor- und Nachbedingungen modelliert. Zum *Feuern* (d.h. zur Anwendung) der Regeln müssen alle Vorbedingungen erfüllt sein. Wird eine Regel angewendet, so werden ihre Nachbedingungen umgesetzt. Die Vor- und Nachbedingungen von Regeln beziehen sich stets auf Attribute von Konzepten oder Zielen und verändern diese gegebenenfalls. Vorbedingungen können zusätzlich über logische Verknüpfungen verbunden sein.

Regeln können einem der in Abschnitt 2.3.3 vorgestellten Level des Situationsbewusstseins nach [Endsley 1995] zugeordnet werden (*L1*, *L2* bzw. *L3*). *L3*-Regeln nutzt COSA² während der Planung zur Prädiktion der Auswirkungen von Umweltverhalten. Die Berücksichtigung von zeitlichen Abhängigkeiten kann dabei zusätzlich über die Dauer (*Duration*) von *L3*-Regeln modelliert

werden. Während der Laufzeit dienen sie COSA² zur Überprüfung der zeitlichen Konsistenz der Handlungsagenda.

Die Modellierung der Interaktionsschnittstelle in COSA² umfasst die Definition von Nachrichtenklassen, die über eine gewählte Interprozesskommunikation (vgl. Abschnitt 4.5.1) gesendet und empfangen werden können. Nachrichtenklassen kapseln eine beliebige Menge von Attributen und dienen der Kommunikation mit externen Systemkomponenten. Eingehende Nachrichten werden über *LI*-Regeln ausgelesen und zur Verwendung im internen Wissen verarbeitet.

Zum Nachweis der Fähigkeiten von COSA² beschreibt [Brüggenwirth 2014] eine effiziente Eigenimplementierung eines PDDL-Planers für die Domäne der Luftfahrt. Da dieser Planer zum Beginn dieser Arbeit noch nicht verfügbar war, wird stattdessen ein PDDL-Planer auf Basis des Metric-FF Algorithmus [Hoffmann 2015] verwendet. Die IBM CPLEX® Bibliothek wird während der Laufzeit als externe Systembibliothek zur zeitlichen Staffelung von Handlungen und Effekten eingebunden. Durch die plattformübergreifende Verfügbarkeit des PDDL Planers sowie der CPLEX® Bibliothek kann die Architektur auch über Systemgrenzen hinweg eingesetzt werden. Die Entkopplung des Kernels von der Shell-Oberfläche ermöglicht in COSA² darüber hinaus die Überwachung und Steuerung der Systemarchitektur von externen Plattformen.

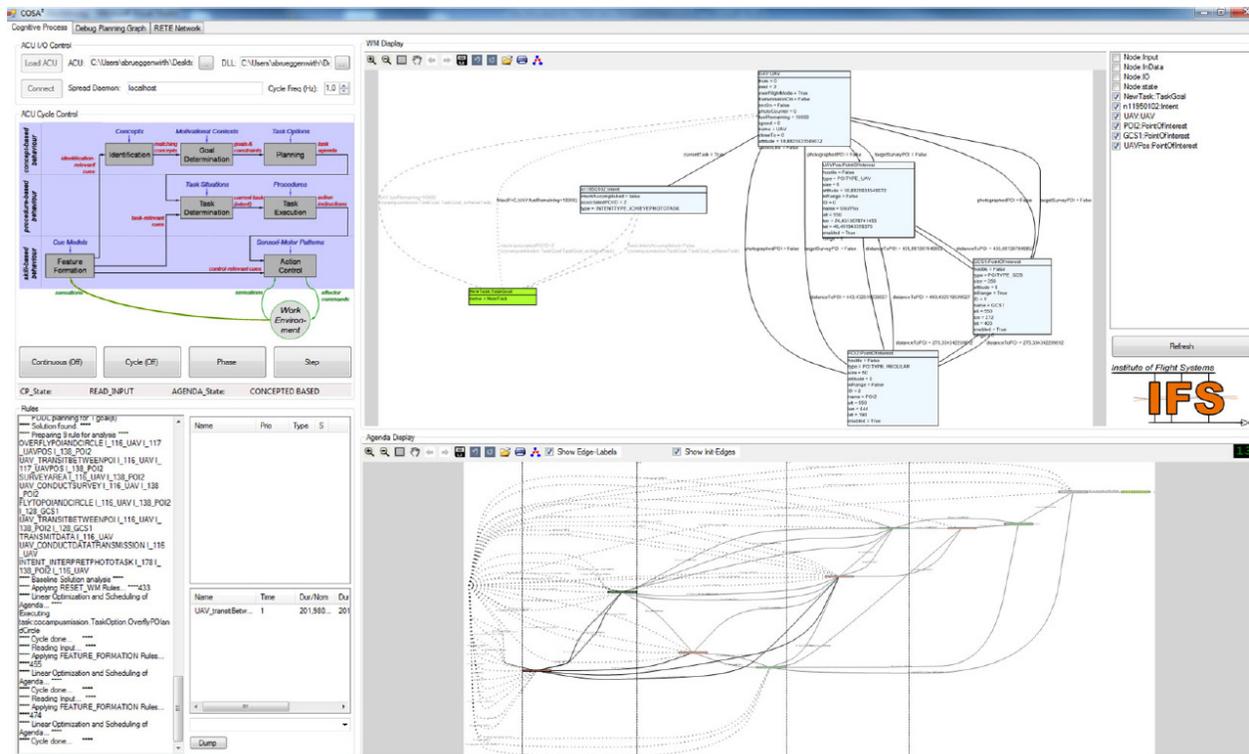


Abbildung 4-2: COSA²-Shell zur grafischen Darstellung des Arbeitsspeicherinhalts, der Auftragsagenda und des kognitiven Verarbeitungsprozesses eines Agenten [Brüggenwirth 2014]

Die COSA²-Shell dient zur grafischen Anzeige des Arbeitsspeicherinhalts (Situationswissen) des COSA²-Kernels sowie der Handlungsagenda zur Erfüllung der aktiven Ziele. Die COSA²-Shell bietet als Debug-Schnittstelle die Möglichkeit zur Steuerung des COSA²-Kernels und zur Überwachung innerer Systemzustände. Abbildung 4-2 zeigt die COSA²-Shell während der Laufzeit

des kognitiven Agenten. Links ist die Debug-Schnittstelle mit einer grafischen Darstellung des Verarbeitungsablaufs anhand der kognitiven Subprozesse des modifizierten Modells menschlichen Verhaltens nach Rasmussen dargestellt. Im oberen Bereich ist das Situationswissen in Form von Instanzen der Konzepte und Ziele mit ihren jeweiligen Attributen dargestellt. Im unteren Bereich befindet sich die grafische Darstellung der Handlungsagenda mit den Handlungsoptionen (als Ovale dargestellt). Links unten wird schließlich die Debug-Ausgabe in Textform dargestellt.

4.1.2 Reaktive Erweiterung für COSA² - ReX

ReX ist als eigenständiges Softwareprogramm in C++ implementiert und dient COSA² zur parametrischen Interaktion mit anderen Prozessen (inkl. der Operateur- Schnittstelle) sowie der Vorverarbeitung eingehender Informationen (Abbildung 4-3). Darüber hinaus implementiert ReX eine Auftragsverwaltung sowie eine Prozesssteuerung für COSA², die es erlaubt, mögliche Fehlerursachen bei der symbolischen Handlungsplanung zu bestimmen und dem Operateur mitzuteilen. Zu diesem Zweck besitzt ReX eine innere Repräsentation des Lagebilds, des Zustands und der Fähigkeiten des UAV sowie aktiver Prozeduren und Aktionen, die mit COSA² synchronisiert werden. Die verfügbaren Prozedurentypen, die im Rahmen der Prozedurenauswahl (*Task Execution*) verarbeitet werden, sind ebenso in ReX hinterlegt, wie die Regeln für das reaktive Verhalten des kognitiven Agenten.

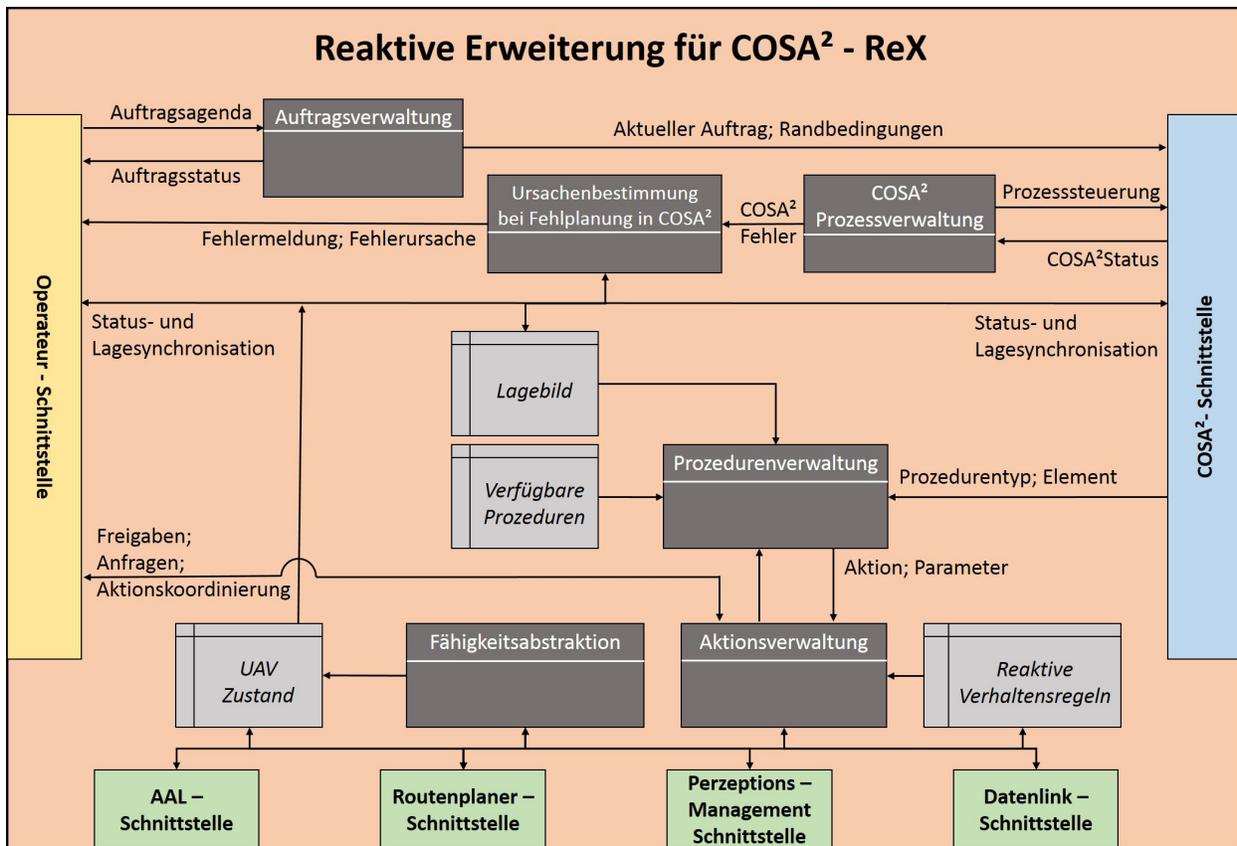


Abbildung 4-3: Funktionelle Darstellung der reaktiven Erweiterung für COSA² - ReX

4.1.2.1 Status- und Lagesynchronisation (Informationsvorverarbeitung)

Um den kognitiven Agenten an äußere technische Prozesse, wie die UAV Automation und die Mensch-Maschine Schnittstelle anzubinden, stellt ReX verschiedene Schnittstellen sowie eine Informationsvorverarbeitung bereit. Die Schnittstellen fungieren als Adapter für den Informationsaustausch und stellen für ReX Informationen über Art und Zustand der gegenüberliegenden Prozesse dar. Hierzu implementieren die Schnittstellen passende Kommunikationsprotokolle und abstrahieren Aktionen als parametrisierbare Kommandos, die durch den Agenten kommandiert werden können (Abschnitt 4.3).

Die Schnittstellen in ReX verarbeiten weitestgehend parametrische Informationen als Eingangswerte und legen diese in einer zentralen UAV-Zustandsrepräsentation ab. Die Informationen dienen zudem als Grundlage der Fähigkeitsabstraktion (Abschnitt 4.1.2.3). Der UAV-Zustand (Zustandsinformationen) und das Lagebild (Lagesynchronisation) werden innerhalb des Agenten, zwischen COSA² und ReX, sowie zwischen dem Agenten und der Mensch-Maschine Schnittstelle synchronisiert.

Die Synchronisation von UAV Zustands- und Lageinformationen zwischen ReX und COSA² verläuft gefiltert und abstrahiert, um den symbolischen Verarbeitungsprozess in COSA² zu entlasten. Die Überführung parametrischer Informationen in symbolische Form wird weitestgehend nach ReX verschoben, sodass die Anzahl der *LI*-Regeln in COSA² zur Umweltwahrnehmung reduziert werden. Die Vorverarbeitung in ReX etabliert dabei eindeutige Bezeichnungen symbolischer Elemente, die zwischen ReX und COSA² ausgetauscht werden, um eine Zuordnung von parametrischen Informationen zu symbolischen Elementen zu ermöglichen.

4.1.2.2 Auftragsverwaltung

Gemäß Abschnitt 3.2.2 erfolgt die Auftragserteilung durch den Operateur in Form einer Auftragsagenda als Liste von Einzelaufträgen. COSA² bietet jedoch keine Möglichkeit sequentielle Ziele bzw. Unterziele als Vorgaben zu formulieren. Vielmehr können lediglich ein einzelnes oder mehrere, parallel zu erreichende Ziele verarbeitet werden. Für die Auftragserteilung in Form einer Auftragsagenda wird demnach eine externe Auftragsverwaltung in ReX realisiert, die aus der eingehenden Auftragsagenda konsekutive Einzelaufträge generiert, die COSA² umsetzen kann.

ReX wandelt hierzu die eingehende Auftragsagenda in eine Liste von Zielen und Zwischenschritten um. Die Ziele werden anschließend sequentiell an COSA² übermittelt und die Zwischenschritte über Randbedingungen formuliert. COSA² betrachtet demnach ausschließlich den aktuell von ReX übertragenen Auftrag und generiert dadurch im Allgemeinen teil-optimale Lösungen bezüglich der gesamten Auftragsagenda. Den Abschluss der Auftragsumsetzung teilt COSA² ReX im Rahmen des Agentenstatus mit und initiiert dadurch die Weiterschaltung der Auftragsagenda.

Trifft eine neue Auftragsagenda im Agenten ein, stellt dieser zunächst den Handlungsbedarf fest und beginnt erst anschließend mit der Verarbeitung. Dies geschieht über einen Abgleich zur Identifizierung von Unterschieden zwischen dem aktuellen Auftrag in COSA² und dem ersten Auftrag der eingehenden Auftragsagenda. Werden Unterschiede festgestellt, wird die aktuelle Auftragsverfolgung eingestellt und mit der Verarbeitung der neuen Agenda begonnen. Andernfalls wird die aktive Agenda lediglich um neue Elemente der eingehenden Auftragsagenda ergänzt und somit die Anpassung für folgende Auftrags Elemente vollzogen.

Als Teil der Leistungsinformationen übermittelt der Agent dem Operateur die Handlungsagenda zur Umsetzung des aktuellen Auftrags sowie den Status der einzelnen Handlungsschritte. Dieser Status umfasst die Kennzeichnung bereits durchgeführter, aktueller und zukünftiger Handlungen des Agenten im Sinne der Forderung nach Nachvollziehbarkeit und Vorhersehbarkeit. Da in COSA² Auftrags Elemente konsekutiv durchlaufen werden, können dem Operateur dabei lediglich Informationen über den aktuellen Auftragsstatus übermittelt werden.

4.1.2.3 Fähigkeitsabstraktion

Die Fähigkeiten des UAV werden in ReX anhand des Zustands von Automationsfunktionen abstrahiert. Die Automationschnittstellen stellen hierzu Informationen über die Verfügbarkeit einzelner Automationsfähigkeiten bereit, welche zentral in ReX verarbeitet und zu UAV-Fähigkeiten zusammengefügt werden. UAV-Fähigkeiten sind Teil des UAV-Status und bilden eine wesentliche Grundlage des Handelns des kognitiven Agenten. Die gestaffelte Abstraktion von Fähigkeiten ist in Abbildung 4-4 exemplarisch dargestellt.

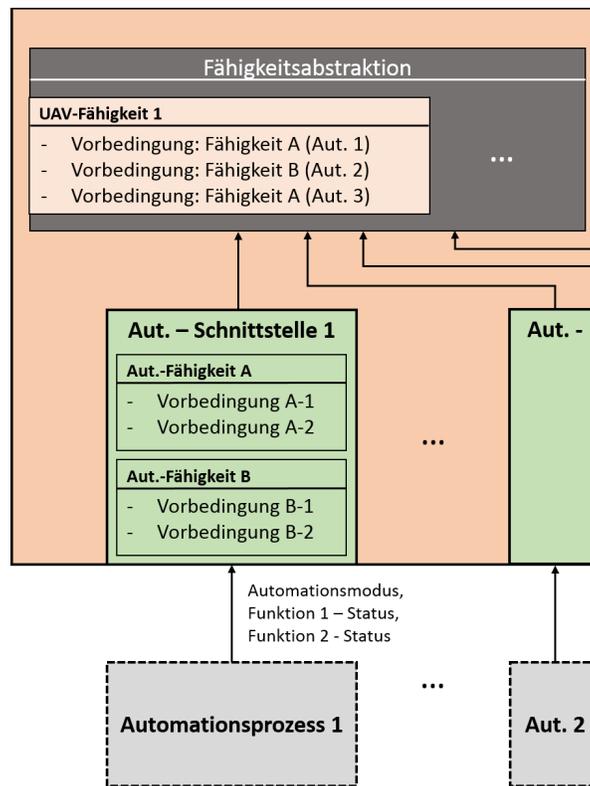


Abbildung 4-4: Gestaffelte Abstraktion von UAV-Fähigkeiten in ReX

Die Abstraktion von Fähigkeiten in den Automationschnittstellen erfolgt über die Ermittlung des Status einzelner Automationsfunktionen. Der Funktionsstatus kann dabei über Abfragen der Automationschnittstelle oder über die Auswertung von Rückgabewerten ausgelesen werden. Die Vorbedingungen für die Verfügbarkeit von Automationsfähigkeiten kann dabei beliebig komplex formuliert werden. Die Kombination von verfügbaren Automationsfähigkeiten beschreibt schließlich die Verfügbarkeit der übergeordneten UAV-Fähigkeiten.

Die Verfügbarkeit von UAV-Fähigkeiten ist in ReX als boolesche Information repräsentiert und wird COSA² zur Handlungsplanung und Systembewertung bereitgestellt. ReX nutzt UAV-

Fähigkeiten bei der regelbasierten Prozedurenauswahl (vgl. Abschnitt 4.1.2.4). UAV-Fähigkeiten werden als Teil der Statusinformationen über die Mensch-Maschine Schnittstelle an den menschlichen Operateur übermittelt.

4.1.2.4 Regelbasierte Prozedurenauswahl & -verwaltung

ReX realisiert die kognitive Subfunktion *Task Execution* über die regelbasierte Auswahl von Prozeduren, passend zu gewünschtem Verhalten, dass durch COSA² vorgegeben wird. Zudem parametrisiert ReX die Einzelaktionen der Prozeduren für die anschließende Kommandierung über die Automationschnittstellen. Hierzu realisiert ReX einen regelbasierten Entscheidungsprozess nach [Laux et al. 2012], der auf Grundlage der vorgegebenen Handlungsanweisung und den jeweiligen Fähigkeiten des UAV aus die bestmögliche der zur Verfügung stehenden Prozeduren auswählt. Das Entscheidungsproblem wird dabei als zweistufiges Problem mit mehreren sicheren Zielgrößen modelliert. Die Zielgrößen sind durch die Verfügbarkeit der Fähigkeiten definiert; die Alternativen umfassen die Menge aller Prozeduren, die prinzipiell die Handlungsanweisung umsetzen können, da sie den passenden Prozedurentyp besitzen. Die Prozeduren umfassen hierbei eine Spezifikation der Fähigkeiten, deren Verfügbarkeit zur Auswahl notwendig ist.

Im ersten Schritt werden aus den Prozeduren solche ausgewählt, für die alle benötigten UAV-Fähigkeiten verfügbar sind. In einem zweiten Schritt folgt der Vergleich dieser Prozeduren anhand einer vordefinierten Metrik. Die Parametrisierung von Prozeduren erfolgt im Anschluss an deren Auswahl anhand der internen Lagerepräsentation in ReX. Hierbei werden die Aktionsparameter in der Prozedur schrittweise befüllt.

Der Erfüllungsgrad einzelner Prozeduren wird nicht an COSA² übermittelt, sondern gekapselt in der Prozedurenverwaltung verarbeitet. COSA² werden lediglich abstrahierte Statusinformationen bezüglich der Handlungsanweisung übermittelt. COSA² vollzieht eine wissensbasierte Umweltverarbeitung lediglich indirekt über eingehende Statusinformationen sowie deren Abgleich mit projiziertem Umweltverhalten (mittels *L3-Regeln*).

4.1.2.5 Aktionsverwaltung

Die Einzelaktionen der Prozeduren werden von ReX über die Automationschnittstellen umgesetzt. Zur Statusverwaltung der Aktionen integriert ReX eine interne Aktionsverwaltung. Die Aktionsverwaltung nutzt dabei kontinuierlich die Daten der Automationschnittstellen für aktive Aktionen. Der erfolgreiche Abschluss einer Aktion wird an die Prozedurenverwaltung weitergeleitet.

Die hierarchische Trennung von Aktionen und Prozeduren ermöglicht die nebenläufige Verwaltung reaktiver Handlungen. Zusätzlich zu Aktionsfolgen aus Prozeduren können reaktiv bestimmte Aktionen zur Aktionsverwaltung hinzugefügt werden. Die Koordination der einzelnen Aktionen wird hierdurch gekapselt (vgl. Abschnitt 4.1.2.6).

Die Aktionsverwaltung realisiert ebenfalls die Auftragsunterstützung durch den menschlichen Operateur gemäß Abschnitt 3.2.4. Hierbei werden Aktionen nicht über die Automationschnittstellen kommandiert, sondern an die Operateur-Schnittstelle als Anfragen geleitet, bzw. die Interaktion mit dem Operateur so adaptiert, dass dieser manuelle Kontrolle über Automationsfunktionen erhält.

Die Interaktion mit dem Menschen umfasst im Rahmen der Auftragsunterstützung die Anfrage von Freigaben für bestimmte UAV-Aktionen, Anfragen für die Bereitstellung menschlicher Fähigkeiten sowie die Kommunikation zur Aktionskoordinierung zwischen Mensch und Agent. Freigaben können dabei erteilt oder verweigert werden, was zum Erfolg oder Misserfolg der jeweiligen Aktion führt. Anfragen zur Unterstützung umfassen die Handlungsabsicht des Agenten sowie das relevante Lageelement. Die Aktionskoordinierung erfolgt auf unterschiedliche Weise. Kann der Agent den erfolgreichen Abschluss einer Aktion über den UAV-Zustand bestimmen (z.B. manueller Start- und Landevorgang), ist keine zusätzliche Kommunikation zwischen Mensch und Agent notwendig. Ist dies, wie im Fall der manuellen Sensordatenauswertung, nicht möglich, muss der Mensch den Abschluss seiner Handlungen manuell bestätigen. Hierzu wird die Mensch-Maschine Schnittstelle auf geeignete Weise adaptiert.

4.1.2.6 Reaktive Verhaltensregeln

ReX verknüpft Zustände und Ereignisse regelbasiert mit Aktionsfolgen, um reaktives Verhalten zu generieren. Reaktionen werden direkt über die Aktionssteuerung von ReX umgesetzt und nehmen keinen Einfluss auf aktive Aufträge oder Prozeduren des Agenten (vgl. Abschnitt 4.1.2.5). Es findet somit keine wissensbasierte Verarbeitung der Handlungen und Effekte statt, vielmehr nimmt COSA² die Auswirkungen der Reaktionen über die Änderungen von Umweltzuständen oder Automationsfunktionen wahr und beurteilt sie hinsichtlich aktueller Handlungspläne.

Im Rahmen dieser Arbeit ist reaktives Verhalten des Agenten beispielhaft als unmittelbare Reaktion auf bevorstehende Gefahrensituationen implementiert. Der Agent kommt durch sein reaktives Verhalten der wissensbasierten Situationsverarbeitung zuvor, da angenommen wird, dass keine zeitgerechte Bearbeitung von Gefahrensituation durch COSA² garantiert werden kann. Die Effekte dieser reflexartigen Reaktionen sind dabei temporär und werden durch das Ergebnis der wissensbasierten Situationsverarbeitung durch COSA² ersetzt.

4.1.2.7 COSA² Prozessverwaltung

Neben den funktionalen Komponenten stellt ReX Funktionen zur Prozesssteuerung von COSA² bereit. ReX überwacht hierfür den Status von COSA² in Form standardisierter Meldungen und PING-Signale. Die Notwendigkeit der Zustandsüberwachung und Prozesssteuerung durch ReX entspringt aus der Implementierung von COSA². COSA² stoppt seinen Verarbeitungszyklus, falls keine Lösung für eine bestehende Zielvorgabe bestimmt werden kann und ist nicht in der Lage, die Verarbeitung ohne äußeren Impuls zu reaktivieren.

Der Neustart des Verarbeitungszyklus benötigt somit die Reaktivierung durch einen äußeren Prozess. ReX übernimmt diese Funktion, indem es den Status von COSA² kontinuierlich überwacht und bei Bedarf eine Reinitialisierung bewirkt. Im Anschluss an den Neustart von COSA² stellt ReX dessen Arbeitsspeicher durch die Übertragung aller aktuellen Status- und Lageinformationen wieder her, sodass in COSA² das gleiche Lagebild existiert wie vor dem Verarbeitungsstopp, jedoch ohne die jeweilige Zielvorgabe, die zum Verarbeitungsabbruch geführt hat.

4.1.2.8 Ursachenbestimmung bei Fehlplanung in COSA²

Für die symbolische Planung nutzt COSA² einen PDDL-basierten Planer, dem die Zielvorgaben sowie die aktuelle Lagerepräsentation in Form einer Problemdatei übergeben werden. Kann keine symbolische Lösung gefunden werden, bricht die Planung ohne Ergebnis ab und COSA² stoppt die

Ausführung seines Verarbeitungszyklus mit einer Fehlermeldung an ReX. Da diese Fehlermeldung jedoch nicht die Ursache für die Fehlplanung beinhaltet, implementiert ReX ein Modul, um nach Planungsabbruch die Ursachen für die Fehlplanung bestimmen zu können. Diese Ursachen können dann als Teil des Agentenfeedbacks an den Operateur zurückgeführt werden.

Für die Ursachenbestimmung implementiert ReX eine Untermenge der Regeln von COSA² sowie der Abhängigkeiten zwischen den Handlungsoptionen und Fähigkeiten des UAV-Systems. Mit diesen Regeln werden die Planungsgrundlage, also die Lagerepräsentation, die UAV-Fähigkeiten und die Zielvorgabe in ReX analysiert. Hierbei wird versucht die Handlungsunfähigkeit des Agenten auf den Zustand eines Lageelementes oder den Zustand und die Fähigkeiten des UAV zurück zu führen. Wird auf diese Weise ein Element als Ursache erkannt, sendet ReX eine entsprechende Fehlermeldung an den Operateur, um ihm eine angepasste Auftragserteilung zu ermöglichen.

Wenn der Auftrag z.B. die Nutzung eines spezifischen Flugkorridors vorgibt, dieser jedoch nicht durchfliegen werden darf, da er in Wirkreichweite von SAM-Stellungen liegt, existiert keine Lösung für das Anwendungsproblem und die symbolische Planung in COSA² bricht ergebnislos ab. Durch die Ursachenbestimmung kann ReX eine Fehlermeldung für den Operateur formulieren, die die Bedrohung des obligatorischen Korridors als Widerspruch identifiziert.

4.1.3 Resultierendes Agentendesign

Die Fähigkeiten und das Verhalten des kognitiven Agenten werden durch die Kombination der kognitiven Systemarchitektur COSA² (Abschnitt 4.1.1) und seiner reaktiven Erweiterung ReX (Abschnitt 4.1.2) bestimmt. Im Folgenden soll die hierarchische Verarbeitung von Auftragsagenden durch die Kopplung beider Softwareelemente zu einem einzelnen Agenten betrachtet werden.

Abbildung 4-5 zeigt ein Schaubild der technisch realisierten Auftragsverarbeitung und benennt Zyklusfrequenzen der Verarbeitungshierarchien. Zunächst bestimmt ReX aus der eingehenden Auftragsagenda den aktuellen Auftrag mit seinen Randbedingungen und leitet diese an COSA² weiter. In COSA² wird anschließend eine Handlungsagenda zur Umsetzung des aktiven Auftrags abgeleitet. COSA² gibt das gemäß der Agenda gewünschte Verhalten ReX vor und spezifiziert zusätzlich ein Lageelement, auf das das gewünschte Verhalten angewendet werden soll. Entsprechend dem gewünschten Verhalten wird in ReX eine optimale Prozedur aus einer Menge an vorhandenen Prozeduren bestimmt und die hinterlegte Sequenz von Aktionen zur Aufgabenerfüllung parametrisiert. Die Einzelaktionen werden in Kommandos an die Automation umgewandelt, kommandiert und ihre Ausführung anschließend überwacht.

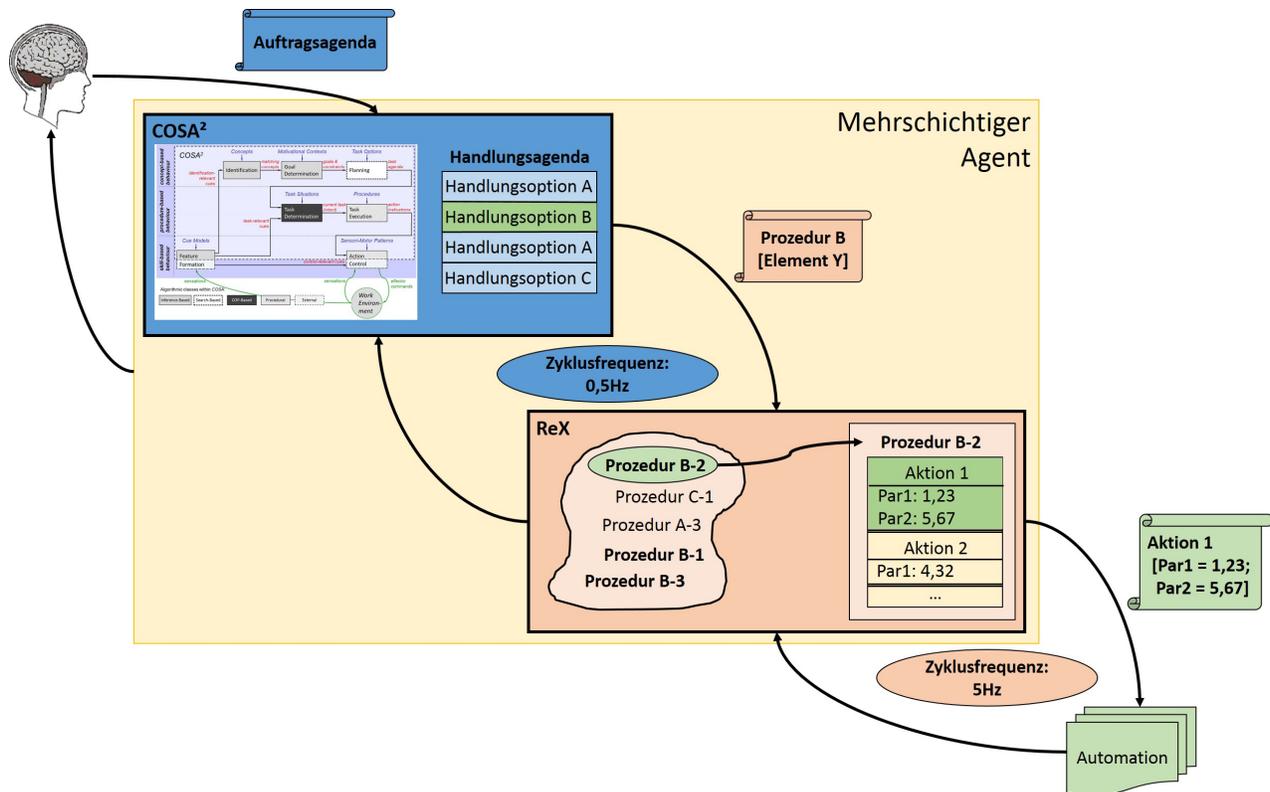


Abbildung 4-5: Hierarchische Verarbeitung von Aufträgen durch den kognitiven Agenten. *COSA²* berücksichtigt und überwacht Handlungsoptionen, die in *ReX* über Prozeduren in Aktionen überführt und parametrisiert werden

Die gestaffelte Verarbeitung von Aufträgen geht einher mit einer gestaffelten Überwachung der Auftragserfüllung. *ReX* überwacht dabei die Umsetzung von Aktionen und Prozeduren, *COSA²* die von gewünschten Handlungen und der Operateur schließlich die Erfüllung der von ihm formulierten Auftragsagenda. Die Zyklusdauer der Überwachung durch *ReX* bzw. *COSA²* unterscheidet sich in einer Größenordnung. *ReX* realisiert eine Überwachungsfrequenz von 5Hz, während *COSA²* mit 0,5Hz Daten von seiner Eingabeschnittstelle liest. Dieser Unterschied spiegelt die Granularität des Überwachungsbereichs von *COSA²* und *ReX* wider. Neben der Aktionsüberwachung vollzieht *ReX* eine Vorverarbeitung von Informationen gemäß Abschnitt 4.1.2.1 und bietet *COSA²* abstrahierte Datensätze für dessen Führungsebene.

Technisch kommunizieren *COSA²* und *ReX* über eine Interprozesskommunikation (vgl. Abschnitt 4.5.1). *ReX* stellt dabei die äußere Schnittstelle des kognitiven Agenten dar (vgl. Abbildung 4-16). Für äußere Prozesse und den Operateur ist die innere Struktur somit gekapselt und nur ein Gesamtverhalten des Agenten beobachtbar.

4.2 Wissensmodellierung in *COSA²*

Die Wissensmodellierung in *COSA²* umfasst die a-priori Wissens Elemente des modifizierten Verhaltensmodells nach Rasmussen (vgl. Abschnitt 2.3.2) unter Nutzung des CML-Editors. Das resultierende Wissenspaket umfasst die Klassen *Concepts*, *Motivational Contexts*, *Task Options* und *Procedures*. Zusätzlich wird das *IO Interface* als Schnittstelle zur Kommunikation mit der reaktiven Erweiterung *ReX* und der *Function Server* für den regelbasierte Versand von Statusmeldungen modelliert.

4.2.1 Darstellung von situativem Wissen

Während der Laufzeit wird in COSA² situatives Wissen aus den a-priori Wissens-elementen instanziiert und in Form gerichteter, gewichteter Graphen hinterlegt [Brüggenwirth 2014]. Abbildung 4-6 zeigt eine Darstellung von situativem Wissen am Beispiel der internen Lagerepräsentation in COSA². Zwei Instanzen vom Typ *Point Of Interest* (POI), deren Attribute mit Werten belegt sind, werden über die gerichtete Kante *distanceToPOI* in Relation gesetzt.

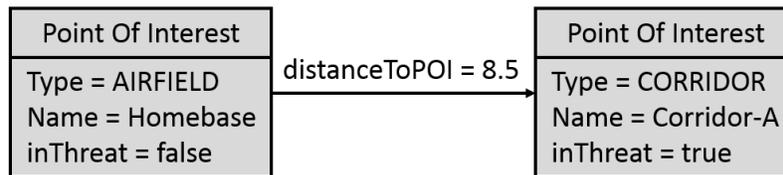


Abbildung 4-6: *Situatives Wissen in COSA² als gewichteter, gerichteter Graph nach [Brüggenwirth 2014]*

Eigenschaften und Beziehungen werden in COSA² als Kanten dargestellt. Kanten sind hierbei über einen Datentypen, einen Bezeichner, einen Ausgangs- und einen Zielknoten definiert. Im Beispiel handelt es sich um eine Beziehung vom Typ *distanceToPOI*, die die Distanz zwischen den Elementen als Fließkommazahl (Datentyp: *real*) abbildet. Die Kante hat den Wert 8.5 und zeigt vom POI mit dem Namen *Homebase* zum Zielknoten mit dem Namen *Corridor-A*. Attribute von Instanzen stellen intern Schleifen im Sinne von Kanten mit gleichem Ausgangs- und Zielknoten dar [Brüggenwirth 2014].

Zur Veranschaulichung der im Folgenden beschriebenen Modellierung von a-priori Wissen, wird exemplarisch situatives Wissen dargestellt, welches während der Laufzeit abgeleitet wird. Die Abbildungen sind dabei der Darstellung des COSA² Arbeitsspeichers aus der COSA²-Shell entnommen.

4.2.2 Concepts

In COSA² dienen Konzepte der Schaffung des internen Lagebildes und somit der Darstellung äußerer Objekte und Zustände. Eingehende Daten können zur Instanziierung neuer Konzepte führen oder bestehende Instanzen modifizieren. Konzepte werden dabei durch Attribute beschrieben und besitzen Regeln, die auf Attribute von Konzeptinstanzen angewendet werden können. Für das Wissensmodell der vorliegenden Arbeit wurden insgesamt sieben Konzepte mit insgesamt 50 Attributen und 71 Regeln in COSA² modelliert. Die Regeln gliedern sich in 54 *L1-Regeln*, acht *L2-Regeln* und neun *L3-Regeln*.

PointOfInterest (POI) beschreibt ein Element der taktischen Lage, mit dem das UAV im Rahmen der Mission interagieren kann. Jedes POI wird eindeutig über die Attribute *ID* und *Name*, die zwischen COSA² und ReX synchronisiert werden synchronisiert. Die Attribute *activeLink*, *onFoeSide*, *hostile* und *inThreat* sind boolesche Werte und treffen Aussagen über den Zustand des Objektes bzw. dessen Lage. Das Attribut *distanceToPOI* wird als Kante zwischen allen Instanzen von POIs erzeugt und gibt die Distanz zwischen zwei POIs als Fließkommazahl an.

Die interne Auftragsverarbeitung nutzt zudem die Attribute *isReconned*, *isScannedForPersons* und *isScannedForVehicles*, die bei der Instanziierung standardmäßig mit *false* belegt werden. Nach erfolgreichem Abschluss des Sensoreinsatzes an einem Lageelement wird das

entsprechende Attribut des POIs auf *true* gesetzt. Randbedingungen für die Auftragsumsetzung werden über die Attribute *specForDeparture* (Abheben), *specForLanding* (Landen), *specForIngress* (Einflug ins Zielgebiet), *specForEgress* (Ausflug aus dem Zielgebiet) und *specForTransmission* (Datenübertragung) gesetzt. Ein POI wird in einem Planungsschritt berücksichtigt, falls das mit der gewünschten Handlung assoziierte Attribut mit *true* belegt ist.

Die Art des POI wird über das Attribut *type* als Enum spezifiziert. *Type* kann einen der folgenden Werte annehmen:

POITYPE_UAV – Dieser Typ wird ausschließlich dem zum UAV gehörigen POI zugewiesen.

Zu Beginn der Laufzeit wird ein POI dieses Typs im Situationswissen instanziiert und dient fortan der Beschreibung der Lage des UAV relativ zu anderen POI.

POITYPE_CORRIDOR – Flugkorridore werden genutzt, um die Überquerung der *Forward Line of Own Troops* (FLOT) zu planen und durchzuführen. Durch die Attribute *specForIngress* und *specForEgress* wird die Nutzbarkeit der Korridore für den Einflug bzw. den Ausflug spezifiziert.

POITYPE_BUILDING – Gebäude sind als punktförmige Objekte hinterlegt, die angefliegen und aufgeklärt werden können.

POITYPE_AIRFIELD - Flugplätze dienen dem Starten und Landen des UAV. Die Spezifikation autorisierter Flugplätze erfolgt über die Attribute *specForDeparture* und *specForLanding*. Das Wissensmodell sieht ausschließlich das Starten und Landen von „nicht-feindlichen“ Flugplätzen (*hostile = false*) vor.

POITYPE_POINT – Punkte sind Lageelemente, die keine Repräsentation physischer Objekte in der realen Welt besitzen. Zumeist werden solche Punkte vom Operateur zur Markierung markanter Positionen verwendet.

POITYPE_ROAD – Über Straßen kann der Agent Aufklärung, Fahrzeug- und Personenerkennung durchführen. Das *inThreat* Attribut bezieht sich auf die gesamte geometrische Ausdehnung der Straße. Für die Bestimmung der Attribute *activeLink* und *onFoeSide* wird dabei der geometrische Schwerpunkt verwendet.

POITYPE_AREA – Gebiete repräsentieren geschlossene Flächen auf der Lagekarte und können äquivalent zu Straßen vom Agenten überfliegen und aufgeklärt werden.

POITYPE_VEHICLE – Instanzen von Fahrzeugen werden dynamisch im Arbeitsspeicher erzeugt, sobald sie im Rahmen einer Fahrzeugerkennung durch die Automation bzw. den Menschen erkannt und der Lage hinzugefügt wurden. Die Bewegung von Fahrzeugen ist über die veränderliche Position des Objekts in ReX modelliert.

POITYPE_PERSON – Personen werden äquivalent zu Fahrzeugen entweder durch einen Personenerkennungsalgorithmus oder manuell durch den Operateur dem Situationswissen hinzugefügt und werden durch punktförmige Objekte repräsentiert.

UAV repräsentiert den Zustand des UAV im Missionskontext. Die Konzeptklasse existiert als Singleton Instanz im Arbeitsspeicher von COSA² und wird beim Start initialisiert. Die Attribute von *UAV* stellen die symbolische Repräsentation der aktuellen Handlungen der UAV-

Automation und deren Zustand dar. Die Regeln in *UAV* modellieren das Verhalten der Automationskomponenten des UAV bzw. die Auswirkungen des Automationseinsatzes auf den Zustand des UAV und andere POIs.

Über das Attribut *remainingFuel* wird die verfügbare Treibstoffmenge des UAV angegeben. Der Treibstoff des UAV wird als Teil des UAV-Status durch ReX übertragen. Bei der Planung optimiert COSA² den Treibstoffverbrauch in Abhängigkeit von der zurückgelegten Distanz. Die Attribute *departingLanding* und *airborne* beschreiben, ob sich das UAV aktuell in der Luft bzw. im Start- oder Landevorgang befindet.

Das Attribut *poiOriginID* verweist im Transitflug zwischen zwei POI auf das Startelement, während *poiGoalID* das Zielelement beschreibt. Befindet sich das UAV nahe eines POI, verweist die Kante *closeTo* auf dieses Objekt.

Die Attribute *reconOn*, *scanningForVehicles* und *scanningForPersons* sind als boolesche Werte modelliert und beschreiben die aktuelle Aufklärungstätigkeit des UAV. *NumberOfRecords* gibt die Anzahl der an Bord aufgezeichneten Datenströme als *int* an. Daten werden übertragen, wenn das Attribut *transmissionOn* auf *true* gesetzt wird. Das Attribut *hasRecceData* wird auf *true* gesetzt, wenn *numberOfRecords* größer null ist.

Das Beispiel in Abbildung 4-7 zeigt den Zustand des UAV *Explorer* im Anflug auf einen Flugplatz und dessen Abstand zur Zielposition. Zur Abbildung dienen Instanzen der Klassen *UAV* und *PointofInterest* für den Zustand des UAV und eine weitere Instanz der *PointOfInterest* Klasse für den Flugplatz. Der Anflug des UAV wird über das Attribut *poiGoalID* = 100 definiert. Die Distanz zwischen dem UAV und dem Flugplatz ist als Kante zwischen den *POI* Instanzen mit *distanceToPOI* = 612,95m angegeben. Während der Transition zwischen zwei taktischen Elementen zeigt die positive *closeTo* Kante von *UAV* stets auf die eigene *POI* Instanz.

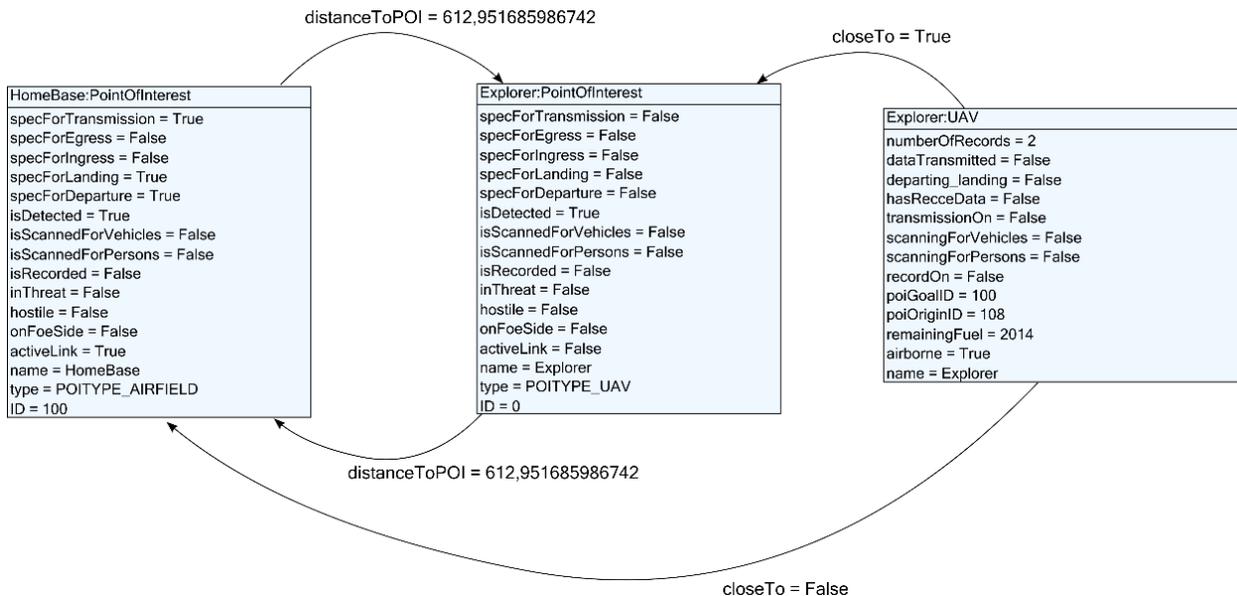


Abbildung 4-7: Repräsentation des Zustands des UAV Explorer im Zielanflug auf einen Flugplatz

Task spezifiziert den aktuellen Auftrag und ist über den Auftragsstypen (*type*) und die ID des Elementes (*poiID*) spezifiziert, auf die der Auftrag angewendet werden soll. Über das Attribut

taskAccomplished ist der Auftragszustand als *bool* modelliert. Ein eingehender Auftrag erzeugt eine *Task* Instanz im Arbeitsspeicher. Ziel ist es fortan, das Attribut *taskAccomplished* des Auftrags auf *true* setzen zu können und den Auftrag damit abzuschließen (vgl. Abschnitt 4.2.3). Das Konzept umfasst die Auftragsstypen und Regeln als Bedingungen für ihre jeweilige Erfüllung. Weitere Regeln setzen Randbedingungen für die Auftragsbefreiung durch Modifikation der Attribute in den POI Instanzen um.

Im Wissensmodell sind folgende Auftragsstypen über das Attribut *type* spezifiziert:

TASKTYPE_DEPART – Der Auftrag *depart* fordert den Start des UAV. Der Abschluss des Startvorgangs ist über das Attribut *airborne* modelliert (*UAV: airborne = true*). Über *poiID* kann der Flugplatz spezifiziert werden, von dem aus das UAV starten soll.

TASKTYPE_LAND – Der Auftrag *land* fordert die Landung des UAV auf einem Flugplatz. Analog zum *depart* Auftrag, gilt *land* als erfüllt, wenn das UAV gelandet ist (*UAV: airborne = false*). Das Attribut *poiID* spezifiziert den Flugplatz für die Landung. Verweist das Attribut *poiID* auf kein POI (*poiID = -1*), so kann die Landung auf einem beliebigen Flugplatz erfolgen.

TASKTYPE_REACH_POI – Der Auftrag fordert den Anflug des durch die *poiID* spezifizierten POI (*UAV: closeTo [POI: ID = poiID] = true*). Hierbei ist auch bei Flugplätzen der reine Anflug gemeint, bei dem sich das Luftfahrzeug immer noch in der Luft befindet (*UAV: airborne = true*). Das Wissensmodell erlaubt dabei ausschließlich den Anflug solcher POI, die nicht durch Feindkräfte bedroht werden (*POI: inThreat = false*).

TASKTYPE_RECON_POI – *Recon POI* fordert die Aufklärung des über die *poiID* spezifizierten POI (*POI: ID = poiID*). Die Aufklärung gilt als erfolgt, wenn das Attribut *isReconned* des POIs auf *true* gesetzt wird. Die Form der Aufklärung wird in COSA² nicht berücksichtigt, sondern durch ReX verarbeitet. Gemäß der Verfügbarkeit des Datenlinks werden die gewonnenen Aufklärungsdaten an Bord gespeichert (*UAV: numberOfRecords += 1*) oder direkt an die Bodenkontrollstation übermittelt.

TASKTYPE_SCAN_POI_FOR_VEHICLES – Der Auftrag gilt als ausgeführt, wenn das spezifizierte POI nach Fahrzeugen abgesehen wurde (*POI: isScannedForVehicles = true*). Hat der Operator die manuelle Auftragsunterstützung spezifiziert (*manual sensorGuidance = true* oder *manual vehicleDetection = true*), so kann der Auftrag nur bei aktivem Datenlink erfolgen. Die Erfüllung des Auftrags ist unabhängig von der Anzahl der dabei detektierten Fahrzeuge.

TASKTYPE_SCAN_POI_FOR_PERSONS – Der Auftrag zur Personenerkennung wird analog zur Fahrzeugerkennung verarbeitet.

TASKTYPE_TRANSMIT_ALL_DATA – Der Auftrag fordert die Übertragung aller aufgezeichneten Videodaten an die Bodenkontrollstation (*UAV: hasRecceData = false*). Bei Spezifikation der *poiID* (*poiID != -1*) muss die Datenübertragung an dem entsprechenden POI erfolgen. Hier ist die Verfügbarkeit des Datenlinks am POI vorausgesetzt. Für die erfolgreiche Auftragsbefreiung müssen alle Aufzeichnungen übermittelt werden (*UAV: nrOfRecords = 0*).

Flightplan wird von COSA² verwendet, um den Zustand der Flugroute des UAV zu beschreiben. Dieser Zustand wird ausgedrückt durch die Attribute *valid* und *inThreat*, die die Integrität der Route und ihre aktuelle Bedrohung beschreiben. Kommandiert der Agent das Abfliegen einer Route, so wird diese aktiviert (*Flightplan: active = true*). Für die Auftragserfüllung wird grundsätzlich eine integre und unbedrohte Flugroute vorausgesetzt. Dies wird über die Vorbedingungen der *Task Options* modelliert (vgl. Abschnitt 4.2.4). Während der Erstellung der Flugrouten, besitzt das Attribut *updating* den Wert *true*.

Eine Instanz von *Flightplan* wird als Singleton Instanz zu Laufzeitbeginn im Arbeitsspeicher von COSA² initialisiert. Über den Missionsverlauf werden lediglich die Attribute dieser Instanz modifiziert. COSA² hält keine geometrische Repräsentation der Flugroute vor, sondern verwaltet sie lediglich über den Zustand der symbolischen Attribute.

Capabilities dient der Beschreibung der Fähigkeiten des UAV. UAV-Fähigkeiten werden in ReX abstrahiert und mit COSA² als boolesche Aussagen synchronisiert (vgl. Abschnitt 4.1.2).

atol_Available – Verfügbarkeit einer automatischen Start- und Landefähigkeit

routeplanning_Available – Verfügbarkeit der Routenplanungsfähigkeit

waypointMode_Available – Verfügbarkeit der Fähigkeit, automatisiert Wegpunkte zu verfolgen

recording_Available – Verfügbarkeit der Sensorik und Sensordatenaufnahme

vehicleDetection_Available – Verfügbarkeit der Fähigkeit zur Fahrzeugerkennung

personDetection_Available – Verfügbarkeit der Fähigkeit zur Personenerkennung

dataTransmission_Available – Verfügbarkeit der Fähigkeit zur Datenübertragung an die BKS

Status beschreibt den Prozessstatus von COSA² und dient in Kombination mit ReX zur Überwachung und Steuerung der Systemarchitektur. Hierfür ist das Attribut *heartbeat* modelliert, welches bei aktivem Verarbeitungszyklus von COSA² den Wert *true* hält. Auf Anfrage durch ReX sendet COSA² ein PING-Signal, welches bestätigt, dass die Systemarchitektur wie erwartet funktioniert. Dieses Signal ist über direkten Zugriff auf den *Function Server* von COSA² realisiert und geschieht regelbasiert, unabhängig von der aktuellen Auftragserfüllung.

Von *Status* wird zu Laufzeitbeginn eine Singleton Instanz initialisiert, die der reinen Reaktion auf Anfragen durch ReX dient.

MissionContent dient als Hilfskonzept zur korrekten Abbildung des Attributs *nrOfRecords* der *UAV*-Instanz. Als technische Notwendigkeit wird in *MissionContent* das Attribut *nrOfRecordsAfterTransmit* vorgehalten. Dieses Attribut beschreibt die Anzahl der Aufnahmen, die im UAV verbleiben, nachdem eine Datenübertragung gemäß der Task Option *TransmitData* abgeschlossen ist. Nach abgeschlossener Datenübertragung wird *nrOfRecords* auf den Wert von *nrOfRecordsAfterTransmit* gesetzt. *nrOfRecordsAfterTransmit* wird vor der nächsten Datenübertragung entsprechend der zu übertragenden Daten angepasst.

Zu Laufzeitbeginn wird von COSA² eine Singleton Instanz von *MissionContent* initialisiert, in der das Attribut *nrOfRecordsAfterTransmit* auf -1 gesetzt wird. Bei der Speicherung von Videodaten wird der Wert zur Laufzeit angepasst.

4.2.3 Motivational Contexts

Zur Auftragsverfolgung erzeugt COSA² Instanzen von Zielen gemäß definierter Regeln, die in Form von *Motivational Contexts* im Wissensmodell hinterlegt sind. Gemäß ihres Typs umfassen die Zielinstanzen dabei ihrerseits Regeln für die Erfüllung der Ziele.

Das Wissensmodell der vorliegenden Arbeit umfasst eine einzelne Klasse in *Motivational Contexts*, über die eingehende Aufträge direkt durch COSA² verarbeitet werden. Das zusätzliche, auftragsübergreifende Ziel der Bedrohungsvermeidung ist nicht explizit als Klasse im *Motivational Context* modelliert, sondern als Vorbedingung der *Task Options* realisiert (vgl. Abschnitt 4.2.4).

Task Goal – Die Erteilung eines neuen Auftrags durch ReX instanziiert in COSA² das interne Ziel zur Auftragsbefreiung. Dieses Ziel ist definiert über die Erfüllung aller im Wissensmodell spezifizierter Vorbedingungen und damit der Belegung des Attributs *taskAccomplished* der *Task* Instanz mit *true* (vgl. Abschnitt 4.2.2). Bei der Instanzierung des Ziels wird auch die Metrik der Optimierungsfunktion für den PDDL-Planer spezifiziert.

Abbildung 4-8 zeigt den Zusammenhang zwischen Auftrag und instanziiertem Ziel im Arbeitsspeicher von COSA². Über die *achieveTask* Regel definiert COSA² die Auftragsumsetzung als Zielsetzung des eigenen Handelns. Für den Auftrag sind der Auftragsstyp (*type*) und das Auftragsselement (*poiID*) als Attribute von *Task* hinterlegt. Die Metrik der Optimierungsfunktion ist über $Max(P+C, UAV.remainingFuel = 10000)$ angegeben und symbolisch mit einem Objekt für die externe Planung verbunden. Die dargestellte Metrik wird einheitlich im Rahmen dieser Arbeit verwendet und meint die Maximierung des verbleibenden Treibstoffs des UAV (UAV: *remainingFuel*) ausgehend von der aktuellen Treibstoffmenge.

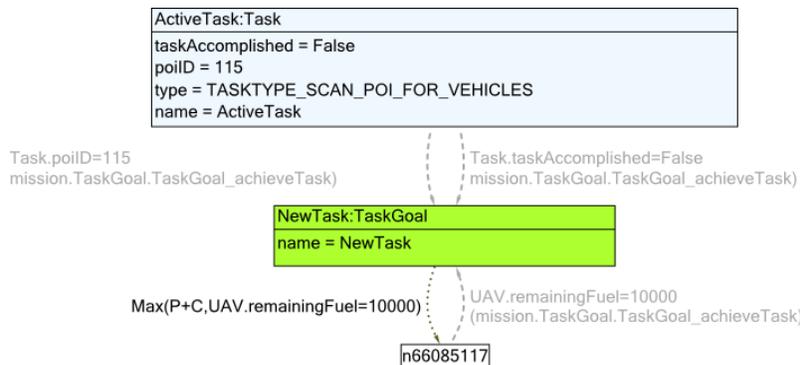


Abbildung 4-8: Repräsentation eines Auftrags und des daraus instanziierten Ziels in COSA². Neben den Eigenschaften von Auftrag und Zielelement wird auch die Optimierungsfunktion spezifiziert

4.2.4 Task Options

Um vorgegebene Ziele zu erreichen, werden in COSA² Handlungsoptionen zu einer Handlungsagenda kombiniert und sind einzeln als Klassen in den *Task Options* modelliert. Die Modellierungsform der Handlungsoptionen ist dabei ähnlich der Modellierung von Regeln in Konzepten, bei der Vorbedingungen mit Nachbedingungen über Handlungen gekoppelt werden.

Nachbedingungen von Handlungsoptionen können dabei auf zwei unterschiedliche Arten modelliert werden. Zum einen können Attribute von Konzeptinstanzen durch Handlungen direkt modifiziert werden. Alternativ initiiert COSA² durch Handlungen die Ausführung externer

Prozesse und modifiziert erst auf Grundlage eingehender Statusdaten die Attribute der Konzeptinstanzen. Die Prädiktion des externen Verhaltens erfolgt dabei über die Modellierung entsprechender *L3-Regeln* in den Konzeptklassen von *POI* bzw. *UAV*. Die externen Prozesse werden dabei über die Ausgangsschnittstelle von COSA² kommandiert (vgl. Abschnitt 4.2.5).

Handlungsoptionen sind dabei fest an die Verfügbarkeit von UAV-Fähigkeiten gebunden. Jede Option bedarf spezifischer Fähigkeiten des UAV, um ausgewählt werden zu können. Die Verfügbarkeit wird hierbei aus der *Capabilities* Instanz im Arbeitsspeicher von COSA² ausgelesen. Die konkret benötigten Fähigkeiten sind im Folgenden für die *Task Options* vermerkt.

Regeln zur Flugsicherheit sind in COSA² als Vorbedingungen der *Task Options* modelliert. Alle Handlungsoptionen verlangen, dass das UAV und die Flugroute unbedroht sind. Ausgenommen ist hiervon die Handlungsoption *CreateFlightplan*, die COSA² zur Erstellung einer integren Flugroute für das UAV nutzt.

TakeOff – Die Handlungsoption dient dem Start des UAV. Unter der Voraussetzung eines integren Flugplans (*Flightplan: valid = true*) und der Verfügbarkeit einer ATOL-Fähigkeit (*Capabilities: atol_Available = true*), wird der Start des UAV initiiert (*UAV: departingLanding = true*). Ist der Start vollzogen, erhält COSA² das entsprechende Signal durch ReX (*eventInput: eventType = EVENTTYPE_DEPARTED*), wodurch die Handlung als abgeschlossen gilt.

Land – *Land* führt zur Landung des UAV am spezifizierten Flugplatz (*POI: specForLanding = true; hostile = false*). Hierfür muss sich das UAV am entsprechenden Flugplatz befinden (*UAV: closeTo[POI] = true*) und über eine ATOL-Fähigkeit verfügen (*Capabilities: atol_Available = true*). Die Landung wird analog zum *TakeOff* initiiert und bei Erhalt des Signals (*eventInput: eventType = EVENTTYPE_LANDED*) abgeschlossen.

ReconPOI – Die Aufklärung von POI erfolgt über *ReconPOI*. Hierzu muss sich das UAV am POI befinden (*UAV: closeTo[POI] = true*) und entweder die Fähigkeit zur Aufzeichnung der Daten (*Capabilities: recording_Available = true*) oder zur Live-Übertragung (*Capabilities: dataTransmission_Available = true*) besitzen. Der Sensor bleibt aktiviert (*UAV: recordOn = true*), bis ReX den Abschluss der Aufklärung meldet (*eventInput: eventType = EVENTTYPE_POI_DATA_COLLECTED*). Bei der Aufzeichnung von Videodaten wird ein zusätzliches Signal (*eventInput: eventtype = EVENTTYPE_DATA_STORED*) empfangen und die Anzahl der Aufzeichnungen (*UAV: numberOfRecords += 1*) angepasst.

ScanPoiForVehicles – Der Vorgang der Fahrzeugerkennung wird in COSA² über die Handlung *ScanForPOIVehicles* umgesetzt. Als Voraussetzung muss sich das UAV an dem zu scannenden POI befinden (*UAV: closeTo[POI] = true*) und über die Fähigkeit zur Fahrzeugerkennung verfügen (*Capabilities: vehicleDetection_Available = true*). Im Anschluss an die Kommandierung befindet sich das UAV im Scanvorgang (*UAV: scanningForVehicles = true*), bis ReX das Terminierungssignal (*eventInput: eventType = EVENTTYPE_POI_SCANNED*) emittiert. Als Resultat wird das POI entsprechend markiert (*POI: isScannedForVehicles = true*).

ScanPoiForPersons – Der Vorgang der Personenerkennung ist in COSA² analog zur Fahrzeugerkennung modelliert, wobei als Voraussetzung die Fähigkeit zur Personenerkennung (*Capabilities: personDetection_Available = true*) benötigt wird. Die Markierung des gescannten POIs erfolgt ebenfalls analog (*POI: isScannedForPersons = true*).

FlyToPOI – Die Handlung *FlyToPOI* wird zum Anflug von POI bzw. der Transition zwischen verschiedenen POI verwendet. Bei integrem Flugplan (*Flightplan: valid = true*) wird das UAV von seiner derzeitigen Position zu dem spezifizierten Ziel POI kommandiert, solange dieser nicht bedroht ist (*POI: inThreat = false*). Das UAV darf ein POI nur dann direkt anfliegen, wenn es entweder auf der gleichen Seite der FLOT liegt, oder Start bzw. Ziel einen Korridor darstellen (*POI: type = POITYPE_CORRIDOR*). Die Korridorauswahl erfolgt dabei anhand der spezifischen Randbedingungen des Auftrags (*POI: specForIngress = true; specForEgress = true*). *FlyToPOI* benötigt die Fähigkeit zur wegpunktbasierten Flugführung (*Capabilities: waypointMode_Available = true*). Sobald das UAV das spezifizierte POI erreicht (*eventInput: eventType = EVENTTYPE_POI_REACHED*), wird dessen Position angepasst (*POI: closeToPOI = poiID*).

TransmitData – Gespeicherte Daten können mittels *TransmitData* vom UAV an die Bodenkontrollstation übertragen werden. Als Vorbedingung werden hierzu aufgezeichnete Daten (*UAV: hasRecceData = true*) und die Fähigkeit zur Datenübertragung (*Capabilities: dataTransmission_Available = true*) benötigt. Der Ort der Datenübertragung kann über ein POI spezifiziert werden (*UAV: specForTransmission = true*), wenn dessen Position die Datenübertragung erlaubt (*POI: activeLink = true*). Der Abschluss der Datenübertragung (*UAV: transmissionOn = false*) wird durch ReX signalisiert (*eventInput: eventType = EVENTTYPE_POI_DATA_TRANSMITTED*).

CreateFlightplan – Die Handlung *CreateFlightplan* dient der Erzeugung eines Flugplans, der die geometrischen Randbedingungen im Missionsgebiet berücksichtigt. Als Fähigkeit wird hierzu die automatisierte Routenplanung vorausgesetzt (*Capabilities: routePlanning_Available = true*). Nach der Kommandierung des Routenplaners (*Flightplan: updating = true*) erwartet COSA² die Meldung der erfolgreichen Flugplanerstellung durch ReX (*eventInput: eventType = EVENTTYPE_FLIGHTPLAN_CREATED*). Der erfolgreich erstellte Flugplan gilt anschließend als integer und unbedroht (*Flightplan: valid = true; inThreat = false*).

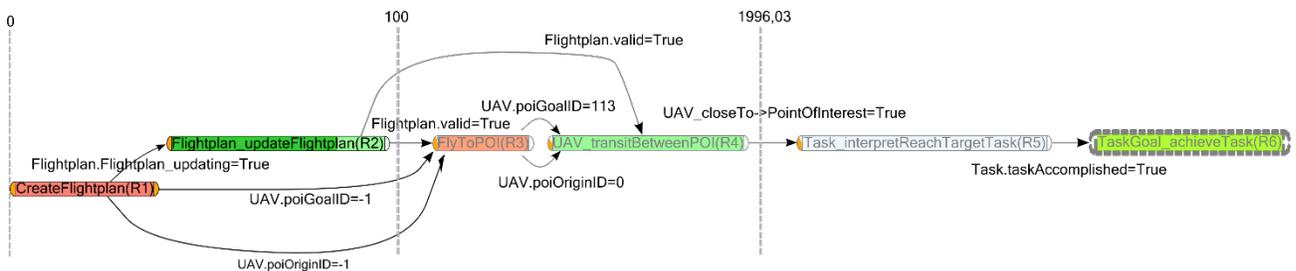


Abbildung 4-9: In COSA² generierte Handlungsagenda zum Anfliegen eines POI im Zielgebiet

Abbildung 4-9 zeigt eine simplifizierte, in COSA² generierte Handlungsagenda zum Anflug des POI 113. Durch die Handlungen (rot) *CreateFlightplan* und *FlyToPOI* werden die Zielbedingungen (*UAV: closeTo[poiID = 113] = true*) der Regel *interpretReachTargetTask* erfüllt und damit der aktuelle Auftrag umgesetzt. Der Handlungsplan enthält *L3-Regeln* (grün) zur Abbildung der erwarteten externen Aktionen. Diese Regeln definieren über ihre Nachbedingungen die Zustände zum Weiterschalten des Plans während der Laufzeit.

4.2.5 Procedures

Prozeduren sind in COSA² direkt an die *Task Options* gekoppelt. Das bedeutet, dass für jede in Abschnitt 4.2.4 genannte Handlungsoption genau eine Prozedur existiert. Die Prozeduren werden dabei in COSA² so modelliert, dass sie das von COSA² gewünschte Verhalten des UAV in C#-Code überführen und über die Ausgangsschnittstelle an ReX übertragen, wo anschließend die regelbasierte Auswahl der Prozeduren erfolgt.

4.2.6 I/O Interface

Das *I/O Interface* stellt die Schnittstelle von COSA² zur reaktiven Erweiterung ReX dar. Die Implementierung umfasst die Spezifikation aller eingehenden und ausgehenden Nachrichten (*Input / Output*) in Form von Klassen. Nachrichten nutzen dabei die gleichen Datentypen wie Konzepte und Handlungsoptionen.

telemetryInput – Überträgt den symbolischen UAV-Status von ReX an COSA².

poiInput – Umfasst Informationen zu den POI Instanzen, die über das Attribut *id* zugeordnet werden. Über die Nachricht können neue POI hinzugefügt oder bestehende POI editiert und gelöscht werden, um die Lagerepräsentation zwischen ReX und COSA² zu synchronisieren.

poiDistanceInput – Spezifiziert die Distanz zwischen zwei POI, deren Instanzen über die Attribute *poiOriginID* und *poiGoalID* adressiert werden. Die Distanz wird als Fließkommazahl über das Attribut *distance_m* als Distanz in Metern übermittelt.

eventInput – Überträgt Signale eines bestimmten Typs (*eventInput: eventType*) mit Bezug auf ein POI (*eventInput: poiID*). COSA² interpretiert die Signaltypen über das Enum *E_EventType*:

taskInput – Dient der Übermittlung von Aufträgen an COSA². Ein Auftrag wird über die *poiID* und seinen Auftragsstyp (*taskInput: taskType*) spezifiziert, welcher von COSA² über das Enum *E_TaskType* interpretiert wird.

constraintsInput – Diese Nachricht kapselt auftragsspezifische Randbedingungen. Randbedingungen sind definiert mit Bezug auf ein POI (*constraintsInput: poiID*) und die relevante Handlung, definiert über das Enum *E_ConstraintActionType*. Randbedingungen der Konzeptklasse *POI* sind in Abschnitt 4.2.2 beschrieben.

heartbeatInput – Dient der Abfrage des Prozessstatus von COSA² durch ReX.

capabilitiesInput – Synchronisiert die UAV-Fähigkeiten zwischen ReX und COSA² bezüglich ihrer Verfügbarkeit gemäß der Konzeptklasse *Capabilities* (vgl. Abschnitt 4.2.2).

taskAgendaOutput – Übermittelt die Handlungsagenda von COSA² an ReX in Form einzelner, sequenzieller Handlungen. Jede Handlung ist über den Prozedurentyp (*taskAgendaOutput: procedureType*) und die *poiID* modelliert und wird in ReX empfangen und zu einer einheitlichen Handlungsagenda zusammengefügt.

commandOutput – Dient der Kommandierung von Prozeduren als gewünschtes Verhalten an ReX. Dies erfolgt über den Prozedurentyp (*commandType: procedureType*) und eine *poiID*. ReX gleicht die Prozedur mit der hinterlegten Handlungsagenda ab und setzt sie regelbasiert um.

taskOutput – In *taskOutput* wird der aktuelle Auftragsstatus an ReX übermittelt. Der Auftragsstatus dient ReX zur Weiterschaltung von Aufträgen und ist spezifiziert über das Enum *E_TaskActionType*.

cosaStatusOutput – Der Prozessstatus von COSA² dient ReX zur Kontrolle der internen Kommunikation des kognitiven Agenten und der Statusmeldung an den Operateur. Der Prozessstatus dient ReX zudem zur Prozesssteuerung von COSA² und der Detektion von Planungsfehlern.

Node:InData poiDistanceInput_poiGoalID = 100 poiDistanceInput_poiOriginID = 0 poiDistanceInput_distance_m = 3612,9516	Node:InData telemetryInput_closeToPOI = 108 telemetryInput_remainingFuel = 18451 telemetryInput_airborne = True telemetryInput_inThreat = False telemetryInput_activeLink = True telemetryInput_uavName = ExplorerUAV	Node:InData poiDistanceInput_poiGoalID = 108 poiDistanceInput_poiOriginID = 100 poiDistanceInput_distance_m = 525,8295
--	---	---

Abbildung 4-10: Eingehende Nachrichten im I/O Interface von COSA²

Abbildung 4-10 zeigt beispielhafte Nachrichten, die COSA² über das I/O Interface empfängt. Neben dem UAV Zustand in Form von Telemetriedaten werden Distanzangaben zwischen POI von ReX übertragen. Die eingehenden Nachrichten werden nach Erhalt über *LI-Regeln* der Konzeptklassen verarbeitet und ihre Informationen in die entsprechenden Instanzen überführt.

4.2.7 Function Server

Der *Function Server* ermöglicht es, mittels COSA², unabhängig von der aktuellen Handlungsagenda, einzelne Aktionen aufzurufen und über die I/O Schnittstelle an ReX zu kommandieren. Dies ermöglicht das regelbasierte Versenden von Nachrichten sowie die Reaktion auf interne Systemzustände, unabhängig vom regulären Verarbeitungsprozess. Die Modellierung des regelbasierten Verhaltens ist dabei nicht über den CML-Editor möglich, sondern erfolgt durch manuelle Programmierung in C#.

sendHeartbeat – Diese Funktion dient der kontinuierlichen Statusmeldung an ReX. Der Status wird als direkte Antwort auf die *heartbeatInput* Nachricht von ReX in Form der *cosaStatusOutput* Nachricht versendet. Ist der Verarbeitungszyklus durch eine Fehlplanung unterbrochen, wird die Funktion nicht durchlaufen und keine Statusmeldung gesendet.

sendTaskComplete – Über die *sendTaskComplete* Funktion wird der erfolgreiche Auftragsabschluss über Nachricht *taskOutput* an ReX gesendet.

4.3 Automationskomponenten des UAV

Im Folgenden werden die vier für diese Arbeit wesentlichen Automationskomponenten als separate, vom kognitiven Agenten entkoppelte Prozesse beschrieben. Die Komponenten dienen dabei der Abbildung des Funktionsumfangs des UAV gemäß Abbildung 3-3. Dieser umfasst Automation zur Routenplanung, zur plattformunabhängigen Flugführung, der Sensorik und der Sensordatenverarbeitung sowie zur Konfiguration und Nutzung des Datenlinks. Neben der Kommandierung durch den Agenten erlauben die Prozesse alternativ und parallel die manuelle Kontrolle durch den Operateur. Neben der Funktionsbeschreibung der Automationskomponenten

werden im Folgenden die spezifischen Schnittstellen zur Kommunikation mit dem kognitiven Agenten über ReX sowie der daraus resultierende Informationsfluss beschrieben.

4.3.1 *Automation Abstraction Layer (AAL)*

Das *Automation Abstraction Layer (AAL)* ist als Software in C++ implementiert und dient der Bereitstellung von FMS Funktionalitäten über eine standardisierte Schnittstelle. AAL erlaubt die Flugsteuerung des UAV auf der Ebene von Wegpunkten und Routen sowie über Autopilotenkommandos. Zur Überwachung von Flugzustands- und Systemdaten stellt AAL standardisierte Datenformate zur Ausgabe bereit. AAL ermöglicht die Anbindung verschiedener FMS in Form von Hardware- und Softwarekomponenten, unter Beibehaltung des verfügbaren Fähigkeitsumfangs.

Für die Flugführung realisiert AAL, unabhängig vom verbundenen FMS, eine interne Wegpunkt- und Routenverwaltung und überträgt erst bei Aktivierung der Wegpunkte und Routen durch den Nutzer (hier der Operateur oder der kognitive Agent) die Daten an das jeweilige Endgerät. AAL übernimmt dabei die Umsetzung der Wegpunkt- und Routeninformationen in die entsprechenden Endformate und überwacht Routenevents (wie z.B. das Erreichen von Wegpunkten) zur Weitergabe an den Nutzer. Alternativ zu Wegpunkten stellen Autopilotenkommandos den direkten Durchgriff auf das Endgerät durch den Nutzer bereit.

AAL umfasst drei Module:

1. Die Remote-Schnittstelle **AAL Com** erlaubt den Zugriff auf AAL Funktionen von anderen Applikationen auf dem lokalen Rechner oder von extern angebundenen Systemen. Die Schnittstelle implementiert die automatische Synchronisation von Wegpunkt- und Routendaten sowie die Übertragung von Telemetrie- und Systemstatusdaten des FMS-Endgeräts. Die Datenkommunikation in der Remote-Schnittstelle wird über Interprozesskommunikation realisiert.
2. Wegpunkt- und Routeninformationen werden intern im **AAL Data Management (ADM)** gespeichert und verarbeitet. Der Nutzer kann Daten dynamisch hinzufügen, editieren oder entfernen. Die Datenverwaltung geschieht dabei unabhängig vom FMS-Endgerät und erlaubt beliebige Größen sowohl für die Anzahl der Routen als auch für die jeweilige Routenlänge.
3. Der Endgeräte-Treiber (**Backend Driver**) verbindet das jeweilige FMS mit AAL. Der Treiber übersetzt Kommandos des Nutzers sowie automatische AAL-Aktionen in das FMS-spezifische Format, welches über eine entsprechende Hardware- oder Softwareverbindung übertragen wird. Der Treiber setzt dabei auch das standardisierte Routenverhalten und die Kommandos der Remote-Schnittstelle in FMS Kommandos um. Schließlich regelt der Treiber den Empfang der Telemetrie- und Systemstatusinformationen vom Endgerät und überführt sie in ein standardisiertes Format.

Die Treiber-Architektur ist in Form einer Fabrikmethode (*Factory Method*) implementiert, die das dynamische Laden des Treibers für das Endgerät möglich macht [Gamma et al. 2011]. Diese Implementierung erlaubt ein Höchstmaß an Entkopplung zwischen der AAL-Software und der technischen Realisierung des angebundenen FMS Systems. Mit AAL können FMS sowohl für Starr- als auch für Drehflügler angebunden werden.

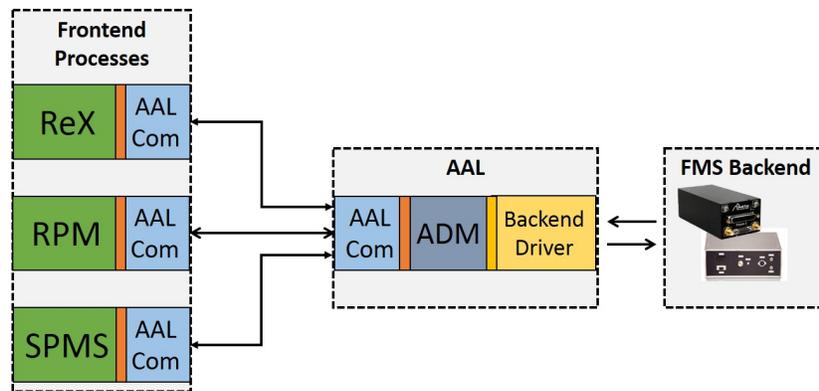


Abbildung 4-11: Schematische Darstellung der Komponenten des Automation Abstraction Layers (AAL) zur Abstraktion von FMS Funktionalitäten für Prozesse an Bord des UAV

Mittels AAL können sich mehrere Nutzer gleichzeitig mit einem FMS-Endgerät verbinden. Die Synchronisation von Wegpunkt- und Routendaten schafft dabei eine gemeinsame Repräsentation aktueller Flugpläne und des Flugzugstands. Abbildung 4-11 zeigt die Prozesse an Bord des UAV, die im Rahmen dieser Arbeit über AAL Zugriff auf das FMS haben. Diese umfassen den kognitiven Agenten über ReX, das Routenplaner Management (RPM) sowie das Sensor- und Perzeptionsmanagement (SPMS). AAL dient im Rahmen dieser Arbeit zur Kommandierung eines einzelnen FMS, bietet jedoch prinzipiell die Möglichkeit zur Verwaltung mehrerer Endgeräte.

4.3.1.2 AAL-Schnittstelle in ReX

AAL ermöglicht dem Agenten die Kommandierung von Wegpunkten, das Abfliegen von Routen sowie den Start und die Landung des Luftfahrzeugs. Das Abfliegen von Routen erfolgt dabei unter Nutzung des über die RPM-Schnittstelle generierten Flugplans (vgl. Abschnitt 4.3.2). Die AAL-Schnittstelle in ReX überwacht die Aktionsdurchführung und verarbeitet Telemetrie- und Systemzustandsdaten, die über AAL bereitgestellt werden. Die über AAL bereitgestellten Funktionen des FMS dienen der Ableitung von Flugführungsfähigkeiten des UAV in ReX und werden u.a. aus dem Typ des Luftfahrzeugs ausgelesen. Folgende Aktionen kann der Agent über die AAL-Schnittstelle abrufen:

FLY_TO_POSITION – Erzeugt einen neuen Wegpunkt an der spezifizierten Position und kommandiert den Anflug des Wegpunktes. Die Position ist dabei frei parametrisierbar und kann unabhängig von Lageelementen gewählt werden. Die Aktion wird z.B. beim reaktiven Ausweichen bei Gefährdung genutzt.

FLY_ROUTE – Wählt einen bestehenden Flugplan in AAL und kommandiert dessen Abflug von einem Startwegpunkt bis zu seinem Ende. Der Startwegpunkt wird dabei anhand der aktuellen Position des UAV, des Routenbeginns oder eines gewünschten Einstiegswegpunkts spezifiziert.

FLY_TO_ELEMENT – Kommandiert den Abflug eines bestehenden Flugplans von einem Startwegpunkt bis zum Erreichen eines spezifizierten Lageelementes. Die Zuordnung des Lageelementes zu den Wegpunkten des bestehenden Flugplans erfolgt intern im ADM.

DEPART – Dient der Kommandierung des Startvorgangs des Luftfahrzeugs, falls das UAV über eine ATOL Fähigkeit verfügt. Die Parametrisierung des Startvorgangs erfolgt über die Vorgabe

von Endbedingungen (Kurs, Höhe, Geschwindigkeit), die sich aus dem aktiven Flugplan des UAV ergeben.

LAND – Dient der Kommandierung des Landevorgangs des Luftfahrzeugs, falls das UAV über eine ATOL Fähigkeit verfügt. Die Parametrisierung der Landung erfolgt über die Daten des Lageelementes (zumeist des Flugplatzes), auf dem die Landung vollzogen werden soll.

4.3.2 Routenplaner Management (RPM)

Die Flugführung des UAV erfolgt prinzipiell routenbasiert, d.h. über die Formulierung von Routen, die das Luftfahrzeug von seiner Ausgangsposition in eine Zielposition überführen. Zur automatisierten Erstellung von Flugrouten existieren verschiedene Routenplaner, die sich in Performanz und Funktionsumfang teilweise stark unterscheiden. Routenplanung kann generell zerlegt werden in die Überführung des realen Fluggebietes in einen Suchraum und die anschließende Bestimmung einer Route im Suchraum [Giesbrecht 2004]. Die Routenbestimmung geht dabei meist einher mit einer Optimierung der Route entsprechend einer Optimierungsfunktion. Die Eigenschaften und Funktionen der Routenplaner werden dabei durch die Algorithmen bestimmt, durch die sie implementiert werden. Hierbei kann kein Algorithmus definiert werden, der für alle Anwendungen den optimalen Routenplaner bereitstellt. Vielmehr eignen sich unterschiedliche Routenplaner für verschiedene Applikationen [Giesbrecht 2004]. Das *Routenplaner Management* (RPM) stellt eine standardisierte Schnittstelle für den Zugriff auf unterschiedliche Routenplaner Algorithmen bereit.

Durch RPM können Routen mit einer beliebigen Anzahl von Stützstellen, spezifiziert über Position und Kurs des UAV, generiert werden. Im Rahmen dieser Arbeit implementiert RPM vier verschiedene Algorithmen zur Routenplanung, die jeweils zwischen zwei Stützstellen optimale Routen generieren und damit für die Gesamtroute teil-optimale Lösungen erreichen. Die Randbedingungen zur Routenplanung sind dabei durch kreisförmige Bedrohungen und polygonförmige Hindernisse und Flugverbotszonen formuliert, die nicht durchfliegen dürfen. Zusätzlich nutzen die Algorithmen Terraindaten über den Zugriff auf eine SRTM-Datenbank. Die in RPM generierten Flugrouten werden im Anschluss an die Erstellung an AAL übertragen und dort für den weiteren Zugriff bereitgestellt. Im Rahmen dieser Arbeit wurden folgende Algorithmen in RPM implementiert:

1. Der A* (*A-Star 3D*) Algorithmus zur Routenplanung auf einem uniformen, dreidimensionalen Gitter, stellt ein heuristisches Suchverfahren dar, das einen garantiert günstigsten Pfad bezüglich einer Kostenfunktion findet [Lunze 2010]. Die Kostenfunktion berücksichtigt dabei ausschließlich die zurückgelegte Distanz des UAV. Knoten innerhalb von Objekten und unterhalb der Terrainhöhe werden für die Suche ausgeschlossen. Als Heuristik wird die euklidische Distanz zwischen der Position des UAV und dem Ziel verwendet. Diese monotone Heuristik unterschätzt für alle Fälle die tatsächlichen Kosten zum Ziel und garantiert damit eine optimale Routenlänge. Die resultierende Route ist eine Liste von Knotenpunkten vom Start bis zum Ziel.

Die Laufzeit für die Erstellung des uniformen Gitters wird im Wesentlichen durch die Anzahl und die Größe der zu berücksichtigenden Hindernisse sowie in Bodennähe durch das Terrain bestimmt. Die Laufzeitkomplexität des A*-Algorithmus hängt dabei von der Differenz zwischen Heuristik und tatsächlichen Kosten ab. Für Fälle, in denen die euklidische Distanz

die tatsächliche Distanz zum Ziel deutlich unterschätzt, müssen wesentlich mehr Knoten untersucht werden. Liegen demnach viele Hindernisse auf dem Weg des UAV zum Ziel, so erhöht sich die Laufzeit des A*-Algorithmus. Unabhängig von der Laufzeit, ist eine optimale Lösung garantiert, solange eine mögliche Verbindung zwischen Start und Ziel existiert.

2. Der Theta* (*Theta-Star 3D*) Algorithmus nach [Nash et al. 2007] stellt eine Erweiterung des A*-Algorithmus dar, der unnötige Knoten einer Route während der Laufzeit eliminiert und eine Route aus Stützstellen zur Richtungsänderung erstellt. Die Suche erfolgt dabei ebenfalls auf einem uniformen, dreidimensionalen Gitter, bei dem Hindernisse wie in Punkt 1 modelliert werden. Das Resultat der Planung entspricht dabei in ihrer Form eher einer Flugroute, da auf Strecken ohne Hindernisse keine zusätzlichen Stützstellen in Form von Knoten existieren. Der Theta*-Algorithmus findet im Allgemeinen günstigere Routen als der A*-Algorithmus. Die Laufzeitkomplexität des Theta*-Algorithmus entspricht der des A*-Algorithmus. Durch die Verarbeitung der Knotenpunktliste während der Laufzeit ergibt sich eine vergleichsweise kleine Erhöhung der Laufzeit [Nash et al. 2007].

3. Der A*-Algorithmus auf einem Quadtree-Gitter (*Quad-A* 2D*) wurde gemäß [Brand 2012] implementiert. Hierbei werden neben der Suchraumerstellung und Routenplanung auch eine Glättung der Route und anschließend eine Eliminierung unnötiger Knotenpunkte vorgenommen. Der Suchraum ist dabei nach [Eichhorn 2007] mittels eines zweidimensionalen Quadtree-Gitters modelliert, das um Objekte ein besonders feines Knotennetz bildet, um eine genauere Planung zur Vermeidung von Hindernissen zu ermöglichen. Das Gitter ist damit nicht unidirektional, sondern an die Struktur des Suchgebietes bzw. die spezifizierten Randbedingungen der Routenplanung angepasst.

Die Kostenfunktion des Algorithmus ist eine Funktion aus der Distanz zwischen Nachbarknoten, Bedrohungskosten und Kosten aus flugdynamischen Randbedingungen [Brand 2012]. Bedrohungskosten ergeben sich dabei aus dem Abstand eines Punktes zu einem Hindernis oder einer Flugverbotszone. Knoten in Objekten werden nicht von der Suche ausgeschlossen, sondern können durchfliegen werden, wenn dies die Gesamtkosten der Route minimiert. Dies ermöglicht z.B. den Durchflug durch vollständig bedrohtes Gebiet und das Auffinden bedrohter Abkürzungen. Kursvorgaben und Wenderadien des UAV werden durch das Einbringen künstlicher, radialer Hindernisse in Form von zusätzlichen Kosten in das Gitter eingebracht. Die resultierenden Gesamtkosten stellen die Grundlage für die A* Suche dar.

Im Anschluss an die Routenerstellung findet zunächst eine Glättung statt. Hierbei werden, ähnlich wie beim Theta*-Algorithmus, solche Knoten entfernt, die für die Hindernisvermeidung nicht notwendig sind. Abschließend wird eine Trajektorie aus der Route erstellt, indem Kurven, die den minimalen Kurvenradius unterschreiten, durch zusätzlich eingebrachte Stützpunkte ausrunden. Das Ergebnis ist eine Liste an Knotenpunkten, die im Rahmen der Randbedingungen des Luftfahrzeugs geflogen werden kann.

Die Laufzeitkomplexität des Algorithmus wird im Wesentlichen durch die Erstellung des Quadtree-Gitters bestimmt. Die Zeit für die Erstellung des Quadtree-Gitters liegt dabei wesentlich über der Dauer für ein unidirektionales Gitter. Die Komplexität des Suchalgorithmus auf dem Gitter ist anschließend mit dem A*-Algorithmus und dem Theta*-Algorithmus vergleichbar.

4. Die Modellierung des Suchraums durch künstliche Potenzialfelder (*PotField 2D*) nach [Horner & Healey 2004] erzeugt über einer zweidimensionalen Repräsentation des Fluggebiets ein

Vektorfeld mit Potenzialen, die Start und Ziel sowie Hindernisse modellieren. Die Routerstellung verläuft dann durch die schrittweise Annäherung an das Ziel auf der Grundlage des Vektorfeldes. Hierbei wird eine feste Schrittweite definiert, mit der, ausgehend von der aktuellen Position, entlang des Gradienten vorangeschritten wird. Das Ziel gilt als erreicht, wenn eine gewisse Distanz zum Ziel unterschritten wird. Die resultierende Route liegt als Liste der Stützpunkte bei der Suche vor.

Das Zielpotenzial wird als schräge Ebene modelliert, die am Start ihre Normale senkrecht zur Start-Ziel-Geraden besitzt. Hindernisse werden dieser Schrägen in Form begrenzter Potenzialberge überlagert. Die Schrittweite für die Suche sowie der Ziel-Offset werden so gewählt, dass sie unterhalb der Größenordnung der Hindernisse liegen.

Durch seine einfache mathematische Form hat der Algorithmus sehr geringe Laufzeiten und kann prinzipiell alle relevanten Hindernisse in Annäherung abbilden. Allerdings besteht die Gefahr von lokalen Minima, die eine korrekte Lösung verhindern. Oszillationen können zudem durch zu geringe Hindernisabstände entstehen [Werner 2013].

Im Rahmen dieser Arbeit nutzt der kognitive Agent das RPM zur Erstellung des Flugplans und zur Berechnung von Distanzen zwischen taktischen Elementen. Das RPM verarbeitet eingehende Anfragen in separaten Verarbeitungsthreads, wie in Abbildung 4-12 dargestellt. Anfragen umfassen hierfür den Typ (Routenplanung oder Entfernungsbestimmung), die ID und die Knotenpunktliste der Stützstellung zur Planung. Außerdem werden der zu verwendende Algorithmus und dessen Genauigkeit in Form digitaler Genauigkeitsstufen (niedrig, mittel, hoch) angegeben. RPM setzt die Genauigkeitsstufen intern in vordefinierte Parametersets um. Die geometrischen Randbedingungen der Planung werden über taktische Elemente übergeben.

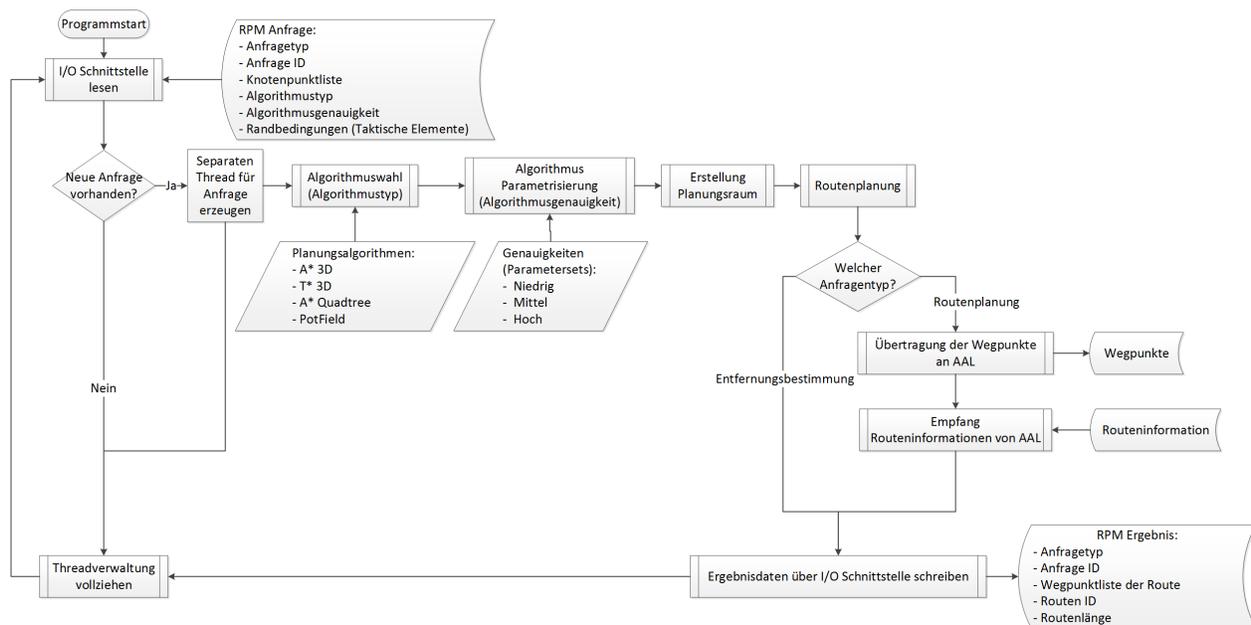


Abbildung 4-12: Verarbeitung einer Routenanfrage in separaten Verarbeitungsthreads des Routenplaner Managements (RPM)

Über den Anfragetyp und die Anfrage ID kann das Ergebnis nach der Ausführung durch ReX eindeutig zugeordnet werden. Bei der Routenplanung entnimmt ReX hierbei die ID des in AAL hinterlegten Flugplans, der im Rahmen der Distanzberechnung die Distanz in Metern übergibt.

4.3.2.2 RPM-Schnittstelle in ReX

Die RPM-Schnittstelle dient ReX zur Distanzberechnung und zur Erstellung von Flugplänen. Hierbei muss zunächst ein geeigneter Planungsalgorithmus gewählt und parametrisiert werden. Für den Sensoreinsatz des UAV muss zudem über dem Zielgebiet bzw. Zielobjekt ein geeignetes Aufklärungsverfahren entsprechend der Eigenschaften des Luftfahrzeugs und der Sensorik sowie der Auftragsanforderungen gewählt werden.

Die Auswahl und Parametrisierung des Planungsalgorithmus‘ folgt dabei einem regelbasierten Entscheidungsprozess, anhand quantitativer und qualitativer Kriterien der Eigenschaften der Algorithmen [Werner 2014a]. Die benötigten Zielgrößen ergeben sich dabei aus der geometrischen Lage sowie der Handlungsagenda des Agenten. Die Parametrisierung der Algorithmen erfolgt über eine von drei vordefinierten Parametersets zur Spezifikation der Planungsgenauigkeit.

Aufklärungsverfahren werden bei der Flugplanerstellung über Wegpunktmuster realisiert, die in ReX ausgewählt und parametrisiert werden. Hierzu dient ein Entscheidungsprozess, der von der Art der Aufklärung, der Aufklärungsziele und der Eigenschaften der Sensorik des UAV abhängt [Borchers 2014a]. Wegpunktmuster werden für die Aufklärung von punkt- und streckenförmigen Zielen sowie Flächenzielen erzeugt und parametrisiert.

Die RPM-Schnittstelle in ReX stellt folgende Aktionen für den Agenten bereit:

CALC_DISTANCE – Dient der Bestimmung der kürzesten Distanz zwischen Lageelementen unter Berücksichtigung geometrischer Randbedingungen. Die Anzahl der Lageelemente ist dabei beliebig. Zur Beschleunigung kann lediglich eine Approximation der Distanz über die Summe der Luftliniendistanzen zwischen den Einzelelementen abgerufen werden.

PLAN_ROUTE – ReX kommandiert die Flugplanerstellung mit dem zuvor gewählten Algorithmus und der gewünschten Genauigkeit. Die Stützpunkte des Flugplans sind dabei durch die Elemente der Handlungsagenda, die durch den Flugplan verbunden werden sollen gegeben und werden durch Wegpunktmuster zur Aufklärung ergänzt. Das RPM vollzieht die Routenplanung zwischen den einzelnen Stützpunkten anhand der geometrischen Randbedingungen und überträgt die resultierende Flugroute an AAL. Durch die Übergabe der Routen-ID an ReX kann im Weiteren direkt auf die Flugroute zugegriffen werden.

Im Anschluss an die Routenerstellung wird durch Verknüpfung der geometrischen Wegpunktinformationen mit den Lageelementen der Handlungsagenda ein einheitlicher Flugplan in ReX erstellt. Im weiteren Verlauf der Auftragserfüllung kann der Agent somit über die Spezifikation von Lageelementen direkt auf geometrische Wegpunktinformationen zugreifen.

4.3.3 Datenlink Management (DLM)

Das *Datenlink Management* (DLM) nach [Böhm & Schulte 2012] stellt Funktionen zur Datenübertragung, -konfiguration und -überwachung des bordseitigen Datenlinks bereit und kann parallel durch verschiedene Prozesse an Bord des UAV gleichzeitig angesprochen werden. Im Allgemeinen verfügt der Datenlink über den Verlauf einer Mission über eine variierende

Bandbreite und steht in einigen Abschnitten des Missionsgebiets wegen seiner begrenzten Reichweite gar nicht zur Verfügung. Der Datenlink wird dabei parallel zur Übertragung von Nutzlast- und Kommandodaten verwendet, die für die Missionsführung des UAV unterschiedlich hohe Prioritäten besitzen. Das DLM bietet zu diesem Zweck Funktionen zur Analyse und zur Konfiguration der Datenströme, die über den Datenlink übermittelt werden sollen.

Das DLM ermöglicht die Vorhersage der Verfügbarkeit des Datenlinks an definierten Orten anhand eines hinterlegten Datenlinkmodells. Für priorisierte Daten ist zudem eine Konfiguration des Datenlinks anhand von *Quality of Service* (QoS) möglich, die eine optimierte Nutzung der verfügbaren Bandbreite erlaubt [Böhm & Schulte 2012]. Unterstützt wird diese Funktion durch dynamisches Umcodieren, z.B. von Videodaten. Generell ermöglicht das DLM die Übertragung von Videodaten zur Mensch-Maschine Schnittstelle. Videodaten werden als Livedaten oder aufgezeichnete Daten durch Funktionen des Perzeptionsmanagements bereitgestellt (vgl. Abschnitt 4.3.4).

4.3.3.1 DLM-Schnittstelle in ReX

Das DLM ermöglicht dem kognitiven Agenten die informierte Berücksichtigung der Eigenschaften des Datenlinks im Rahmen der Auftragsumsetzung. Der Agent nutzt das DLM zur Prädiktion der Datenlinkverfügbarkeit an den Positionen der Lageelemente zum Zwecke des Versands bzw. der Speicherung von Perzeptionsergebnissen und des manuellen Kontrolleingriffs durch den Menschen. Die Kombination des Datenlinks mit der Möglichkeit zur Datenaufzeichnung trägt zur Fähigkeit des Systems bei, auch solche Aufklärungsaufträge zu erfüllen, die den Einsatz außerhalb der Datenlinkreichweite des Systems verlangen. ReX überwacht hierfür kontinuierlich die Ausgangsdaten des DLM und verarbeitet sie für die weitere Berücksichtigung in Form der Datenlinkfähigkeit (*Datalink Capability*) des UAV.

4.3.4 Sensor- und Perzeptionsmanagement System (SPMS)

Der Einsatz bildgebender Sensorik und der dazugehörigen Verarbeitungsalgorithmen wird durch ein System unterschiedlicher Softwaremodule implementiert, welches im Folgenden als Sensor- und Perzeptionsmanagement (SPMS) bezeichnet wird. Die Implementierung des SPMS beruht auf dem Perzeptionskonzept nach [Russ et al. 2013], welches durch [Hellert et al. 2012] im Hinblick auf die Adaption von Bildverarbeitungsketten weiterentwickelt wurde.

Das SPMS integriert eine beliebige Anzahl von Sensoren sowie Vorverarbeitungs- und Verarbeitungsalgorithmen, denen intern individuelle Attribute zugeordnet werden. Wird ein gewünschtes Verhalten durch einen Nutzer aufgerufen, so generiert das Perzeptionsmanagement passend zur Laufzeit Verarbeitungsketten als Kombination aus Sensoren und Verarbeitungsalgorithmen. Diese Kettengenerierung bewertet die Attribute der Verarbeitungselemente gemäß der Anforderungen aus der vorliegenden Situation und dem gewünschten Verhalten, um eine bestmögliche Lösung zu erzielen. Das SPMS ist hierzu zweischichtig implementiert. Zunächst empfängt eine Auftragsebene sog. Perzeptionsaufträge vom Nutzer, die in Listen mit Anforderungen für den Sensoreinsatz und die Verarbeitung gewandelt werden. Diese Listen werden auf einer weiteren Ebene verarbeitet, auf der der tatsächliche Sensor und die Algorithmen gewählt und ihr Einsatz parametrisiert werden. Die Verarbeitung der Anforderungslisten soll dabei durch eine KI-Architektur gemäß [Hellert & Stütz 2014] realisiert werden, die zum Zeitpunkt der Anfertigung dieser Arbeit jedoch noch nicht implementiert ist.

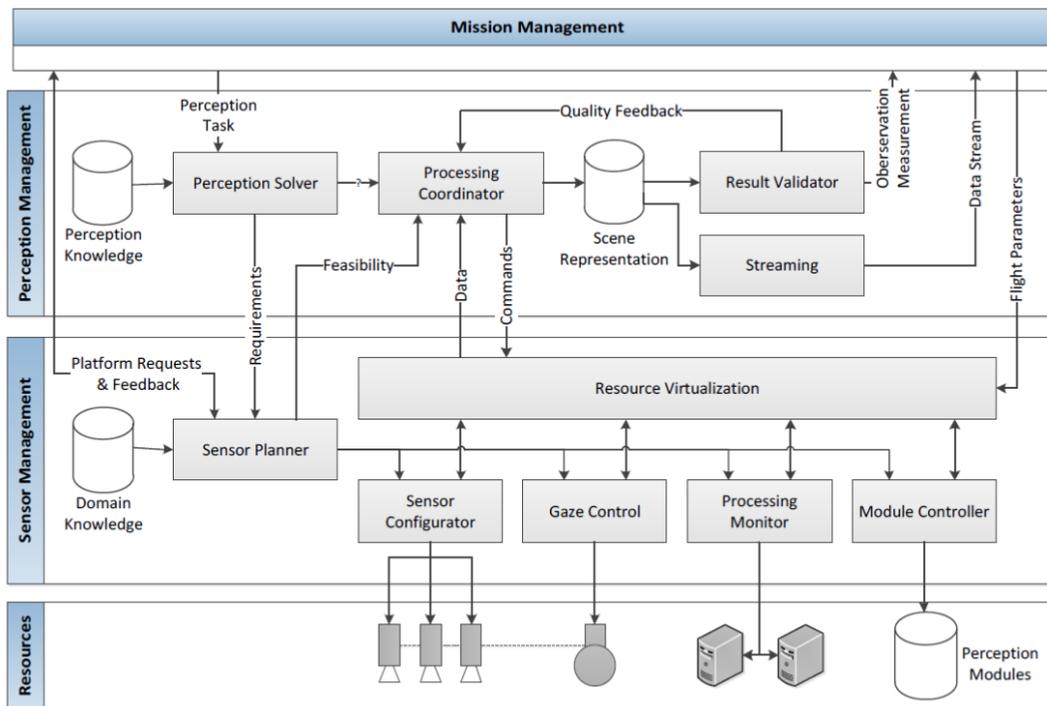


Abbildung 4-13: Systemarchitektur des Sensor- und Perceptionsmanagement Systems (SPMS) [Hellert et al. 2012]

Abbildung 4-13 stellt hierzu die Systemarchitektur des SPMS anhand von drei hierarchischen Verarbeitungsebenen dar. Perzeptionsaufträge werden vom Perceptionsmanagement (*Perception Management*) in Anforderungen umgesetzt und führen zur gezielten Konfiguration von Sensorik, Sensorführung und Sensordatenverarbeitung durch das *Sensor Management*. Die eigentliche Umsetzung erfolgt dann auf der Ressourcenebene (*Resources*). Die Ausgaben der Ressourcen werden dem *Processing Coordinator* zugeführt, der Ergebnisse in Form von Videodaten oder Objekterkennungen erzeugt und an die beauftragende Ebene zurück meldet.

Das SPMS wird durch den kognitiven Agenten zum Einsatz der Wahrnehmungsfähigkeiten des UAV verwendet. Hierzu ist die getrennte Spezifikation von Datengewinnung, -verarbeitung und -speicherung möglich. Zunächst spezifiziert der Agent die gewünschte Sensorart und die Sensorausrichtung. Prinzipiell sind hierbei die Fixierung von Positionen, das Verfolgen von Objekten sowie Winkelvorgaben für die Ausrichtung möglich. Für die Sensordatenverarbeitung kann eine automatisierte Personen- sowie eine Fahrzeugerkennung auf dem gewonnen Videomaterial gewählt werden. Schließlich können Videos in Abhängigkeit von der Verfügbarkeit des Datenlinks direkt zur Bodenkontrollstation übertragen oder, zur späteren Übertragung, an Bord aufgezeichnet werden.

Das SPMS liefert Ausgaben in zwei separaten Formen. Zum einen werden Videodaten zur Speicherung und zur Anzeige bereitgestellt, zum anderen werden Perzeptionsergebnisse in Form erkannter Personen oder Fahrzeuge an den Agenten übertragen. Die Anzeige der Videodaten erfolgt in der Bodenkontrollstation auf dem Sensorinterface gemäß Abschnitt 4.4.2. Ergebnisse der Personen- bzw. Fahrzeugerkennung werden mit der Mensch-Maschine Schnittstelle synchronisiert und für den Operateur auf dem Missionsinterface dargestellt (vgl. Abschnitt 4.4.1). Die manuelle Kontrolle der Sensorführung bzw. die Sensordatenauswertung durch den Operateur wird über das

Sensorinterface ausgeführt und setzt die kontinuierliche Verfügbarkeit des Datenlinks zur Übertragung der Videodaten voraus.

4.3.4.1 SPMS-Schnittstelle in ReX

Die SPMS-Schnittstelle in ReX bietet dem Agenten Aktionen zum Sensoreinsatz, zur Sensordatenverarbeitung, zur Speicherung und zur Wiedergabe von Daten. Die einzelnen Aktionen können dabei frei miteinander kombiniert werden, um individuelle Perzeptionsergebnisse zu erzielen. Bilddaten werden hierbei in der Verarbeitung des Agenten von ReX nicht berücksichtigt, sondern dienen ausschließlich der Ansicht durch den Menschen. ReX verwaltet hierzu die Daten als geschlossene Pakete sowie die Erkennungsergebnisse (Fahrzeuge und Personen) als einzelne Objekte.

START_SENSOR_GUIDANCE – Aktiviert den Aufklärungssensor des UAV unter Vorgabe der Ausrichtung als Datenquelle für weitere Prozesse. Ausrichtungsmodi sind hierbei Winkel- und Positionsvorgaben sowie die intelligente Sensorsteuerung gemäß der Sensordatenverarbeitung durch SPMS. Der Sensor kann als Datenquelle über die Aktion *STOP_SENSOR_GUIDANCE* deaktiviert werden.

START_FPV_SENSOR – Aktiviert den *First Person View* (FPV) Sensor des UAV (falls vorhanden) für die Aufzeichnung sowie die Übertragung. Die Sensordaten des FPV Sensors können im SPMS keiner weiteren Verarbeitung zugeführt werden. Der FPV-Sensor wird über *STOP_FPV_SENSOR* deaktiviert.

START_PROCESSING_SENSOR_OUTPUT – Startet die Verarbeitung von Sensordaten. Als Datenquellen können wahlweise der Aufklärungssensor oder aufgezeichnete Daten spezifiziert werden. Mögliche Arten der Datenverarbeitung sind die Fahrzeug- und die Personenerkennung. Detektionen werden als Ausgabe an ReX übergeben, wo sie der taktischen Lage hinzugefügt und mit den anderen Prozessen synchronisiert werden. Die Sensordatenverarbeitung wird durch *STOP_PROCESSING_SENSOR_OUTPUT* gestoppt.

LIVE_STREAM – Das Streaming von Sensordaten zur Anzeige im Sensordisplay wird über diese Aktion aktiviert. Als Datenquelle dient hierbei der aktivierte Aufklärungs- bzw. FPV-Sensor. Die Sensordaten werden so lange übertragen, wie die Datenquelle aktiviert bleibt. Bei Deaktivierung des Sensors stoppt automatische die Datenübertragung.

START_RECORDING – Startet die lokale Speicherung von Sensordaten auf dem UAV. Als Datenquelle dienen hierbei analog zum Streaming der Aufklärungs- und der FPV-Sensor. Der Zugriff auf die gespeicherten Daten erfolgt über eine eindeutige Identifikationsnummer. Die Aufnahme wird über die Aktion *STOP_RECORDING* gestoppt.

START_TX_STREAM – Startet die Übertragung aufgezeichneter Daten zur Anzeige im Sensordisplay. Voraussetzung für die Datenübertragung ist die Datenlinkverfügbarkeit. Die gespeicherten Daten werden über ihre ID ausgewählt und bleiben auch nach der Übertragung auf dem UAV gespeichert. Die Aktion endet automatisch mit dem Übertragungsende.

4.4 Mensch-Maschine Schnittstelle

Für die Interaktion des Operateurs mit dem UAV-System wird eine zweigeteilte Mensch-Maschine Schnittstelle implementiert. Das Missionsinterface dient dem Operateur zur Kommandierung und

Überwachung des kognitiven Agenten innerhalb einer Missionsumgebung unter Berücksichtigung von Umweltbedingungen und dynamischen Lageelementen. Ein separates Sensorinterface ermöglicht die Steuerung und Überwachung der Sensorik des UAV und der Sensordatenverarbeitung.

4.4.1 Missionsinterface

Das Missionsinterface nutzt eine Implementierung nach [Theißing & Schulte 2013] auf Basis der grafischen Kartendarstellung des Einsatzgebietes, mit Lageelementen als taktischen Symbolen gemäß [Department of Defense 2008] (vgl. Abbildung 4-14). Die Lageelemente sind durch den Operateur dynamisch editierbar und repräsentieren als Ganzes die taktische Lage, deren Inhalt im Rahmen der Lagesynchronisation mit dem kognitiven Agenten synchronisiert wird.

Die Auftragserteilung erfolgt über Kontextmenüs, die durch Interaktion mit den Lageelementen dargestellt werden, und alle verfügbaren Auftragsoptionen (1) anzeigen. Auftragsoptionen werden dabei dynamisch entsprechend dem Typ und Status des Lageelements sowie der Verfügbarkeit der Fähigkeiten des UAV-Systems generiert. Das Kontextmenü dient zudem der Anzeige von Informationen über den Zustand des UAV, wie z.B. dessen Position, Geschwindigkeit oder des Treibstoffvorrats.

Die einzeln formulierten Aufträge werden im Bereich *Task Agenda* (2) in Form der konsekutiven Auftragsagenda angezeigt. Die manuelle Übernahme von UAV-Funktionalitäten in Form der Fähigkeitssubstitution bzw. -addition kann für die Aufträge einzeln spezifiziert werden. Über die Schaltfläche *Commit* wird die Auftragsagenda an den kognitiven Agenten kommandiert. Zur Verknüpfung der Auftragsagenda mit den Elementen der Lagedarstellung dienen Pfeile, die mittels zusätzlicher Symbole die Auftragsreihenfolge darstellen.

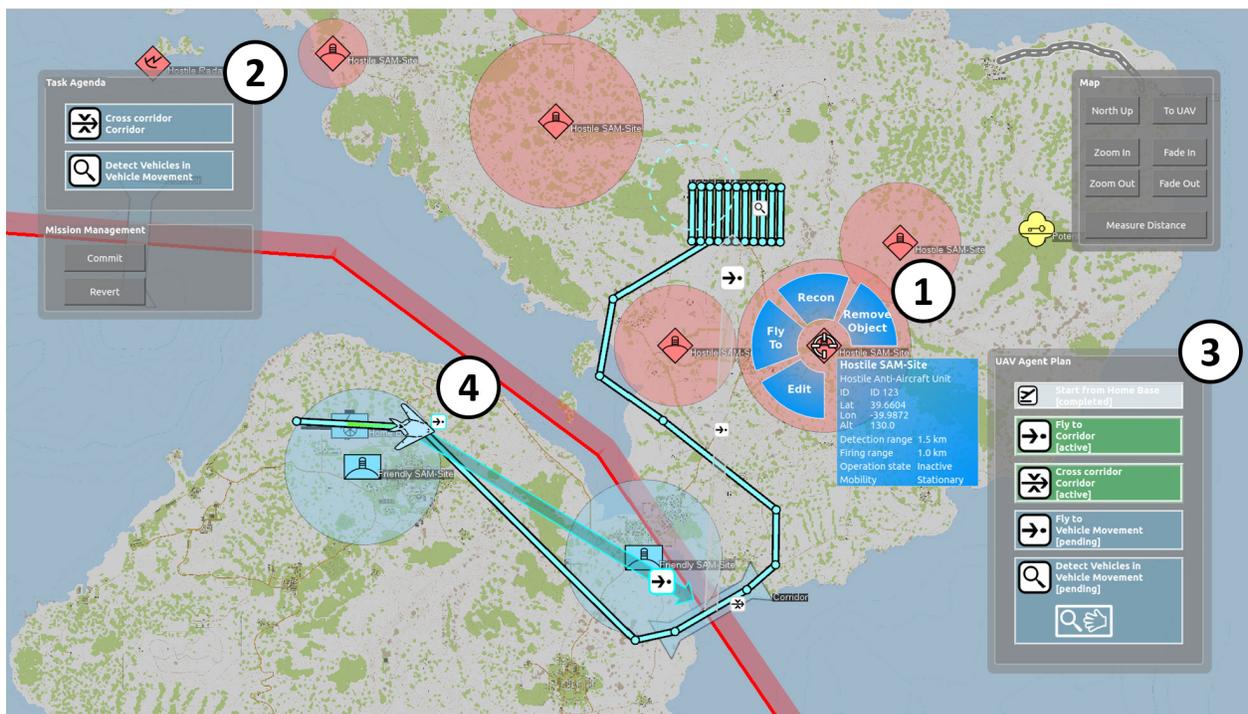


Abbildung 4-14: Das Missionsinterface nach [Theißing & Schulte 2013] zur Kommandierung und Überwachung des UAV-Systems im Missionskontext

Nach erfolgreicher Verarbeitung der Auftragsagenda führt der Agent die Handlungsagenda an den Operateur zurück (vgl. Abschnitt 3.2.2). Die Handlungsagenda wird im Bereich *UAV Agent Plan* (3) abgebildet und beinhaltet die textuelle und farbliche Beschreibung (ausstehend – blau; aktiv – grün; abgeschlossen – grau) des Erfüllungsgrads einzelner Handlungsschritte. Manuelle Kontrolleingriffe sind symbolisch in der Handlungsagenda dargestellt. In Abbildung 4-14 übernimmt der Operateur die manuelle Sensorführung bei der Fahrzeugerkennung.

Freigaben, Anfragen und Nachrichten zur Aktionskoordinierung werden, die Kartendarstellung überlagernd, als Textnachrichten angezeigt. Gleiches gilt für die Anzeige von Fehlern bei der Handlungsplanung und Auftragsumsetzung durch den Agenten. Der Operateur kann Textnachrichten bestätigen oder ablehnen.

Die Position und der Flugplan des UAV (4) sind auf der taktischen Lagekarte geometrisch repräsentiert. Für das UAV dient ein Symbol, dessen Form der Art des Luftfahrzeugs entspricht. Als Flugplan dient eine Route als verbundene Wegpunktliste und wird bei Aktivierung farblich hervorgehoben.

Zur geografischen Verortung von Sensordaten dienen Polygone, die in der Kartendarstellung zur Abbildung des Kamerasichtfeldes eingezeichnet werden. Fahrzeuge und Personen, die vom Operateur über das Sensorinterface oder direkt vom Agenten identifiziert werden, werden automatisch der taktischen Lage hinzugefügt und auf dem Missionsinterface dargestellt.

Das Missionsinterface bietet schließlich ein Werkzeug zur Distanzmessung zwischen beliebigen Punkten auf der Lagekarte. Über den Menüpunkt *Measure Distance* werden Stützpunkte definiert, zwischen denen die Distanz berechnet und angezeigt wird.

4.4.2 Sensorinterface

Das *Sensorinterface* (SIF) nach [Smirnov & Stütz 2014] bietet dem Operateur die Anzeige von Videodaten verschiedener Sensordatenquellen des UAV, die Speicherung und Auswertung dieser Daten sowie die manuelle Steuerung der Sensorik. Abbildung 4-15 zeigt die Oberfläche des SIF, in der Datenquellen über den Bereich *Cameras* (1) ausgewählt und im zentralen Anzeigebereich (2) wiedergegeben werden.

Standbilder der georeferenzierten Videodaten können über *Take Video Screen* erstellt werden. Der Zugriff auf Standbilder erfolgt über eine Bildergalerie unter der Videoanzeige (3). Standbilder dienen dem Operateur zur späteren Betrachtung relevanter Videodaten sowie zur Auswertung (u.a. zur Objekterkennung). Im Bereich *Image Information* (4) werden dem Operateur Informationen bezüglich der Lage des UAV und des Sensors zum Zeitpunkt der Aufnahme des Standbildes angezeigt. Bei der manuellen Objekterkennung annotiert der Operateur Standbilder mit Fahrzeugen oder Personen. Die Annotation erfolgt über die Spezifizierung des Objekttyps und die Bestätigung der Auswahl über *Commit Annotation*. Auf diese Art spezifizierte Lageelemente werden an das Missionsinterface übertragen und dort zur Anzeige gebracht.

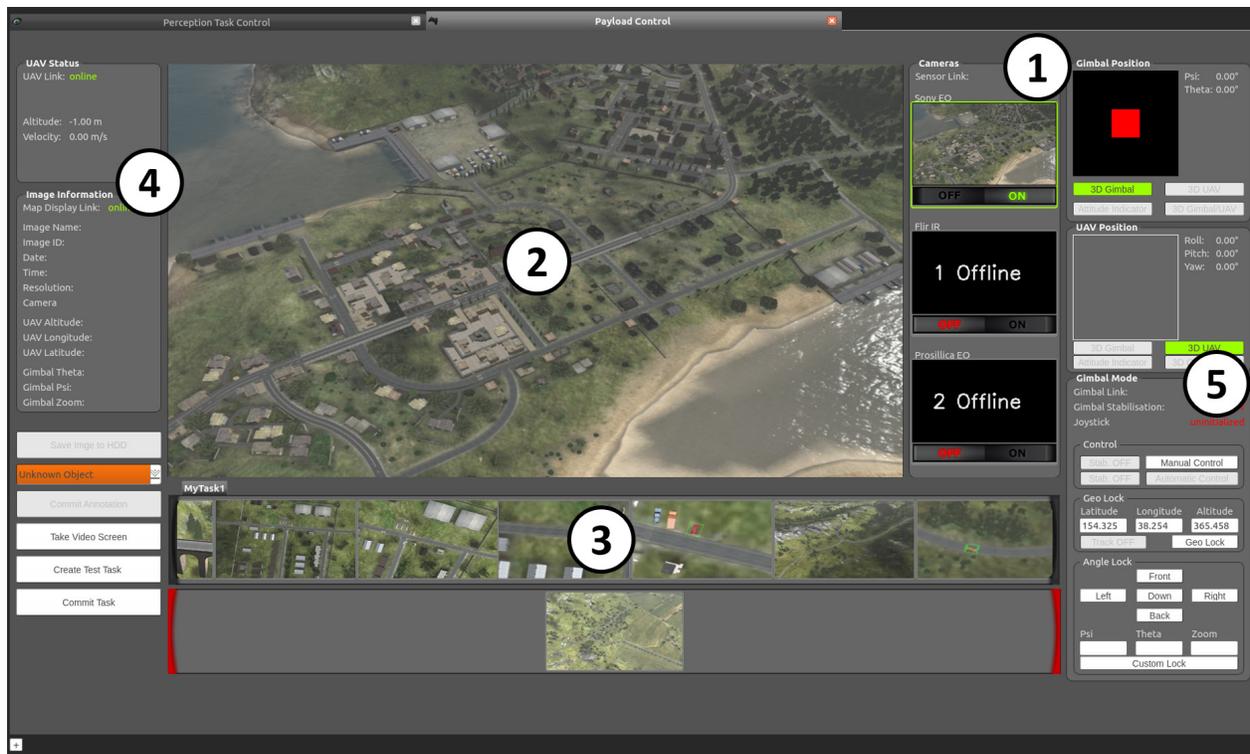


Abbildung 4-15: Das Sensorinterface nach [Smirnov & Stütz 2014] als Bedien- und Anzeigeoberfläche, die dem Operateur die Überwachung des Sensoreinsatzes des UAV ermöglicht

Bei der Anzeige von Live-Sensordaten werden im Bereich *UAV Status* Informationen zur Lage des UAV sowie zum Status der Datenlinkverbindung angezeigt. Live-Sensordaten können nur bei bestehender Datenlinkverbindung vom UAV an die Bodenkontrollstation übertragen werden. Gleiches gilt für die manuelle Steuerung des bildgebenden Sensors. Im Bereich *Gimbal Mode* (5) können Steuereingaben manuell getätigt werden. Eingaben umfassen Winkelvorgaben, eine Geolock Funktion zur Fixierung eines Fixpunktes am Boden sowie die kontinuierliche Steuerung des Sensors per Joystick.

Über die Position und Orientierung des UAV sowie die Orientierung des aktivierten Sensors ist eine Georeferenzierung der angezeigten Bilddaten möglich. Die Anzeige dieser georeferenzierten Daten als Abbild des Erfassungsbereiches des Sensors erfolgt in Form eines Polygons im Missionsinterface (vgl. Abschnitt 4.4.1).

4.5 Plattformunabhängigkeit der technischen Lösung

Die Implementierung des kognitiven Agenten ist nicht auf eine spezifische fliegende Plattform oder auf ein festes Set von Eigenschaften des Fluggeräts oder seiner Nutzlast beschränkt, sondern ist plattform- und hardwareunabhängig. Das UAV-System, bestehend aus dem kognitiven Agenten, seinen fähigkeitsbildenden Automationskomponenten und der Mensch-Maschine Schnittstelle (vgl. Abbildung 4-16) kann ohne grundlegende Anpassungen auf unterschiedlichen Plattformen eingesetzt werden. In Kapitel 5 wird die Portabilität der technischen Lösung durch die Evaluierung in der Simulation und auf zwei Flugplattformen im Flugversuch demonstriert.

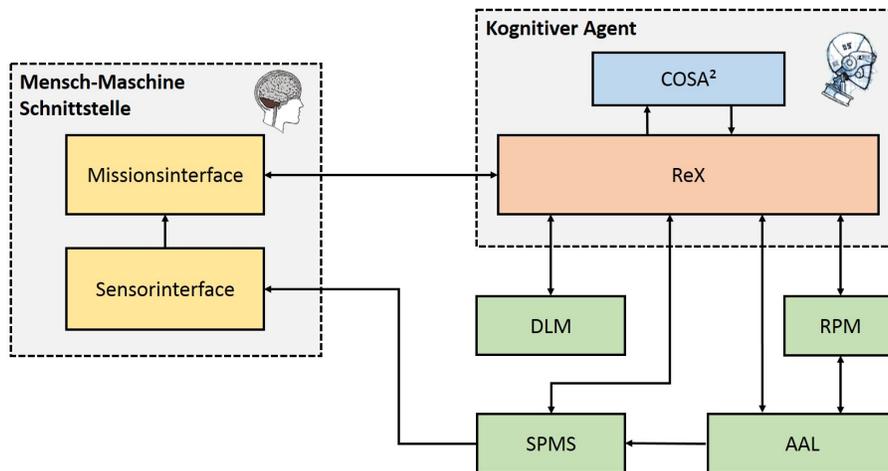


Abbildung 4-16: Softwaredarstellung der plattformunabhängigen Implementierung

Die Portabilität der technischen Lösung wird dabei auf mehreren Ebenen der Implementierung adressiert. Die eingesetzte Interprozesskommunikation entkoppelt die kommunizierenden Prozesse von der technischen Realisierung der Datenübertragung und ermöglicht dabei die Überwindung von Prozess- und Systemgrenzen. Abschnitt 4.5.1 beschreibt die hierfür eingesetzte Middleware *AnyCom*.

Auf der Ebene der Automationsfunktionen gemäß Abschnitt 4.3 besteht eine lose Kopplung zur Hardware, da die Automationskomponenten nicht für eine spezifische Systemkonfiguration, sondern für die Steuerung und das Management unterschiedlicher Endgeräte ausgelegt sind. Insbesondere gilt dies für AAL, das für beliebige FMS Implementierungen eine einheitliche Form der wegpunktbasierten Führung ermöglicht. Ebenso ermöglicht das SPMS die Verwendung verschiedener Typen von Sensoren und den Einsatz unterschiedlicher Algorithmen zur Bilddatenverarbeitung.

Auf der Ebene der Missionsführung wird durch das Fähigkeitsbewusstsein des kognitiven Agenten eine Entkopplung zwischen Handlungsoptionen und Funktionsimplementierungen erreicht. Die Verfügbarkeit von UAV-Fähigkeiten wird entsprechend ihrer Definition schrittweise aus Automationsfunktionen und ihren Fähigkeiten abgeleitet (vgl. Abschnitt 4.1.2.3). Eine Veränderung der Hardware- oder Automationskomponenten bewirkt lediglich eine Anpassung der verfügbaren UAV-Fähigkeiten und kann entsprechend durch den kognitiven Agenten berücksichtigt werden.

4.5.1 Interprozesskommunikation

Als Interprozesskommunikation (engl. *Interprocess Communication* – IPC) zwischen den in Abbildung 4-16 dargestellten Prozessen wird die Middleware *AnyCom* verwendet. *AnyCom* wurde am IFS entwickelt und abstrahiert die IPC für die kommunizierenden Prozesse, indem es eine einheitliche Schnittstelle für spezifische Kommunikationsarten und Plattformen bietet. Den kommunizierenden Prozessen werden Single- und Multicast Funktionen bereitgestellt, die sich dynamisch an die Verfügbarkeit der verwendeten Datenverbindung anpassen. Abbildung 4-17 zeigt eine schematische Darstellung der Verarbeitungskomponenten von *AnyCom*.

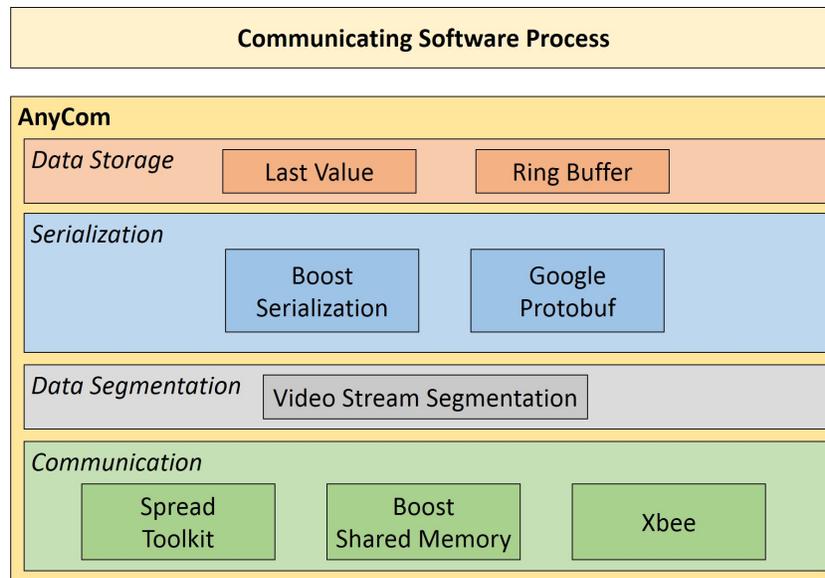


Abbildung 4-17: Schematische Darstellung der Middleware AnyCom zur Interprozesskommunikation

Die Entkopplung der Kommunikations- und der Prozessschicht bei der IPC ermöglicht den dynamischen Austausch der Kommunikationsschnittstellen und damit die Anpassung an Hardware und Software-Anforderungen ohne die Anpassung der kommunizierenden Softwareprozesse. Die Verarbeitungsstruktur von *AnyCom* leistet darüber hinaus eine Angleichung der Fähigkeiten der unterschiedlichen Kommunikationsarten und schafft mit dieser Standardisierung eine weitere Entkopplung.

Über die Instanziierung eines *AnyCom*-Objektes wird die IPC mit einer spezifizierten Art der Datenkommunikation initialisiert (vgl. Abschnitt 4.5.1.1). Aus beliebigen Datentypen werden Nachrichtenklassen geformt und an das *AnyCom*-Objekt übergeben oder von diesem empfangen und über einen eindeutigen Namen identifiziert. Die Form der Nachrichtenserialisierung wird als Teil der Nachrichtenklasse deklariert (vgl. Abschnitt 4.5.1.3). Über die Zuordnung von Nachrichten zu Gruppen wird zudem das Versenden mittels Multicast ermöglicht. Die Spezifikation der Datenspeicherung erfolgt ebenfalls über die Deklaration in den Nachrichten (vgl. Abschnitt 4.5.1.4).

4.5.1.1 Datenkommunikation

AnyCom implementiert drei verschiedene Arten der Kommunikation, die jeweils unterschiedliche Anwendungsfälle abdecken. Daten werden dabei grundsätzlich als serialisierte Byteströme versendet und empfangen.

- 1. Spread Toolkit:** Das *Spread Toolkit* [Spread Concepts LLC 2014] bietet einen zuverlässigen Kommunikationsbus zwischen verteilten Systemen, der eine skalierbare und gruppenorientierte Kommunikation zwischen Prozessen ermöglicht. Ein Serverprozess verarbeitet hierzu versendete Nachrichten von Clientprozessen und verteilt diese an die jeweiligen Empfänger unter garantierter Beibehaltung der Datenreihenfolge. Vernetzte Serverprozesse sorgen darüber hinaus für eine intelligente Verteilung von Datenströmen in Netzwerken und damit für eine Entlastung der Datenverbindung. Das *Spread Toolkit* nutzt eine *Transmission Control Protocol* (TCP) Datenübertragung über einen IP-fähigen Datenlink.

- 2. Boost Shared Memory:** Die Implementierung der *Shared Memory* Kommunikation in AnyCom nutzt die *Boost Shared Memory Library* [Gaztanaga 2014]. *Shared Memory* beschreibt den gemeinsamen Zugriff mehrerer Prozesse auf den gleichen Speicherbereich im Arbeitsspeicher eines Computers und ermöglicht damit eine schnelle Distribution großer Datenmengen. Durch die speicherinterne Kommunikation ist *Shared Memory* jedoch auf lokale Systeme begrenzt.
- 3. XBee:** Zur Implementierung der Kommunikation über einen Schmalbanddatenlink implementiert AnyCom die Anbindung an XBee-Pro 868 Kommunikationsmodule [Digi International Inc. 2014], die über eine Frequenz von 868MHz kommunizieren. Der XBee Datenlink wird in 5.2.1 als Kommandolink an Bord von UAV verwendet.

4.5.1.2 Datensegmentierung

Die Länge der Byteströme ist bei der Kommunikation eingeschränkt. Große Datenklassen überschreiten nach der Serialisierung diese Länge und müssen für die Übertragung zunächst segmentiert, einzeln übertragen und nach der Übertragung erneut zusammengefügt werden. AnyCom implementiert eine solche Segmentierung für Videodatenströme. Einzelne Bildframes werden hierzu segmentiert und einzeln übertragen. Auf der Empfängerseite werden zunächst die einzelnen Frames zusammengefügt und anschließend als Video deserialisiert.

4.5.1.3 Datenserialisierung

Für die Serialisierung von Daten als reversible Erzeugung von Byteströmen aus beliebigen Datentypen nutzt AnyCom zwei alternative Implementierungen. Auf der Empfangsseite werden die Daten nach der Übertragung zunächst deserialisiert und anschließend zum Abruf vorgehalten.

- 1. Boost Serialization:** Die *Boost Serialization Library* [Boost 2004] ermöglicht die Serialisierung und Deserialisierung beliebiger Datentypen in C++. Nachrichtendefinitionen werden zudem versioniert und können somit aufwärtskompatibel verarbeitet werden.
- 2. Google Protobuf:** Zur Serialisierung von Datentypen in Prozessen, die nicht in C++ implementiert sind, nutzt AnyCom die *Google Protocol Buffers* (ProtoBuf) [Google Inc. 2008]. ProtoBuf bietet ähnliche Funktionen wie die *Boost Serialization Library*, ist jedoch programmiersprachenunabhängig. Im Rahmen dieser Arbeit wird ProtoBuf für die IPC zwischen ReX und COSA² verwendet, da COSA² in C# implementiert ist.

4.5.1.4 Datenverwaltung

AnyCom bietet unterschiedliche Arten der Nachrichtenverwaltung für den Empfänger. Vorhandene Nachrichten werden entweder durch neue Nachrichten überschrieben (*Last Value*) oder in einem Ringpuffer (*Circular Buffer*) mit fester Größe gespeichert, in dem neue Nachrichten an ältere angehängt werden. Wird die Speicherkapazität des Ringpuffers überschritten, wird die älteste Nachricht im Puffer verworfen und durch die neue ersetzt. Der Abruf der Nachrichten ist als verbrauchendes Lesen implementiert, bei dem die Daten nach dem *First-in-First-Out* (FiFo) Prinzip aus dem Puffer gelesen werden.

Die *Last Value* Speicherung wird vor allem für Statusnachrichten verwendet, während sich der *Circular Buffer* besonders für Kommandonachrichten und für Verarbeitungsprozesse eignet.

5 Evaluierung

Die experimentelle Evaluierung der technischen Implementierung des kognitiven Agenten und des automatisierten UAV-Systems gliedert sich in zwei Teile. Zunächst wird der Zusammenhang zwischen dem Leistungs- und Interaktionsverhalten des kognitiven Agenten und verschiedenen Leistungsparametern des Arbeitssystems anhand eines Mensch-Maschine Versuchs im Rahmen einer UAV-Missionssimulation bewertet. In einem zweiten Abschnitt wird über Realflugversuche mit zwei verschiedenartigen UAV-Plattformen der Nachweis des Reifegrades sowie der Plattformunabhängigkeit der Implementierung erbracht.

5.1 Evaluierung im Mensch-Maschine Versuch

Die Mensch-Maschine Versuchsreihe erfolgt in Form eines (Labor-)Experiments zur intern validen Hypothesenüberprüfung. Das Ziel der Versuchsreihe ist die Verifizierung der zentralen Forschungsfragestellungen gemäß Abschnitt 2.4 im Hinblick auf den Einfluss des Interaktionsverhaltens des Agenten auf die leitende Kontrolle des Operateurs durch die Gegenüberstellung zweier Agentenkonfigurationen (vgl. Abschnitt 5.1.1). Hierzu werden in Abschnitt 5.1.2 zunächst Versuchshypothesen formuliert. Anschließend werden die Erhebungsmethoden des Experiments in Form der Stichprobe, des Aufbaus, des Materials sowie die Versuchsdurchführung selbst beschrieben (vgl. Abschnitt 5.1.3). Schließlich werden die Ergebnisse der Versuchsreihe in Abschnitt 5.1.4 präsentiert und hinsichtlich der zugrundeliegenden Hypothesen diskutiert (vgl. Abschnitt 5.1.5).

Wesentliche Teile des Versuchsentwurfs und der Versuchsdurchführung sind im Rahmen der Betreuung zweier Masterarbeiten durch Studenten entstanden, die sich mit Aspekten der Kommandierung und der Überwachung des kognitiven Agenten auseinandersetzen. Für eine genauere Beschreibung des Versuchskontexts und der Messdatenerhebung sei an dieser Stelle auf die Arbeiten von [Werner 2014b] und [Borchers 2014b] verwiesen. Eine kompakte Darstellung ausgewählter Ergebnisse der Mensch-Maschine Versuchsreihe geben [Denk et al. 2015].

5.1.1 Agentenkonfigurationen

Zur Bewertung des Interaktionsverhaltens zwischen Mensch und kognitivem Agent wird eine Vergleichsstudie mit zwei unterschiedlichen Agentenkonfigurationen durchgeführt. Die beiden Konfigurationen sind bezüglich der Leistungsfähigkeit der internen Auftragsverarbeitung identisch (vgl. Abschnitt 3.2.2), unterscheiden sich jedoch hinsichtlich der an den Operateur zurückgeführten Informationen. Tabelle 5-1 zeigt eine Gegenüberstellung des Interaktionsverhaltens der beiden Agentenkonfigurationen.

In **Konfiguration A** ist die Rückführung von Statusinformationen weitestgehend auf subsymbolische Informationen beschränkt. So werden die Position sowie der Flugplan des UAV an den Operateur übertragen, nicht jedoch symbolische Informationen bezüglich der Verfügbarkeit von UAV-Fähigkeiten. Zudem entfällt die Rückführung des zur Auftragserfüllung abgeleiteten Agentenplans, wodurch dem Operateur lediglich der Flugplan zum Erschließen der Absicht des Agenten und zur Vorhersehbarkeit von Aktionen zur Verfügung steht. Bei der symbolischen Fehlplanung des Agenten erfolgt eine reduzierte Rückmeldung an den Operateur, in der lediglich ein Fehlerereignis gemeldet, nicht jedoch dessen Ursache benannt wird.

Konfiguration B stellt den Agenten gemäß des in Abschnitt 3.2 beschriebenen Konzepts mit uneingeschränktem Interaktionsverhalten dar. Zusätzlich zur Position und dem Flugplan des UAV wird dem Operateur die Verfügbarkeit von UAV-Fähigkeiten symbolisch dargestellt. Zudem erfolgen eine Rückführung des Agentenplans und die Darstellung des Auftragsstatus. Bei Fehlermeldungen auf Grund von Fehlplanung des Agenten wird in einem Dialogfenster zusätzlich die Fehlerursache dargestellt, um die Anpassung der Operateureingaben zu unterstützen.

Tabelle 5-1: Gegenüberstellung des Interaktionsverhaltens der Versuchskonfigurationen

Interaktionsverhalten	Konfiguration A	Konfiguration B
	Eingeschränkte Rückführung von Statusinformationen	Uneingeschränktes Interaktionsverhalten
Grafische Darstellung der UAV-Position und des Flugplans	✓	✓
Rückführung des Agentenplans	⊗	✓
Symbolische Darstellung der Verfügbarkeit von UAV-Fähigkeiten	⊗	✓
Fehlermeldungen ohne Fehlerursachen	✓	⊗
Fehlermeldungen mit Fehlerursachen	⊗	✓

Technische Umsetzung

Technisch entspricht die Agentenkonfiguration B der in Abschnitt 4.1 dargestellten Implementierung des kognitiven Agenten. Für die Umsetzung der Agentenkonfiguration A wird im Wesentlichen die Operateur-Schnittstelle in ReX angepasst, sodass Inhalte der Komponenten *Auftragsverwaltung* (Agentenplan und Auftragsstatus) und *Ursachenbestimmung bei Fehlplanung in COSA²* (Fehlerursachen) nicht an die Mensch-Maschine Schnittstelle übertragen werden. Zudem werden die Inhalte der Komponente *UAV-Status* hinsichtlich der Verfügbarkeit von UAV-Fähigkeiten gefiltert. Die Agentenkonfigurationen unterscheiden sich somit ausschließlich hinsichtlich der Rückführung von Informationen an den Menschen. Die Verarbeitung innerhalb der hierarchischen Agentenarchitektur ist in beiden Agentenkonfigurationen somit identisch realisiert.

5.1.2 Hypothesen

Die in Abschnitt 2.4 formulierte Forschungsfragestellung suggeriert einen positiven Zusammenhang zwischen dem angepassten (Interaktions-)verhalten eines kognitiven Agenten und der hierdurch erzielbaren Unterstützung des menschlichen Operateurs. Zur experimentellen Untersuchung dieser These ist es notwendig, empirisch-inhaltliche Hypothesen abzuleiten, die in Form statistischer Vorhersagen bestätigt bzw. widerlegt werden können [Hussy et al. 2013].

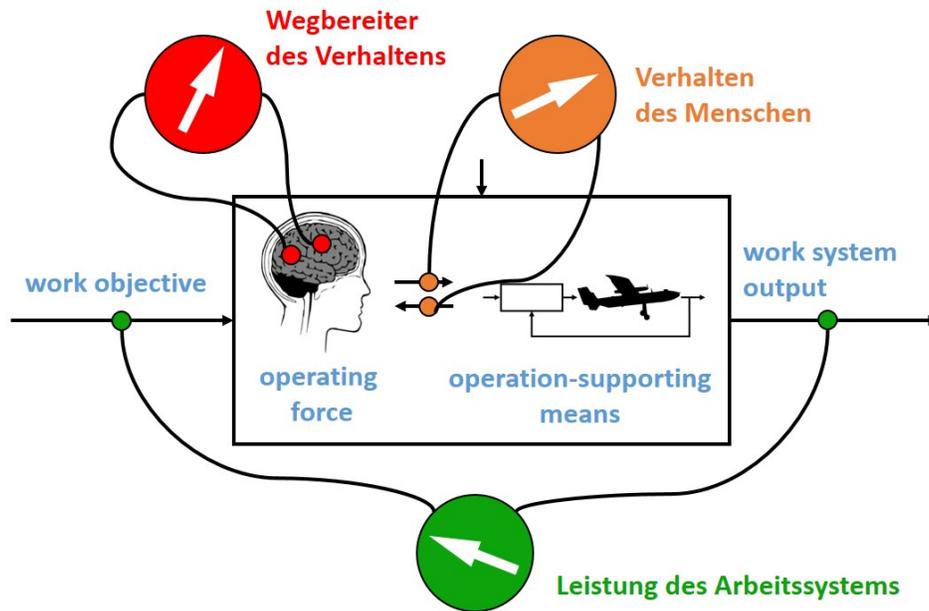


Abbildung 5-1: Kategorisierung von Messgrößen im Arbeitssystem bei der leitenden Kontrolle [Schulte 2014]

Zur Evaluation des Einflusses spezifischer Agentenkonfigurationen auf das resultierende Arbeitssystem lassen sich nach [Schulte 2014] drei Kategorien von Messgrößen identifizieren (vgl. Abbildung 5-1):

Wegbereiter des Verhaltens umfassen interne Zustände des Operators, die als subjektive Messgrößen seine kognitive Leistung beeinflussen. Zu den kognitiven Wegbereitern zählen u.a. die Beanspruchung, die Aufmerksamkeit und das Situationsbewusstsein des Operators. Mit Bezug auf Abschnitt 2.3.1.2 hat zudem das Vertrauen des Operators in das Verhalten des kognitiven Agenten Einfluss auf dessen Leitung.

Als **Verhalten** wird die objektiv messbare Interaktion des Operators mit seiner Umgebung und den Einsatz der OSM, insbesondere des kognitiven Agenten, bezeichnet. Für die Leitung des agentengeführten UAV lässt sich menschliches Verhalten über die Reaktionszeit oder die Bearbeitungsdauer und Fehlerhäufigkeit bei Aufgaben quantifizieren. *Verhalten* wird wesentlich durch die *Wegbereiter des Verhaltens* beeinflusst.

Die **Leistung** des Arbeitssystems beschreibt schließlich die Beurteilung des Gesamtsystems bestehend aus OF und OSM. Diese wird dabei maßgeblich durch das menschliche *Verhalten* beeinflusst. Die Bewertung der Leistung erfolgt dabei mit Bezug auf das vorgegebene Arbeitsziel, welches im Folgenden in Form von Missionszielen vorgegeben wird (vgl. Abschnitt 5.1.3.2). Leistungskriterien sind u.a. die Missionsdauer bzw. die Einhaltung von Missionskriterien.

Entsprechend dieser Kategorisierung können nun überprüfbare Hypothesen formuliert werden, die die beschriebenen Variablen als Kriterien verwenden. Die Überprüfung der Hypothesen erfolgt mit Bezug auf die Leitungsfunktionen des Operators bei der UAV Führung. Konkret werden dazu Daten bei der Kommandierung (*teach* und *intervene*) und der Überwachung (*monitor*) bei der Initialplanung und während der Missionsdurchführung erhoben. Die Operationalisierung der Variablen wird in Abschnitt 5.1.3.3 beschrieben.

Die einzelnen Hypothesen lauten:

Wegbereiter des Verhaltens

H1 – Beanspruchung

Der Operateur wird bei der UAV Führung in Konfiguration B weniger stark beansprucht als in Konfiguration A.

H2 – Situationsbewusstsein

Das Situationsbewusstsein wird in Konfiguration B besser ausgeprägt sein als in Konfiguration A.

H3 – Vertrauen in den Agenten

Das Vertrauen des Operateurs in die Fähigkeiten des kognitiven Agenten ist in Konfiguration B höher als in Konfiguration A.

H4 – Akzeptanz

Konfiguration B erzeugt beim Operateur eine höhere Akzeptanz in Bezug auf a) die Systeminteraktion, b) das Systemverhalten bei Lageänderung, c) die systemseitig bereitgestellten Informationen und d) das Gesamtsystem.

H5 – Aufmerksamkeitsverteilung

In Konfiguration B lenkt der Operateur einen größeren Anteil seiner Aufmerksamkeit auf die durch den Agenten zurückgeführten, symbolischen Informationen als in Konfiguration A.

Verhalten

H6 – Anzahl von Arbeitsschritten

In Konfiguration B benötigt der Operateur gegenüber der Konfiguration A eine geringere Anzahl von Arbeitsschritten zur Formulierung der Aufträge an den kognitiven Agenten.

H7 – Interaktionszeit

Konfiguration B verringert gegenüber Konfiguration A die Interaktionszeit des Operateurs bei der Formulierung von Aufträgen an den kognitiven Agenten.

H8 – Fehlerhäufigkeit

Konfiguration B verringert gegenüber Konfiguration A die Fehlerhäufigkeit des Operateurs bei der Formulierung von Aufträgen an den kognitiven Agenten.

Leistung

H9 – Umsetzung der Arbeitsaufträge

In Konfiguration B kann der Operateur seine Arbeitsaufträge, unter Berücksichtigung des Treibstoffvorrats des UAV, eher umsetzen als in Konfiguration A.

Im Mensch-Maschine Versuch sind die Hypothesen *H1*, *H5* und *H6* zwei Mal, getrennt für eine initiale Kommandierungsphase und eine manuelle Umplanungsphase, zu überprüfen. Die Testszenarien bei der Messwerterhebung (vgl. Abschnitt 5.1.3.2) sind entsprechend gestaltet, um eine getrennte Auswertung der Hypothesen sicherzustellen.

5.1.3 Methode

5.1.3.1 Personenstichprobe

Die Personenstichprobe des Versuchs umfasste 14 Masterstudenten der Luft- und Raumfahrttechnik an der Universität der Bundeswehr München (UniBwM). 13 der Probanden waren Offiziere aus allen Teilstreitkräften der Bundeswehr, der vierzehnte Proband ein Austauschstudent aus den USA mit geringen Deutschkenntnissen. Von den Probanden waren 12 männlich und zwei weiblich in einer Altersspanne zwischen 21 und 27 Jahren mit einem Altersdurchschnitt von 24,2 Jahren.

Alle Versuchspersonen verfügten über ein ausgeprägtes technisches Grundlagenwissen ohne gesonderte fliegerische oder taktische Ausbildung zur UAV-Führung oder Vorkenntnisse zur Interaktion mit dem kognitiven Agenten im Missionssimulator. Die Einweisung der Probanden in die Operateurtätigkeiten erfolgte ohne Vorlauf direkt zu Beginn der Versuchsdurchführung (vgl. Abschnitt 5.1.3.4). Die Einweisung und Versuchsleitung erfolgte auf Deutsch und wurde lediglich für den amerikanischen Probanden auf Englisch durchgeführt.

Die Teilnahme der Probanden an den Versuchen erfolgte freiwillig und ohne Aufwandsentschädigung. Es existierten keine Ausschlusskriterien für Versuchsteilnehmer. Ein Versuchsabbruch durch die Probanden war prinzipiell jederzeit möglich, wurde jedoch nicht in Anspruch genommen. Während der Versuchsdurchführung standen den Probanden Getränke und Snacks zur Verfügung.

5.1.3.2 Aufbau

Die Datenerhebung erfolgte anhand zweier in ihrer Form und Komplexität vergleichbaren Testszenarien (*Szenario I* und *Szenario II*), die nacheinander jeweils mit einer der beiden Agentenkonfigurationen A und B von den Probanden durchlaufen wurden. Jeder Proband durchlief im Rahmen des Experiments zwei Messdurchläufe. Zur Vermeidung von Sequenzeffekten, hierbei insbesondere Positionseffekten und Übertragungseffekten, durchlief die erste Hälfte der Probanden *Szenario I* in Konfiguration A und anschließend *Szenario II* in Konfiguration B, während die zweite Hälfte *Szenario I* in Konfiguration B und anschließend *Szenario II* in Konfiguration A durchlief. Durch die große Ähnlichkeit der Testszenarien wurde mit der Variation der Agentenkonfigurationen ein ausreichend interindividuelles Ausbalancieren erreicht [Hager 1987].

Zur Homogenisierung der Versuchsgruppe und ihrer Befähigung zur Szenariendurchführung erfolgte vor den Messdurchläufen eine Einweisungs- und Übungsphase. Hierbei wurden den Probanden zunächst die Aufgaben und Tätigkeiten für den Versuch erläutert. Es folgte eine Beschreibung der Anzeige- und Eingabelemente der Mensch-Maschine Schnittstelle des UAV-Missionssimulators. Schließlich wurden anhand eines Übungsszenarios die grundlegende Agenteninteraktion sowie wesentliche Szenarienelemente demonstriert und erläutert. Während der Messdurchläufe wurde zur Vermeidung der Verfälschung von Messwerten auf Erläuterungen durch den Experimentator verzichtet. Eingriffe durch den Experimentator erfolgten nur zur Behebung systembedingter Fehler.

Während des Versuchs erfolgte die Interaktion hauptsächlich über das Missionsinterface. Ständige Nebenaufgabe während der Messdurchläufe war die Beobachtung der Kamerabilder im Sensorinterface. Hierbei sollte von den Probanden auffälliges Verhalten von Fahrzeugen erfasst und notiert werden. Die Nebenaufgabe diente im Versuch ausschließlich zur Generierung einer zusätzlichen Belastung der Probanden, wurde jedoch nicht ausgewertet.

Szenarien

Abbildung 5-2 zeigt die gemeinsame Ausgangslage der Szenarien und alle szenarienrelevanten Lageelemente in einer Kartendarstellung des Missionsinterface, wie sie auch den Probanden im Simulator dargestellt wurde.

Die Lage umfasst zwei Inseln, von denen die kleinere, süd-westlich gelegene Insel, von eigenen Kräften kontrolliert wird und den Heimatflughafen (*Home Base*) beheimatet, welcher als Ausgangspunkt aller Szenarien diente. Die größere, nord-östlich gelegene Insel, wird durch feindliche Kräfte kontrolliert. Die FLOT verläuft zwischen beiden Inseln von West nach Ost und darf durch das UAV ausschließlich unter Nutzung dynamisch verfügbarer Flugkorridore überquert werden. Auf der feindlich kontrollierten Insel befinden sich ein Gebiet (*Northern Bay*), eine Straße (*Coastal Road*) sowie zwei Objekte (*Potential Headquarters* und *Potential Ammunition Depot*), für die dem UAV im Rahmen der Szenarien Aufklärungsaufgaben übertragen werden. Die Insel ist dabei durch dynamisch veränderliche SAM Stellungen befestigt, die das UAV bei der Auftragsbefehlsausführung bedrohen.



Abbildung 5-2: Ausgangslage der Übungs- und Testszenarien im UAV-Missionssimulator

In beiden Szenarien operiert das UAV mit begrenztem Treibstoffvorrat, der zu Beginn auf einen Ausgangswert initialisiert wird und anschließend proportional mit der zurückgelegten Flugdistanz abnimmt. Alle Aufträge müssen mit dem vorgegebenen Treibstoffvorrat abgeschlossen werden. Ist der Treibstoff des UAV aufgebraucht, so wird dies im Kontextmenu des UAV grafisch dargestellt, ohne die Simulation zu beeinträchtigen oder zu unterbrechen. Zum Abschluss aller Szenarien wird das UAV an die *Home Base* zurückgeführt und gelandet.

Zum Schutz des UAV im Flug müssen die Wirkungsbereiche von SAM Stellungen gemieden werden. Diese Vorgabe korreliert mit der Beschreibung der in Abschnitt 4.2 modellierten Motivation des kognitiven Agenten zum Eigenschutz. Sie wird jedoch als zusätzliches Führungsziel der Operateure in die Rahmenlage aufgenommen.

Trainingsszenario

Zum erstmaligen Erleben und Verstehen der Bedienung und des Verhaltens des kognitiven Agenten umfasst das Trainingsszenario eine doppelte Aufklärung des *Potential Headquarter*, bei deren Durchführung unterschiedliche Ereignisse eingespielt werden. Zur gezielten Anleitung der Probanden werden die einzugebenden Einzelaufträge vorformuliert und lediglich die Eingabe und Überwachung der Aufträge an die Probanden übertragen. Das Trainingsszenario wird grundsätzlich unter Nutzung des kognitiven Agenten mit vollem Interaktionsumfang (*Agentenkonfiguration B*) durchlaufen.

Im ersten Teil wird die Aufklärung des Ziels und die anschließende Rückkehr zur *Home Base* kommandiert. Bei der Rückkehr erscheint eine SAM Stellung auf dem geplanten Flugweg des UAV, woraufhin eine automatische Umplanung durch den Agenten erfolgt. Hierbei werden ein angepasster Agentenplan und eine Alternativroute erzeugt, welche die Bedrohung vermeiden und schließlich zum sicheren Erreichen der *Home Base* führen.

Im zweiten Teil wird ebenfalls die Aufklärung des Ziels und die Rückkehr zur *Home Base*, diesmal jedoch über einen spezifischen Flugkorridor (*Corridor-B*) kommandiert. Nach Abschluss der Aufklärung wird der gewählte Korridor durch den Experimentator blockiert, woraufhin der Agent umplanen muss und auf Grund eines Konflikts der Auftragsvorgaben keine valide Handlungsagenda ableiten kann. Der Operateur erhält eine Fehlermeldung unter Angabe der Fehlerursache (*Corridor-B is unreachable*). Der Proband ist daraufhin gezwungen, seinen Auftrag umzuformulieren, indem er einen Alternativkorridor vorgibt bzw. von der Spezifikation eines Korridors absieht. Der Agent erstellt daraufhin den Agentenplan sowie den Flugplan und führt abschließend das UAV zur *Home Base* zurück, um dort die Landung zu vollziehen.

Szenario I

In Szenario I sollen die vier Lageelemente auf der vom Feind kontrollierten Insel in einer gegebenen Reihenfolge aufgeklärt werden. Ziel ist es zunächst, eine Fahrzeugerkennung über dem *Northern Bay* Gebiet und entlang der *Coastal Road* durchzuführen. Anschließend sollen das *Potential Ammunition Depot* sowie das *Potential Headquarter* aufgeklärt werden. Als Missionsabschluss erfolgt die Landung des UAV auf der *Home Base*.

Während der Missionsdurchführung werden zu festgelegten Zeitpunkten durch den Experimentator Ereignisse abgerufen. So wird im Anflug auf die *Coastal Road* die UAV-Fähigkeit zur Fahrzeugerkennung deaktiviert (simulierter Systemausfall im UAV) und das *Potential Ammunition*

Depot durch eine neu erzeugte SAM Stellung bedroht. Hierdurch entsteht ein Auftragskonflikt auf Seiten des Agenten, der ein Eingreifen des Operateurs erfordert. Der Operateur ist dabei gezwungen, direkt mit der Aufklärung des *Potential Headquarter* fortzufahren.

Im späteren Verlauf der Mission wird der Ausflugkorridor des UAV während des Anflugs gesperrt, wodurch eine Umplanung notwendig ist. Abhängig von der Kommandierung durch den Operateur kann hierbei der Agent selbstständig eine Umplanung durchführen (keine explizite Vorgabe des Korridors), oder der Proband muss erneut korrigierend eingreifen (explizite Vorgabe des Korridors durch den Operateur).

Zur Erhöhung der Missionsdynamik werden während des Szenarios zusätzlich, in unregelmäßigen Abständen, SAM Stellungen hinzugefügt und entfernt bzw. Flugkorridore aktiviert und deaktiviert. Hierbei wird eine Beeinflussung der Auftrags Erfüllung durch den Agenten vermieden.

Szenario II

Szenario II sieht die Aufklärung von drei Zielen auf der feindlich kontrollierten Insel vor. Zunächst soll eine Fahrzeugerkennung über dem *Northern Bay* Gebiet durchgeführt werden. Es folgt die Behandlung der in *Szenario I* nicht erreichten Zielobjekte in Form einer Fahrzeugerkennung entlang der *Coastal Road* und der anschließenden Aufklärung des *Ammunition Depot*. Zum Abschluss soll das UAV, wie in Szenario I, an der *Home Base* gelandet werden.

Im Anflug auf die *Northern Bay* wird eine SAM Stellung so platziert, dass ihr Wirkungsbereich die Flugroute des UAV blockiert. Hieraus ergibt sich die Notwendigkeit zur Umplanung der Flugroute, was selbstständig durch den Agenten initiiert wird. Beim Anflug auf das *Potential Ammunition Depot* wird die UAV-Fähigkeit zur Sensorführung vom Experimentator deaktiviert, sodass der Auftrag zur Aufklärung des Objekts nicht vollzogen werden kann. Hierdurch muss der Operateur korrigierend eingreifen und seine Auftragsformulierung anpassen.

Die Abschätzung der verbleibenden Flugdistanz auf der Basis des restlichen Treibstoffvorrats des UAV erfolgt durch den Probanden über ein Werkzeug zur Distanzmessung im Missionsinterface (vgl. Abschnitt 4.4.1).

Kontrollvariablen

Aus den beschriebenen Testszenarien lassen sich Kontrollvariablen für das Experiment identifizieren. Hierbei handelt es sich um „Stellschrauben“ des Experiments, die der Experimentator gezielt am System einstellen kann, und die Einfluss auf die Ausprägung und den Wert der abhängigen Variablen (vgl. Abschnitt 5.1.3.3) nehmen. Die unabhängigen Variablen des Experiments sind durch die Agentenkonfigurationen A und B gegeben.

Treibstoff

Im Experiment können die Ausgangsmenge an Treibstoff sowie der Proportionalitätsfaktor zur Bestimmung des Treibstoffverbrauchs vom Experimentator angepasst werden. Für die Datenerhebung wurde die initiale Treibstoffmenge einheitlich mit 10000 Einheiten festgelegt. Zur vereinfachten Berechnung der UAV-Reichweite durch die Probanden (Distanzen werden in Meter angezeigt) wurde der Treibstoffverbrauch mit 1 Einheit pro Meter fixiert.

Bedrohungslage

Die Anzahl und Größe der Wirkungsbereiche von SAM Stellungen waren durch den Experimentator während der Versuche frei veränderbar. Die Menge der Wirkbereiche ergab während der Missionsdurchführung die relevante Bedrohungslage, die vom Probanden zu berücksichtigen war. In *Szenario I* variierte die Anzahl der SAM Stellungen dabei zwischen zehn und zwölf, in *Szenario II* zwischen neun und zehn. Die Wirkungsbereiche waren jeweils halbkugelförmig mit Radien zwischen 2000m und 5000m vorgegeben.

Flugkorridore

Die Anzahl und Position der Flugkorridore zur Überquerung der FLOT blieben über den Verlauf der Experimente unverändert. Jedoch wurde die Möglichkeit diese zu nutzen durch die Vorgaben „gesperrt“, „nicht gesperrt“ des Experimentators beeinflusst. In *Szenario I* wurde über die Sperrung eines Ausflugkorridors direkt die Auftragserfüllung verhindert.

UAV-Fähigkeiten

Das UAV wurde im Experiment voll funktionsfähig initialisiert. Während des Experiments wurde die Funktionsfähigkeit des UAV über die Deaktivierung von Fähigkeiten dynamisch durch den Experimentator beeinflusst. Systemeigene Einflussfaktoren auf die Verfügbarkeit von UAV-Fähigkeiten, z.B. durch stochastische Systemausfälle, wurden im Experiment nicht abgebildet.

5.1.3.3 Material

Als Versuchsumgebung wurde im Mensch-Maschine Versuch ein UAV-Missionssimulator genutzt, der über die technische Infrastruktur und Ausrüstung verfügt, um Messdaten zu erheben und zu protokollieren. Zusätzlich zu technischen Komponenten wurden im Versuch Fragebögen eingesetzt, die zur Erhebung subjektiver Beanspruchungs- und Empfindungsdaten dienen. Das genutzte Material ermöglichte die Operationalisierung der in Abschnitt 5.1.2 zur Hypothesenformulierung verwendeten abhängigen Variablen.

UAV-Missionssimulator und Bodenkontrollstation

Der UAV-Missionssimulator umfasst vier Komponenten: Die Bodenkontrollstation als Arbeitsplatz der Probanden, die Missionssimulation zur Sensorbildgenerierung, den Arbeitsplatz des Experimentators zur Steuerung und Manipulation der Simulation und den Arbeitsplatz zur Kalibrierung und Überwachung der Blickbewegungsmessung. Alle Komponenten sind über ein lokales Netzwerk miteinander verbunden. Abbildung 5-3 zeigt die Bodenkontrollstation mit Eingabegeräten und Anbauten sowie den Arbeitsplatz des Experimentators für die Überwachung der Blickbewegungsmessung.



Abbildung 5-3: UAV-Missionssimulator mit Eingabegeräten und Ausstattung zur Blickbewegungsmessung (links)

Die Bodenkontrollstation ist ein einzelner Operateur-Arbeitsplatz, baugleich mit denen der mobilen Bodenkontrollstation des IFS (vgl. Abschnitt 5.2.3). Die Mensch-Maschine Schnittstelle wird über zwei vertikal angeordnete Multi-Touch Bildschirme realisiert. Der untere Bildschirm zeigt dabei das Missionsinterface, der obere das Sensorinterface. Das Videosignal der Monitore wird jeweils über separate Computer erzeugt, die hinter einer Blende in den Simulator integriert sind. Zur Eingabe dienen den Probanden neben der Touch-Funktionalität der Monitore eine Maus und Tastatur zur Steuerung der Bedienoberflächen sowie ein Joystick zur Steuerung des Kamerabildes im Sensorinterface. Als Sitz dient ein Pilotensitz der C-160 Transall. Stellwände sollen die Ablenkung der Probanden durch visuelle Einflüsse von außen verhindern und werden während der Datenerhebung auch zwischen Experimentator und Bodenkontrollstation montiert. Als akustische Abschirmung während der Versuchsdurchführung dienen schalldämmende Kopfhörer.

Zur Erhebung der Blickbewegungsdaten der Probanden inkludiert die Bodenkontrollstation das Eyetracking-System *faceLAB* [Seeing Machines 2015]. Für die berührungslose Messung ist zwischen den beiden Monitoren eine Stereokamera verbaut, welche Infrarot-Lichtimpulse verarbeitet, die von dem unter dem Missionsinterface angebrachten IR-Pod emittiert werden. Mit diesem Aufbau kann die Richtung von Blickbewegungen im Bereich des unteren Bildschirms des Simulators (Missionsinterface) wahrgenommen werden. Die Kalibrierung und Steuerung des *faceLAB* Systems erfolgt im Versuch durch den Experimentator Arbeitsplatz neben der Bodenkontrollstation.

Zur Missionssimulation dient die kommerzielle Simulationssoftware *Virtual Battlespace 2* (VBS2) [Bohemia Interactive Simulations 2015], wobei die Missionsmodellierung die Terraingenerierung und die Erstellung abbildbarer Objekte und Kräfte umfasst. Während der Versuche dient VBS2 ausschließlich zur Generierung der Außenbilddarstellung zwecks Anzeige im Sensorinterface.

Hierbei wird das UAV durch die Ausgabe der Positions- und Zustandsdaten in VBS2 positioniert und mit der Kameraorientierung verrechnet.

Die Missionssimulation wird durch den Experimentator an einem externen Arbeitsplatz kontrolliert. Hierzu gehört die Steuerung der Flugdynamiksimulation sowie einer Simulationssoftware des UAV-FMS zur Abbildung einer wegpunktbasierter UAV-Flugführung. Während der Versuchsdurchführung können taktische Lageelemente und UAV-Fähigkeiten durch den Experimentator manipuliert werden.

Datenerhebungsmethoden zur Operationalisierung abhängiger Variablen

Zur Datenerhebung wurden zunächst die technisch messbaren Systemeingaben der Probanden sowie der zeitliche Verlauf der Missionssimulation genutzt. Zusätzlich wurde die Missionsdurchführung zu festen Messzeitpunkten vom Experimentator unterbrochen, um subjektive Einschätzungen über Fragebögen zu erheben.

Die technische Datenerhebung erfolgte über eine zeitsynchrone Aufnahme von Missionsdaten, der Operateureingaben und der Blickbewegungsdaten der Probanden. Hierzu wurde ein Werkzeug der *AnyCom* IPC verwendet. Die integrierte Datenerhebung ermöglicht die Korrelation des Systemverhaltens mit dem Verhalten des Operateurs sowie eine automatisierte Auswertung der Messwerte. Die abhängigen Variablen der in Abschnitt 5.1.20 vorgestellten Hypothesen *H5 – H9* können damit wie folgt operationalisiert werden:

Als *Interaktionszeit* der Probanden mit dem System wird die Zeit des Systemeingriffs bei der Initialplanung (*Kommandieren*) bzw. bei späteren Eingriffen (*Intervenieren*) verstanden. Bei der Initialplanung dient hierzu die Zeit zwischen der ersten Systeminteraktion der Probanden und der abschließenden Kommandierung an den Agenten. Bei der Umplanung startet die Zeitmessung mit der Anzeige der auslösenden Nachricht des Agenten („*Planning Failed*“) und endet ebenfalls bei Abschluss der Kommandierung. Die *Anzahl der Arbeitsschritte* des Operateurs wird als Anzahl der Klicks (über Maus oder Touchscreen) während der *Interaktionszeit* operationalisiert. Als Fehler des Operateurs wird die Kommandierung einer Auftragsagenda verstanden, die vom Agenten nicht durchgeführt werden kann. Die *Fehlerhäufigkeit* wird damit über die Anzahl der Auftragsagenden, die vom Operateur kommandiert aber vom Agenten nicht ausgeführt werden können, bevor eine korrekte Kommandierung erfolgt, operationalisiert. Der verbleibende *Treibstoffvorrats* des UAV wird bei der abschließenden Landung erhoben.

Die Bestimmung der *Aufmerksamkeitsverteilung* erfolgt für das Missionsinterface gemäß [Just & Carpenter 1980] bzw. [Deubel & Schneider 1996] anhand der Position und Dauer der Fixationspunkte des Operateurs. Hierzu wird der über faceLAB ermittelte Blickpunkt der Probanden mit dem Zustand der dynamischen Karten- und Menüdarstellung korreliert. Zur Bestimmung der Aufmerksamkeit der Probanden hinsichtlich der vom Agenten bereitgestellten Informationen werden hierbei die Verweildauer ihres Blickes auf angezeigten Dialogfenstern und dem Agentenplan verwendet. Eine genaue Beschreibung zur Messwerterhebung mittels faceLAB für dieses Experiment gibt [Werner 2014b].

Zur Abbildung der *Wegbereiter menschlichen Verhaltens* werden ausschließlich intrusive, subjektive Erhebungsmethoden über Fragebögen genutzt. Der zeitliche Ablauf der Befragungen im Experiment ist in Abschnitt 5.1.3.4 dargestellt.

Zur Ermittlung der *subjektiven Beanspruchung* der Probanden im Experiment wird der NASA-Task Load Index (NASA-TLX) nach [Hart & Staveland 1988] verwendet. NASA-TLX stellt ein multidimensionales Bewertungsverfahren dar, welches aus einem gewichteten Mittel der Bemessungsgrößen von sechs Subskalen eine Beanspruchung ermittelt. Für das Ausfüllen des NASA-TLX Fragebogens ist eine Unterbrechung der Simulation und der Auftragserfüllung notwendig. Zum besseren Verständnis der Dimensionen und ihrer Bewertungen wurde für die deutschsprachigen Probanden im Versuch eine deutsche Übersetzung des originalen NASA-TLX Fragebogens eingesetzt (vgl. Anhang C.1). Der englischsprachige Proband füllte hingegen den NASA-TLX Fragebogen im englischen Original aus.

Das *Situationsbewusstsein* von Probanden im Experiment umfasst, analog zur Kategorisierung in Abschnitt 2.3.4, die Wahrnehmung von Sinneseindrücken aus der Umwelt (Ebene 1), das Verständnis dieser Sinneseindrücke (Ebene 2) und die Nutzung dieses Verständnisses zur Projektion zukünftiger Umgebungszustände (Ebene 3). Zur Quantifizierung des Situationsbewusstseins der Probanden wird die in [Endsley 1988] für Piloten beschriebene *Situation Awareness Global Assessment Technique* (SAGAT) verwendet. Die Datenerhebung erfolgte entsprechend der SAGAT Vorgaben. Zu definierten Zeitpunkten wurde die Simulation gestoppt und die Monitore in der Bodenkontrollstation abgeschaltet. Anschließend wurden den Probanden situationspezifische Fragen zum aktuellen Umgebungszustand vorgelegt. Hierbei waren die Fragen mit *Ja* oder *Nein* zu beantworten, Zahlenwerte anzugeben oder Zeichnungen anzufertigen. Für jede Mission wurden auf diese Weise zwei Fragebögen mit jeweils acht Fragen von den Probanden befüllt, wobei pro Fragebogen maximal 18 Punkte zu erreichen waren. Die Punktevergabe richtete sich dabei nach der jeweiligen Menge des abgefragten Wissens. SAGAT wurde gemeinsam in einer Unterbrechung mit dem NASA-TLX Fragebogen ausgefüllt, um die Anzahl an Unterbrechungen gering zu halten. Anhang C.2 listet die im Rahmen der Datenerhebung verwendeten Fragebögen auf.

Um Messdaten bezüglich des *Vertrauens* und der *Akzeptanz* der Probanden in den kognitiven Agenten zu erheben, wurde zudem ein individuell erstellter subjektiver Fragebogen verwendet, der abschließend im Anschluss an die Szenarien von den Probanden zu befüllen war. Dieser in Anhang C.3 dargestellte Fragebogen umfasste die vier Bewertungskategorien *Systeminteraktion*, *Systemverhalten bei Lageänderung*, *systemseitig bereitgestellte Informationen* und *Gesamtsystem* sowie einen Interviewteil zum Notieren mündlicher Aussagen der Probanden. Eine genaue Beschreibung des Fragebogens und der Klassifizierung von Antwortmöglichkeiten findet sich in [Borchers 2014b].

Tabelle 5-2 zeigt mit Bezug auf die Beschreibung der Bewertungskategorien in Anhang C.3 Kriterien bei der Operationalisierung der Variable *Vertrauen*. Die Vertrauensbildung erfolgt bei der Systeminteraktion im Wesentlichen durch die Verarbeitung von Leistungsinformationen, die der Operateur vom System erhält. Der Operationalisierung der Variable *Vertrauen* dienten damit die Angaben der Probanden über die Leistungsinformationen hinsichtlich des Verhaltens und der Interaktion des Systems. Die Quantifizierung der Variable erfolgte über den Mittelwert der insgesamt zehn Bewertungskriterien gemäß Tabelle 5-2.

Tabelle 5-2: Bewertungskriterien zur Operationalisierung der abstrakten Variable „Vertrauen“

Kategorie	Kriterium
Systeminteraktion	zielführend
Systemverhalten bei Lageänderungen	verständlich informativ vorhersehbar nachvollziehbar zielführend
Systemseitig bereitgestellte Informationen	verständlich
Gesamtsystem	Das System führte meine Aufträge immer richtig aus. Ich fühlte mich über den gegenwärtigen Missionsverlauf stets gut informiert. Das System stellt die Mission realitätsnah dar.

Die komplexe Variable *Akzeptanz* beschreibt die Bereitschaft der Probanden, das System für die Auftragserfüllung einzusetzen. Sie stellt damit auf der Grundlage der Erfahrungen in der Interaktion mit dem System eine Wertung und positive Einstellung hinsichtlich zukünftigen Verhaltens dar. *Akzeptanz* wurde im Versuch allgemein über alle Bewertungskategorien des subjektiven Fragebogens kategorisiert. Die Quantifizierung erfolgte damit als Mittelwert der Einzelbewertungen aus allen vier Kategorien. Die Variable *Akzeptanz* wurde damit separat für jede Kategorie abgebildet.

5.1.3.4 Ablauf

Die Versuchsdurchführung folgte einem definierten zeitlichen Ablaufplan, der die Reihenfolge und Dauer von Handlungen festlegte. Die Handlungen gliederten sich dabei in fünf Phasen, die insgesamt eine Versuchsdauer von 162 Minuten ergaben. Tabelle 5-3 zeigt den Ablaufplan des Experiments mit seinen Phasen, Handlungen und angesetzten Handlungsdauern. Variationen bei der Versuchsdauer lassen sich dabei auf individuelle Unterschiede im Verhalten der Probanden zurückführen.

Während des gesamten Experiments betreuten zwei Experimentatoren die Versuchsdurchführung. Der erste Experimentator nahm die Betreuung der Probanden und die Bedienung des faceLAB Systems wahr, der zweite Experimentator kontrollierte die Missionssimulation und die Einspielung szenarienspezifischer Ereignisse.

Die Vorbereitungsphase begann mit der Begrüßung der Probanden und der Vorstellung des Testablaufs durch den Experimentator. Es folgte eine Erhebung statistischer Daten (*Angaben zur Person*) der Probanden. Im Anschluss justierte und kalibrierte der Experimentator das faceLAB System und erläuterte die Grundlagen der Bedienung der Bodenkontrollstation und des kognitiven Agenten. Der Proband erhielt daraufhin Zeit, sich mit dem System bekannt zu machen und führte anschließend die Trainingsmission selbstständig durch. Abschließend erfolgte die Klärung offener Fragen des Probanden durch den Experimentator.

Tabelle 5-3: Ablaufplan der Mensch-Maschine Versuche

Phase	Handlung / Szenarienphase	Dauer [min]	Phasendauer [min]	Summierte Dauer [min]
Vorbereitung	Begrüßung	2		2
	Vorstellung des Testablaufs durch Experimentator	5		7
	Ausfüllen Fragebogen "Angaben zur Person"	1		8
	Anpassen des Arbeitsplatzes, Kalibrierung faceLAB	12		20
	Einweisung in Bedienelemente, Anzeigen und Verhalten des Systems	13		33
	Briefing Testszenario	2		35
	Durchführung Testszenario	14		49
	Nachbesprechung und Klärung offener Fragen	2	51	51
Szenario I	Briefing	2		53
	Planung & Kommandierung gemäß Lage und Auftrag	1		54
	Simulationsunterbrechung	0		54
	Erhebung Fragebogen NASA-TLX	2		56
	Take-Off von <i>Home Base</i> und Transit zu <i>Northern Bay</i>	8		64
	Erstellung von SAM-Stellungen abseits der Flugroute und über <i>Potential Ammunition Depot</i>	0		64
	Fahrzeugaufklärung über <i>Northern Bay</i> und Transit zu <i>Coastal Road</i>	2		66
	Während des Transits: Deaktivierung der Fähigkeit Fahrzeugerkennung	0		66
	Manuelle Umplanung durch Operateur (Auslassen von Zielen: <i>Coastal Road</i> und <i>Potential Ammunition Depot</i>)	2		68
	Simulationsunterbrechung nach Abschluss Umplanung	0		68
	Erhebung Fragebogen SAGAT	5		73
	Erhebung Fragebogen NASA-TLX	2		75
	Aufklärung von <i>Potential Headquarter</i> und Transit zu <i>Home Base</i>	4		79
	Sperrung des auf Flugroute liegenden Korridors und weitest entfernten Korridors	0		79
	Simulationsunterbrechung (vor automatischer Umplanung durch Agenten)	0		79
	Erhebung Fragebogen SAGAT	5		84
	Erhebung Fragebogen NASA-TLX	2		86
	Transit zu <i>Home Base</i> und Landung	6		92
	Debriefing mit Erhebung Fragebogen "Subjektive Bewertung"	7	48	99
	Szenario II	Briefing	2	
Planung & Kommandierung gemäß Lage und Auftrag		1		102
Simulationsunterbrechung		0		102
Erhebung Fragebogen NASA-TLX		2		104
Take-Off von <i>Home Base</i> und Transit zu <i>Northern Bay</i>		4		108
nach Passage des Korridors: Erstellung von SAM-Stellung auf der Flugroute		0		108
Simulationsunterbrechung (vor automatischer Umplanung durch Agenten)		0		108
Erhebung Fragebogen SAGAT		5		113
Erhebung Fragebogen NASA-TLX		2		115
Fahrzeugaufklärung über <i>Northern Bay</i> und <i>Coastal Road</i>		10		125
Transit zu <i>Potential Ammunition Depot</i>		1		126
Sperrung von <i>Corridor-A</i> und Deaktivierung der Fähigkeit Fahrzeugerkennung		0		126
Manuelle Umplanung durch Operateur (Rückkehr zu <i>Home Base</i>)		2		128
Simulationsunterbrechung nach Abschluss Umplanung		0		128
Erhebung Fragebogen SAGAT		5		133
Erhebung Fragebogen NASA-TLX		2		135
Transit zu <i>Home Base</i> und Landung		7		142
Debriefing mit Erhebung Fragebogen "Subjektive Bewertung"	7	50	149	
Szenario III	Briefing	2		151
	Durchführung Szenario III	20		171
	Debriefing	5	27	176
Nachbereitung	Abschlussbesprechung	6		182
	Verabschiedung	1	7	183

Szenario I wurde im Anschluss an ein eingehendes Briefing von einer Hälfte der Probanden in Systemkonfiguration A und der anderen Hälfte in Systemkonfiguration B durchlaufen. Im Versuch wurden zu zwei Zeitpunkten SAGAT und drei Mal NASA-TLX Messdaten erhoben. Die erste NASA-TLX Messung erfolgte im direkten Anschluss an die Kommandierung der initialen Auftragsagenda durch den Probanden. Im Verlauf der Mission wurde nach erfolgter Umplanung durch den Probanden (als Reaktion auf den Ausfall der Aufklärungsfähigkeit des UAV) die Missionssimulation zur Erhebung von SAGAT und NASA-TLX unterbrochen. Eine weitere Unterbrechung erfolgte unmittelbar vor der automatischen Umplanung durch den kognitiven Agenten. Auch hier wurden zu beiden Fragebögen Messdaten erhoben. Im abschließenden Debriefing wurde der Fragebogen *Subjektive Bewertung* durch den Probanden befüllt.

Szenario II wurde von den Probanden in jeweils umgekehrter Reihenfolge mit den beiden Systemkonfigurationen durchlaufen. Die Anzahl der Messpunkte entsprach dabei der in *Szenario I*. Nach der initialen Kommandierung erfolgte eine erste Unterbrechung für das Ausfüllen eines NASA-TLX Fragebogens. Unmittelbar vor der automatischen Routenumplanung durch den Agenten erfolgten eine weitere Unterbrechung und die Erhebung von SAGAT und NASA-TLX Messdaten. Im Anschluss an die Umplanung durch den Probanden (als Reaktion auf den Ausfall der Fähigkeit zur Fahrzeugerkennung des UAV) wurde die Missionssimulation unterbrochen und Messdaten erhoben. Auch in *Szenario II* erfolgte eine Erhebung des Fragebogens *Subjektive Bewertung* als Teil des Debriefings.

Im Rahmen der Versuchsdurchführung wurde von den Probanden anschließend ein drittes *Szenario* durchlaufen, welches zur Evaluierung von separaten Assistenzfunktionen diente, jedoch für diese Arbeit keine Bedeutung hat und deswegen im Folgenden nicht weiter betrachtet wird.

In der Nachbereitungsphase erfolgten eine Abschlussbesprechung und die Verabschiedung der Probanden durch den Experimentator. Die Nachbereitung diente der abschließenden Aufnahme von Anmerkungen der Versuchspersonen sowie der Besprechung der Probandenleistung.

5.1.4 Ergebnisse

Die Versuchsergebnisse werden im Folgenden im Abgleich mit den in Abschnitt 5.1.2 beschriebenen Hypothesen dargestellt. Hierzu erfolgt zu jeder Hypothese eine Darstellung der Messergebnisse in Form einer deskriptiven Statistik über den Mittelwert (M) und die Standardabweichung (engl. *Standard Deviation* – SD). Abweichende Stichprobengrößen in den Ergebnistabellen ergeben sich durch systeminterne Fehler des Versuchsaufbaus durch die Daten teilweise nicht korrekt protokolliert werden konnten. Zudem erfolgen eine mathematische Auswertung der abhängigen Variablen und eine Bewertung der zugehörigen Hypothesen. Die grafische Darstellung der Ergebnisse erfolgt farblich kodiert, wobei Konfiguration A jeweils orange und Konfiguration B grün dargestellt wird.

Die statistische Auswertung der Messergebnisse erfolgt mittels der Software SPSS [IBM Corp. 2015b], mit der ein nicht-parametrischer Wilcoxon-Rangsummentest auf die verbundenen Stichproben angewendet wird [Borchers 2014b]. Die Messdurchläufe der beiden Agentenkonfigurationen werden dabei als verbundene Stichproben behandelt, wobei ein signifikanter Unterschied zwischen den Messreihen ab einem exakten Signifikanzwert von $p < .05$ angenommen wird. Trotz der einseitig formulierten Versuchshypothesen (vgl. Abschnitt 5.1.2) erfolgt die Signifikanzprüfung in jedem Fall zweiseitig, da ein jeweils andersartiger oder umgekehrter Zusammenhang nicht per se ausgeschlossen werden kann.

„[...] a one-tailed test (a) [...] can never lead us to conclude that we are wrong about the direction of the population effect and (b) is therefore never appropriate in a research setting.“ [Harris 1997]

Soweit wie möglich erfolgt eine Differenzierung zwischen der Ausprägung der Messergebnisse in der initialen Planungs- und Kommandierungsphase (PuK) und der späteren automatischen Umplanung (AU) bzw. manuellen Intervention (MInt) während der Missionsdurchführung.

H1 – Beanspruchung

Der Ermittlung des Beanspruchungswertes der Probanden liegen pro Agentenkonfiguration jeweils drei Erhebungen des NASA-TLX Fragebogens zu Grunde (vgl. Tabelle 5-3). Dies erlaubt neben der Gegenüberstellung der beiden Konfigurationen einen Vergleich der drei Missionsphasen. Die mathematische Statistik bezüglich der dabei gewonnenen Ergebnisse des NASA-Task Load Index ist in Tabelle 5-4 dargestellt. Hinsichtlich der Beanspruchung der Probanden können keine signifikanten Unterschiede zwischen den beiden Agentenkonfigurationen identifiziert werden.

Tabelle 5-4: Mathematische Statistik zu den Ergebnissen des NASA-Task Load Index

NASA-Task Load Index						
	Initiale Planung & Kommandierung		Automatische Umplanung		Manuelle Intervention	
	A (n=14)	B (n=14)	A (n=14)	B (n=14)	A (n=14)	B (n=14)
M [%]	17.57	20.50	29.71	31.29	38.79	37.86
SD [%]	12.377	17.531	17.331	16.868	17.125	12.996
Exakte Sig. (2-seitig)	.414		.599		.946	

Es ergibt sich ein Anstieg des mittleren Beanspruchungswertes zwischen den Missionsphasen von $19.04\% \pm 14.965\%$ (PuK) über $30.5\% \pm 16.801\%$ (AU) zu $38.33\% \pm 14.925\%$ (MInt), der damit über alle Messphasen hinweg in der unteren Belastungshälfte bleibt. Große Interpersonelle Unterschiede bei NASA-TLX führen in allen drei Phasen zu einer starken Streuung der Ergebnisse, die eine Unterscheidung der beiden Agentenkonfigurationen unmöglich machen. Abbildung 5-4 zeigt hierzu die Bandbreite der Messergebnisse. Bei der PuK in Konfiguration A wurden z.B. Werte zwischen 3% und 45% gewonnen, bei Konfiguration B Werte zwischen 2% und 52%.

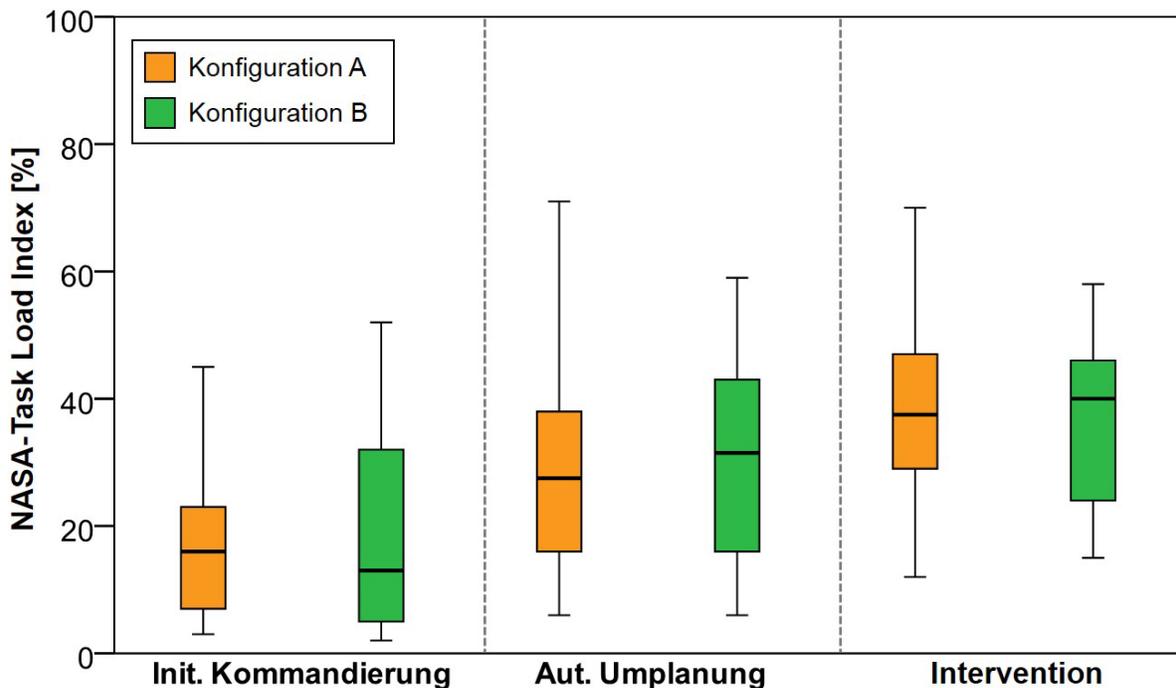


Abbildung 5-4: NASA-Task Load Index bei Konfiguration A und B für die drei Missionsphasen

Es kann somit keine Aussage bezüglich der Beeinflussung der Beanspruchung der Probanden durch die Ausprägung der Agenteninteraktion gemacht werden. Versuchshypothese *H1* bezüglich der Beanspruchung der Probanden in Abhängigkeit von der Agentenkonfiguration wird somit nicht bestätigt.

H2 – Situationsbewusstsein

Die Bewertung des Situationsbewusstseins der Probanden erfolgt über die Ergebnisse der SAGAT Fragebögen, welche gemäß Tabelle 5-3 in den Testszenarien jeweils zum Zeitpunkt der AU und der MInt erhoben wurden. Die statistische Auswertung der Ergebnisse des SAGAT Tests ist in Tabelle 5-5 dargestellt. Sie zeigt keine signifikanten Unterschiede zwischen Konfiguration A und Konfiguration B.

Tabelle 5-5: Mathematische Statistik zu den Ergebnissen des SAGAT Fragebogens

SAGAT Gesamtergebnisse				
	Automatische Umplanung		Manuelle Intervention	
	A (n=14)	B (n=14)	A (n=14)	B (n=14)
M ^a	13.54	12.61	11.14	12.46
SD	2.098	1.933	1.975	2.756
Exakte Sig. (2-seitig)	.194		.122	
^a . Die Bewertungsskala reicht von 0 bis 18 Punkten				

Tabelle 5-5 zeigt einen signifikanten Abfall des Situationsbewusstseins von der AU zur MInt, mit einer Verringerung der mittleren Gesamtpunktzahl des SAGAT Fragebogens von 13.07 ± 2.04 (AU) auf 11.80 ± 2.45 (MInt). Mit Bezug auf die Agentenkonfiguration lässt sich diese Verringerung hauptsächlich auf eine Minderung des Situationsbewusstseins in Konfiguration A zurückführen, deren Wert von $13,54 \pm 2.098$ auf 11.14 ± 1.975 schrumpft. Die Ergebnisse bei Konfiguration B bleiben nahezu konstant.

Der direkte Vergleich der beiden Agentenkonfigurationen ergibt keinen signifikanten Unterschied. Insgesamt liegen die erzielten Punktzahlen zwischen 9.5 und 17.5 (AU) bzw. 8.0 und 16.0 (MInt) und damit überwiegend in der oberen Hälfte der Bewertungsskala. Die große interpersonelle Streuung erschwert zudem die Bewertung der Agentenkonfigurationen. Versuchshypothese *H2* bezüglich des Situationsbewusstseins der Probanden in Abhängigkeit der Agentenkonfiguration kann durch die erhobenen Messdaten nicht bestätigt werden.

H3 – Vertrauen in den Agenten

Das Vertrauen der Probanden in den Agenten wird als komplexe Variable gem. Abschnitt 5.1.3.3 über verschiedene Kriterien des Fragebogens „Subjektive Bewertung“, welcher jeweils zum Ende der beiden Testszenarien erhoben wurde, operationalisiert.

Tabelle 5-6 zeigt die statistische Auswertung der dabei gewonnenen Ergebnisse als Gegenüberstellung der beiden Agentenkonfigurationen. Es ergibt sich ein signifikanter Unterschied zwischen beiden Stichproben.

Table 5-6: Mathematical statistics to the results of the questionnaire „Subjective evaluation“ regarding the criteria for determining the abstract variable „Trust in the agents“

Vertrauen in den Agenten		
	A (n=14)	B (n=14)
M ^a	4.44	5.06
SD	1.084	0.772
Exakte Sig. (2-seitig)	.008*	
^a . Der Wertebereich umfasst 1 bis 7 Punkte, wobei mehr Punkte ein größeres Vertrauen in den Agenten beschreiben *. Stichprobe B ist signifikant größer als Stichprobe A mit $p < .05$		

In Configuration A the trust in the agents was evaluated by the participants with 4.44 ± 1.084 , in Configuration B with 5.06 ± 0.772 , where ratings between one and seven points were possible. It results in an exact significance (two-sided) of $p = .008$ and thus a significantly higher result for sample B than for sample A.

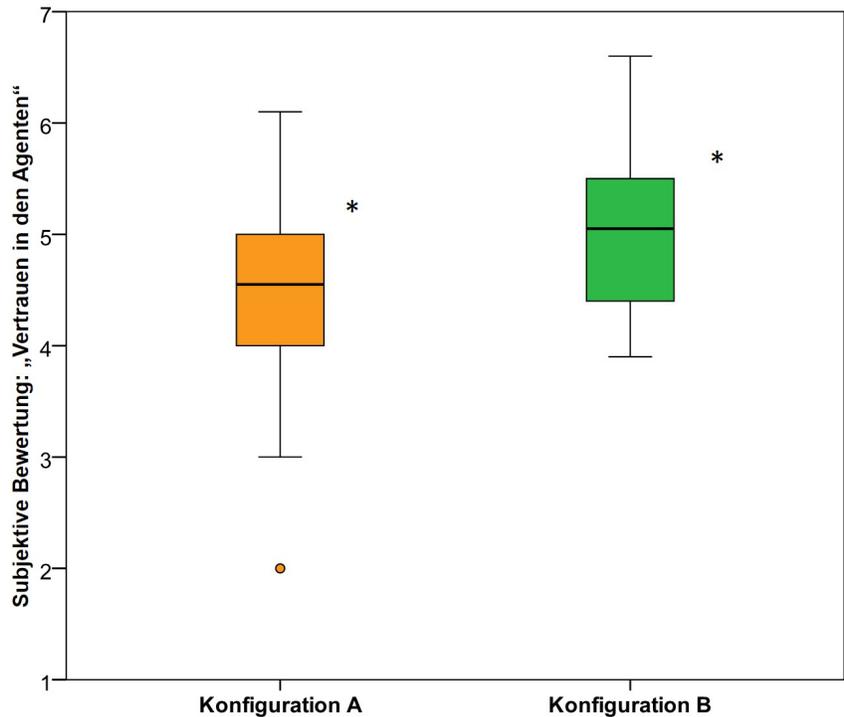


Abbildung 5-5: Bewertung des Vertrauens in den Agenten durch die Probanden in Konfiguration A und B.

Abbildung 5-5 stellt die Bewertungsergebnisse grafisch dar. Insgesamt wurde das Vertrauen in Konfiguration A zwischen 2.00 und 6.10 bewertet, während Konfiguration B Bewertungen zwischen 3.90 und 6.60 erhielt. Der signifikante Unterschied zwischen Konfiguration A und B bestätigt Hypothese H3 hinsichtlich des Einflusses der Agentenkonfiguration auf das Vertrauen der Probanden in den kognitiven Agenten.

H4 – Akzeptanz

Die Bewertung der Akzeptanz des Systems durch die Probanden gliedert sich gemäß Abschnitt 5.1.2 in die vier Kriterien *Systeminteraktion* (SI), *Lageänderung* (LÄ), *Bereitgestellte Informationen* (BI) und *Gesamtsystem* (GS). Tabelle 5-7 zeigt die statistische Auswertung der Kriterien gemäß des Fragebogens „Subjektive Bewertung“. Die Auswertung zeigt ein geteiltes Ergebnis. Bei den Kriterien LÄ und GS zeigen beide signifikante Unterschiede zwischen Konfiguration A und Konfiguration B. Bei den Kriterien SI und BI zeigen sich keine signifikanten Unterschiede.

Tabelle 5-7: Mathematische Statistik zu den Ergebnissen des Fragebogens „Subjektive Bewertung“ bezüglich der Kriterien zur Bestimmung der abstrakten Variable „Akzeptanz“

Akzeptanz								
	System- interaktion		Lageänderung		Bereitgestellte Informationen		Gesamtsystem	
	A (n=14)	B (n=14)	A (n=14)	B (n=14)	A (n=14)	B (n=14)	A (n=14)	B (n=14)
M ^a	5.16	5.45	3.95	4.70	4.32	4.50	4.63	5.42
SD	1.085	1.121	1.461	1.121	0.529	0.340	1.021	0.775
Exakte Sig. (2-seitig)	.142		.011*		.283		.005*	
^a . Der Wertebereich umfasst jeweils 1 bis 7 Punkte, wobei mehr Punkte einen höheren Akzeptanzwert beschreiben *. Stichprobe B jeweils signifikant größer als Stichprobe A, da $p < 0.05$								

Der Akzeptanzwert liegt für SI bei 5.16 ± 1.085 (A) bzw. 5.45 ± 1.121 (B), für LÄ bei 3.95 ± 1.461 (A) bzw. 4.70 ± 1.121 (B), für BI bei 4.32 ± 0.529 (A) bzw. 4.50 ± 0.340 (B) und für GS bei 4.63 ± 1.021 (A) bzw. 5.42 ± 0.775 (B). Für die Bewertung der Akzeptanz sind dabei zwischen eins und sieben Punkte möglich. Dies führt zu exakten Signifikanzwerten von $p_{SI} = .142$, $p_{LÄ} = .011$, $p_{BI} = .283$ bzw. $p_{GS} = .005$. Für LÄ und GS ergeben sich somit für Stichprobe B signifikant höhere Akzeptanzwerte als für Stichprobe A.

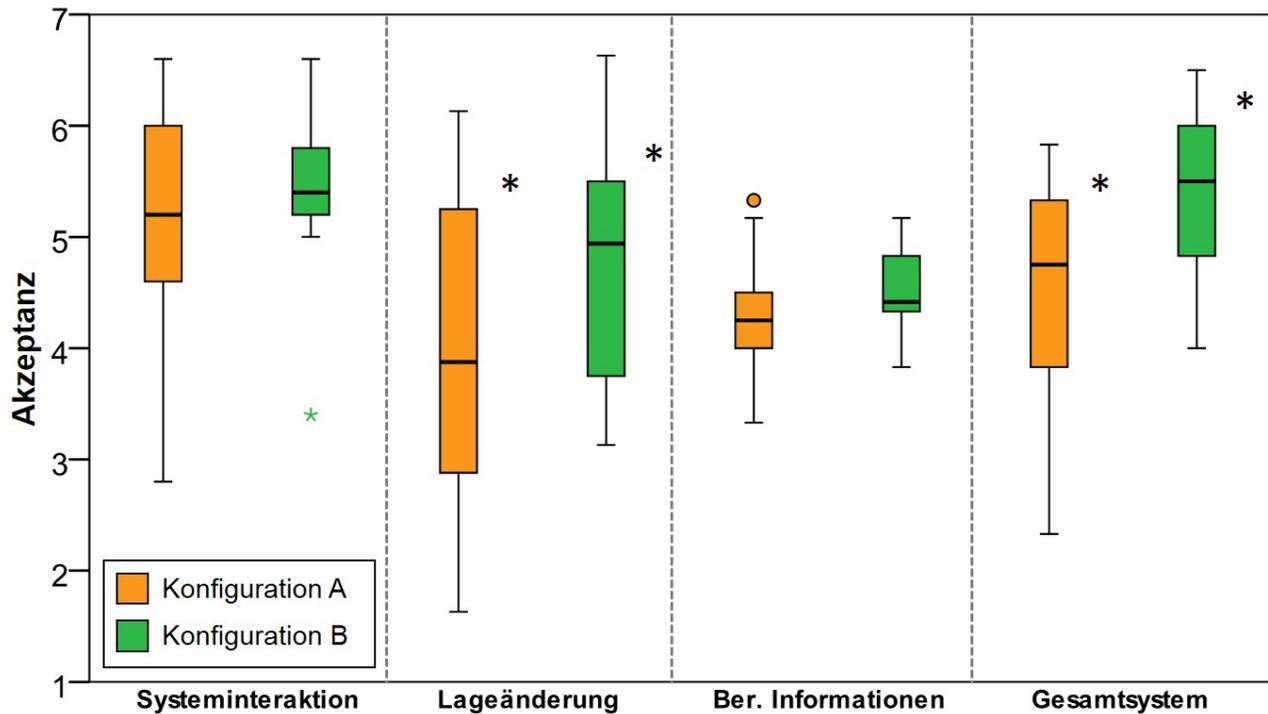


Abbildung 5-6: Bewertung der Akzeptanz der Systemkonfigurationen anhand von vier Bewertungskriterien

Die Akzeptanzwerte sind in Abbildung 5-6 für die einzelnen Kriterien in Form einer Gegenüberstellung der Agentenkonfigurationen grafisch dargestellt. Hierbei ist zu erkennen, dass besonders für Konfiguration A große interpersonelle Unterschiede bei der Bewertung der jeweiligen Akzeptanzkriterien vorliegen. Für das Kriterium LÄ liegen z.B. Bewertungen zwischen 1.63 und 6.13 Punkten vor. Gleiches gilt für das Kriterium GS mit Werten zwischen 2.33 und 5.83 Punkten.

Hypothese *H4* hinsichtlich des Einflusses der Agentenkonfiguration auf die Akzeptanz des Systems durch die Probanden wird damit für die Kriterien b) Lageänderung und d) Gesamtsystem bestätigt. Für die Kriterien a) Systeminteraktion und c) Bereitgestellte Informationen wird die Hypothese hingegen nicht bestätigt.

H5 – Aufmerksamkeitsverteilung

Zur Bestimmung der Aufmerksamkeitsverteilung werden gemäß Abschnitt 5.1.3.3 Fixationspunkte und –dauern auf dem Missionsinterface verwendet, welche über *faceLAB* erhoben wurden. Zur Bestimmung der Intensität der Aufmerksamkeit auf die vom Agenten bereitgestellten Informationen, diente die Messung der Verweildauer auf dem Agentenplan und angezeigten Dialogfenstern. Tabelle 5-8 zeigt hierzu die statistische Auswertung der Blickbewegungsdaten mit den Verweildauern im relevanten Bildschirmbereich anteilig an der Szenariendauer.

Tabelle 5-8: Mathematische Statistik hinsichtlich der mittels *faceLAB* gemessenen Fixationsdauern der Probanden auf Bereiche mit vom Agenten bereitgestellten Informationen

Fixationsdauern auf Bildschirmbereiche zur Anzeige von Informationen des kognitiven Agenten		
	A (n=13)	B (n=14)

M [%] ^a	3.91	2.59
SD [%] ^a	2.695	1.722
Exakte Sig. (2-seitig)	.635	
^a . Fixationsdauern sind anteilig an der jeweiligen Szenariendauer		

Die mittlere Fixationsdauer des relevanten Bildschirmbereichs liegt in Konfiguration A bei $3.91 \pm 2.695\%$, in Konfiguration B bei $2.59 \pm 1.722\%$. Hieraus ergibt sich kein signifikanter Unterschied der Stichproben. Die Fixation der vom Agenten bereitgestellten Informationen ist in beiden Konfigurationen mit jeweils unter fünf Prozent gering. [Werner 2014b] gibt eine ausführliche Auswertung der Fixationsdauern über allen Bildschirmbereichen, wobei für keinen Bereich ein signifikanter Unterschied identifiziert werden kann.

Hypothese *H5* hinsichtlich der Abhängigkeit der Aufmerksamkeitsverteilung der Probanden von der jeweiligen Agentenkonfiguration wird somit nicht bestätigt.

H6 – Anzahl von Arbeitsschritten

Die Systemeingaben der Probanden wurden über den Zeitraum der Testszenarien protokolliert. Die Erhebung der Anzahl von Arbeitsschritten zur Auftragsumsetzung erfolgte unter Nutzung der Systemeingaben der Probanden im Verlauf der PuK bzw. der MInt. Für die manuelle Intervention ergibt sich dabei ein signifikanter Unterschied zwischen Konfiguration A und Konfiguration B (vgl. Tabelle 5-9).

Tabelle 5-9: Mathematische Statistik zur Anzahl von Arbeitsschritten, die zur Auftragsumsetzung in den beiden Agentenkonfigurationen benötigt werden

Anzahl von Arbeitsschritten				
	Initiale Planung & Kommandierung		Manuelle Intervention	
	A (n=13)	B (n=14)	A (n=13)	B (n=12)
M	28.85	37.64	68.46	23.00
SD	12.047	33.790	51.312	21.612
Exakte Sig. (2-seitig)	.554		.000*	
*. Stichprobe B benötigt bei der Intervention signifikant weniger Arbeitsschritte als Stichprobe A, da $p < 0.05$				

Bei der PuK werden in Konfiguration A 28.85 ± 12.047 und in Konfiguration B 37.64 ± 33.790 Arbeitsschritte zur Auftragsumsetzung benötigt. Bei der MInt ergibt sich in Konfiguration A eine Anzahl von 68.46 ± 51.312 und in Konfiguration B von 23 ± 21.612 Arbeitsschritten. Hieraus ergibt sich für die MInt eine exakte Signifikanz von $p = .000$. In Konfiguration B werden somit signifikant weniger Arbeitsschritte zur Auftragsumsetzung benötigt als in Konfiguration A.

Abbildung 5-7 stellt die Anzahl der Systeminteraktionen für die PuK sowie die MInt grafisch dar. Die Abbildung zeigt für die MInt neben der Verringerung der mittleren Interaktionsanzahl insbesondere auch eine Verkleinerung der Streuung der Ergebnisse. Während in Konfiguration A zwischen 12 und 185 Arbeitsschritte benötigt werden, reduziert sich diese Anzahl in Konfiguration B auf 4 bis 27 Schritte (mit einem Ausreißer bei 85 Arbeitsschritten). Bei der PuK zeigt die Erhöhung der Anzahl von Arbeitsschritten keine Signifikanz.

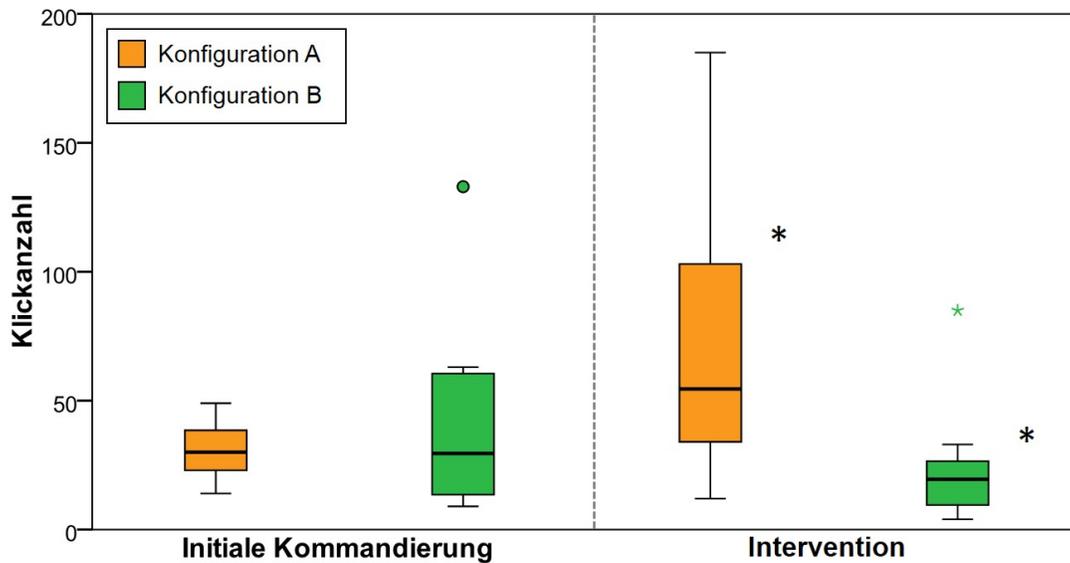


Abbildung 5-7: Anzahl der zur Auftragsumsetzung von den Probanden benötigten Arbeitsschritte

Hypothese *H6* hinsichtlich der Abhängigkeit der Anzahl der benötigten Arbeitsschritte zur Auftragsumsetzung der Agentenkonfiguration wird für die manuelle Intervention, nicht jedoch für die initiale Phase der Planung und Kommandierung bestätigt.

H7 – Interaktionszeit

Die Erhebung der Interaktionszeit der Probanden erfolgt über die Protokollierung der Systeminteraktionen über den Verlauf der Testszenarien. Die Erhebung der Interaktionszeit ergibt sich dabei gemäß Abschnitt 5.1.3.3. Tabelle 5-10 zeigt die statistische Auswertung der Messergebnisse, aus der sich ein signifikanter Unterschied zwischen Konfiguration A und Konfiguration B ergibt.

Tabelle 5-10: Mathematische Statistik zu der Interaktionsdauer, die zur Auftragsumsetzung in den beiden Agentenkonfigurationen benötigt wird

Interaktionszeit				
	Initiale Planung & Kommandierung		Manuelle Intervention	
	A (n=13)	B (n=14)	A (n=13)	B (n=12)
M [s]	33.84	38.88	112.57	50.03
SD [s]	16.763	27.158	58.380	44.544
Exakte Sig. (2-seitig)	.588		.012*	
*. Stichprobe B erzielt bei der Intervention signifikant kürzere Zeiten als Stichprobe A, da $p < 0.05$				

Die Interaktionszeit beträgt bei der PuK in Konfiguration A 33.84 ± 16.763 Sekunden und in Konfiguration B 38.88 ± 27.158 Sekunden. Für die MInt beträgt die Interaktionsdauer in Konfiguration A 112.57 ± 58.380 Sekunden und in Konfiguration B 50.03 ± 44.544 Sekunden. Hieraus ergibt sich für die MInt eine exakte Signifikanz von $p = .012$, wodurch sich eine signifikante

Verringerung der Interaktionszeit in Konfiguration B ergibt. Für die PuK kann hingegen kein signifikanter Unterschied identifiziert werden.

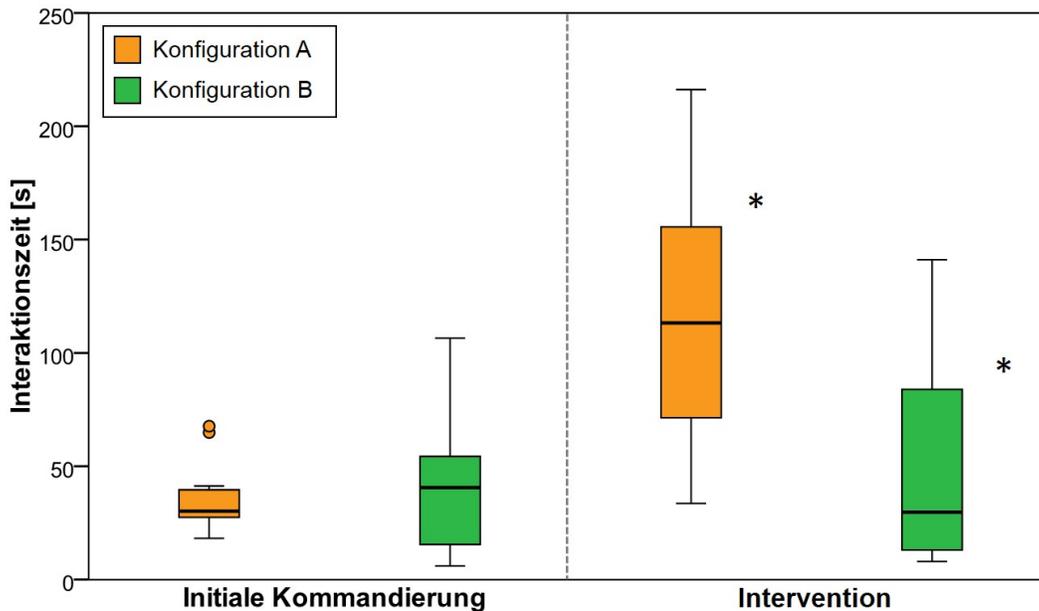


Abbildung 5-8: Dauer der Systeminteraktion zur Auftragsumsetzung durch die Probanden

Bei der Betrachtung der Ergebnisse in Abbildung 5-8 ergibt sich ein vergleichbares Bild. Bei der MInt ist sowohl eine signifikante Verringerung der Interaktionszeit als auch eine Verringerung der Streuung zu identifizieren. In Konfiguration A werden Dauern zwischen 33.6 und 216.2 Sekunden, in Konfiguration B Dauern zwischen 8.0 und 141.1 Sekunden für die Auftragsumsetzung erhoben.

Hypothese *H7* hinsichtlich der Abhängigkeit der Interaktionsdauer für die Auftragsumsetzung der Agentenkonfiguration wird für die manuelle Intervention bestätigt, nicht jedoch für die initiale Planung und Kommandierung.

H8 – Fehlerhäufigkeit

Die Erhebung der Fehlerhäufigkeit erfolgt über die Messung der fehlerhaft kommandierten Aufträge an den kognitiven Agenten im Rahmen der MInt. Hierdurch ergibt sich pro Testszenario jeweils ein Messpunkt zur Datenerhebung. Tabelle 5-11 zeigt die mathematische Statistik zu den Ergebnissen der fehlerhaften Systemeingaben. Die Auswertung zeigt einen signifikanten Unterschied zwischen den beiden Agentenkonfigurationen.

Tabelle 5-11: Mathematische Statistik zu der Anzahl der fehlerhaften Systemeingaben bei der Auftragsumsetzung in den beiden Agentenkonfigurationen

Fehlerhäufigkeit		
	A (n=13)	B (n=13)
M	2.15	0.62
SD	0.899	0.961
Exakte Sig. (2-seitig)	.002*	
*. Die Fehlerhäufigkeit ist bei Stichprobe B signifikant geringer als bei Stichprobe A mit $p < .05$		

In Konfiguration A treten im Mittel 2.15 ± 0.899 Fehler bei der MInt auf, in Konfiguration B im Mittel 0.62 ± 0.961 . Dies ergibt eine exakte Signifikanz von $p = .002$, womit in Konfiguration B die Fehlerhäufigkeit im Vergleich zu Konfiguration A signifikant gesenkt wird. Abbildung 5-9 zeigt eine grafische Darstellung der zugrunde liegenden Eingabefehler.

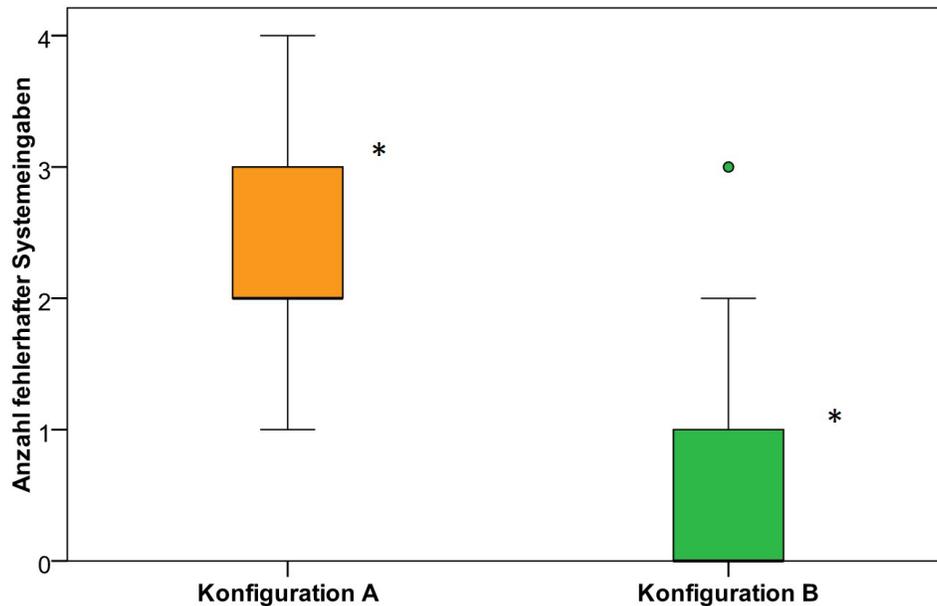


Abbildung 5-9: Anzahl fehlerhafter Systemeingaben bei der manuellen Intervention durch den Operateur

In Konfiguration A werden zwischen einem und vier fehlerhafte Aufträge kommandiert, in Konfiguration B zwischen null und zwei fehlerhafte Aufträge (mit einem Ausreißer bei drei). Insgesamt können in Konfiguration B acht Probanden die manuelle Intervention fehlerfrei durchführen, während dies in Konfiguration A keinem Probanden gelingt.

Hypothese H_8 hinsichtlich der Abhängigkeit der Fehlerhäufigkeit bei der Auftragskommandierung wird somit bestätigt.

H9 – Umsetzung der Arbeitsaufträge

Zur Bewertung der Leistung des Gesamtsystems wird die Anzahl der Probanden betrachtet, die alle Arbeitsaufträge unter Einhaltung der vorgegebenen Treibstoffmenge absolvieren konnten. Die Ergebnisse dieser Gegenüberstellung sind in Tabelle 5-12 in Form einer Kreuztabelle dargestellt.

Tabelle 5-12: Gegenüberstellung der beiden Agentenkonfigurationen hinsichtlich der Anzahl von Probanden, welche die Szenarien mit der vorgegebenen Menge Treibstoff absolvieren konnten

Treibstoffvorrat ausreichend		
Konfiguration A	Konfiguration B	
	Nein	Ja
Nein	2	5
Ja	1	5
Exakte Sig. (2-seitig)	.219 ^a	
^a . basierend auf McNemar-Test unter Verwendung einer Binomialverteilung		

In fünf Fällen konnten Probanden, die in Konfiguration A die Vorgaben für den Treibstoffvorrat nicht einhalten konnten, dies in Konfiguration B erfüllen. Umgekehrt konnten nur ein einzelnes

Mal die Vorgaben in Konfiguration A erfolgreich umgesetzt werden, nachdem sie in Konfiguration B nicht eingehalten werden konnten. Eine statistische Untersuchung der Ergebnisdaten führt zu einem nicht signifikanten Unterschied zwischen den beiden Agentenkonfigurationen. Hypothese *H9* hinsichtlich der Abhängigkeit der Umsetzung von Arbeitsaufträgen von der jeweiligen Agentenkonfiguration wird somit nicht bestätigt.

5.1.5 Diskussion

Die Ergebnisse des Mensch-Maschine Experiments belegen den signifikanten Einfluss der Agentenkonfiguration auf Aspekte des menschlichen Verhaltens sowie auf das Vertrauen und die Akzeptanz des Operators in den kognitiven Agenten. Dieser Einfluss lässt sich insbesondere für die Intervention, d.h. den manuellen, korrigierenden Eingriff des Operators während der Missionsdurchführung identifizieren. Die zur Korrektur getätigten Arbeitsschritte sowie die darauf verwendete Zeitdauer konnten durch das in dieser Arbeit beschriebene Interaktionskonzept signifikant gemindert werden. Gleiches gilt für die bei der Intervention getätigten fehlerhaften Systemeingaben. Das Vertrauen in den kognitiven Agenten wurde hingegen signifikant erhöht. Gleiches gilt für die Akzeptanz des Systemverhaltens bei Lageänderung und die Akzeptanz des Gesamtsystems.

Andere Parameter zeigen dagegen weder Signifikanzen noch Trends. Die Ergebnisse bezüglich der Beanspruchung (NASA-TLX) und des Situationsbewusstseins (SAGAT) der Probanden sind stark gestreut, was auf große interpersonelle Unterschiede beim Ausfüllen der subjektiven Fragebögen zurück zu führen ist. Die Ergebnisse zur Aufmerksamkeitsverteilung der Probanden zeigen rückläufige Aufmerksamkeitsanteile bezüglich Informationen des kognitiven Agenten, wobei auch hier keine Signifikanz festgestellt werden kann.

Die Ergebnisse des Experiments belegen den positiven Effekt des beschriebenen Interaktionskonzepts auf die mögliche Unterstützung des Operators (vgl. Forschungsfragestellung in Abschnitt 2.4). Die Bildung angepassten Vertrauens wird durch die Rückführung von Leistungsinformationen des kognitiven Agenten gemäß [Lee & See 2004] gefördert. Die Beschreibung von Zuständen des Systems und des Agenten sowie die Kommunikation von Gründen für das eigene Handeln (bzw. Nicht-Handeln aufgrund von Fehlerzuständen) verringern negative Effekte wie *Opacity*, erhöhen vielmehr das Verständnis des Operators für das aktuelle Geschehen und liefern ihm Möglichkeiten zum Eingriff. Dieses Verständnis senkt die wahrgenommene Komplexität des UAV-Systems für den Operator und führt schließlich zu einer höheren Akzeptanz des Gesamtsystems. [Spain & Madhavan 2009] beschreiben vergleichbare Resultate und assoziieren das zusätzliche Vertrauen mit einer Steigerung der wahrgenommenen Zuverlässigkeit des Systems.

Die Unterstützung des Operators durch den Agenten ist dabei nicht konstant, sondern situationsbezogen, da Informationen zielgerichtet dargestellt werden, wenn der Operator Entscheidungen zu treffen hat bzw. eigene Aufträge (um)formulieren muss. Besonders bei der Intervention, in der die Beanspruchung des Operators erhöht ist, leistet der Agent durch die Bereitstellung entscheidungsrelevanter Informationen Unterstützung. Dies korrespondiert mit der Definition von Etikette nach [Parasuraman & Miller 2004] als Form der Interaktion, die die Unterstützung und das daraus resultierende Vertrauen maßgeblich positiv beeinflusst. Die Ergebnisse der Hypothesen *H6*, *H7* und *H8* zeigen eine signifikante Reduktion von Zeit- und

Interaktionsbedarf sowie von Fehlern bei der Intervention. Das vorgestellte Interaktionskonzept führt insbesondere zur teilweisen Eliminierung von Ausreißern in den Ergebnissen (insbesondere der Anzahl von Arbeitsschritten sowie der Interaktionsdauer).

Diese Verschiebung der Ergebnisse in einen akzeptablen Bereich spiegelt sich im Leistungsniveau des Gesamtsystems, abgebildet über die Einhaltung des verfügbaren Treibstoffvorrats, wider. Durch die Darstellung der Leistungsinformationen konnte die Anzahl der Probanden, die die Vorgaben nicht einhalten konnten, von sieben auf drei reduziert werden. Obwohl diese Verbesserung nicht statistisch validiert werden konnte (fehlende statistische Signifikanz), ist dies ein weiteres Indiz für die Notwendigkeit der Rückführung geeigneter Informationen bei der Führung eines kognitiven Agenten, um die beabsichtigte Leistungssteigerung mittels kognitiver Automation zu erzielen. Ergebnisse nach [Uhrmann 2013], die bereits für das Führungskonzept der auftragsbasierten UAV-Führung gegenüber konventionellen Ansätzen eine Leistungssteigerung nachweisen konnten, werden hierdurch bestätigt und ergänzt.

Die NASA-TLX Erhebung gemäß [Hart & Staveland 1988] liefert keine aussagekräftigen Ergebnisse, da die interpersonelle Streuung die intrapersonellen Unterschiede überschatten. Ähnliche Ergebnisse bezüglich der Beanspruchung zeigte bereits [Uhrmann 2013], der diesbezüglich ebenfalls keine statistisch belastbaren Ergebnisse erzielen konnte, obwohl bereits eine Normierung der NASA-TLX Ergebnisse nach [Donath 2011] erfolgte, die darauf abzielt, interpersonelle Unterschiede zu egalisieren. Im Gegensatz zu diesen Erkenntnissen benennt [Hart 2006] eine Vielzahl von Studien, in denen NASA-TLX erfolgreich zur Messung der Beanspruchung von Probanden angewendet wurde. Und [Schipani 2003] z.B. beschreibt im Zusammenhang mit der Führung von UGVs durch einen Operateur eine hohe Korrelation der NASA-TLX Messung zu anderen Erhebungsmethoden. Unterschiede zu den in [Hart 2006] beschriebenen Beispielen lassen sich in der Komplexität der zugrunde liegenden Aufgaben identifizieren. Die im Experiment untersuchte Aufgabe stellt keine singuläre Aufgabenstellung, sondern ein komplexes Zusammenspiel verschiedener Aktionen und Faktoren dar. Im Gegensatz dazu beschreiben z.B. [Rubio et al. 2004] die Anwendung von NASA-TLX bei der Messung der Beanspruchung von Probanden bei einfachen kognitiven und motorischen Aufgaben.

Das Situationsbewusstsein der Probanden wurde im Versuch über einen SAGAT Fragebogen gemäß [Endsley 1988] erhoben, zeigt jedoch ebenfalls keine signifikanten Unterschiede zwischen den Agentenkonfigurationen. [Endsley 2000] beschreibt hingegen belastbare Ergebnisse für die Erhebung von Situationsbewusstsein mittels SAGAT und benennt hierbei explizit Beispiele im Rahmen der Evaluierung von Automationskonzepten, u.a. bei der Automatisierung von Flugzeug-Cockpits. [Denford et al. 2004] verwenden SAGAT als leistungsfähiges Nachweismittel des Situationsbewusstseins von Fluglotsen. Obwohl der Versuchsaufbau dieses Mensch-Maschine Experiments den Vorgaben bezüglich der Anwendung von SAGAT [Endsley 2000] entspricht, konnten die Ergebnisse die beschriebene Sensitivität und Genauigkeit der Datenerhebung nicht bestätigen, wobei nicht ausgeschlossen werden kann, dass keine signifikanten Unterschiede aus den untersuchten Agentenkonfigurationen hinsichtlich des Situationsbewusstseins resultieren.

Die Erhebung der Aufmerksamkeitsverteilung der Probanden über die Messung der Position von Fixationen auf dem Missionsinterface mittels faceLAB zeigt ebenfalls keine nachweisbaren Zusammenhänge. Der prozentuale Anteil der Fixationspunkte auf vom Agenten bereitgestellten

Informationen liegt bei beiden Stichproben unter fünf Prozent und zeigt keine signifikanten Unterschiede. Andere Arbeiten setzen hingegen erfolgreich Blickbewegungsmessung zur Zuordnung der Aufmerksamkeit von Operateuren ein, wie z.B. in [Donath 2011], wo die Aufmerksamkeit von UAV-Operateuren unterschiedlichen Aufgaben und Handlungen zugeordnet wird. Als begrenzender Faktor wirkt sich in dieser Arbeit die Beschränkung auf die Messung der durchschnittlichen Fixationsdauer im Rahmen der Blickbewegungsmessung aus. Andere Größen, wie z.B. zielführende Bewegungen (Sakkaden) [Kaufmann & Steffen 2012], wurden nicht erhoben. Die geforderte Kalibrierungsgenauigkeit des faceLAB Systems konnte zudem lediglich für die Hälfte der Versuchspersonen vor der Datenerhebung erzielt werden. Von den erhobenen Messungen wurden daher zwischen 20% und 25% von faceLAB verworfen, da ihre Genauigkeit als unzureichend bewertet wurde [Werner 2014b]. Gründe hierfür waren neben der teilweise unzureichenden Kalibrierungsgenauigkeit die Körper- und Kopfbewegungen der Probanden, welche sich während der Datenerhebung zum Teil aus dem Erfassungsbereich entfernten. Ein weiterer Grund stellte die unzureichende Öffnung der Augen einzelner Probanden dar.

Insgesamt lässt sich feststellen, dass durch die geringe Stichprobengröße mit 12 Probanden für die statistische Auswertung der Ergebnisse Einschränkungen hinzunehmen sind. Ähnliche Untersuchungen wie z.B. in [Uhrmann 2013] und [Schipani 2003] sind auf Grund der Komplexität des Versuchsaufbaus auf vergleichbare Stichprobengrößen beschränkt. Dem Einfluss von Sequenzeffekten wurde durch die Variation der Zuordnung von Agentenkonfigurationen und Szenarien entgegengewirkt. Trotzdem können Beeinflussungen z.B. durch Lerneffekte für die Erhebungen im zweiten Testszenario nicht ausgeschlossen werden. Eine strikte Trennung war durch den Versuchsaufbau und die inhaltliche Nähe der Szenarien nicht möglich.

Prinzipiell konnte die Rückführung und Anzeige gezielter Informationen, bei gleicher Funktionalität des Agenten, als wesentlich für die Gesamtleistung des Mensch-Maschine Systems identifiziert werden. Das entwickelte Interaktionskonzept stellt dabei eine Lösung für die in Abschnitt 2.4 formulierte Forschungsfragestellung dieser Arbeit dar. Die gewonnenen Aussagen beziehen sich hierbei auf den Einfluss des gesamten Interaktionsverhaltens und beschreiben nicht den Einfluss einzelner Parameter auf das erzielbare Leistungsniveau. Die quantitative Beschreibung des Einflusses einzelner Aspekte der Form und des Inhalts von Agenteninformationen kann somit nicht erfolgen.

Im Mensch-Maschine Versuch konnten die Effekte unterschiedlicher Agentenkonfigurationen in einer kontrollierten Simulationsumgebung wiederholt ermittelt werden. Um die technische Reife des kognitiven Agenten und die in Abschnitt 2.4 geforderte Plattformunabhängigkeit der technischen Lösung zu demonstrieren, wird im Folgenden die Integration und Demonstration der Agentenfähigkeiten auf unterschiedlichen Flugplattformen beschrieben.

5.2 Demonstration im Flugversuch

Zum Nachweis des Reifegrads der Implementierung des kognitiven Agenten und der Automationskomponenten sowie der Plattformunabhängigkeit der technischen Lösung, wurde, aufbauend auf den Ergebnissen der Mensch-Maschine Evaluierung in der Simulation, eine Systemdemonstration im Realflug durchgeführt. Hierzu wurde der kognitive Agent auf zwei UAV-Plattformen mit unterschiedlichen Eigenschaften und Systemkomponenten integriert und im Rahmen einer gemeinsamen übergeordneten Aufgabenstellung separat im Flugversuch zur

Anwendung gebracht [Rudnick et al. 2014]. Die Durchführung der Flugversuche erfolgte auf dem Gelände der UniBwM. Zu diesem Zweck wurde die in Abschnitt 5.1.3.2 beschriebene Ausgangslage an die räumlichen Gegebenheiten des Versuchsgeländes der Universität angepasst und Bereiche zur Anwendung der beiden UAV-Demonstratoren spezifiziert (vgl. Abbildung 5-10).



Abbildung 5-10: Die Flugbereiche der UAV-Demonstratoren Explorer (links) und OktoXL (rechts) auf dem Campus der UniBwM. Die Einschränkung der Datenlinkverfügbarkeit ist symbolisch dargestellt.

Das Starrflügel-UAV *Explorer* dient der Abbildung von Fähigkeiten zur Aufklärung über größere Distanzen und Flächen unter Berücksichtigung einer äußeren taktischen Lage (vgl. Abschnitt 5.2.1). Hierfür integriert *Explorer* Komponenten zur Aufnahme von Videodaten mit der Möglichkeit zur zeitversetzten Steuerung des zu betrachtenden Bildausschnitts. Der integrierte Videodatenlink dient der Übertragung von Einzelbildern und Videos und ist nicht über den gesamten Missionsbereich verfügbar (vgl. Symbole in Abbildung 5-10).

Das Oktokopter-UAV *OktoXL* bildet zusätzlich Fähigkeiten zur gezielten Aufklärung von Objekten und Gebieten ab und bietet die Möglichkeit der kontinuierlichen Sensorführung bzw. Sensordatenauswertung durch den Operator im Rahmen der Auftragserfüllung (vgl. Abschnitt 5.2.2). Hierzu ermöglicht der integrierte Datenlink eine kontinuierliche Übertragung von Video- und Kommandodaten über den gesamten Einsatzbereich des *OktoXL*.

Beide UAV werden über eine einheitliche mobile Bodenkontrollstation (mBKS) geführt (vgl. Abschnitt 5.2.3), welche die in Abschnitt 5.1.3.3 beschriebene Bedienkonsole integriert und um einen Arbeitsplatz für den Experimentator erweitert. Die UAV-Führung aus der mBKS und die Übertragung von Sensordaten erfolgen für beide UAV über eine IP-basierte Funkkommunikation in Form eines WLAN Netzes bei einer Frequenz von 5GHz. Bei *Explorer* wird der Kommandolink separat über ein *XBee 868* Funkmodul [Digi International Inc. 2014] bei 868MHz realisiert, das über *AnyCom* an die übrigen Prozesse angebunden ist.

Die unterschiedlichen Flugeigenschaften der beiden UAV-Demonstratoren werden dabei vom kognitiven Agenten über AAL-Parameter verarbeitet und bei der Auftragserfüllung genutzt. Dabei werden von beiden UAV-Demonstratoren Funktionen proprietärer FMS über die AAL-Architektur

abstrahiert bereitgestellt. Die Anbindung der FMS wird dabei über entsprechende Treiber realisiert, die die Kommunikation über die Hardware-Schnittstelle des Computers sowie die Kommandierung und das Management von Wegpunkt- und Autopilotenkommandos steuern.

Die Anbindung der Hardwarekomponenten des Datenlinks und der Sensorik an den kognitiven Agenten erfolgt über Module der in Abschnitt 4.5.1 beschriebenen Interprozesskommunikation *AnyCom*. Auf Seiten der Aufklärungssensorik verwenden beide UAV-Demonstratoren Fischaugenobjektive zur Abbildung großer Bildausschnitte. Zur Anzeige und Entzerrung von Bildbereichen wird die *AnyCom* Software *Electronic Camera Gimbal* (ECG) verwendet [Böhm et al. 2013]. ECG bildet in einem separaten Prozess eine digitale Sensorführung nach, die auf den Daten des gesamten (verzerrten) Bildausschnitts operiert. kann an allen Stellen des Systems verwendet werden, an denen die ursprünglichen Sensordaten verfügbar sind. In dieser Arbeit wird ECG zum einen an Bord des *OktoXL* im Sinne einer klassischen Sensorführung eingesetzt, um die zu übertragende Datenmenge möglichst gering zu halten. Zum anderen wird ECG bei *Explorer* innerhalb der mBKS genutzt, um den gesamten Bildausschnitt am Boden vorzuhalten und zeitverzögert eine Sensorführung abbilden zu können.

5.2.1 UAV-Demonstrator – Explorer

Der UAV-Demonstrator *Explorer* basiert auf der Zelle des *Bormatec Explorer* Modellbausatzes [BORMATEC 2015], erweitert durch ein Spornradfahrwerk, verstärkte Motorgondeln und Versteifungen der Strukturkomponenten durch Kohlefaserelemente (vgl. Abbildung 5-11). *Explorer* nutzt das *Cloud Cap Piccolo* Autopiloten System [Cloud Cap Technology 2015], welches Flugsensorik und Flugsteuerung anbindet sowie Flugregelung und Autopilotenfunktionen bis auf die Ebene der Wegpunkt- und Routenkommandos bereitstellt. Der proprietäre Datenlink des *Piccolo* Autopiloten operiert bei einer Frequenz von 2,4GHz, wird im Folgenden jedoch ausschließlich als Sicherheitsdatenlink verwendet und über den Experimentator Arbeitsplatz in der mBKS adressiert. Der Datenaustausch im Flugversuch ist über einen WLAN-Adapter in Form der *Ubiquiti SR71-E Highpower* mPCI-E Karte [Ubiquiti Networks 2015] realisiert, die über zwei Antennen auf 5GHz operiert. Zusätzlich verfügt *Explorer* über einen *xBee 868 Funkmodul* auf 868MHz als zusätzlichen Kommandodatenlink.



Abbildung 5-11: UAV-Demonstrator *Explorer* bei Flugversuchen auf dem Testgelände der UniBwM

Explorer integriert zwei *Microsoft LifeCam Studio FullHD Webcams* [Microsoft 2015] als EO Sensoren. Eine ist im Sinne einer FPV-Kamera in Richtung der Längsachse des *Explorer* auf dem Rumpf montiert. Die zweite Kamera ist durch ein Fischaugenobjektiv modifiziert und senkrecht

unter dem Rumpf als Aufklärungssensor montiert. Für die im Folgenden beschriebenen Flugversuche werden lediglich die Ausgaben des Aufklärungssensors verwendet.

Der kognitive Agent sowie die Automationskomponenten operieren auf einem *COMMELL LS-37BS* Mainboard mit einem Intel® Core i7 Prozessor [COMMELL 2015]. Der *Piccolo* Autopilot ist dabei über eine serielle RS232 Schnittstelle, die beiden Kameras sind über USB 2.0 Schnittstellen angebunden. Eine detaillierte Beschreibung der Einzelkomponenten des *Explorer* sowie eine Betrachtung der elektromagnetischen Abschirmung und thermischen Belastung des Demonstrators finden sich in [Schneemilch 2013].

AAL stellt als Schnittstelle zwischen kognitivem Agenten und *Piccolo* Autopiloten Telemetrie- und Systemstatusdaten zur weiteren Verarbeitung bereit. Die Flugeigenschaften des UAV-Demonstrators werden zu Fähigkeiten gemäß Abschnitt 4.3.1 abstrahiert und wirken sich zudem u.a. in der Größe der Kurvenradien sowie der Fluggeschwindigkeit aus. Die Größenbeschränkung programmierbarer Routen durch *Piccolo* wird von *AAL* gegenüber dem Agentensystem durch die interne Wegpunktverwaltung abstrahiert.

RPM nutzt für die Routenplanung die von *AAL* bereitgestellten Informationen zur Flugdynamik des *Explorer* und erstellt entsprechend angepasste Flugpläne für den kognitiven Agenten. *RPM* integriert eine Terrain Datenbank, um Überflüge und Aufnahmen in konstanten Höhen über Grund zu ermöglichen.

Die Verfügbarkeit des Datenlinks im Einsatzgebiet wird durch *DLM* verarbeitet. *DLM* kombiniert die interne Repräsentation der Datenlinkreichweite mit den missionsspezifischen Spezifikationen der Verfügbarkeit (vgl. Abbildung 5-10) und steuert dementsprechend den WLAN Datenlink. Der *xBee 868* Kommandodatenlink wird auf Grund seiner großen Reichweite von *DLM* als konstant verfügbar über dem gesamten Einsatzgebiet angenommen.

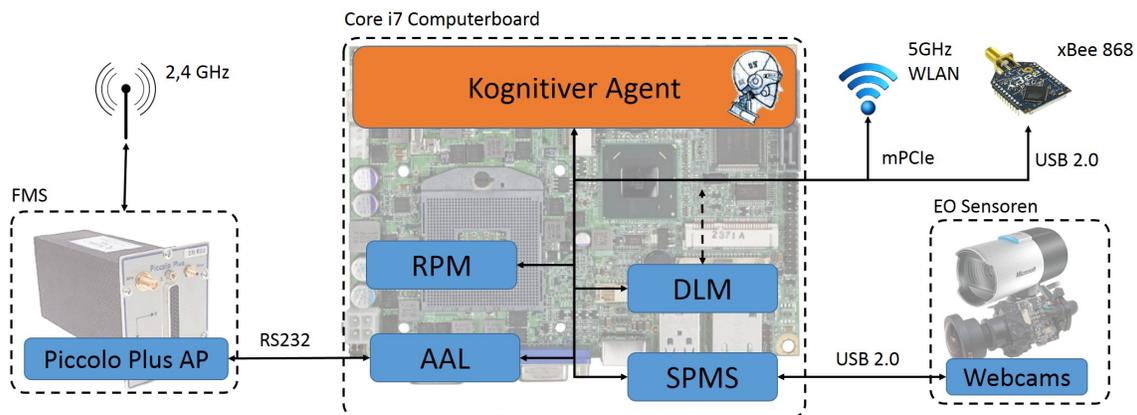


Abbildung 5-12: Hardwarekomponenten des UAV-Demonstrators Explorer, verbunden über ein zentrales Computerboard und gesteuert durch den kognitiven Agenten und die Automationskomponenten

Das *SPMS* integriert bei *Explorer* keine Perzeptionsalgorithmen, sondern stellt ausschließlich Bild- und Videodaten zur Aufnahme und Datenübertragung bereit, da die Qualität der Full-HD Sensordaten bei der Flughöhe des *Explorer* unzureichend für eine automatisierte Personen- und Fahrzeugerkennung ist. Daher ist eine manuelle Auswertung der Sensordaten durch den Operateur in der mBKS erforderlich. Hierzu werden Sequenzen der Sensordaten zunächst in spezifizierten

Zeiträumen aufgezeichnet und zu einem späteren Zeitpunkt gemeinsam über das *DLM* zur *mBKS* übertragen.

Abbildung 5-12 zeigt das Systemdesign des UAV-Demonstrators *Explorer* mit den Softwarekomponenten auf dem zentralen Rechnerboard sowie die angebundene Hardwarekomponenten.

5.2.2 UAV-Demonstrator – OktoXL

Der UAV-Demonstrator *OktoXL* ist ein am IFS entwickeltes Oktokopter Modell, basierend auf einem Kohlefaserrumpf. *Mikrokopter FlightCtrl* wird zur Motor- und Flugregelung, *Mikrokopter NaviCtrl* als Autopilotensystem des Oktokopters genutzt [HiSystems GmbH 2015]. *NaviCtrl* verfügt über eine proprietäre Funkschnittstelle bei 2,4GHz, die als Sicherheitsdatenlink eingesetzt wird. Der Datenaustausch erfolgt, wie bei *Explorer*, über einen WLAN-Adapter bei einer Frequenz von 5GHz. Ein separates Datenfunkmodul für Kommandodaten ist nicht integriert. Abbildung 5-13 zeigt den UAV-Demonstrator *OktoXL* mit Anbauteilen im Flugversuch.

Die modulare Nutzlast-Box unter dem *OktoXL* Rumpf trägt zwei baugleiche *COMMELL LS-37BS* Mainboards (vgl. Abschnitt 5.2.1) und eine *XIMEA* 4MP Kamera mit einem Fischaugenobjektiv als Aufklärungssensor, der über eine USB 3.0 Schnittstelle angesteuert wird. Der Einsatz von zwei separaten Rechnerboards ermöglicht die Dislokation der Automationskomponenten anhand ihres Leistungsbedarfs. *OktoXL* integriert Algorithmen zur automatisierten Personen- und Fahrzeugerkennung im Rahmen des SPMS auf der Grundlage der Daten des Aufklärungssensors. SPMS wird daher auf ein separates Rechnerboard, den *Perception Control Computer* (PCC), ausgelagert und dort, in Zusammenarbeit mit ECG, zur Bildauswahl und -entzerrung betrieben. Sonstige Automationskomponenten sowie der kognitive Agent operieren wie bei *Explorer* gemeinsam auf dem *Mission Management Computer* (MMC). MMC integriert den WLAN Adapter und fungiert an Bord als Router für den via Ethernet verbundenen PCC.



Abbildung 5-13: UAV-Demonstrator *OktoXL* bei Flugversuchen auf dem Testgelände der UniBwM

AAL bildet die Schnittstelle zu *NaviCtrl* und stellt hierfür Fähigkeiten u.a. zum Schwebeflug bzw. für positionsneutrale Richtungswechsel bereit. Die Kommunikation erfolgt dabei ausschließlich zwischen *AAL* und *NaviCtrl*, welches intern die Steuerung der *FlightCtrl* realisiert.

RPM unterliegt im Einsatzgebiet des *OktoXL* keinen Vorgaben einer taktischen Rahmenlage und keiner Restriktionen durch Flugeigenschaften des Oktokopters. Vorgaben existieren für *RPM* bei der Aufklärung auf Grund der geringen Flughöhe des *OktoXL* im Wesentlichen durch den Abbildungsbereich des Aufklärungssensors, welcher über *SPMS* übergeben wird.

Für den gesamten Einsatzbereich des *OktoXL* wird eine konstante Verfügbarkeit des Datenlinks angenommen. *DLM* dient damit ausschließlich der separaten Übertragung von Video- und Systemstatus- bzw. Kommandodaten.

Das *SPMS* integriert Algorithmen zur Personen- und Fahrzeugerkennung auf Grundlage der mittels *ECG* rektifizierten Bilddaten des Aufklärungssensors. Die Algorithmen werden dem kognitiven Agenten von *SPMS* in Form von Erkennungsfähigkeiten bereitgestellt. Die Steuerung des Bildausschnittes mittels *ECG* erfolgt manuell durch den Operateur über einen Joystick, dessen Signale über das Sensorinterface und die WLAN Verbindung an den *PCC* übermittelt werden. Die Anzeige des aktiven Bildausschnittes sowie der Erkennungsleistung erfolgen ebenfalls über das Sensorinterface (vgl. Abschnitt 5.2.3). Erkennungsprozeduren und Eingriffe des Operateurs werden dabei gemäß Abschnitt 4.1.2.5 im Rahmen der Aktionsverwaltung des kognitiven Agenten koordiniert.

Da zum Zeitpunkt der Versuchsdurchführung die Erkennungsleistung der Algorithmen für Realflugganwendungen unzureichend war, werden die erkannten Personen und Fahrzeuge über fest in *SPMS* hinterlegte Positionsdaten simuliert.

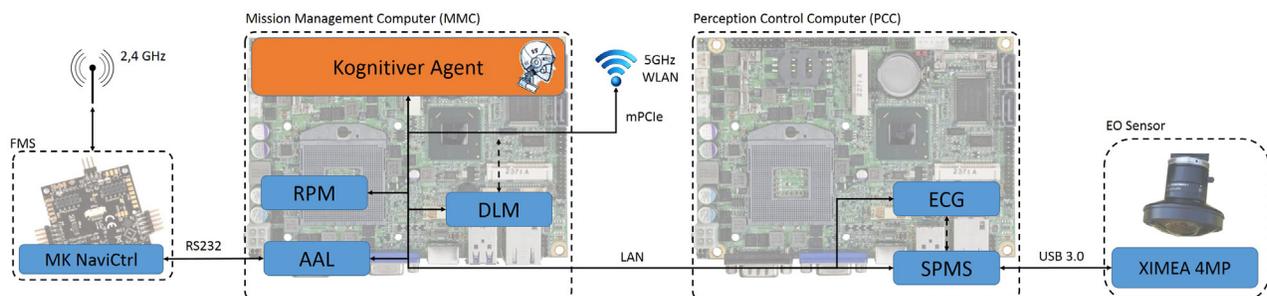


Abbildung 5-14: Hardwarekomponenten des UAV-Demonstrators *OktoXL* mit zwei Computerboards (*MMC* und *PCC*), die über eine lokale Netzwerkverbindung kommunizieren

Abbildung 5-14 zeigt die resultierende Zuordnung der Automationsprozesse und des kognitiven Agenten auf die eingesetzte Rechnerarchitektur (*MMC* und *PCC*). Zusätzlich sind *NaviCtrl*, der Aufklärungssensor und der WLAN-Adapter als Kommunikationsschnittstelle zur *mBKS* dargestellt.

5.2.3 Mobile Bodenkontrollstation

Die mobile Bodenkontrollstation (*mBKS*) dient der einheitlichen Steuerung und Überwachung der beiden UAV-Demonstratoren *Explorer* und *OktoXL* (vgl. [Rudnick et al. 2014]). Die *mBKS* beinhaltet hierfür die Mensch-Maschine Schnittstelle gemäß Abschnitt 4.4, ähnlich dem UAV-Missionssimulator im Mensch-Maschine Versuch (vgl. Abschnitt 5.1.3.3). Zusätzlich dient ein Arbeitsplatz für den Experimentator zur Überwachung der Hardware- und Softwarekomponenten während des Flugversuchs und zur Bereitstellung einer alternativen Datenverbindung für Notfallsituationen (siehe Abbildung 5-15).



Abbildung 5-15: Mobile Bodenkontrollstation (mBKS) mit den Arbeitsplätzen für den Bediener (links) und den Experimentator (rechts). Mittels Joystick steuert der Operateur den Bildausschnitt auf dem Sensorinterface (oben)

Die mBKS basiert auf einem modifizierten Mercedes Sprinter Kastenwagen mit einer integrierten Bedienkonsole, Bordfunk und GPS Systemen sowie einer internen Spannungsversorgung und stellt eine Erweiterung der in [Kriegel 2011] vorgestellten mBKS dar. Zur Datenkommunikation mit den UAV dient eine steuerbare WLAN Richtantenne, welche den Demonstratoren über GPS-Positionsdaten im Flug nachgeführt wird, um die maximale Übertragungreichweite des Datenlinks zu erhöhen. Für die *Explorer* Flugversuche wurde die mBKS zusätzlich durch eine *xBee 868* Bodenstation am Experimentatorarbeitsplatz erweitert.

5.2.4 Demonstrationsdesign

Die Demonstrationsflüge mit den UAV-Demonstratoren wurden gemäß der in Abbildung 5-10 dargestellten Flugbereiche zeitlich separiert vollzogen. Neben dem UAV-Operateur, der die UAV über die gemeinsame Mensch-Maschine Schnittstelle aus der mBKS führte, begleitete ein Sicherheitspilot die Flugversuche. Auf Grund mangelnder ATOL Fähigkeiten von *Explorer* und *OktoXL* vollzog der Sicherheitspilot im Flugversuch Start und Landung der UAV und übernahm bei Notfällen die Flugsteuerung über den proprietären Datenlink der Autopiloten.

Die Flugversuche sollten die Anpassungsfähigkeit des Agenten auf der Grundlage der verfügbaren UAV-Fähigkeiten und der äußeren Rahmenbedingungen demonstrieren. Insbesondere wurden hierbei Unterschiede in der Auftragsumsetzung in Form variierender Agentenpläne (u.a. der Anpassung von Handlungen gemäß der Datenlinkverfügbarkeit) bzw. Unterschiede in der subsymbolischen Durchführung von Aktionen (z.B. Anwendung angepasster Flugmuster zur Gebietsaufklärung) demonstriert. Auf Seiten der Mensch-Maschine Interaktion wurde im Fall des UAV-Demonstrators *OktoXL* die manuelle Sensorführung durch den Operateur demonstriert.

Die Flugversuche begannen jeweils an einem definierten Startpunkt und endeten mit der Landung der UAV am Ausgangspunkt. Start und Landung wurden vom Operateur kommandiert und vom kognitiven Agenten unter Berücksichtigung der nicht-vorhandenen ATOL Fähigkeiten verarbeitet. Der Agent interpretierte den UAV-Start als erfolgreich abgeschlossen, wenn der Sicherheitspilot die UAV-Flugführung in der Luft an den Autopiloten übergab. Die Landung wurde wiederum mit der Übernahme der Flugführung durch den Sicherheitspiloten vollzogen. Im Folgenden werden die gedachten Demonstrationsverläufe der UAV-Demonstratoren beschrieben:

Gedachter Demonstrationsverlauf - *Explorer*

Der Auftrag des *Explorer* umfasst die Aufklärung der Gebiete *Area_North* und *Area_East* sowie der Straße *Public_Road* mit der anschließenden Übertragung der Aufklärungsdaten zur *mBKS* bei *Transmission_Point* (vgl. Abbildung 5-10). Hierzu startet *Explorer* ausgehend von *Home_Base* und sammelt nacheinander Sensordaten von *Area_North* und *Public_Road*. Während des Anflugs auf *Area_East* fällt die Aufklärungsfähigkeit des UAV aus, wodurch der Agent unfähig ist den aktuellen Auftrag umzusetzen. Der Agent informiert den Operateur daraufhin über den Fähigkeitsverlust als Grund für den Auftragsabbruch, woraufhin dieser eine manuelle Umplanung vollzieht, die den direkten Anflug von *Transmission_Point* zur Übertragung der gesammelten Daten definiert.

Die Auswertung der Sensordaten zeigt eine Person im Missionsgebiet, die erneut durch *Explorer* aufgeklärt werden soll. Nach der Kommandierung überfliegt das UAV die Person, sammelt Aufklärungsdaten und übersendet diese anschließend vom *Transmission_Point* an die *mBKS*. Der Operateur veranlasst nun die Rückkehr und Landung des UAV an der *Home_Base*. Die Ausführung des hierfür erstellten Agentenplans wird durch die Sperrung des Egress-Korridors unterbrochen, woraufhin der Agent selbstständig umplant und einen alternativen Korridor nutzt, um abschließend an der *Home_Base* die Landung zu vollziehen.

Gedachter Demonstrationsverlauf - *OktoXL*

OktoXL operiert in der von *Explorer* nicht aufgeklärten *Area_East* im östlichen Bereich des Einsatzgebietes (vgl. Kartenausschnitt in Abbildung 5-10). Der Auftrag umfasst die Fahrzeugaufklärung über der Straße (*Road*) sowie die Aufklärung von Personen über der Rasenfläche (*Area*). Hierzu startet *OktoXL* nördlich der *Forward-Operating Base* (FOB), fliegt die Straße an und vollzieht eine Fahrzeugerkennung anhand der Daten des Aufklärungssensors, welcher manuell durch den Operateur in der *mBKS* gesteuert wird. Hierbei wird ein Fahrzeug auf der Straße detektiert und auf dem Missionsinterface zur Anzeige gebracht. Der Operateur kommandiert daraufhin die erneute Aufnahme des Fahrzeugs, um anschließend mit der Aufklärung der Rasenfläche fortzufahren. Der Agent erstellt hierzu ein geeignetes Aufklärungspattern zur Abdeckung der Rasenfläche mittels des Aufklärungssensors und wendet einen Personenerkennungsalgorithmus auf die Ausgabedaten an. Die Sensorführung erfolgt erneut manuell durch den Operateur. Zwei Personen werden hierbei auf der Rasenfläche detektiert, woraufhin beide zur erneuten Aufklärung überflogen werden. Abschließend wird *OktoXL* zur Landung zurück zur *Forward-Operating Base* kommandiert. Das UAV fliegt die Ausgangsposition an und vollzieht die Landung.

5.2.5 Ergebnisse

Im Jahr 2014 wurden insgesamt 12 Versuchsflüge mit *Explorer* und 11 Versuchsflüge mit *OktoXL* durchgeführt. Die im Folgenden präsentierten Ergebnisse stellen exemplarisch jeweils einen Flugversuch pro UAV-Demonstrator vor. Anschließend werden Unterschiede bei der Auftragsumsetzung durch den kognitiven Agenten beschrieben und die Anwendung des *Electronic Camera Gimbal* (ECG) am Beispiel der Fahrzeugerkennung bei *OktoXL* erläutert. Aspekte der Mensch-Maschine Interaktion, welche bereits in Abschnitt 5.1 betrachtet wurden, werden hier außer Acht gelassen.

Demonstrationsflug – *Explorer*

Ein Überblick über den zeitlichen Verlauf des *Explorer* Demonstrationsfluges mit den relevanten Ereignissen und Handlungen der beteiligten Akteure ist in Tabelle 5-13 gegeben. Der Demonstrationsflug erfolgte zwischen Start und Landung in einer konstanten Flughöhe von $h_{AGL}=150\text{m}$ bei einer mittleren Geschwindigkeit von $v=17\text{m/s}$. Der Kurvenradius des *Explorer* wurde dabei mit $r=500\text{m}$ u.a. bei der Dimensionierung des Kreisens um einen Haltepunkt berücksichtigt.

Für die Durchführung des Demonstrationsfluges wurde *Explorer* auf der Start- und Landefläche östlich der FLOT (siehe Wiese mit Handgranatenwurfständen in Abbildung 5-16) positioniert. Die mBKS verweilte für den gesamten Zeitraum des Flugversuchs stationär neben der Start- und Landefläche. Zum Zeitpunkt $t=-2:07\text{min}$ begann der Operateur mit der Missionsplanung und Kommandierung zur Umsetzung seines Auftrags gemäß Abschnitt 5.2.4. Abbildung 5-16 zeigt die Auftragsagenda des Operateurs (oben links) sowie die vom Agenten erstellte Handlungsagenda (unten rechts) zur Umsetzung des ersten Hauptauftrags (*Detect Persons in Area_North*). Nach der Erstellung des Flugplans erbat der Agent Startfreigabe sowie, auf Grund fehlender ATOL Fähigkeiten, die manuelle Durchführung des Startvorgangs. Nach Erteilung der Startfreigabe führte der Sicherheitspilot den UAV-Start bei $t=0:00\text{min}$ durch, steuerte *Explorer* auf seine Missionsflughöhe von $h_{AGL}=200\text{m}$ und flog anschließend den Ausgangspunkt für die Missionsführung durch den Agenten, die *Home Base* an (vgl. Anflug durch *Explorer* in Abbildung 5-16). Die Übergabe der Flugsteuerung vom Sicherheitspiloten an den Autopiloten signalisierte dem Agenten das Ende der Startprozedur, woraufhin der Anflug auf *Corridor_North* initiiert wurde.

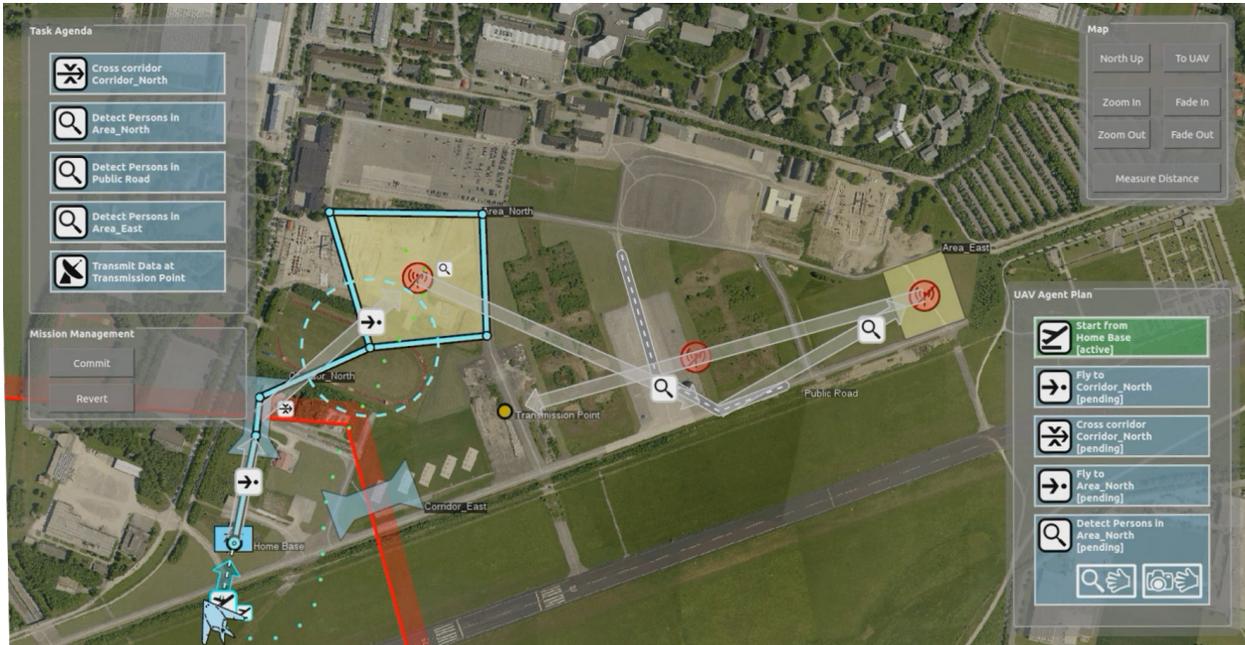


Abbildung 5-16: Missionsinterface mit Auftragsagenda, Agentenplan, Flugplan und Flughistorie während des Explorer Demonstrationsfluges

Tabelle 5-13: Protokoll des Flugversuchs des UAV-Demonstrators Explorer

Time [min]	Action	Actor
-2:07	Mission Planning	Operator
-0:02	Takeoff Clearance	Ground Crew
0:00	Takeoff	Safety Pilot
1:06	Crossing Corridor_North	Agent
1:24	Recon Area_North	Agent
2:06	FlyTo Road	Agent
2:38	Recon Road	Agent
2:53	FlyTo Area_East	Agent
3:03	Disable Sensor Cap.	Experimenter
3:09	Report Planning Failure	Agent
3:18	Manual Replanning	Operator
3:21	FlyTo Transmit_Point	Agent
3:40	Start Transmission	Agent
5:33	Person Detected	Operator
6:23	Task: Recon Target	Operator
7:39	FlyTo Target	Agent
7:58	Recon Target	Agent
8:43	FlyTo Transmission_Point	Agent
8:53	Start Transmission	Agent
9:36	Task: Land at Home Base	Operator
10:26	Disable Corridor_East	Experimenter
10:28	New Flightplan created	Agent
11:16	Crossing Corridor_North	Agent
12:01	Landing at Home Base	Safety Pilot

Bei $t=1:06\text{min}$ überquerte *Explorer* die FLOT über *Corridor_North* zum Anflug auf das Zielgebiet *Area_North*. Dieses wurde ab Zeitpunkt $t=1:24\text{min}$ entlang seiner äußeren Begrenzung umflogen, und parallel wurden Videodaten des Aufklärungssensors aufgezeichnet. Im Anschluss erstellte der Agent einen Aktionsplan zur Umsetzung des zweiten Hauptauftrags (*Detect Persons in Public Road*). Bei $t=2:06\text{min}$ begann *Explorer* den Anflug auf *Public Road*, entlang deren Verlauf der Aufklärungsflug erfolgte. Ab $t=2:38\text{min}$ wurden Sensordaten aufgezeichnet.

Die Umsetzung des Auftrags *Detect Persons in Area_East* begann mit dem Anflug auf das Zielgebiet ab $t=2:53\text{min}$. Hierbei deaktivierte der Experimentator die Sensorfähigkeit des *Explorer* ($t=3:03\text{min}$), wodurch die Auftragserfüllung für den Agenten nicht mehr möglich war (vgl. Abbildung 5-17). Das reaktive Verhalten des Agenten

fürhte zum Kreisen des UAV um seine aktuelle Position. Der Agent meldete dem Operateur den Planungsfehler mit Angabe des Fähigkeitsausfalls. Der Operateur kommandierte daraufhin die unmittelbare Übertragung der bisher gewonnenen Videodaten bei *Transmission Point* ($t=3:18\text{min}$). Der Agent flog den Zielpunkt an und initiierte bei $t=3:40\text{min}$ die Übertragung der Videodaten zur mBKS. Die Übertragung endete mit der Erkennung und der Eingabe der Person in *Area_North* durch den Operateur bei $t=5:33\text{min}$. Die Kommandierung zur Aufklärung der detektierten Person ($t=6:23\text{min}$) bewirkte deren Überflug und Aufnahme durch *Explorer* bei $t=7:58\text{min}$, die Rückkehr zu *Transmission Point* ab $t=8:43\text{min}$ und die Übertragung der Sensordaten zur mBKS ab $t=8:53\text{min}$.



Abbildung 5-17: Abbruch der Auftragsumsetzung nach Ausfall der Sensorfähigkeit des UAV und Notifikation des Operateurs. Während der Umplanung durch den Operateur kreist das UAV um seinen letzten Wegpunkt

Abschließend kommandierte der Operateur die Rückkehr und Landung bei *Home Base* ($t=9:36\text{min}$). Der hierzu erstellte Agentenplan nutzte *Corridor_East* zur Überquerung der FLOT, welcher jedoch bei $t=10:26\text{min}$ durch den Experimentator gesperrt wurde. Der Agent initiierte daraufhin erneut ein Kreisen um die aktuelle UAV-Position und erstellte einen Alternativplan unter Nutzung von *Corridor_North*. Bei $t=12:01\text{min}$ erbat der Agent Unterstützung bei der Landung des UAV durch den Sicherheitspiloten, der im Anschluss den Autopiloten deaktivierte und *Explorer* manuell auf der Rasenfläche landete.

Demonstrationsflug – *OktoXL*

Die wesentlichen Ereignisse des *OktoXL* Demonstrationsflugs im östlichen Abschnitt des Flugversuchsbereiches (vgl. Abbildung 5-10) sind in Tabelle 5-14 in seinem zeitlichen Verlauf tabellarisch dargestellt. *OktoXL* wurde im Flugversuch auf einer konstanten Flughöhe von $h_{AGL}=15\text{m}$ und bei einer mittleren Geschwindigkeit von $v=10\text{m/s}$ geführt. Wesentliches Element des *OktoXL* Demonstrationsflugs gemäß Abschnitt 5.2.4 war die Auftragsunterstützung des kognitiven Agenten bei der Objektdetektion durch die kontinuierliche manuelle Sensorführung durch den Operateur. Die Ergebnisse zeigen eine erfolgreiche Umsetzung dieser manuellen Kontrollaufgabe durch die Verwendung des ECG an Bord des *OktoXL* und die kontinuierliche Übertragung und Anzeige des berechneten Bildausschnitts in der mBKS.

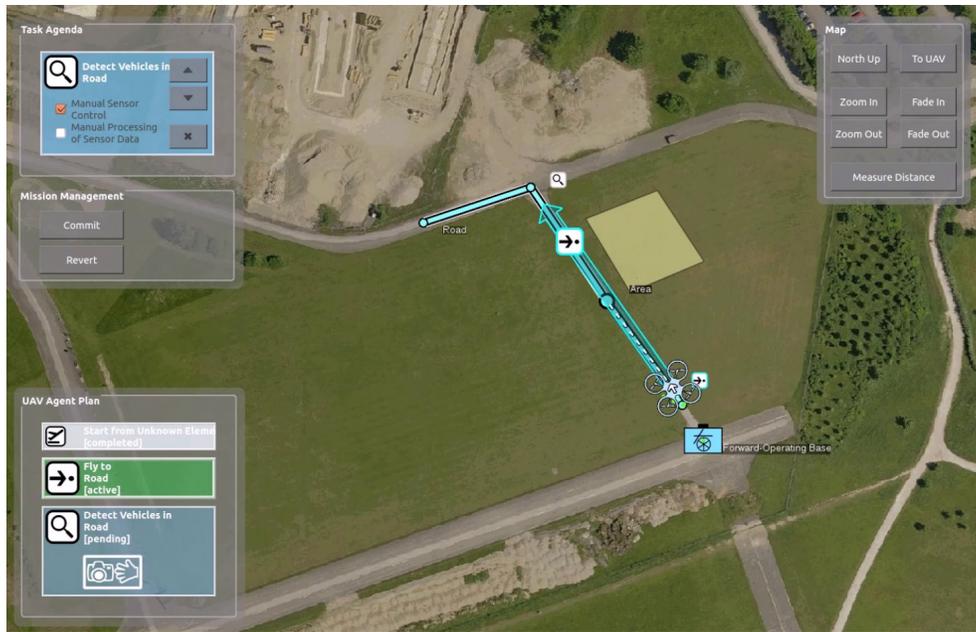


Abbildung 5-18: Auftragserteilung für die Fahrzeugerkennung auf Road durch OktoXL, bei manueller Steuerung des betrachteten Bildausschnitts durch den Operateur (Manual Sensor Control)

Tabelle 5-14: Protokoll des Flugversuchs des UAV-Demonstrator OktoXL

Time [min]	Action	Actor
-0:20	Task: Vehicle Detection on Road	Operator
0:00	Takeoff	Safety Pilot
0:34	Perform Vehicle Detection	Agent, Operator
1:20	Vehicle Detection	Agent
1:23	Task: Recon Vehicle	Operator
1:30	FlyTo Vehicle	Agent
1:46	Recon Vehicle	Agent
2:08	Task: Person Detection on Area	Operator
2:19	FlyTo Area	Agent
2:32	Perform Person Detection	Agent, Operator
5:15	Persons Detected	Agent
5:25	Task: Recon Person 1	Operator
5:36	FlyTo Person 1	Agent
5:49	Recon Person 1	Agent, Operator
6:16	Task: Recon Person 2	Operator
6:24	FlyTo Person 2	Agent
6:43	Recon Person 2	Agent, Operator
7:18	Task: Land at FOB	Operator
7:48	Landing at FOB	Safety Pilot

Für den Versuchsstart wurde *OktoXL* in unmittelbarer Nähe der *FOB* auf einem Straßenabschnitt platziert. Zum Zeitpunkt $t=-0:20$ min begann der Operateur mit der Kommandierung der Fahrzeugerkennung auf dem Straßenabschnitt im nördlichen Bereich des Flugversuchsbereiches. Hierbei wurde die manuelle Sensorführung durch den Operateur spezifiziert, wodurch die resultierende Handlungsagenda durch ein entsprechendes Symbol ergänzt wurde (vgl. Abbildung 5-18). Nach Erstellung des Flugplans forderte der Agent die Unterstützung für den Start des UAV, welcher bei $t=0:00$ min durch den Sicherheitspiloten durchgeführt wurde. Mit der Übergabe der Flugsteuerung an den Autopiloten initiierte der Agent den Anflug auf den spezifizierten Straßenabschnitt, wo bei $t=0:34$ min die Fahrzeugerkennung begann.

Während des Erkennungsvorgangs übertrug der Agent dem Operateur die manuelle Steuerung des

Aufklärungssensors über das Sensorinterface. Die Flugführung und Sensordatenverarbeitung wurden weiterhin zentral über den kognitiven Agenten koordiniert. Abbildung 5-21 zeigt die hierbei vollzogene Entzerrung des Sensorbildes im Rahmen der Fahrzeugerkennung über *ECG*.

Bei Abschluss des Erkennungsvorgangs ($t=1:20\text{min}$) meldete der Agent das über *SPMS* erkannte Fahrzeug auf der Straße, indem es in die taktische Lage als zusätzliches Objekt eingefügt wurde. Der Operateur kommandierte daraufhin die erneute Aufklärung des Fahrzeugs, wodurch der Agent den Anflug der spezifizierten Position initiierte und ab $t=1:46\text{min}$ für 20 Sekunden Sensoraufnahmen des Fahrzeugs an die mBKS übertrug.

Es folgte die Personenerkennung über der östlichen Rasenfläche (vgl. Abbildung 5-19). Im Anschluss an die Kommandierung durch den Operateur initiierte der Agent bei $t=2:19\text{min}$ den Anflug des Gebiets und begann dort bei $t=2:32\text{min}$ mit dem Erkennungsvorgang entlang eines zuvor generierten Flugmusters. Die Sensorführung wurde, wie zuvor bei der Fahrzeugerkennung, manuell durch den Operateur vollzogen. Die Personenerkennung endete nach Abflug des Flugmusters und mit Einfügen der zwei erkannten Personen in die taktische Lage durch den kognitiven Agenten bei $t=5:15\text{min}$.

Der Operateur kommandierte nacheinander die Aufklärung der beiden erkannten Personen, wodurch bei $t=5:49\text{min}$ und $t=6:43\text{min}$ jeweils 20-sekündige Aufnahmen an den spezifizierten Positionen durch den Agenten initiiert und zur mBKS zur Anzeige übermittelt wurden. Die Flugdemonstration endete mit der Landung des *OktoXL* nahe der *FOB*, wobei die Kommandierung durch den Operateur bei $t=7:18\text{min}$ erfolgte. Nach Erreichen des Landepunktes und der Anfrage zur Unterstützung des kognitiven Agenten bei der UAV-Landung wurde sie bei $t=7:48$ durch den Sicherheitspiloten vollzogen.



Abbildung 5-19: Flugplan zur Durchführung einer Personenerkennung in Area

Plattformunabhängigkeit

Die Plattformunabhängigkeit des Agentensystems zeigt sich maßgeblich durch die Anpassung der Auftragsumsetzung an die Fähigkeiten des jeweiligen UAV-Demonstrators mit Unterschieden im symbolischen Agentenplan als auch in der subsymbolischen Aktionsimplementierung. Zur beispielhaften Darstellung dieser Anpassungsfähigkeit als Grundlage der Plattformunabhängigkeit des Systems werden folgende Unterschiede beschrieben:

1. Die Sensordatenverarbeitung und -übertragung in Abhängigkeit von der Qualität der Sensordaten, der verfügbaren UAV-Fähigkeiten sowie der Datenlinkverfügbarkeit,
2. die Umsetzung von Flugmustern zur Aufklärung über einem Gebiet in Abhängigkeit von den Sensor- und Flugeigenschaften des UAV und
3. die Anwendung des ECG als plattformunabhängigen Sensorführungsprozess am Beispiel der Fahrzeugerkennung des *OktoXL*.

Sensordatenverarbeitung und -übertragung

Durch den Aufklärungskontext der Demonstrationsflüge spielte die Verarbeitung und Übertragung von Sensordaten eine zentrale Rolle bei der Auftrags Erfüllung durch den kognitiven Agenten. Einflussfaktoren waren hierbei die Qualität der Sensordaten, die verfügbaren UAV-Fähigkeiten zur Verarbeitung der Daten und schließlich die Verfügbarkeit des Datenlinks zwischen UAV und *mBKS*.

Beim UAV-Demonstrator *Explorer* wurden die unverarbeiteten Sensordaten zunächst lokal gespeichert und von einem geeigneten Übertragungspunkt aus gemeinsam zur *mBKS* übertragen. Abbildung 5-16 zeigt hierzu die separate Kommandierung von Aufklärung und Datenübertragung in der Auftragsagenda. Tabelle 5-13 führt im Ablauf der *Explorer* Flugdemonstration ebenfalls zeitlich getrennte Vorgänge zur Datenerhebung und -speicherung sowie zur Datenübertragung auf. Gleiches gilt für die nachträgliche Aufklärung der detektierten Person im Einsatzgebiet, ohne dass in diesem Fall die Datenübertragung explizit formuliert wurde. Durch das Wissen des Agenten um die Notwendigkeit einer aktiven Datenverbindung zur Übertragung von Sensordaten, wurden die hierfür benötigten Handlungen selbstständig ergänzt.

Im Demonstrationsflug des *OktoXL* fanden die Datenverarbeitung und -übertragung unmittelbar während des Fluges statt. Die Sensordaten wurden an Bord durch den *ECG* entzerrt und zur *mBKS* übertragen. Der resultierende Agentenplan enthielt daher keine separate Handlung zur Übermittlung der Sensordaten (vgl. Abbildung 5-18 und Abbildung 5-19). Die kontinuierliche Anzeige des Sensorbildes in der *mBKS* ermöglichte dem Operateur dabei die manuelle Kontrolle der Sensorführung an Bord des UAV. Abbildung 5-18 zeigt hierzu die Spezifikation der manuellen Kontrolle als Teil der Auftragsagenda.

Unterschiede bei der Auftragsumsetzung sind auf die spezifische Verfügbarkeit von UAV-Fähigkeiten von *Explorer* und *OktoXL* zurückzuführen, die gemäß Abschnitt 4.2.2 im kognitiven Agenten hinterlegt sind. *Explorer* besaß im Gegensatz zu *OktoXL* keine Fähigkeiten zur Objekterkennung (*vehicleDetection_Available* bzw. *personDetection_Available*), da zum einen der genutzte Sensor unzureichende Daten lieferte und SPMS nicht die entsprechenden Algorithmen implementierte. Hierdurch erkannte der kognitive Agent die Notwendigkeit der Datenauswertung durch den Operateur. Die Datenübertragung setzte jedoch die Datenlinkverfügbarkeit voraus, die

vom Agenten als Fähigkeit (*dataTransmission_Available*) verarbeitet wurde. Die Verfügbarkeit war im Einsatzgebiet jedoch nicht uneingeschränkt gegeben, weswegen der *Transmission Point* zur Datenübertragung genutzt wurde.

Die manuelle Sensorführung wurde bei *OktoXL* realisiert, indem die UAV-eigene Fähigkeit zur Sensorführung (als Teil des *SPMS*) durch menschliche Steuereingaben ersetzt wurde. Die Anzeige eines entsprechenden Symbols im Agentenplan indizierte hierbei Unterschiede in dessen interner Verarbeitung. Auf Seiten des *ECG* wurden Steuereingaben von *SPMS* bzw. des Joysticks während des Erkennungsvorgangs gleichwertig verarbeitet.

Flugmuster

Die Umsetzung spezifischer Flugmuster zur Aufklärung von Objekten, Personen und Gebieten erfolgte subsymbolisch als Teil der Flugplanerstellung über die Auswahl und Parametrisierung hinterlegter Aufklärungsmuster durch den kognitiven Agenten (vgl. Abschnitt 4.3.2). In der symbolischen Umsetzung des Agentenplans fand hierbei keine Anpassung statt.



Abbildung 5-20: Individuelle Umsetzung von Mustern zur Gebietsaufklärung für die UAV-Demonstratoren *Explorer* (links) und *OktoXL* (rechts)

Die Grundlage für Auswahl und Parametrisierung der Flugmuster waren gemäß [Borchers 2014a] u.a. der Öffnungswinkel des Aufklärungssensors sowie Flughöhe über Grund und Flugdynamik des UAV-Demonstrators. Der Demonstrationsflug des *Explorer* wurde in einer Höhe von $h_{AGL}=150\text{m}$ durchgeführt. Für *Explorer* als Flächenflieger wurde zudem ein Kurvenradius von 500m angenommen. *OktoXL* hingegen flog im Demonstrationsflug bei $h_{AGL}=15\text{m}$ und konnte als Multikopter Richtungsänderungen positionsfest umsetzen. Das Fischaugenobjektiv bot bei beiden UAV-Demonstratoren einen großen Öffnungswinkel der Aufklärungssensorik, wobei nicht der gesamte abgebildete Bereich ausreichend gute Daten zur Auswertung lieferte, da die Pixeldichte nach außen hin abnahm.

Die interne Bewertung der Parameter durch den kognitiven Agenten führte zur Erstellung der abgebildeten Flugmuster. Die Stützpunkte des Musters bei *Explorer* wurden dabei den Begrenzungen des aufzuklärenden Gebiets entnommen. Bei *OktoXL* wurde der Abstand zwischen den Stützpunkten auf der Grundlage der Flughöhe über Grund und einem angenommenen horizontalen Winkelbereich von $\alpha=30^\circ$ bestimmt. Unabhängig vom jeweiligen Flugmuster endete

der Erkennungsvorgang bei beiden Demonstratoren jeweils mit dem Erreichen des abschließenden Wegpunkts im Flugplan.

Electronic Camera Gimbal (ECG)

Der *ECG* als Werkzeug zur Sensorführung wurde bei *Explorer* in der mBKS, bei *OktoXL* unmittelbar an Bord des UAV in die Sensordatenverarbeitung integriert. Abbildung 5-21 zeigt die Anwendung des *ECG* am Beispiel der Fahrzeugerkennung durch *OktoXL*. Zu erkennen sind das verzerrte Ausgangsbild des Aufklärungssensors mit Fischaugenobjektiv (links) sowie der entzerrte Bildausschnitt (rechts) als Grundlage der weiteren Verarbeitung. Die Abbildung zeigt ein klar zu erkennendes Fahrzeug auf dem definierten Straßenabschnitt.

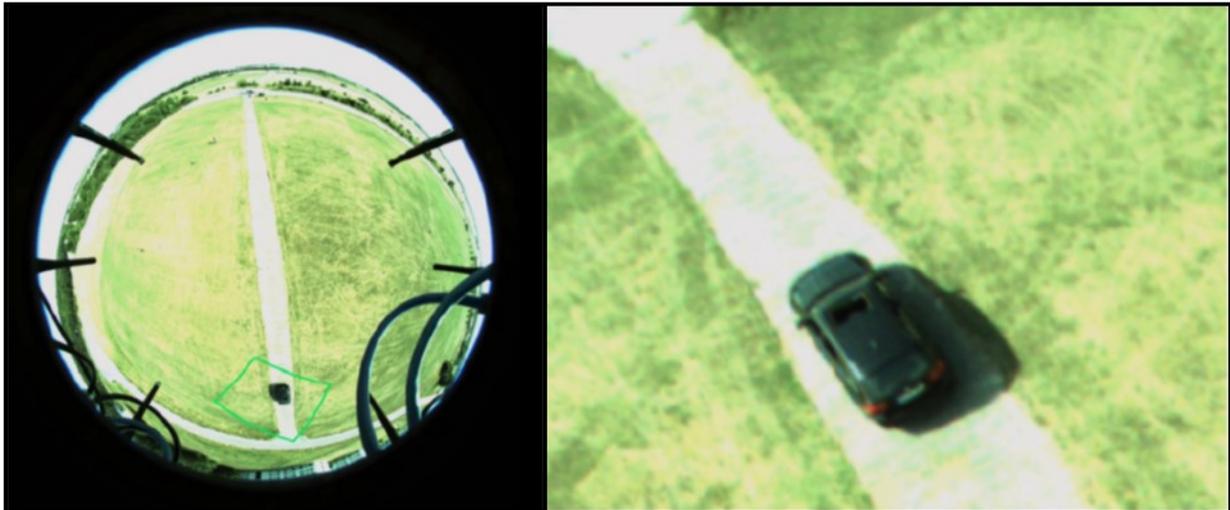


Abbildung 5-21: Anwendung des Electronic Camera Gimbal (ECG) auf die Ausgabedaten des Aufklärungssensors des UAV-Demonstrators *OktoXL*. Zu sehen sind das Ausgangsbild (links) und der entzerrte Bildausschnitt (rechts) [Rudnick et al. 2014]

Das Ausgangsbild zeigt die Originalaufnahme mit einem eingezeichneten Trapez, welches den aktiven Bildausschnitt markiert und im Flugversuch über Joystickeingaben in der mBKS manipuliert werden konnte. Neben Zoomen (Größenänderung des Trapezes) war auch die freie Bewegung des Bildausschnitts (über Drehung und Verschiebung des Trapezes) möglich.

Die Auflösung des entzerrten Bildausschnitts ergab sich aus der zugrundeliegenden Pixeldichte sowie der jeweiligen Zoomstufe [Böhm et al. 2013]. Im Flugversuch waren die Auflösung und damit die Datenmenge des Bildausschnitts deutlich geringer als die der zugrunde liegenden Originalaufnahme, woraus (bei *OktoXL*) die Möglichkeit zur Liveübertragung der Sensordaten an die mBKS resultierte.

Die Flugversuche zeigen die sinnvolle Ergänzung der Plattformunabhängigkeit des vorgestellten Automationsansatzes durch den *ECG*, der in Abhängigkeit von den verfügbaren Verarbeitungskapazitäten und der Datenlinkverfügbarkeit an verschiedenen Stellen des Arbeitssystems eingebracht werden kann. Die Abbildung einer UAV-eigenen Fähigkeit zur Sensorführung durch einen geeigneten Führungsalgorithmus in *SMPS* wurde im Rahmen der Erprobung bereits in der Simulation erfolgreich umgesetzt, war jedoch zum Zeitpunkt der Demonstrationsflüge nicht für den Realflug verfügbar.

6 Zusammenfassung und Ausblick

Die Befähigung eines einzelnen Operators zur alleinverantwortlichen Führung von hochautomatisierten UAV in komplexen Anwendungen ist eine erklärte Forderung militärischer und ziviler Nutzer und Inhalt dieser Dissertation. Die aktuelle Forschung konzentriert sich hierbei weitestgehend unabhängig voneinander auf die Teilaspekte der Autonomiesteigerung von UAV und der Schaffung geeigneter Mensch-Maschine Schnittstellen für die Abbildung der Aufgaben heutiger Multi-Operateur Konfigurationen durch einen einzelnen Operateur. Diese Ansätze lassen jedoch außer Acht, dass eine gesamtheitliche Betrachtung des gemeinsamen Handelns von Mensch und Automation und ihrer gegenseitigen Abhängigkeit und Beeinflussung notwendig ist.

In Abschnitt 2.1.5 wurden die begrenzten kognitiven Fähigkeiten des Menschen als limitierender Faktor der durch einen Operateur beherrschbaren Systemkomplexität identifiziert. Diese Begrenzung kann auch durch zusätzliche konventionelle Automationskomponenten nicht aufgehoben werden. Aus diesem Grund wurde zur Unterstützung des Operators und zur Leistungssteigerung der UAV in dieser Arbeit die Integration kognitiver Automation in Form intelligenter KI-Agenten (sog. Kognitive Agenten) als zusätzliche Automationskomponente an Bord des Luftfahrzeugs gewählt.

Zunächst wurde hierzu der Ansatz der *Agent Supervisory Control* (ASC) vorgestellt, der den kognitiven Agenten gleichermaßen als Untergebenen des Operators sowie als Leiter des nachgeordneten UAV Systems betrachtet. Über die Etablierung einer Etikette für die Mensch-Agenten Interaktion wurde versucht, die Wahrnehmung des Agenten durch den Operateur positiv zu beeinflussen und das Delegationsverhalten des Operators sowie dessen mögliche Entlastung bei der UAV-Führung über die Schaffung kalibrierten Vertrauens zu verbessern. Ausgehend vom Ziel der Entkopplung des Agenten von den nachgeordneten Automationskomponenten und der UAV-Plattform wurden UAV-Fähigkeiten aus Automationsfunktionen abgeleitet, deren Verfügbarkeit über Systemressourcen und –zustände ermittelt wurde.

In der weiterführenden Betrachtung wurde der Ansatz der ASC mit dem Konzept der auftragsbasierten UAV-Führung nach [Uhrmann 2013] zur Delegation von Aufgaben verknüpft. Die Auftragsverarbeitung des Agenten wurde dabei über eine dreistufige Verarbeitungskette realisiert, die eigenständiges Handeln innerhalb des eigenen Handlungsrahmens umsetzt und situations- sowie auftragspezifisches Agentenfeedback ermöglicht. Die Rolle des Menschen als oberste Entscheidungsinstanz im UAV-System blieb dabei unverändert und wurde in Abschnitt 3.2.4 durch die Möglichkeit zum manuellen Eingriff in Automationsfunktionen im Sinne einer Fähigkeitssubstitution und Fähigkeitsaddition ergänzt.

Für die prototypische Implementierung des Kognitiven Agenten wurde die bestehende Systemarchitektur COSA² durch die reaktive Erweiterung ReX ergänzt, um eine regelbasierte Prozedurauswahl sowie reaktives Agentenverhalten in zeitkritischen Situationen zu ermöglichen. Es folgte die Implementierung prototypischer Automationskomponenten und ihre Anbindung an den kognitiven Agenten als UAV-Fähigkeiten. Die Integration der Komponenten verfolgte dabei den Ansatz der Plattformunabhängigkeit durch eine modulare Anbindung von Endgeräten und eine hierarchische Abstraktion der Schnittstellen.

Das entwickelte Gesamtsystem wurde anschließend zunächst in Mensch-Maschine-Experimenten in einer komplexen Missionssimulation evaluiert. Hierbei konnte nachgewiesen werden, dass der Kognitive Agent den Menschen befähigt ein UAV, in einer militärischen Aufklärungsmission, alleinverantwortlich zu führen. Zudem konnte der positive Effekt des Interaktionsverhaltens des Agenten durch eine signifikante Steigerung von Vertrauen und Akzeptanz des Menschen in den Agenten sowie durch eine signifikante Senkung der Interaktionszeit und Fehlerhäufigkeit bei korrigierenden Eingriffen während der Missionsdurchführung gezeigt werden. Effekte auf andere intrinsische Größen des Menschen, wie z.B. die Beanspruchung und das Situationsbewusstsein, konnten in den Experimenten dagegen lediglich unzureichend untersucht werden. Abschließend wurden der Reifegrad und die Plattformunabhängigkeit der Implementierung im Realflugversuch demonstriert. Hierzu wurde das Agentensystem auf zwei unterschiedlichen UAV-Demonstratoren mit spezifischen Automationskomponenten integriert und in komplementären Missionen erfolgreich zum Einsatz gebracht.

Die Ergebnisse dieser Dissertation stellen durch die Verknüpfung der Mensch-Maschine Evaluation in der Simulation mit der Demonstration im Realflug eine breite Ausgangsbasis für weiterführende Untersuchungen dar.

Aufbauend auf Ergebnissen nach [Uhrmann 2013] bietet sich zukünftig die Untersuchung einer multi-UAV-Führung durch einen einzelnen Operateur hinsichtlich der Mensch-Agenten Kommunikation an. In diesem Rahmen könnten durch umfangreichere Personenstichproben auch aussagekräftige Ergebnisse bezüglich des Einflusses des Interaktionsverhaltens des Agenten in verschiedenen Variationen auf die Beanspruchung und das Situationsbewusstseins von Probanden gewonnen werden. Durch die plattformunabhängige Gestaltung des Systemkonzepts sind hierbei sowohl weitere Laborexperimente als auch Realflugversuche denkbar. Schließlich sind sogenannte Mixed-Reality Versuche möglich, bei denen aus einer Simulationsumgebung heraus gleichzeitig simulierte UAV und reale UAV-Demonstratoren über eine gemeinsame Schnittstelle geführt werden.

Ergänzend kann das Arbeitssystem durch das Einbringen weiterer kognitiver Automation in Form eines Assistenzsystems (*Mode 2 Cognitive Automation*) ergänzt werden, um menschliche Fehler bei der UAV-Führung zu reduzieren und die Leistung des Gesamtsystems weiter zu steigern. Durch die Integration der Assistenzfunktionen in die vorhandene Mensch-Maschine Schnittstelle ist die Untersuchung eines Arbeitssystems mit zwei hierarchisch gegliederten kognitiven Agenten möglich [Theißing & Schulte 2014]. Hierbei gilt es zu untersuchen, in welcher Form der Informationsfluss zwischen den Agenten bzw. zwischen dem Mensch und den Agenten angepasst werden muss, um größtmöglichen Nutzen aus den Agentenfähigkeiten zu ziehen. Einen weiteren Aspekt wird hierbei der Eingriff des menschlichen Operateurs auf niedrigeren Führungsebenen und die Verarbeitung des resultierenden Systemverhaltens durch die Agenten darstellen. In diesem Zusammenhang ist zu klären, inwiefern das Assistenzsystem ebenfalls die Autorität erhält, selbstständig Eingaben, auch auf nachgeordneten Eingriffsebenen zu tätigen.

A Literaturverzeichnis

- AEROVIRONMENT, INC. (2016). *Unmanned Aircraft Systems (UAS)* [online]. *The Future is Unmanned*. Silicon Valley (CA): AeroVironment Inc. [Zugriff am: 10.02.2016]. Verfügbar unter: <http://www.avinc.com/uas/>
- AIRROBOT GMBH & CO. KG (2016). *AirRobot – Produkte* [online]. *Die Universelle Plattform für die Überwachung, Inspektion und Dokumentation aus der Luft*. Arnsberg: AirRobot GmbH & Co. KG [Zugriff am: 10.02.2016]. Verfügbar unter: <http://www.airrobot.de/index.php/produkte.html>
- AMERICAN PSYCHOLOGICAL ASSOCIATION (2016). *Glossary of Psychological Terms* [online]. Washington (DC): American Psychological Association [Zugriff am: 05.11.2014]. Verfügbar unter: <http://www.apa.org/research/action/glossary.aspx>
- ANDERSON, John R., Daniell BOTHELL, Michael D. BYRNE, Scott DOUGLASS, Christian LEBIERE und Yulin QIN (2004). An Integrated Theory of the Mind. In: K. J. HOLYOAK (Hrsg.) *Psychological Review*. **111** (4), S. 1036-1060. ISSN 0033-295X
- ANDERSON, John R. (2014). *Cognitive Psychology and Its Implications*. 8. Auflage. New York, NY: Worth. ISBN 978-14641148910
- AURICH, Pierre (2011). *Entwicklung eines kognitiven Agenten zur auftragsbasierten UAV-Führung unter Nutzung von „Optimal Control“* [Masterarbeit]. Neubiberg: Universität der Bundeswehr München
- AUSTIN, Reg (2010). *Unmanned Aircraft Systems: UAVs Design, Development and Deployment*. Chicester (West Sussex): John Wiley & Sons. ISBN 978-0-470-05819-0
- BAINBRIDGE, Lisanne (1983). Ironies of Automation. In: *Automatica*. **19** (6), S. 775-779. ISSN 0005-1098
- BARBIER, Magali und Elodie CHANTHERY (2004). Autonomous Mission Management for Unmanned Aerial Vehicles. In: *Aerospace Science and Technology*. **8** (4), S. 359-368. ISSN 1270-9638
- BENJAMIN, D. Paul, Damian LYONS und Deryle LONSDALE (2004). ADAPT: A Cognitive Architecture for Robots. In: *Proceedings of the Sixth International Conference on Cognitive Modeling (ICCM 2004 - Integrating Models)*. Pittsburgh (PA), 30. Juli – 01. August. Mahwah (NJ): Lawrence Erlbaum Associates, S. 337-338. ISBN 0-8058-5426-6
- BILLINGS, Charles E. (1997). *Aviation Automation: The Search for a Human-Centered Approach*. Mahwah (NJ): Lawrence Erlbaum Associates. Human Factors in Transportation. ISBN 978-0805821277
- BLISS, James P. (2003). Investigation of Alarm-Related Accidents and Incidents in Aviation. In: *The International Journal of Aviation Psychology*. **13** (3), S. 249-268. ISSN 1050-8414
- BOHEMIA INTERACTIVE SIMULATIONS (2015). *Virtual Battlespace 2 | BISim* [online]. Orlando (FL): Bohemia Interactive Simulations [Zugriff am: 14.03.2015]. Verfügbar unter: <https://bisimulations.com/virtual-battlespace-2/>

- BÖHM, Florian und Axel SCHULTE (2012). UAV Autonomy Research – Challenges and Advantages of a fully distributed System Architecture. In.: *International Telemetering Conference (ITC 2012)*. San Diego (CA), 22. – 25. Oktober
- BÖHM, Florian, Stefan SCHNEEMILCH und Axel SCHULTE (2013). The Electronic Camera Gimbal. In: *AIAA Infotech@Aerospace Conference 2013*. Boston (MA), 19. – 22. August. Reston (VA): American Institute for Aeronautics and Astronautics (AIAA), S. 907-914. ISBN 978-1-62993-152-4
- RAMEY, Robert (2004). *Boost Serialization* [online]. Zugriff am: 09.09.2014. Verfügbar unter: http://www.boost.org/doc/libs/1_51_0/libs/serialization/doc/index.html
- BORCHERS, Annike (2014a). *Konzeption und Entwicklung eines Programms zur automatisierten Auswahl und Parametrisierung von Aufklärungsverfahren* [Masterprojekt]. Neubiberg: Universität der Bundeswehr München
- BORCHERS, Annike (2014b). *Konzeptionierung und Durchführung einer Versuchsreihe zur Evaluierung eines kognitiven Agenten in Bezug auf die Planung und Kommandierung von Aufklärungsmissionen* [Masterarbeit]. Neubiberg: Universität der Bundeswehr München
- BORMATEC (2015). *Bormatec - Unmanned Vehicles* [online]. Ravensburg: BORMATEC [Zugriff am: 17.04.2015]. Verfügbar unter: <http://www.bormatec.com/>
- BORST, Jelmer P. und John R. ANDERSON (2014). Using the ACT-R Cognitive Architecture in combination with fMRI data. In: B. U. FORSTMANN und E.-J. WAGENMAKERS (Hrsg.) *An Introduction to Model-Based Cognitive Neuroscience*. New York (NY): Springer. ISBN 978-1-4939-2236-9
- BRAND, Yannik (2012). *Konzeption und Implementierung eines Missionsplaners für bemannte-unbemannte Operationen* [Masterprojekt]. Neubiberg: Universität der Bundeswehr München
- BROCKHAUS, Rudolf, Wolfgang ALLES und Robert LUCKNER (2011). *Flugregelung*, 3. Auflage. Berlin, Heidelberg: Springer. ISBN 978-3-642-01443-7
- BROOKS, Rodney A. (1990). Elephants don't play chess. In: *Robotics And Autonomous Systems – Designing Autonomous Agents*. 6 (1-2), S. 3-15. Amsterdam: Elsevier Science. ISSN 0921-8890
- BROOKS, Rodney A. (1999). *Cambrian Intelligence: The Early History of the New AI*. Cambridge (MA): MIT Press. ISBN 978-0-262-02468-6
- BROWER, Holly H., F. David SCHOORMAN und Hwee Hoon TAN (2000). A Model of Relational Leadership: The Integration of Trust and Leader-Member Exchange. In: *The Leadership Quarterly* [online]. 11 (2), S. 227-250. ISSN 1048-9843
- BRÜGGENWIRTH, Stefan und Axel SCHULTE (2012). COSA² - A Cognitive System Architecture with Centralized Ontology and Specific Algorithms. In: *2012 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Seoul, 14.-17. Oktober. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 307-314. ISBN 978-1-4673-1713-9

- CHEN, Jessie Y. C., Katelyn PROCCI, Michael BOYCE, Julia WRIGHT, Andre GARCIA und Michael BARNES (2014) *Situation Awareness-Based Agent Transparency*. (ARL-TR-6905). Aberdeen Proving Ground (MD): U.S. Army Research Laboratory (ARL)
- CHOI, Dongkyu, Tolga KONIK, Negin NEJATI, Chunki PARK und Pat LANGLEY (2007). A Believable Agent for First-Person Shooter Games. In: *Proceedings of the 3rd Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2007)*. Stanford (CA), 06. – 08. Juni. Menlo Park (CA): AAAI Press, S. 71-73. ISBN 978-1-57735-325-6
- CHONG, Hui Qing, Ah-Hwee TAN und Gee-Wah NG (2007). Integrated cognitive architectures: a survey. In: *Artificial Intelligence Review*. **28** (2), S. 103-130. ISSN 0269-2821
- CLAUB, Sebastian, Pierre AURICH, Stefan BRÜGGENWIRTH, Vladimir DOBROKHODOV, Isaac KAMINER und Axel SCHULTE (2012). Design and Evaluation of a UAS combining Cognitive Automation and Optimal Control. In: *AIAA Infotech@Aerospace Conference 2012*. Garden Grove (CA), 19. – 21. Juni. Reston (VA): American Institute of Aeronautics and Astronautics (AIAA)
- CLAUB, Sebastian und Axel SCHULTE (2014a). Implications for Operator Interactions in an Agent Supervisory Control Relationship. In: *Proceedings of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS 2014)*. Orlando (FL), 27. – 30. Mai. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 703-714. ISBN 978-1-47992-377-9
- CLAUB, Sebastian und Axel SCHULTE (2014b). Leitende Kontrolle eines UAV durch einen Kognitiven Agenten. In: *3. Interdisziplinärer Workshop – Kognitive Systeme: Mensch, Teams, Systeme und Automaten*. Magdeburg, 25. – 27. März
- CLAUB, Sebastian, Elisabeth DENK und Axel SCHULTE (in Druck): Experimental Evaluation of Varying Feedback of a Cognitive Agent System for UAV Mission Management. In: *Advances in Aviation Psychology*, Volume 2. Farnham (Surrey): Ashgate. Ashgate Studies in human factors for flight operations
- CLOT, André J. (1999). Communications, Command and Control – The Crowded Spectrum. In: *Development and Operation of UAVs for Military and Civil Applications*. S. 2B-1 – 2B-8. Neuilly-Sur-Seine: NATO Research and Technology Organization (RTO). ISBN 92-837-1033-9
- CLOUD CAP TECHNOLOGY (2015). *Piccolo Autopilots | Cloud Cap Technology* [online]. Hood River (OR): Cloud Cap Technology [Zugriff am: 02.01.2015]. Verfügbar unter: <http://www.cloudcaptech.com/products/auto-pilots/>
- CLOUGH, Bruce T. (2002a). Metrics, Schmetrics! How the Heck Do You Determine A UAV's Autonomy Anyway?. In: *Proceedings of the 2002 Performance Metrics for Intelligent Systems Workshop (PerMIS-02)*. Gaithersburg (MD), 13. – 15. August
- CLOUGH, Bruce T. (2002b). Unmanned Aerial Vehicles: Autonomous Control Challenges, A Researcher's Perspective. In: R. MURPHEY und P. M. PARDALOS (Hrsg.) *Cooperative Control and Optimization*. New York (NY): Kluwer Academic, S.35-52. ISBN 1-4020-0549-0

- COMMELL (2015). *Industrial PC, Surveillance, OEM/ODM Service* [online]. New Taipei City: Taiwan Commate Computer Inc. [Zugriff am: 17.04.2015].
Verfügbar unter: <http://www.commell.com.tw/>
- CUMMINGS, Mary L., Sylvain BRUNI, S. MERCIER und P. J. MITCHELL (2009). Automation Architecture for Single Operator-Multiple UAV Command and Control. In: *International Command and Control Journal*. **1** (2), S. 1-24. ISSN 1938-6044
- DEFENSE INDUSTRY DAILY (2011). *It's better to Share: Breaking Down UAV GCS Barriers* [online]. Defense Industry Daily, LLC [Zugriff am: 06.11.2014]. Verfügbar unter: <http://www.defenseindustrydaily.com/uav-ground-control-solutions-06175/>
- DEFENSE SCIENCE BOARD (2012). *Task Force Report: The Role of Autonomy of DoD Systems*. Washington, DC: Office of the Secretary of Defense
- DEKKER, Sidney W. A. und David D. WOODS (2002): MABA-MABA or Abracadabra? Progress on Human-Automation Co-ordination. In: *Cognition, Technology & Work*. **4** (4), S. 240-244. ISSN 1435-5558
- DENFORD, James, John A. STEEL, Roger ROY und Eugenia KALANTZIS (2004). Measurement of Air Traffic Control Situational Awareness Enhancement through Radar Support toward Operating Envelope Expansion of an Unmanned Aerial Vehicle. In: *WSC '04 Proceedings of the 36th Conference on Winter Simulation*. Washington (DC), 05. – 08. Dezember. Washington (DC): Winter Simulation Conference, S. 1017-1025. ISBN 0-7803-8786-4
- DENK, Elisabeth, Sebastian CLAUB, Annike BORCHERS, Josef WERNER und Axel SCHULTE (2014). Experimental Evaluation of varying Feedback of a Cognitive Agent System for UAV Mission Management. In: *18th International Symposium on Aviation Psychology (ISAP 2015)*. Dayton (OH), 04. – 07. Mai. Dayton (OH): Wright State University Conferences and Events, S. 171-176. ISBN 978-1-51080-408-1
- DEPARTMENT OF DEFENSE (2008). *MIL-STD-2525C, Department of Defense Interface Standard: Common Warfighting Symbolology*. Washington, DC: Office of the Secretary of Defense
- DEPARTMENT OF DEFENSE (2011). *Unmanned Systems Integrated Roadmap FY2011-2036*. (Reference Number 11-S-3613). Washington, DC: Office of the Secretary of Defense
- DEUBEL, Heiner und Werner X. SCHNEIDER (1996). Saccade target selection and object recognition: Evidence for a common attentional mechanism. In: *Vision Research*. **36** (12), S. 1827-1837. ISSN 0042-6989
- DEUTSCHE GESELLSCHAFT FÜR LUFT- UND RAUMFAHRT (2008). *DGLR: L3.1 Unmanned Aerial Vehicles (UAV)* [online]. Bonn: Deutsche Gesellschaft für Luft- und Raumfahrt [Zugriff am: 24.02.2015]. Verfügbar unter: <http://www.dglr.de/?id=2502/>
- DEWHIRST, H. D., V. METTS und R. T. LADD (1987). Exploring the delegation decision: Managerial responses to multiple contingencies. In: *Annual Meeting of the Academy of Management*. New Orleans (LA), 09. – 12. August. Briarcliff Manor (NY): Academy of Management
- DIGI INTERNATIONAL INC. (2014). *XBee-PRO 868 – Digi International* [online]. *Long-Range RF Module for Europe*. Minnetonka (MN): Digi International Inc. [Zugriff am: 18.08.2014]. Verfügbar unter: <http://www.digi.com/products/xbee-rf-solutions/modules/xbee-pro/868/>

- DIMPERIO, Eric und Glenn GUNZELMANN (2008). An Initial Evaluation of a Cognitive Model of UAV Reconnaissance. In: J. HANSBERGER (Hrsg.) *Proceedings of the Seventeenth Conference on Behavior Representation in Modeling and Simulation*. Providence (RI), 14. – 17. April. Orlando (FL): Simulation Interoperability Standards Organization (SISO), S. 165-173. ISBN 978-1-60560-124-3
- DIXON, Stephen R., Christopher D. WICKENS und Dervon CHANG (2004). Unmanned Aerial Vehicle Flight Control: False Alarms versus Misses. In: *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting*. Saint Louis (MO), 20. – 24. September. Santa Monica (CA): Human Factors and Ergonomics Society (HFES), S. 152-156. ISSN 1071-1813
- DOBROKHODOV, Vladimir, Ioannis KITSIOS, Isaac KAMINER, Kevin D. JONES, Enric XARGAY, Naira HOVAKIMYAN, Chengyu CAO, Mariano I. LIZARRAGA, und Irene M. GREGORY (2008). Flight Validation of a Metrics Driven L1 Adaptive Control. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. Honolulu (HI), 18. – 21. August. Reston (VA): American Institute of Aeronautics and Astronautics (AIAA), S. 3877-3898. ISBN 978-1-60560-808-2
- DONATH, Diana (2011). *Verhaltensanalyse der Beanspruchung des Operateurs in der Multi-UAV-Führung* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- DUDENVERLAG (2014). *Duden | Fähigkeit | Rechtschreibung, Bedeutung, Definition, Synonyme* [online]. Berlin: Bibliografisches Institut GmbH – Dudenverlag [Zugriff am: 26.01.2015]. Verfügbar unter: <http://www.duden.de/rechtschreibung/Faehigkeit>
- EICHHORN, Mike-Joachim (2007). *Intelligentes Führungskonzept für ein Autonomes Unterwasserfahrzeug in Sondersituationen* [Dissertation]. Ilmenau: Technische Universität
- EISENBEISS, Henri (2004). A mini Unmanned Aerial Vehicle (UAV): System Overview and Image Acquisition. In: *WG V/6 International Workshop on Processing and Visualization using High-Resolution Imagery*. Pitsanulok, 18. – 20. November. Hannover: International Society for Photogrammetry and Remote Sensing (ISPRS)
- EMT INGENIEURGESELLSCHAFT (2014). *ALADIN Drohnensystem – EMT Penzberg* [online]. Penzberg: EMT Ingenieurgesellschaft [Zugriff am: 10.11.2014]. Verfügbar unter: <http://www.emt-penzberg.de/produkte/aladin/aladin.html>
- ENDSLEY, Mica R. (1988). Situation Awareness Global Assessment Technique (SAGAT). In: *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference (NAECON 1988)*. Dayton (OH), 23. – 27. Mai. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 789-795
- ENDSLEY, Mica R. (1995). Measurement of Situation Awareness in Dynamic Systems. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society*. **37** (1), S. 65-84. ISSN 0018-7208
- ENDSLEY, Mica R. (2000). Direct measurement of situation awareness. Validity and use of SAGAT. In: M. R. ENDSLEY und D. J. GARLAND (Hrsg.) *Situation Awareness Analysis and Measurement*. Mahwah (NJ): Lawrence Erlbaum Associates, S. 147-173. ISBN 0-8058-2133-3

- ENDSLEY, Mica R. und David Kaber (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. In: *Ergonomics*. **42** (3), S. 462-492. ISSN 0014-0139
- FITTS, Paul M. (1951). Human Engineering for an Effective Air-Navigation and Traffic-Control System. In: *Air Navigation Development Board*. Columbus (OH): Ohio State University Research Foundation
- FRANKLIN, Stan und Art GRAESSER (1997). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: *Intelligent Agents III. Agent Theories, Architectures, and Languages*. Berlin, Heidelberg: Springer, S. 21-35. Lecture Notes in Artificial Intelligence. ISBN 978-3-540-62507-0
- FREY, Andreas (2004): *Überwachung und Kontrolle in einem künstlichen kognitiven System für die autonome Fahrzeugführung* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- GAMMA, Erich, Richard HELM, Ralph JOHNSON und John VLISSIDES (2011). *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software*. Deutsche Übersetzung von D. Riehle. München: Addison-Wesley. ISBN 978-3-8273-3043-7
- GANGL, Stefan (2015). *Kooperative Führung mehrerer unbemannter Luftfahrzeuge aus einem einsitzigen Kampflugzeug* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- GAZTANGA, Ion (2014). *Sharing memory between processes* [online]. *Boost C++ Libraries*. [Zugriff am: 13.08.2014]. Verfügbar unter: http://www.boost.org/doc/libs/1_56_0/doc/html/interprocess/sharedmemorybetweenprocesses.html
- GENERAL ATOMICS (2014). *Aircraft Platforms* [online]. Poway (CA): General Atomics Aeronautical Systems, Inc. [Zugriff am: 12.11.2014]. Verfügbar unter: <http://ga-asi.com/aircraft-platforms/>
- GEORGEFF, Michael P. und Francois F. INGRAND (1989). Decision Making in an Embedded Reasoning System. In: *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI 1989)*. Detroit (MI), 20. – 26. August. International Joint Conferences on Artificial Intelligence, S. 972-978. ISBN 1-55860-094-9
- GEORGEFF, Michael P. und Anand S. RAO (1996). A profile of the Australian AI Institute. In: *IEEE Expert*. **11** (6), S. 89-92. ISSN 0885-9000
- GIESBRECHT, Jared (2004). *Global Path Planning for Unmanned Ground Vehicles*. (DRDC Suffield TM 2004-272). Suffield (AB): Defense Research and Development Canada
- GIESE, Stefanie, David CARR und Javaan CHAHL (2013). Implications for Unmanned Systems Research of Military UAV Mishap Statistics. In: *2013 IEEE Intelligent Vehicles Symposium (IV 2013)* [CD-ROM]. Gold Coast (CA), 23. – 26. Juni. New York (NY): Institute of Electrical and Electronics Engineers (IEEE). ISBN 978-1-46732-754-1
- GLUCK, Kevin A., Jerry T. BALL, Glenn GUNZELMANN, Michael A. KRUSMARK, Don R. LYON und Nancy J. COOKE (2005). A Prospective Look at a Synthetic Teammate for UAV Applications. In: *Infotech@Aerospace*. Arlington (VA), 26. – 29. September. Reston (VA): American Institute of Aeronautics and Astronautics (AIAA)

- GLUCK, Kevin A., Jerry T. BALL, Michael A. KRUSMARK, Stuart M. RODGERS und Mathew D. PURTEE (2003). A Computational Process Model of Basic Aircraft Maneuvering. In: F. DETJE, D. DORNER und H. SCHAUB (Hrsg.) *Proceedings of the 5th International Conference on Cognitive Modeling: The Logic of Cognitive Systems*. Bamberg, 10. – 12. April. Bamberg: Universitas, S. 117-122. ISBN 978-3-93346-315-9
- GOERZEN, C., Z. KONG und B. METTLER (2010). A survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. In: *Journal of Intelligent and Robotic Systems*. **57** (1-4), S. 65-100. ISSN 0921-0296
- GOOGLE INC. (2008). *Protocol Buffers – Google’s Data Interchange Format* [online]. Mountain View (CA): Google Inc. [Zugriff am: 15.05.2015]. Verfügbar unter: <https://github.com/google/protobuf/>
- GRAEN, George B. und Mary UHL-BIEN (1995). Relationship-based approach to leadership: Development of leader-member exchange (LMX) theory of leadership over 25 years: Applying a multi-level multi-domain perspective. In *Leadership Quarterly*. **6** (2), S. 219-247. ISSN 1048-9843
- GROTE, Gudela, Steffen WEIK, Toni WÄFLER, Martina ZÖLCH und Cornelia RYSER (1999). KOMPASS (Komplementäre Analyse und Gestaltung von Produktionsaufgaben in soziotechnischen Systemen). In: H. DUNCKEL (Hrsg.) *Handbuch psychologischer Arbeitsanalyseverfahren*. Zürich: vdf Hochschulverlag, S. 255-284. Mensch, Technik, Organisation. ISBN 3-7281-2238-6
- GUNETTI, Paolo, Haydn THOMPSON und Tony DODD (2013). Simulation of Soar-Based Autonomous Mission Management System for Unmanned Aircraft. In: *Journal of Aerospace Information Systems*. **10** (2), S. 53-70. ISSN 2327-3097
- GUPTA, Suraj G., Mangesh M. GHONGE und Pradip JAWANDHIYA (2013). Review of Unmanned Aircraft System (UAS). In: *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*. **2** (4), S. 1646-1658. ISSN 2278-1323
- HACKER, Winfried und Pierre SACHSE (2014). *Allgemeine Arbeitspsychologie: Psychische Regulation von Tätigkeiten*. 3. vollständig überarbeitete Auflage. Göttingen: Hogrefe. ISBN 978-3-80172-540-2
- HAGER, Willi (1987). Grundlagen einer Versuchsplanung zur Prüfung empirischer Hypothesen in der Psychologie. In: G. LÜER (Hrsg.) *Allgemeine Experimentelle Psychologie*. Stuttgart: Fischer, S. 43-264. ISBN 978-3-43700-477-3
- HANCOCK, Peter A. und Stephen F. SCALLEN (1996). The Future of Function Allocation. In: *Ergonomics in Design: The Quarterly of Human Factors Applications (EID)*. **4** (4), S. 24-29. ISSN 1064-8046
- HARRIS, Richard J. (1997). Significance Tests Have Their Place. In: *Psychological Science*. **8** (1), S. 8-11. ISSN 0956-7976
- HART, Sandra G. (2006). NASA-Task Load Index (NASA-TLX): 20 Years Later. In: *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*. San Francisco (CA), 16. – 20. Oktober. Santa Monica (CA): Human Factors and Ergonomics Society (HFES), S. 904-908. ISSN 1071-1813

- HART, Sandra G. und Lowell E. STAVELAND (1988). Development of the NASA-TLX: Results of Empirical and Theoretical Research. In: P. HANCOCK und N. MESHKATI (Hrsg.) *Human Mental Workload*. Amsterdam: North Holland, S. 139-183. ISBN 978-0-44470-388-0
- HARTLEY, Ralph und Frank PIPITONE (1991). Experiments with the subsumption architecture. In: *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*. Sacramento (CA), 09. – 11. April. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 1652-1658. ISBN 0-818-62163-X
- HASSLACHER, Brosl und Mark W. TILDEN (1995). Living machines. In: *Robotics and Autonomous Systems*. **15** (1-2), S. 143-169. ISSN 0921-8890
- HELLERT, Christian, Denis SMIRNOV, Martin RUSS und Peter STÜTZ (2012). A High Level Active Perception Concept for UAV Mission Scenarios. In: *Deutscher Luft- und Raumfahrtkongress 2012*. Berlin, 10. – 12. September. Bonn: Deutsche Gesellschaft für Luft- und Raumfahrt
- HELLERT, Christian und Peter STÜTZ (2014). A Concept for Adaptation of Perceptual Capabilities for UAV Platforms using Case-Based Reasoning. In: *52nd AIAA Aerospace Sciences Meeting*. Reston (VA), 13. – 17. Januar. Reston (VA): American Institute of Aeronautics and Astronautics (AIAA), S. 7725-7732. ISBN 978-1-63266-928-5
- HERCZEG, Michael (2014). *Prozessführungssysteme: Sicherheitskritische Mensch-Maschine-Systeme und Interaktive Medien zur Überwachung und Steuerung von Prozessen in Echtzeit*. München: Oldenbourg Wissenschaftsverlag. ISBN 978-3-48658-445-5
- HILL Jr., Randall W., Johnny CHEN, Jonathan GRATCH, Paul ROSENBLOOM und Milind TAMBE (1998). Soar-RWA: Planning, Teamwork, and Intelligent Behavior for Synthetic Rotary Wing Aircraft. In: *Proceedings of the 7th Conference on Computer Generated Forces & Behavioral Representation*. Orlando (FL), 12. – 14. Mai. Orlando (FL): University of Central Florida - Institute for Simulation and Training, S. 177-188
- HiSYSTEMS GMBH (2015). *MikroKopter – Universelle UAV Lösung* [online]. Moormerland: HiSystems GmbH [Zugriff am: 18.04.2015]. Verfügbar unter: <http://www.mikrokoetter.de/>
- HOFFMANN, Jörg (2015): *Metric-FF Homepage* [online]. Saarbrücken: Universität Saarland – Foundations of Artificial Intelligence Group [Zugriff am: 04.02.2015]. Verfügbar unter: <https://fai.cs.uni-saarland.de/hoffmann/metric-ff.html>
- HOFFMANN, Robert R., Paul J. FELTOVICH, Kenneth M. FORD, David D. WOODS, Gary KLEIN und Anne FELTOVICH (2002). A Rose by Any Other Name ... Would Probably Be Given an Acronym. In: *IEEE Intelligent Systems*. **17** (4), S. 72-80. ISSN 1541-1672
- HOLLINGSWORTH, A. T. und Abdul Rahman A. AL-JAFARY (1983). Why supervisors don't delegate and employees won't accept responsibility. In: *Supervisory Management*. **28** (4), S. 12-17.
- HOLLNAGEL, Erik (1993). *Human Reliability Analysis Context and Control*. London: Academic Press. Computers and People. ISBN 978-0-12352-658-8
- HOLLNAGEL, Erik (2004). Automation and Human Work. In: C. SANDOM und R. HARVEY (Hrsg.) *Human Factors for Engineers*. London: The Institution of Engineering and Technology, S. 113-150. ISBN 978-0-86341-329-2

- HORNER, Douglas P. und Anthony J. HEALEY (2004). Use of Artificial Potential Fields for UAV Guidance and Optimization of WLAN Communication. In: *2004 IEEE/OES Autonomous Underwater Vehicles*. Monterey (CA), 17. – 18. Juni. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 88-95. ISBN 078-0-38543-8
- HOWES, Andrew, David REITTER und Christian LEBIERE (2011): How groups develop a specialized domain vocabulary: A cognitive multi-agent model. In: *Cognitive Systems Research*. **12** (2), S. 175-185. ISSN 1389-0417
- HOWES, Andrew und Richard M. YOUNG (1997). The Role of Cognitive Architecture in Modeling the User: Soar's Learning Mechanism. In: *Human-Computer Interaction*. **12** (4), S. 311-343. ISSN 1044-7318
- HOYOS, C. (1990). Menschliches Handeln in technischen Systemen. In: C. HOYOS und B. ZIMOLONG (Hrsg.) *Ingenieurpsychologie. Enzyklopädie der Psychologie. Themenbereich D, Serie III*. Band 2. Göttingen: Hogrefe, S. 1-32. ISBN 978-3-80171-508-3
- HUANG, Hui-Min (2004): *Autonomy Levels for Unmanned Systems (ALFUS) Framework – Volume I: Terminology*. (NIST Special Publication 1011-I-2.0). Gaithersburg (MD): National Institute of Standards and Technology (NIST)
- HUANG, Hui-Min, Kerry PAVEK, Brian NOVAK, James ALBUS und Elena MESSINA (2005). A Framework for Autonomy Levels For Unmanned Systems (ALFUS). In: *AUVSI's Unmanned Systems North America 2005 Proceedings*. Baltimore (MD), 28. – 30. Juni. Arlington (VA): Association for Unmanned Vehicle Systems International (AUVSI)
- HUSSY, Walter, Margrit SCHREIER und Gerald ECHTERHOFF (2013). *Forschungsmethoden in Psychologie und Sozialwissenschaften für Bachelor*. Berlin, Heidelberg: Springer. ISBN 978-3-642-34362-9
- HWANG, Yong K. und Narendra AHUJA (1992). Gross Motion Planning – A Survey. In: *ACM Computing Surveys (CSUR)*. **24** (3), S. 219-291. ISSN 0360-0300
- IBM CORP. (2015a). *IBM CPLEX Optimizer – United States* [online]. *CPLEX Optimizer – High-Performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming*. Armonk (NY): International Business Machines Corp. [Zugriff am: 13.11.2014]. Verfügbar unter: <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>
- IBM CORP. (2015b). *IBM SPSS Software* [online]. *SPSS software – Predictive analytics software and solutions*. Armonk (NY): International Business Machines Corp. [Zugriff am: 24.03.2015]. Verfügbar unter: <http://www-01.ibm.com/software/analytics/spss/>
- ISRAEL AEROSPACE INDUSTRIES (2014). *Unmanned Air Systems – Heron Family* [online]. Lod: Israel Aerospace Industries (IAI) [Zugriff am: 15.11.2014]. Verfügbar unter: <http://www.iai.co.il/2013/18900-16382-en/IAI.aspx>
- JOHANNSEN, Gunnar (1993). *Mensch-Maschine-Systeme*. Berlin, Heidelberg: Springer. ISBN 987-3-54056-152-1

- JOHANNSEN, Gunnar (1994). Leitende Kontrolle und integrierte Automation in Mensch-Maschine-Systemen. In: K.-P. GÄRTNER, W. STEIN und H. WIDDEL (Hrsg.) *Mensch-Maschine Systeme und Neue Informationstechnologien*. Aachen: Augustinus Buchhandlung, S. 245-251. ISBN 978-3-86073-335-6
- JOHNSON, Matthew, Jeffrey M. BRADSHAW, Paul J. FELTOVICH, Robert R. HOFFMAN, Catholijn M. JONKER, Birna VAN RIEMSDIJK und Maarten SIERHUIS (2011a). Beyond Cooperative Robotics: The Central Role of Interdependence in Coactive Design. In: *IEEE Intelligent Systems*. **26** (3), S. 81-88. ISSN 1541-1672
- JOHNSON, Matthew, Jeffrey M. BRADSHAW, Paul J. FELTOVICH, Catholijn M. JONKER, Birna VAN RIEMSDIJK und Maarten SIERHUIS (2011b). The Fundamental Principle of Coactive Design: Interdependence Must Shape Autonomy. In: M. de VOS, N. FORNARA, J. V. PITT und G. VOUIROS (Hrsg.) *Coordination, Institutions, and Norms in Agent Systems VI*. Berlin, Heidelberg: Springer, S. 172-191. ISBN 978-3-64221-687-3
- JONES, Randolph M., John E. LAIRD, Paul E. NIELSON, Karen J. COULTER, Patrick KENNY und Frank V. KOSS (1999). Automated Intelligent Pilots for Combat Flight Simulation. In: *AI Magazine*. **20** (1), S. 27-42. ISSN 0738-4602
- JONSSON, Jon E. und Wendell R. RICKS (1995). *Cognitive Models of Pilot Categorization and Prioritization of Flight-Deck Information*. (NASA Technical Paper 3528). Hampton (VA): NASA-Langley Research Center
- JOSLIN, Robert (2015). Synthesis of Unmanned Aircraft Systems Safety Reports. In: *Journal of Aviation Technology and Engineering (JATE)*. **5** (1), S. 2-6. ISSN 2159-6670
- JUST, Marcel A. und Patricia A. CARPENTER (1980). A theory of reading: From eye fixations to comprehension. In: *Psychological Review*. **87** (4), S. 329-354. ISSN 0033-295X
- KARIM, Samin, Clint HEINZE und Shane DUNN (2004). Agent-based mission management for a UAV. In: *Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference*. Melbourne, 14. – 17. Dezember. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S.481-486
- KAUFMANN, Herbert und Heimo STEFFEN (2012). *Strabismus*. 4. Auflage. Stuttgart: Thieme. ISBN 978-3-13129-724-2
- KIERAS, David E. und David E. MEYER (1997). An Overview of the EPIC Architecture for Cognition and Performance with Application to Human-Computer Interaction. In: *Human-Computer Interaction*. **12** (4), S. 391-438. ISSN 1044-7318
- KIERAS, David E. und David E. MEYER (2000). The Role of Cognitive Task Analysis in the Application of Predictive Models of Human Performance. In: J. M. SCHRAAGEN, S. F. CHIPMAN und V. L. SHALIN (Hrsg.) *Cognitive Task Analysis*. Mahwah (NJ): Lawrence Erlbaum Associates, S. 237-260. ISBN 978-0-80583-383-6
- KIM, Jong-Hyuk, Salah SUKKARIEH und Stuart WISHART (2006). Real-time navigation, guidance and control of a UAV using low-cost sensors. In: P. CORKE und S. SUKKARIAH (Hrsg.) *Field and service Robotics – Results of the 5th International Conference*. Berlin, Heidelberg: Springer, S. 299-309. Springer Tracts in Advanced Robotics. Volume 25. ISBN 978-3-54033-452-1

- KLEIN, Gary (2008). Naturalistic Decision Making. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society*. **50** (3), S. 456-460. ISSN 0018-7208
- KLEIN, Gary, David D. WOODS, Jeffrey M. BRADSHAW, Robert R. HOFFMAN und Paul FELTOVICH (2004). Ten Challenges for Making Automation a “Team Player” in Joint Human-Agent Activity. In *IEEE Intelligent Systems*. **19** (6), S. 91-95. ISSN 1541-1672
- KRIEGEL, Michael (2011). *Wissensbasierte Konfiguration eines unbemannten Fluggerätes als Architekturansatz zur kognitiven Flugführung* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- KUFFNER Jr., James J. und Steven M. LAVALLE (2000). RRT-connect: An efficient approach to single-query path planning. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA 2000)*. San Francisco (CA), 24. – 28. April. Piscataway (NJ): IEEE Robotics and Automation Society, S.995-1001
- LAIRD, John E., Allen NEWELL und Paul S. ROSENBLOOM (1987). SOAR: An Architecture for General Intelligence. In: *Artificial Intelligence*. **33** (1), S. 1-64. ISSN 0004-3702
- LAMONTE, Jon (2009). The Future of UAVs: Concepts and Considerations. In: *RUSI Air Power Conference 2009*. London, 08. – 09. Juli. London: Royal United Services Institute (RUSI)
- LANC, Otto (1982). *Ergonomie. Psychologie der technischen Welt*. Stuttgart: Kohlhammer. ISBN 978-3-17001-673-6
- LANDRY, Steven J. (2012). Human Factors and Ergonomics in Aviation. In: G. SALVENDY (Hrsg.) *Handbook of Human Factors and Ergonomics*. 4. Auflage. Hoboken (NJ): John Wiley & Sons, S. 1667-1688. ISBN 978-0-47052-838-9
- LANGLEY, Pat und Dongkyu CHOI (2006). A Cognitive Architecture for Physical Agents. In: *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*. Boston (MA), 16. – 20. Juli. Palo Alto (CA): Association for the Advancement of Artificial Intelligence (AAAI), S. 1469 - 1474. ISBN 978-1-57735-281-5
- LAUX, Helmut, Robert M. GILLENKIRCH und Heike Y. SCHENK-MATHES (2012). *Entscheidungstheorie*. 8. Auflage. Berlin: Springer Vieweg. ISBN 978-3-64223-510-8
- LAVALLE, Steven M. (2006). *Planning Algorithms*. Cambridge (MA): Cambridge University Press. ISBN 0-521-86205-1
- LEANA, Carrie R. (1986). Predictors and Consequences of Delegation. In: *The Academy of Management Journal*. **29** (4), S. 754-774. ISSN 0001-4273
- LEE, John D. und Neville MORAY (1992). Trust, control strategies and allocation of function in human-machine systems. In: *Ergonomics*. **35** (10), S. 1243-1270. ISSN 0014-0139
- LEE, John D. und Katrina A. SEE (2004). Trust in Automation: Designing for Appropriate Reliance. In *Human Factors: The Journal of the Human Factors and Ergonomics Society*. **46** (1), S. 50-80. ISSN 0018-7208
- LEE, John D. und Bobbie D. SEPPELT (2009). Human Factors in Automation Design. In: *Springer Handbook of Automation*. Berlin, Heidelberg: Springer, S. 417-436. ISBN 978-3-54078-830-0

- LEES, M. N. und John D. LEE (2007). The influence of distraction and driving context on driver response to imperfect collision warning systems. In: *Ergonomics*. **50** (8), S. 1264-1286. ISSN 0014-0139
- LEHMAN, Jill F., John LAIRD und Paul S. ROSENBLOOM (2006). *A Gentle Introduction to SOAR: An Architecture for Human Cognition: 2006 Update* [online]. [Zugriff am: 12.02.2015]. PDF e-Book. Verfügbar unter:
<http://alpha.tmit.bme.hu/speech/docs/education/GentleIntroduction-2006.pdf>
- LEttl, Benjamin und Axel SCHULTE (2013). Self-explanation capability for cognitive agents on-board of UCAVs to improve cooperation in a manned-unmanned fighter team. In: *AIAA Infotech@Aerospace Conference 2013*. Boston (MA), 19. – 22. August. Reston (VA): American Institute for Aeronautics and Astronautics (AIAA), S.1047-1057. ISBN 978-1-62993-152-4
- LEWANDOWSKY, Stephan, Michael MUNDY und Gerard P. A. TAN (2000). The dynamics of trust: Comparing humans to automation. In: *Journal of Experimental Psychology: Applied*. **6** (2), S. 104-123. ISSN 1076-898X
- LUFTVG (2013): „Luftverkehrsgesetz vom 1. August 1922 (RGBl. 1922 I S. 681), das zuletzt durch Artikel 2 Absatz 175 des Gesetzes vom 7. August 2013 (BGBl. I S. 3154) geändert worden ist“
- LUFTVO (2012): „Luftverkehrs-Ordnung vom 10. August 1963 (BGBl. I S. 652), die zuletzt durch Artikel 3 des Gesetzes vom 8. Mai 2012 (BGBl. I S. 1032) geändert worden ist“
- LUNZE, Jan (2010). *Künstliche Intelligenz für Ingenieure*. München: Oldenbourg Wissenschaftsverlag. ISBN 978-3-48670-222-4
- MAES, Pattie (1993). Behavior-Based Artificial Intelligence. In: *Proceedings of the 15th Annual Conference of the Cognitive Science Society*. Bolder (CO), 18. – 21. Juni. Austin (TX): Cognitive Science Society, S. 74-83. ISBN 080-5-81487-6
- EL MAFKOUK, Rachid, Jean-Francois GABARD und Catherine TESSIER (2006). A Supervision Architecture to Deal with Disruptive Events in UAV Missions. In: *DX'06 17th International Workshop on Principles of Diagnosis*. Burgos, 26. – 28. Juni. S. 163-169
- MARSH, Luke, G. CALBERT, J. TU, D. GOSSINK und H. KWOK (2005). Multi-Agent UAV Path Planning. In: A. ZERGER und R. M. ARGENT (Hrsg.) *MODSIM 2005 International Congress on Modelling and Simulation*. Melbourne (Victoria), 12. – 15. Dezember. Modelling and Simulation Society of Australia and New Zealand, S. 2188-2194. ISBN 097-5-84002-9
- MATTHEWS, Stephen (1980). The gentle art of delegation. In: *Accountancy*. **91**, S. 104-106, 122, 124
- MATZNER, Alexander (2010). *Weiterentwicklung einer Kognitiven Systemarchitektur auf Basis von Graphtransformationen* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- MCCARLEY, Jason S. und Christopher D. WICKENS (2004). *Human Factors Concerns in UAV Flight*. (AHFD-05-05/FAA-05-1). Champaign (IL): University of Illinois - Human Factors Division

- MCMILLAN, G. R., D. BEEVIS, W. STEIN, M. H. STRUB, E. Salas, R. Sutton und K. C. Reynolds (1991). *A Directory of Human Performance Models for System Design*. (AC/243(Panel 8)TR/1). Brüssel: NATO Defence Research Group
- MEITINGER, Claudia (2008). *Kognitive Automation zur Kooperativen UAF-Flugführung* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- MIALL, R. C. und J. K. JACKSON (2006). Adaptation to visual feedback delays in manual tracking: Evidence against the Smith Predictor model of human visually guided action. In: *Experimental Brain Research*. **172** (1), S. 77-84. ISSN 0014-4819
- MICROSOFT (2015). *Microsoft Webcams: LifeCam Products* [online]. Redmond (WA): Microsoft [Zugriff am: 17.04.2015]. Verfügbar unter: <http://www.microsoft.com/hardware/en-us/webcams>
- MILEWSKI, Allen E. und Steven H. LEWIS (1997). Delegating to Software Agents. In: *International Journal of Human-Computer Studies*. **46** (4), S. 485-500. ISSN 1071-5819
- MILLER, Christopher A. (2002). Definitions and dimensions of etiquette. In: *AAAI 2002 Fall Symposium on Etiquette for Human-Computer Work*. North Falmouth (MA), 15. – 17. November. Palo Alto (CA): Association for the Advancement of Artificial Intelligence (AAAI)
- MILLER, Christopher A. (2005). Delegation Architectures: Playbooks and Policy for Keeping Operators in Charge. In: *ICAPS 2005 Workshop on Mixed-Initiative Planning and Scheduling (MIPAS)*. Monterey (CA), 07. Juni. Palo Alto (CA): Association for the Advancement of Artificial Intelligence (AAAI)
- MILLER, Christopher A. (2014a). Delegation and Transparency: Coordinating Interactions So Information Exchange Is No Surprise. In: *Virtual Augmented and Mixed Reality. Designing and Developing Virtual and Augmented Environments*. 8525. Heidelberg, Berlin: Springer, S. 191-202. Lecture Notes in Computer Science. ISBN 978-3-31907-457-3
- MILLER, Christopher A. (2014b). Trust and Delegation: Achieving Robust Interactions with Subordinates. In: *2014 AAAI Spring Symposium Series*. Palo Alto (CA), 24. – 26. März. Palo Alto (CA): Association for the Advancement of Artificial Intelligence (AAAI)
- MILLER, Christopher A., Harry FUNK, Peggy WU, Robert GOLDMAN, Jon MEISNER und Marc CHAPMAN (2005). The Playbook Approach to Adaptive Automation. In: *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting*. Orlando (FL), 26. – 30. September. Human Factors and Ergonomics Society (HFES), S. 15-19. ISSN 1071-1813
- MILLER, Christopher A. und Raja PARASURAMAN (2003): Beyond Levels of Automation: An Architecture for More Flexible Human-Automation Collaboration. In: *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting*. Denver (CO), 13. – 17. Oktober. Santa Monica (CA): Human Factors and Ergonomics Society (HFES), S. 182-186. ISSN 1071-1813
- MILLER, Christopher A. und Raja PARASURAMAN (2007). Designing for Flexible Interaction Between Humans and Automation: Delegation Interfaces for Supervisory Control. In: *Human Factors*. **49** (1), S. 57-75. ISSN 0018-7208

- MYERS, Karen L. und David N. MORLEY (2001). Human Directability of Agents. In: *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001)*. Victoria (BC), 21. – 23. Oktober. Aarhus C.: Interactive Design Foundation, S. 108-115. ISBN 158-1-13380-4
- NAKASHIMA, Hideyuki und Itsuki NODA (1998). Dynamic Subsumption Architecture for Programming Intelligent Agents. In: *Proceedings of the International Conference on Multi Agent Systems*. Paris, 3. – 7. Juli. Washington (DC): IEEE Computer Society, S. 190-197. ISBN 081-8-68500-X
- NASH, Alex, Kenny DANIEL, Sven KOENIG und Ariel FELNER (2007). Theta*: Any-Angle Path Planning on Grids. In: *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*. Vancouver (BC), 22. – 26. Juli. Palo Alto (CA): Association for the Advancement of Artificial Intelligence (AAAI), S. 1177-1183. ISBN 978-1-57735-323-2
- NERDINGER, Friedemann W., Gerhard BLICKLE und Niclas SCHAPER (2011). *Arbeits- und Organisationspsychologie*. 2. Auflage. Berlin, Heidelberg: Springer Medizin. ISBN 978-3-64216-971-7
- NEWELL, Allen (1990). *Unified Theories of Cognition*. Cambridge (MA): Harvard University Press. ISBN 978-0-67192-101-6
- NONAMI, Kenzo, Farid KENDOUL, Satoshi SUZUKI, Wei WANG und Daisuke NAKAZAWA (2010). *Autonomous Flying Robots, Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Heidelberg, Berlin: Springer. ISBN 978-4-43154-687-0
- NORMAN, Donald A. (1986). Cognitive Engineering. In: D. A. NORMAN und S. W. DRAPER (Hrsg.) *User-Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale (NJ): Lawrence Erlbaum Associates, S. 31-61. ISBN 978-0-89859-872-8
- NORTHROP GRUMMAN CORP. (2014). *Global Hawk* [online]. Falls Church (VA): Northrop Grumman Corp. [Zugriff am: 12.11.2014]. Verfügbar unter: <http://www.northropgrumman.com/globalhawk/>
- NUI, Michael C. Y. (2011). *Airframe Structural Design: Practical Design Information and data on Aircraft Structures*. 2nd Edition. Northridge (CA): Adaso/Adatra Engineering Center. ISBN 978-9-62712-809-0
- OFFICE OF THE SECRETARY OF DEFENSE (2005). *Unmanned Aircraft Systems (UAS) Roadmap: 2005-2013*. Washington (DC): Office of the Secretary of Defense
- ONCU, Mehmet (2014). *An Analysis of Human Causal Factors in Unmanned Aerial Vehicle (UAV) Accidents* [Dissertation]. Monterey (CA): Naval Postgraduate School
- ONG, M., F. J. BRYAN, M. K. HOOK, J. C. BOSSARD, R. TODD, A. J. FORSYTH und H. THOMPSON (2012). Intelligent Agent Based Power Management for UAV Systems. In: *6th IET International Conference on Power Electronics, Machines and Drives (PEMD 2012)*. Bristol, 27. – 29. März. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 1-6. ISBN 978-1-84919-616-1
- ONKEN, Reiner und Axel SCHULTE (2010): *System-Ergonomic Design of Cognitive Automation: Dual-Mode Cognitive Design of Vehicle Guidance and Control Work Systems*. Heidelberg, Berlin: Springer. Studies in Computational Intelligence. ISBN 978-3-64203-134-2

- OPPERMANN, Reinhard, Hrsg. (1994). *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*. Hillsdale (NJ): Lawrence Erlbaum Associates. ISBN 080-5-81655-0
- OUMA, Joseph A., Wayne L. CHAPPELLE und Amber SALINA (2011). *Facets of Occupational Burnout Among U.S. Air Force Active Duty and National Guard/Reserve MQ-1 Predator and MQ-9 Reaper Operators*. Wright-Patterson AFB (OH): Air Force Research Laboratory, 711th Human Performance Wing
- PARASURAMAN, Raja (2000). Designing Automation for Human Use: Empirical Studies and Quantitative Models. In: *Ergonomics*. **43** (7), S. 931-951. ISSN 0014-0139
- PARASURAMAN, Raja, Touflik BAHRI, John E. DEATON, Jeffrey G. MORRISON und Michael BARNES (1992). *Theory and Design of Adaptive Automation in Aviation Systems*. Warminster (PA): Air Vehicle and Crew Systems Technology Department, Naval Air Warfare Center
- PARASURAMAN, Raja und Christopher A. MILLER (2004). Trust and Etiquette in High-Criticality Automated Systems. In: *Communications of the ACM*. **47** (4), S. 51-55. ISSN 0001-0782
- PARASURAMAN, Raja und Victor RILEY (1997). Humans and Automation: Use, Misuse, Disuse, Abuse. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society*. **39** (29), S. 230-253. ISSN 0018-7208
- PARASURAMAN, Raja, Thomas B. SHERIDAN und Christopher D. WICKENS (2000). A Model for Types and Levels of Human Interaction with Automation. In: *IEEE Transactions on Systems, Man, and Cybernetics*. **30** (3), S. 286-297. ISSN 2168-2216
- PARSONS, H. McIlvaine (1995). Automation and the Individual: Comprehensive and Comparative Views. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society*. **27** (1), S. 99-111. ISSN 0018-7208
- PASTOR, E., J. Lopez und P. ROYO (2007). UAV Payload and Mission Control Hardware/Software Architecture. In: *IEEE Aerospace and Electronic Systems Magazine*. **22** (6), S. 3-8. ISSN 0885-8985
- PECHER, Wolfgang, Stefan BRÜGGENWIRTH und Axel SCHULTE (2010). Using Cognitive Automation for Aircraft General Systems Management. In: *5th International Conference on System of Systems Engineering (SoSE)*. Loughborough, 22. – 24. Juni. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 179-186. ISBN 978-1-42448-197-2
- PETERMANN, Thomas und Reinhard GRÜNWARD (2011). *Stand und Perspektiven der militärischen Nutzung unbemannter Systeme*. Arbeitsbericht Nr. 144. Berlin: Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag (TAB)
- PIAGET, Jean (2000). *Psychologie der Intelligenz*. 10. Auflage. Stuttgart: Klett-Cotta. ISBN 978-3-60894-014-5
- PRICE, Harold E. (1985). The Allocation of Functions in Systems. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society*. **27** (1), S. 33-45. ISSN 0018-7208

- PUTZER, Henrik J. (2004). *Ein uniformer Architekturansatz für Kognitive Systeme und seine Umsetzung in ein operatives Framework* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- RAO, Anand S. und Michael P. GEORGEFF (1995). BDI Agents: From Theory to Practice. In: *Proceedings of the First International Conference on Multiagent Systems (ICMAS-95)*. San Francisco (CA), 12. – 14. Juni. Cambridge (MA): MIT Press, S. 312-319. ISBN 026-2-62102-9
- RASMUSSEN, Jens (1983). Skills, Rules, and Knowledge; Signals, Signs and Symbols, and other Distinctions in Human Performance Models. In: *IEEE Transactions on Systems, Man and Cybernetics*. **SMC-13** (3), S. 257-266. ISSN 0018-9472
- RASMUSSEN, Jens, Annelise M. PEJTERSEN und L. P. GOODSTEIN (1994): *Cognitive Systems Engineering*. New York (NY): John Wiley & Sons. ISBN 047-1-01198-3
- RAUSCHERT, Andreas (2013). *Kognitives Assistenzsystem zur Führung unbemannter Luftfahrzeuge in bemannt-unbemannten Flugmissionen* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- REICHEL, Simon (2013). Künftige Beschaffungsplanung von UAS in der Luftwaffe. In: *DGLR Symposium/Workshop „UAV Autonomie“ - Automatisierung unbemannter Luftfahrzeuge* [PowerPoint Präsentation]. Ottobrunn, 19. – 21. März. Bonn: Deutsche Gesellschaft für Luft- und Raumfahrt
- RILEY, Victor (1996). What Avionics Engineers Should Know About Pilots and Automation. In: *IEEE Aerospace and Electronic Systems Magazine*. **11** (5), S. 3-8. ISSN 0018-9251
- ROUSE, William B. (1988). Adaptive Aiding for Human/Computer Control. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society*. **30** (4), S. 431-443. ISSN 0018-7208
- RUBIO, Susana, Eva DÍAZ, Jesús MARTÍN und José M. PUENTE (2004). Evaluation of Subjective Mental Workload: A Comparison of SWAT, NASA-TLX, and Workload Profile Methods. In: *Applied Psychology: An International Review*. **53** (1), S. 61-86. ISSN 0269-994X
- RUDNICK, Georg, Sebastian CLAUB und Axel SCHULTE (2014). Flight Testing of Agent Supervisory Control on Heterogeneous Unmanned Aerial System Platforms. In: *IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC 2014)*. Colorado Springs (CO), 05. – 09. Oktober. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 795-804. ISBN 978-1-47995-000-3
- RUSS, Martin, Marc SCHMITT, Christian HELLERT und Peter STÜTZ (2013). Airborne Sensor and Perception Management: Experiments and Results for Surveillance UAS. In: *AIAA Infotech@Aerospace Conference 2013*. Boston (MA), 19. – 22. August. Reston, VA: American Institute for Aeronautics and Astronautics (AIAA), S. 1506-1521. ISBN 978-1-62993-152-4
- RUSSEL, S. J. und Peter NORVIG (2012). *Künstliche Intelligenz: Ein moderner Ansatz*. Deutsche Übersetzung der 3. Aktualisierten Auflage von F. Langenau. München: Pearson, Higher Education. ISBN 978-3-86894-098-5

- SACERDOTI, Earl D. (1974). Planning in a Hierarchy of Abstraction Spaces. In: *Artificial Intelligence*. **5** (2), S. 115-135. ISSN 0004-3702
- SCERBO, S. (2001). Adaptive Automation. In: W. KARWOWSKI (Hrsg.) *International Encyclopedia of Ergonomics and Human Factors*. 3rd Edition. London: Taylor Francis, S. 1077-1079. ISBN 074-8-40847-9
- SCHAEFER, R. (2003). *Unmanned Aerial Vehicle Reliability Study*. Washington (DC): Office of the Secretary of Defense
- SCHIPANI, Salvatore P. (2003). An Evaluation of Operator Workload, During Partially-Autonomous Vehicle Operations. In: Proceedings of the Performance Metrics for Intelligent Systems Conference (PerMIS '03). Gaithersburg (MD), 16. – 18. September. New York (NY): IEEE Control Systems Society
- SCHNEEMILCH, Stefan (2013). *Aufbau und Integration eines UAV-Demonstrators für den Einsatz eines kognitiven Agenten als Mission-Management System* [Masterarbeit]. Neubiberg: Universität der Bundeswehr München
- SCHNEEWEIß, Christoph A. (1994). Elemente einer Theorie hierarchischer Planung. In: *Operations-Research-Spektrum*. **16** (2), S. 161-168. ISSN 0171-6468
- SCHRIESHEIM, Chester A., Linda L. NEIDER und Terri A. SCANDURA (1998). Delegation and Leader-Member-Exchange: Main Effects, Moderators, and Measurement Issues. In: *The Academy of Management Journal*. **41** (3), S. 298-318. ISSN 0001-4273
- SCHULTE, Axel (2014). *Flugführung & Automation* [Vorlesungsskript]. Neubiberg: Universität der Bundeswehr München
- SEEING MACHINES (2014). *Seeing Machines | Safety in Motion* [online]. Canberra: Seeing Machines [Zugriff am: 14.03.2015]. Verfügbar unter: <http://www.seeingmachines.com/>
- SHERIDAN, Thomas B. (1988). Task Allocation and Supervisory Control. In: M. G. HELANDER (Hrsg.) *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier Science, S. 159-173. ISBN 044-4-81862-6
- SHERIDAN, Thomas B. (1992). *Telerobotics, automation, and human supervisory control*. Cambridge (MA): The MIT Press. ISBN 978-0-26251-547-4
- SHERIDAN, Thomas B. (2000). Function allocation: Algorithm, alchemy or apostasy?. In: *International Journal of Human-Computer Studies*. **25** (2), S. 203-216. ISSN 1071-5819
- SHERIDAN, Thomas B. (2002). *Humans and Automation: System Design and Research Issues*. New York (NY): Wiley & Sons. ISBN 047-1-23428-1
- SHERIDAN, Thomas B. (2011). Adaptive Automation, Level of Automation, Allocation Authority, Supervisory Control, and Adaptive Control: Distinctions and Modes of Adaptation. In: *IEEE Transactions on Systems, Man, and Cybernetics*. **41** (4), S. 662-667. ISSN 2168-2216
- SHERIDAN, Thomas B. (2012). Human Supervisory Control. In G. SALVENDY (Hrsg.) *Handbook of Human Factors and Ergonomics*. 4. Auflage. Hoboken (NJ): John Wiley & Sons, S. 990-1015. ISBN 978-0-47052-838-9
- SHERIDAN, Thomas B. und Raja PARASURAMAN (2005). Human-Automation Interaction. In: *Reviews of Human Factors and Ergonomics*. **1** (1), S. 89-129. ISSN 1557-234X

- SHERIDAN, Thomas B. und William L. VERPLANK (1978). Human and Computer Control of Undersea Teleoperators. In: *14th Annual Conference on Manual Control*. Los Angeles (CA), 25. – 27. April. Moffett Field (CA): NASA-Ames Research Center, S. 343-357
- SMIRNOV, Denis und Peter STÜTZ (2014). Knowledge elicitation and representation for module based perceptual capabilities onboard UAVs. In: *52nd Aerospace Sciences Meeting*. National Harbor (MD), 13. – 17. Januar. Reston (VA): American Institute of Aeronautics and Astronautics (AIAA), S. 12297-12306. ISBN 978-1-63266-928-5
- SMITH, Kip und Peter A. HANCOCK (1995). Situation awareness is adaptive, externally-directed consciousness. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society*. **37** (1), S. 137-148. ISSN 0018-7208
- SPAIN, Randall D. und Poornima MADHAVAN (2009). The Role of Automation Etiquette and Pedigree in Trust and Dependence. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. **53** (4). San Antonio (TX), 19. – 23. Oktober. Santa Monica (CA): Human Factors and Ergonomics Society (HFES), S. 339-343. ISSN 1541-9312
- SPREAD CONCEPTS LLC (2014). *The Spread Toolkit* [online]. Savage (MD): Spread Concepts LLC [Zugriff am: 12.08.2014]. Verfügbar unter: <http://www.spread.org/>
- SPRINGER, Paul J. (2013). *Military Robots and Drones: A Reference Handbook*. Santa Barbara (CA): ABC-CLIO. Contemporary World Issues. ISBN 978-1-59884-732-1
- STEELS, Luc (1995). When are robots intelligent autonomous agents? In: *Robotics and Autonomous Systems*. **15** (1-2), S. 3-9. ISSN 0921-8890
- STIEGLITZ, Klaus-Peter (2007). *The Luftwaffe Perspective on UAVs*. In: *RUSI Defense Systems*. **9** (3), S. 34-36.
- SULLIVAN, Don, Joseph TOTAH, Steve WEGENER, Francis ENOMOTO, Chad FROST, John KANESHIGE und Jeremy FRANK (2004). Intelligent Mission Management for Uninhabited Aerial Vehicles. In: *Remote Sensing Applications of the Global Positioning System (SPIE 5661)*. Honolulu (HI), 08. November. Bellingham (WA): The International Society for Optical Engineering, S. 121-131
- TAATGEN, Niels, Christian LEBIERE und John ANDERSON (2006). Modeling Paradigms in ACT-R. In: R. SUN (Hrsg.) *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*. Cambridge (MA): Cambridge University Press, S. 29-52. ISBN 978-052183-964-8
- TANTRAIRATN, Suradet und Sandor M. VERES (2015). A Rational Agent Framework for Adaptive Flight Control of UAVs. In: *Proceedings of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS 2015)*. Denver (CO), 9. – 12. Juni. New York (NY): Institute of Electrical and Electronics Engineers (IEEE), S. 147-156. ISBN 978-1-47996-009-5
- THEIBING, Nikolaus, Gerrit KAHN und Axel SCHULTE (2012). Cognitive Automation Based Guidance and Operator Assistance for Semi-Autonomous Mission Accomplishment of the UAV Demonstrator Sagitta. In: *Deutscher Luft- und Raumfahrtkongress 2012*. Berlin, 10. – 12. September. Bonn: Deutsche Gesellschaft für Luft- und Raumfahrt

- THEIBING, Nikolaus und Axel SCHULTE (2013). Intent-Based UAV Mission Management Using an Adaptive Mixed-Initiative Operator Assistant System. In: *AIAA Infotech@Aerospace Conference 2013*. Boston (MA), 19. – 22. August. Reston (VA): American Institute for Aeronautics and Astronautics (AIAA), S. 674-690. ISBN 978-1-62993-152-4
- THEIBING, Nikolaus und Axel SCHULTE (2014). A UAS Mission Assistant System Employing a Theory-of-Mind about the Human Operator. In: *International Conference on Human-Computer Interaction in Aerospace (HCI-Aero 2014)*. Santa Clara (CA), 30. Juli – 01. August
- TIMPE, Klaus-Peter, Thomas JÜRGENSOHN und Harald KOLREP (2002). *Mensch-Maschine-Systemtechnik: Konzepte, Modellierung, Gestaltung, Evaluation*. 2. Auflage. Düsseldorf: Symposium. ISBN 978-3-93381-420-3
- TURING, Alan M. (1950). Computing Machinery and Intelligence. In: *Mind: A Quarterly Review of Philosophy*. **59** (236), S. 433-460. ISSN 0026-4423
- TURING, Alan M. (1967). Kann eine Maschine denken?. In: H. M. ENZENSBERGER (Hrsg.) *Kursbuch 8: Neue Mathematik/Grundlagenforschung/Theorie der Automaten*. Deutsche Übersetzung von P. Gänßler. Frankfurt a. M.: Suhrkamp, S. 106-137
- U.S. DEPARTMENT OF TRANSPORTATION (2014). *Unmanned Aircraft System (UAS) Service Demand 2015-2035: Literature Review and Projections of Future Usage, Version 1.0*. (DOT-VNTSC-DoD-13-01). Cambridge (MA): U.S. Department of Transportation
- UBIQUITI NETWORKS, INC. (2015). *Ubiquiti Networks - Wireless networking products for broadband and enterprise* [online]. San Jose (CA): Ubiquiti Networks, Inc. [Zugriff am: 17.04.2015]. Verfügbar unter: <https://www.ubnt.com/>
- UHRMANN, Johann (2013). *Auftragsbasierte Multi-UAV-Führung aus dem Helikoptercockpit durch kognitive Automation* [Dissertation]. Neubiberg: Universität der Bundeswehr München
- UHRMANN, Johann und Axel SCHULTE (2011). Task-based Guidance of Multiple UAV Using Cognitive Automation. In: *COGNITIVE 2011: The Third International Conference on Advanced Cognitive Technologies and Applications*. Nizza, 22. – 27. Juli. Wilmington (DE): International Academy, Research, and Industry Association, S. 47-52. ISBN 978-1-61208-155-7
- VACHTSEVANOS, George, Liang TANG, Graham DROZESKI und Luis GUTIERREZ (2004). From mission planning to flight control of unmanned aerial vehicles: Strategies and implementation tools. In: *Annual Reviews in Control*. **29** (1), S. 101-115. ISSN 1367-5788
- VALAVANIS, KIMON P. (Hrsg.) (2007). *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*. Dordrecht: Springer. Intelligent Systems, Control and Automation: Science and Engineering. ISBN 978-1-40206-113-4
- VALENTI, Mario, Tom SCHOUWENAARS, Yoshiaki KUWATA, Eric FERON, Jonathan HOW und James PAUNICKA (2004). Implementation of a Manned Vehicle – UAV Mission System. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Providence (RI), 16. – 19. August. Reston (VA): American Institute of Aeronautics and Astronautics (AIAA), S. 16-19

- VAN BLYENBURGH, Peter (2014). *RPAS Yearbook 2014* [online]. *Remotely Piloted Aircraft Systems: The Global Perspective – Edition 2014/2015*. Paris: Blyenburgh & Co [Zugriff am: 03.10.2015]. Verfügbar unter: <http://www.uvs-info.com/index.php/yearbooks/yearbook-2014/>
- VAN ERP, J. B. F. und L. Van Breda (1999). *Human Factors Issues and Advanced Interface Design in Maritime Unmanned Aerial Vehicles: A Project Review*. (Report TNO TM-99-A004). Amsterdam: TNO
- VERNON, D., G. Metta und G. SANDINI (2007). A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. In: *IEEE Transactions on Evolutionary Computation*. **11** (2), S. 151-179. ISSN 1089-778X
- VINCENT, Patrick und Izhak RUBIN (2004). A Framework and Analysis for Cooperative Search Using UAV Swarms. In: *The 2004 ACM Symposium on Applied Computing (SAC'04)*. Nikosia, 14. – 17. März. New York (NY): ACM, S. 79-86. ISBN 158-1-13812-1
- WEGENER, Ingo (2003). *Komplexitätstheorie – Grenzen der Effizienz von Algorithmen*. Berlin, Heidelberg: Springer. ISBN 978-3-54000-161-4
- WERNER, Josef (2013). *Konzeption und Entwicklung eines Routenplaners unter Verwendung der Potentialfeldmethode* [Bachelorarbeit]. Neubiberg: Universität der Bundeswehr München
- WERNER, Josef (2014a). *Konzeption und Entwicklung eines Programms zur automatisierten Auswahl und Parametrisierung von Algorithmen zur Routenplanung* [Masterprojekt]. Neubiberg: Universität der Bundeswehr München
- WERNER, Josef (2014b). *Konzeptionierung und Durchführung einer Versuchsreihe zur Evaluierung eines kognitiven Agenten in Bezug auf die Überwachung von Aufklärungsmissionen* [Masterarbeit]. Neubiberg: Universität der Bundeswehr München
- WIENER, Earl L. (1988). Cockpit Automation. In: E. L. WIENER, D. C. NAGEL (Hrsg.) *Human Factors in Aviation*. San Diego (CA): Academic Press, S. 433-462. ISBN 012-7-50030-8
- WILKINS, David E. (1985). *Hierarchical Planning: Definition and Implementation*. (Technical Note 370). Menlo Park (CA): SRI International
- WILLIAMS, Kevin W. (2004). *A Summary of Unmanned Aircraft Accident/Incident Data: Human Factors Implications*. (DOT/FAA/AM-04/24). Oklahoma City (OK): FAA Civil Aerospace Medical Institute
- WILSON, Robert A. und Frank C. KEIL, Hrsg. (2001). *The MIT Encyclopedia of the Cognitive Sciences (MITECS)*. Cambridge (MA): The MIT Press. ISBN 026-2-73144-4
- WINOGRAD, Terry und David D. WOODS (1997). Challenges for Human-Centered Design. In: *Human-Centered Systems: Information, Interactivity, and Intelligence*. Arlington (VA), 17. – 19. Februar. Washington (DC): National Science Foundation
- WOHLER, Marcus, Fabian LOY und Axel SCHULTE (2014). Mental Models as Common Ground for Human-Agent Interaction in Cognitive Assistant Systems. In: *International Conference on Human-Computer Interaction in Aerospace (HCI-Aero 2014)*. Santa Clara (CA), 30. Juli – 01. August

- WOHLER, Marcus und Axel SCHULTE (2014). Development of a Cognitive Airborne Assistant System for UAS Integration into Civil Airspace. In: *31st EAAP Conference*. Valletta, 22. – 26. September. Groningen: European Association for Aviation Psychology (EAAP)
- WOHLERS, Klaus und Christian BLOHM (2007). Flugführungskonzept der KZO Drohne: Autonomie und Verantwortlichkeiten [PowerPoint Präsentation]. In: *Workshop der DGLR Fachausschüsse*. Neubiberg, 15. Februar. Bonn: Deutsche Gesellschaft für Luft- und Raumfahrt
- WRAY, Robert E., John E. LAIRD, Andrew NUXOLL, Devvan STOKES und Alex KERFOOT (2004). Synthetic Adversaries for Urban Combat Training. In: *Proceedings of the 17th Innovative Applications of Artificial Intelligence Conference (IAAI-2007)*. San Jose (CA), 27. – 29. Juli. Palo Alto (CA): Association for the Advancement of Artificial Intelligence (AAAI), S. 923-930. ISBN 978-0-26251-183-4

B Abkürzungsverzeichnis

AAL.....	Automation Abstraction Layer
ACT-R.....	Adaptive Control of Thought-Rational
ADAPT.....	Adaptive Dynamics and Active Perception for Thought
ALFUS.....	Autonomy Levels for Unmanned Systems
ASC.....	Agent Supervisory Control
ATOL.....	Automatic Take-Off and Landing
AU.....	Automatische Umplanung
BDI.....	Belief-Desire-Intention
BKS.....	Bodenkontrollstation
CML.....	Cognitive Modelling Language
CoCampus.....	Cooperative Cognitive Automation through Mathematically optimized Path-Following of UAS
COSA.....	Cognitive System Architecture
COSA ²	Cognitive System Architecture with Centralized Ontology and Specific Algorithms
COSY-flight.....	Cognitive System for the Flight Domain
CPL.....	Cognitive Programming Language
DLM.....	Datenlink Management
ECG.....	Electronic Camera Gimbal
EPIC.....	Executive Process Interactive Control
FCS.....	Flight Control System
FLOT.....	Forward Line of Own Troops
FMS.....	Flight Management System
FOB.....	Forward Operating Base, Forward Operating Base
FPV.....	First Person View
HDD.....	Head-Down Display
HMI.....	Human-Machine Interface
HSC.....	Human Supervisory Control
HUD.....	Head-Up Display
I/O Interface.....	Input / Output Interface
IFS.....	Institut für Flugsysteme
IPC.....	Interprocess Communication
KCV.....	Knowledge Configured Vehicle
KI.....	Künstliche Intelligenz
KP.....	Kognitiver Prozess
LFZ.....	Luftfahrzeug
LoA.....	Levels of Automation
M.....	Mittelwert
MABA-MABA.....	Men Are Better At - Machines Are Better At
MALE.....	Medium Altitude Long Endurance
mBKS.....	mobile Bodenkontrollstation

MEA	More Electric Aircraft
MInt	Manuelle Intervention
MMC	Mission Management Computer
MUM-T	Manned-Unmanned Teaming
NASA APEX.....	NASA Advanced Plant Experiments
NASA-TLX.....	NASA-Task Load Index
OCU	Operating Cognitive Unit
OF.....	Operating Force
OODA	Observe, Orient, Decide, Act
OSM	Operation Supporting Means
p.....	Wahrscheinlichkeit im Sinne eines Signifikanzwertes
PCC	Perception Control Computer
PDDL	Planning Domain Definition Language
POI	Point of Interest
PuK.....	Planungs- und Kommandierungsphase
QOS.....	Quality of Service
ReX.....	Reactive Extension for COSA ²
RPAS.....	Remotely Piloted Aircraft System
RPD	Recognition-Primed Decision
RPM	Routenplaner Management
SAGAT.....	Situation Awareness Global Assessment Technique
SAM	Surface-to-Air-Missile
SCU	Supporting Cognitive Unit
SD.....	Standard Deviation
SIF	Sensorinterface
SOAR	State, Operator and Result
SPMS.....	Sensor- und Perzeptionsmanagement
TCP.....	Transmission Control Protocol
UAS.....	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UniBwM.....	Universität der Bundeswehr München
ZertAP	Zertifizierbarer Autopilot

C Dokumente zur Datenerhebung im Mensch-Maschine Experiment

Im Rahmen der in Abschnitt 5.1 beschriebenen Mensch-Maschine Versuche werden die Wahrnehmung des Verhaltens des kognitiven Agenten sowie Wegbereiter des Probandenverhaltens mittels subjektiver Fragebögen erhoben, die während und im Anschluss an die Datenerhebung durch die Probanden ausgefüllt wurden. Die hierbei eingesetzten Fragebögen werden im Folgenden vorgestellt.

Anhang C.1 zeigt den deutschsprachigen NASA-TLX Fragebogen als Übersetzung des englischen Originals nach [Hart & Staveland 1988]. Der Fragebogen ist in zwei Teile gegliedert und dient der Bewertung der sechs Belastungsdimensionen *Geistige Anforderung*, *Körperliche Anforderung*, *Zeitliche Anforderung*, *Leistung*, *Anstrengung* und *Frustration*. Im ersten Teil wird die subjektive Bedeutung der Dimensionen durch paarweise Gegenüberstellung bewertet. Im zweiten Teil wird die Beanspruchung der einzelnen Dimensionen mit Angaben zwischen null und 20 Punkten bewertet.

Die SAGAT Fragebögen zur Erhebung des Situationsbewusstseins in den Szenarien I und II sind in Anhang C.2 dargestellt. In jedem Szenario wurden dabei jeweils zwei Fragebögen erhoben. Die Fragebögen bestehen jeweils aus acht Fragen mit einer zu erreichenden Gesamtpunktzahl von 18 Punkten. Jedem Fragebogen lag bei der Datenerhebung zusätzlich eine Kartendarstellung des Missionsbereichs gem. Abbildung 5-2 bei, auf der Teile der Antworten von den Probanden eingezeichnet wurden. Die Punktevergabe für die Beantwortung einzelner Fragen entsprach ihrer jeweiligen Komplexität und variierte zwischen ein und drei Punkten.

Anhang C.3 zeigt abschließend den Fragebogen *Subjektive Bewertung*, der zur Bewertung des Verhaltens und der Interaktion der jeweiligen Agentenkonfiguration zum Ende der Datenerhebung eingesetzt wurde. Der Fragebogen umfasst die vier Kategorien *Systeminteraktion*, *Systemverhalten bei Lageänderung*, *Systemseitig bereitgestellte Informationen* und *Gesamtsystem*. In jeder Kategorie wird die Richtigkeit verschiedener Aussagen bezüglich des Agentensystems subjektiv auf einer Skala von *Überhaupt Nicht* bis *Voll und Ganz* zutreffend bewertet. Zwischenwerte sind möglich und ergeben eine Bewertungsskala zwischen einem und sieben Punkten.

C.1 NASA-TLX Fragebogen

Versuchsperson Nr.:

Testphase:

Datum:

NASA-TLX

1. Geben Sie an, welche Beanspruchungsdimension für die Gesamtbeanspruchung, die Sie empfunden haben, bedeutsamer war!

- | | | | |
|-------------------------|--------------------------|--------------------------|-------------------------|
| Geistige Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Körperliche Anforderung |
| Geistige Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Zeitliche Anforderung |
| Geistige Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Leistung |
| Geistige Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Anstrengung |
| Geistige Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Frustration |
| Körperliche Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Zeitliche Anforderung |
| Körperliche Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Leistung |
| Körperliche Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Anstrengung |
| Körperliche Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Frustration |
| Zeitliche Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Leistung |
| Zeitliche Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Anstrengung |
| Zeitliche Anforderung | <input type="checkbox"/> | <input type="checkbox"/> | Frustration |
| Leistung | <input type="checkbox"/> | <input type="checkbox"/> | Anstrengung |
| Leistung | <input type="checkbox"/> | <input type="checkbox"/> | Frustration |
| Anstrengung | <input type="checkbox"/> | <input type="checkbox"/> | Frustration |

Abbildung C-1: Deutschsprachiger NASA-TLX Fragebogen – Seite 1

C.2 SAGAT Fragebogen

Versuchsperson Nr.:	Testphase:	Datum:
---------------------	------------	--------

SAGAT (Mission I, Nr. 1)

1. Zeichnen Sie bitte die letzte Position des UAV in die anliegende Karte ein.
2. Zeichnen Sie bitte die Positionen und den Status (*active, blocked*) der Korridore in die anliegende Karte ein.
3. Zeichnen Sie bitte die Positionen Ihrer Hauptziele für diese Mission in die anliegende Karte ein.
4. Was war Ihr zuletzt aufgeklärtes Teilmissionsziel (Name und ID)?
Welchen Auftrag hatten Sie dort?

5. Warum konnte(n) Ihr(e) vorhergehende(s/n) Teilmissionsziel(e) nicht erfüllt werden?

6. In welcher Flugphase befand sich das UAV, bevor Sie umplanen mussten?

Person Detection*	<input type="checkbox"/>	Recon (Aufklärung)*	<input type="checkbox"/>
Vehicle Detection*	<input type="checkbox"/>	Cross Corridor*	<input type="checkbox"/>
Departure	<input type="checkbox"/>	Transit Flight*	<input type="checkbox"/>
Landing	<input type="checkbox"/>	Holding Pattern	<input type="checkbox"/>

* Ziel angeben:

7. Hat sich die Anzahl der feindlichen SAM-Site Stellungen seit Flugbeginn verändert? Wenn ja, um wie viel?

8. Wird die aktuelle Route bis zum voraussichtlichen Erreichen des nächsten Missionszieles noch einmal umgeplant werden müssen (nach Ihrer jetzigen Informationslage)? Wenn ja, warum?

Abbildung C-3: SAGAT Fragebogen bei manueller Umplanung in Szenario I

Versuchsperson Nr.:

Testphase:

Datum:

SAGAT (Mission I, Nr. 2)

1. Zeichnen Sie bitte die Position des UAV in die anliegende Karte ein.
2. Zeichnen Sie bitte die noch nicht eingezeichneten feindlichen SAM-Site Stellungen in die anliegende Karte ein.
3. Zeichnen Sie bitte die Position der Home-Base in die anliegende Karte ein.
4. Wie viele eigene SAM-Site Stellungen befinden sich auf der Karte? Hat sich deren Anzahl, seitdem Sie Ihren Teilauftrag an der Northern Bay (ID: 101) erfüllt haben, verändert?

-
5. In welcher Flugphase befindet sich das UAV momentan?
- | | | | |
|--------------------|--------------------------|---------------------|--------------------------|
| Person Detection* | <input type="checkbox"/> | Recon (Aufklärung)* | <input type="checkbox"/> |
| Vehicle Detection* | <input type="checkbox"/> | Cross Corridor* | <input type="checkbox"/> |
| Departure | <input type="checkbox"/> | Transit Flight* | <input type="checkbox"/> |
| Landing | <input type="checkbox"/> | Holding Pattern | <input type="checkbox"/> |

* Ziel angeben:

6. Wird die Route bis zum Erreichen der Home-Base noch einmal umgeplant werden müssen(nach Ihrer jetzigen Informationslage)? Wenn ja, warum?
-

7. Werden Sie die Home-Base erreichen können? Falls nicht, warum?
-

8. Wie viele Korridore haben den Status „blocked“?
-

Abbildung C-4: SAGAT Fragebogen bei automatischer Umplanung in Szenario I

Versuchsperson Nr.:

Testphase:

Datum:

SAGAT (Mission II, Nr. 1)

1. Zeichnen Sie bitte die noch nicht eingezeichneten feindlichen SAM-Site Stellungen in die anliegende Karte ein.
2. Zeichnen Sie bitte die aktuelle Position des UAV in die anliegende Karte ein.
3. Zeichnen Sie bitte die Positionen Ihrer Hauptziele für diese Mission in die anliegende Karte ein.
4. Wird die aktuelle Route bis zum voraussichtlichen Erreichen des ersten Missionszieles umgeplant werden müssen (nach Ihrer jetzigen Informationslage)? Wenn ja, warum?

5. Wie viele feindliche SAM-Site Stellungen befinden sich auf der Karte?

6. Wie viele Korridore haben im Moment den Status „blocked“?

7. Kann Ihr aktuelles Teilmissionsziel noch erfüllt werden? Falls nicht, warum?

8. Was ist Ihr aktuelles Teilmissionsziel (Name und ID). Welchen Auftrag haben Sie dort?

Abbildung C-5: SAGAT Fragebogen bei automatischer Umplanung in Szenario II

Versuchsperson Nr.:

Testphase:

Datum:

SAGAT (Mission II, Nr. 2)

1. Zeichnen Sie bitte die noch nicht eingezeichneten feindlichen SAM-Site Stellungen in die anliegende Karte ein.
2. Zeichnen Sie bitte die Position des UAV in die anliegende Karte ein.
3. Zeichnen Sie bitte die eigenen SAM-Site Stellungen in die anliegende Karte ein.
4. Wird die aktuelle Route bis zum voraussichtlichen Erreichen der Home-Base umgeplant werden müssen (nach Ihrer jetzigen Informationslage)? Wenn ja, warum?

-
5. Warum konnte Ihr letztes Teilmissionsziel nicht erfüllt werden?

-
6. Können Sie das optionale Missionsziel noch bearbeiten? Falls nicht, warum?

-
7. Hat sich die FLOT (Forward Line of Own Troops – rote Line) seit Bearbeitung des letzten Teilmissionsziels verschoben?

-
8. In welcher Flugphase befindet sich das UAV momentan?

Person Detection*	<input type="checkbox"/>	Recon (Aufklärung)*	<input type="checkbox"/>
Vehicle Detection*	<input type="checkbox"/>	Cross Corridor*	<input type="checkbox"/>
Departure	<input type="checkbox"/>	Transit Flight*	<input type="checkbox"/>
Landing	<input type="checkbox"/>	Holding Pattern	<input type="checkbox"/>

* Ziel angeben

Abbildung C-6: SAGAT Fragebogen bei manueller Umplanung in Szenario II

C.3 Subjektiver Fragebogen

Versuchsperson Nr.:	Testphase:	Datum:
---------------------	------------	--------

Subjektive Bewertung

1. Systeminteraktion

Bitte geben Sie an, wie gut die folgenden Adjektive Ihre Interaktion mit dem System beschreiben.

Die Bedienung des Systems ist ...

... intuitiv	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... zielführend	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... einfach	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... zeitsparend	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... angenehm	überhaupt nicht	<input type="checkbox"/>	voll und ganz							

2. Systemverhalten bei Lageänderungen

Bitte geben Sie an, wie gut die folgenden Adjektive das Verhalten des Systems bei einer Änderung der taktischen Lage beschreiben.

Das Systemverhalten bei taktischen Lageänderungen ist ...

... hilfreich	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... verständlich	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... informativ	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... vorhersehbar	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... nachvollziehbar	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... zielführend	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... zeitsparend	überhaupt nicht	<input type="checkbox"/>	voll und ganz							
... angenehm	überhaupt nicht	<input type="checkbox"/>	voll und ganz							

Abbildung C-7: Fragebogen „Subjektive Bewertung“ – Seite 1

