

Assistierte auftragsbasierte Führung von unbemannten Luftfahrzeugen aus dem Kampfflugzeugcockpit

Marius Dudek, M. Sc.

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

Doktor-Ingenieurs (Dr.-Ing.)

angenommenen Dissertation.

Gutachter/Gutachterin:

1. Univ.-Prof. Dr.-Ing. Axel Schulte
2. Univ.-Prof. Dr. Dipl.-Psych. Annette Kluge

Diese Dissertation wurde am 23.04.2025 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 23.07.2025 angenommen. Die mündliche Prüfung fand am 01.08.2025 statt.

Kurzzusammenfassung

Das militärische Einsatzkonzept des Manned-Unmanned Teaming (MUM-T) integriert bemannte und unbemannte Luftfahrzeuge (UAVs) in ein Team, um ihre jeweiligen Stärken bestmöglich zu nutzen: UAVs sind kosteneffizient, überwinden menschliche Leistungsgrenzen und können gefährliche Aufgaben übernehmen, während menschliche Entscheidungsträger die Führungsrolle übernehmen und so einen taktisch sinnvollen und ethisch verantwortbaren Einsatz der UAVs sicherstellen.

Diese Arbeit beschäftigt sich mit der Entwicklung eines Führungs- und Interaktionskonzepts, das es einem Nutzer ermöglicht, diese Führungsrolle einzunehmen und aus dem Cockpit eines Kampfflugzeuges mehrere unbemannte Luftfahrzeuge effektiv zu kommandieren.

Das Konzept basiert auf einer auftragsbasierten Führung der UAVs und unterstützt den Nutzer durch zwei zentrale Funktionen in seiner Führungsaufgabe: Erstens bietet es schnell anwendbare und flexibel einsetzbare Interaktionsmethoden zur Delegation von Aufträgen. Zweitens integriert es eine Kritikfähigkeit in den Delegationsprozess, die den Nutzer auf mögliche Probleme hinweist und alternative Lösungsvorschläge unterbreitet.

Zur Umsetzung des Konzepts wurden vier Interaktionsmethoden entwickelt, die eine Auftragsdelegation unter Nutzung von diskreten Touchscreen-Interaktionen, Touchscreen-Gesten, Sprachbefehlen sowie einer Kombination von Blickbewegungen und manuellen Knöpfen ermöglichen. Die Kritikfähigkeit wurde exemplarisch anhand verschiedener Anwendungsfälle umgesetzt. Beide Komponenten wurden in den Kampfflugzeugsimulator der Professur für Flugmechanik und Flugführung an der Universität der Bundeswehr München integriert.

Das entwickelte System wurde in vier Mensch-Maschine-Versuchen mit Kampfflugzeugpiloten der Luftwaffe erprobt. Zwei Studien untersuchten das Gesamtsystem, während die anderen beiden die Interaktionsmethoden und die Kritikfähigkeit im Detail evaluierten. Die Ergebnisse bestätigen die Eignung des Konzepts zur Führung mehrerer unbemannter Luftfahrzeuge. Die Untersuchungen zeigten, dass eine Wahlfreiheit zwischen mehreren Interaktionsmethoden sinnvoll ist, da deren Eignung stark von situativen Faktoren abhängt. Hinsichtlich der Kritikfähigkeit ergab die Untersuchung, dass eine solche Fähigkeit die Wahrnehmung von Systemfehlern verbessern kann, jedoch mit höheren Interaktionszeiten einhergeht als eine vorschlagsbasierte Form der Unterstützung. Daher sollte je nach Anwendung abgewogen werden, ob kritikbasierte oder vorschlagsbasierte Ansätze besser geeignet sind.

Die Ergebnisse dieser Arbeit liefern somit wichtige Erkenntnisse für die Gestaltung der UAV-Führung im Kontext von Manned-Unmanned Teaming.

Abstract

The operational concept of Manned-Unmanned Teaming (MUM-T) integrates manned and unmanned aerial vehicles (UAVs) into a team to exploit their respective strengths: UAVs are cost-effective, can overcome human physiological limitations, and can perform dangerous tasks, while human decision-makers take on the role of mission managers and thus ensure a tactically sound and ethically responsible use of UAVs.

This thesis is about the development of a guidance and interaction concept that enables a user to assume this command role and effectively guide multiple unmanned aerial vehicles from the cockpit of a fighter-jet.

The concept is based on a task-based guidance paradigm and supports the user in guiding their UAVs with two main capabilities: First, it offers fast and flexible interaction methods for task delegation. Second, it integrates a critiquing capability into the delegation process, informing the user of potential problems with delegated tasks and suggesting alternative solutions.

To implement the concept, four interaction methods were developed that allow task delegation using discrete touchscreen interactions, touchscreen gestures, voice commands, and a combination of gaze and manual button inputs. The critiquing capability was implemented for multiple use cases. Both components were integrated into the fighter-jet simulator at the Chair of Aircraft Dynamics and Flight Guidance at the University of the Bundeswehr Munich.

The developed system was tested in four human-in-the-loop simulator studies with fighter-jet pilots of the German Air Force. Two studies examined the full system, while the other two evaluated the interaction methods and the critiquing capability in detail. The results confirm the suitability of the concept for guiding multiple unmanned aerial vehicles. The studies showed that it is beneficial to offer multiple interaction methods, as the effectiveness of each method is highly dependent on situational factors. Regarding the critiquing capability, the study results indicate that while it can improve detection of system errors, it is associated with longer interaction times compared to a proposal-based approach. Therefore, it should be assessed on a case-by-case basis whether a critiquing-based or a proposal-based approach is more suitable for a specific application.

The findings of this research provide valuable insights into how UAVs can be guided by a user in the context of Manned-Unmanned Teaming.

Danksagung

Diese Arbeit fasst die Ergebnisse meiner Forschung zusammen, die ich von 2019 bis 2025 an der Professur für Flugmechanik und Flugführung der Universität der Bundeswehr München durchgeführt habe. Im Folgenden möchte ich mich bei all jenen Menschen bedanken, die mich während dieser Zeit unterstützt haben und somit einen wesentlichen Beitrag zum Gelingen dieser Arbeit geleistet haben.

Ein besonderer Dank gebührt meinem Doktorvater Herrn Prof. Dr.-Ing. Axel Schulte. In meiner Zeit als wissenschaftlicher Mitarbeiter hat er mich stets bei der Durchführung meiner Forschungsarbeiten unterstützt, eine hervorragende Laborumgebung zur Verfügung gestellt und mir viele bereichernde Konferenzreisen ermöglicht. Mein Dank gilt außerdem Frau Dr. Diana Donath, die mich allzeit in organisatorischen Belangen unterstützt und durch zahlreiche Hintergrundarbeiten wesentlich zu einem erfolgreichen Forschungsbetrieb beitragen hat. Ebenfalls danken möchte ich Frau Prof. Dr. Dipl.-Psych. Annette Kluge für ihre Beiträge als Zweitgutachterin dieser Arbeit.

Meinen Kollegen aus dem Jet-Simulator möchte ich einen großen Dank für die gemeinsame Zeit aussprechen. Dieses großartige Team hat mir viel beigebracht und ich habe es sehr geschätzt, wie konsequent alle Teammitglieder auf die gemeinsamen Ziele hingearbeitet und dabei auch persönliche Interessen zurückgestellt haben. Dankbar bin ich dem Team auch für die vielfältige Unterstützung während der Anfertigung dieser Arbeit. Hervorheben möchte ich dabei insbesondere Herrn Dr. Dennis Mund und Herrn Prof. Dr. Simon Schwerd, die mich beide unermüdlich mit wertvollen Ratschlägen zu meiner Dissertation begleitet haben.

Ein großer Dank gilt auch den Kampfflugzeugpiloten der Luftwaffe sowie von Airbus Defense & Space. Mit großem Enthusiasmus haben mir diese Piloten das für meine Forschung notwendige Fachwissen vermittelt und uns durch zahlreiche Verbesserungsvorschläge bei der Weiterentwicklung des Kampfflugzeugsimulators unterstützt. Erwähnenswert sind außerdem das ausgeprägte Interesse an unserer Forschung und die unermüdliche Ausdauer als Probanden während unserer experimentellen Untersuchungen.

Zum Abschluss möchte ich mich bei meinen Eltern, meinem Bruder und meiner Partnerin bedanken, die mich in entscheidenden Phasen der Promotion mit wertvollen Ratschlägen und motivierenden Worten unterstützt haben.

Inhaltsverzeichnis

1. Einleitung	1
2. Hintergrund.....	3
2.1 Einsatzszenarien bemannt-unbemannter Teams	3
2.1.1 Missionsvignette A: Terrorismusbekämpfung	3
2.1.2 Missionsvignette B: Bekämpfung eines Launchers	4
2.1.3 Missionsvignette C: Zerstörung eines C2-Kommandoposten	6
2.2 Mensch-Maschine-Aufgabenteilung	8
2.2.1 Funktionale Aufgabenallokation	8
2.2.2 Automationsgrade	10
2.2.3 Arbeitsprozess- und Arbeitssystemmethode	12
2.3 Mensch-Maschine-Interaktion.....	17
2.3.1 Interaktionsmodalitäten.....	17
2.3.2 Interaktionsmethoden für Kampfflugzeuge	22
2.3.3 Feedback-Taxonomie	25
2.4 Agentenarchitekturen	26
3. Konzept assistierte auftragsbasierte Führung	28
3.1 Definition der Anforderungen.....	28
3.1.1 Anwendung der Arbeitsprozessanalyse	28
3.1.2 Anforderungen an das Mensch-Maschine-System	30
3.2 Systementwurf	31
3.2.1 Kognitives Systemdesign	31
3.2.2 Rollen und Beziehungen der Akteure.....	36
3.2.3 Funktionale Architektur.....	38
3.3 Aufträge als Kommunikationsmedium.....	39
3.3.1 Auftragsdefinition.....	39
3.3.2 Aufträge für militärische Kampfflugzeugmissionen	40
3.3.3 Auftragsdelegation	41
3.4 Delegationsagent	44
3.4.1 Aufgaben.....	44
3.4.2 Aufbau	47
3.5 Plattformsteuerungsagent	49

3.5.1 Aufgaben.....	49
3.5.2 Aufbau	50
4. Implementierung und Integration	52
4.1 Simulationsumgebung.....	52
4.2 Gestaltung der Nutzerschnittstelle	55
4.2.1 Interaktionsmethoden zur Auftragsdelegation.....	55
4.2.2 Richtlinienmanagement	65
4.2.3 Auftragskritik	67
4.2.4 UAV-Feedback	70
4.2.5 Nutzerdialoge	73
4.3 Funktionale Umsetzung der Nutzerschnittstelle	74
4.3.1 Sprachsteuerung.....	74
4.3.2 Sprachsynthese.....	77
4.3.3 Objektauswahl mittels Blickbewegung	78
4.4 Umsetzung der Agentenfunktionen.....	80
4.4.1 Missionsplanung.....	80
4.4.2 Auftragsvervollständigung.....	82
4.4.3 Auftragsanalyse und -bewertung	83
4.4.4 Verhaltensgenerierung.....	85
4.5 Simulator-Integration.....	97
5. Validierungsstudien	100
5.1 Vorstudien	101
5.1.1 Vergleich von Delegationsmethoden in verschiedenen Missionsphasen.....	101
5.1.2 Vergleich von Delegationsmethoden für dynamische Auftragserteilung.....	101
5.1.3 Vergleich von Modalitäten von Nutzerdialogen	102
5.2 Gesamtsystemerprobung: Nutzung von Führungsfunktionen	103
5.2.1 Studiendesign	105
5.2.2 Ergebnisse.....	107
5.2.3 Diskussion	111
5.2.4 Fazit.....	113
5.3 Gesamtsystemerprobung: Bewertung des Systems	114
5.3.1 Studiendesign	116
5.3.2 Ergebnisse.....	118

5.3.3 Diskussion	130
5.3.4 Fazit.....	131
5.4 Detailstudie: Delegationsmethoden	132
5.4.1 Studiendesign	134
5.4.2 Ergebnisse.....	139
5.4.3 Diskussion	145
5.4.4 Fazit.....	147
5.5 Detailstudie: Kritikfähigkeit in Auftragsdelegation	148
5.5.1 Studiendesign	150
5.5.2 Ergebnisse.....	155
5.5.3 Diskussion	161
5.5.4 Fazit.....	162
6. Zusammenfassung und Ausblick.....	165
6.1 Beiträge dieser Arbeit	165
6.2 Kritische Reflexion.....	167
6.3 Weiterer Forschungsbedarf	170
A. Literaturverzeichnis	173
B. Abkürzungsverzeichnis	188
C. Abbildungsverzeichnis	189
D. Tabellenverzeichnis	193
E. Anhang	195
E.1 Logische Objekte	195
E.2 Liste von UAV-Aufträgen	196
E.3 Vokabular der Delegationsmethode Voice	198
E.4 Auftragsvervollständigung	200
E.4.1 Ermittlung des Auftragsstyps	200
E.4.2 Ermittlung des Erfolgskriteriums	201
E.5 Modellierung des BehaviorTree und resultierendes Agentenverhalten	202
E.6 Anhang zu Validierungsstudien	207
E.6.1 Detaillierte Missionsbeschreibung zur Studie „Nutzung von Führungsfunktionen“	207
E.6.2 Detaillierte Missionsbeschreibung zur Studie „Bewertung des Systems“	211

1. Einleitung

Der Krieg in der Ukraine verdeutlicht die Rolle unbemannter Luftfahrzeuge (Unmanned Aerial Vehicles; UAV) als Mittel der modernen Kriegsführung. Beide Konfliktparteien setzen in großem Umfang UAVs ein, um sich einen Vorteil auf dem Gefechtsfeld zu verschaffen. In den Medien machte die russische Seite vor allem durch die massiven Angriffe auf die ukrainische Energieinfrastruktur Schlagzeilen, die unter anderem mit iranischen Shahed-Drohnen durchgeführt wurden, welche sich mithilfe von GPS- und Trägheitsnavigation auf ein vorprogrammiertes Ziel stürzen (Wolkov et al., 2024). Demgegenüber sind die ukrainischen Streitkräfte insbesondere für die Bekämpfung von Bodentruppen und Panzern durch sprengstoffbeladene FPV-Drohnen¹ bekannt, ein Eindruck, der nicht zuletzt durch die gezielte Verbreitung von Filmaufnahmen verschiedener Drohneneinsätze über offizielle und inoffizielle Kanäle verstärkt wurde (Zafra et al., 2024). Diese medial geprägte Wahrnehmung kann den Anschein erwecken, dass die militärische Zukunft von unbemannten Luftfahrzeugen vor allem in vorprogrammierten oder händisch gesteuerten Drohnen liegt.

Die Forschungs- und Entwicklungsaktivitäten im Bereich *Manned-Unmanned Teaming* (MUM-T) zeichnen dagegen ein anderes Bild davon, welche Rolle unbemannte Luftfahrzeuge in zukünftigen militärischen Konflikten spielen könnten. Entwicklungsprogramme wie das US-Amerikanische Next Generation Air Dominance (NGAD) und das europäische Future Combat Air System (FCAS) streben eine neue Form der Luftkriegsführung an, in der bemannte und unbemannte Kampfflugzeuge als Team agieren, um ein Missionsziel zu erfüllen (BDLI, 2021; U. S. Naval Institute, 2024). Das Ziel dieses Ansatzes ist es, die Stärken von bemannten und unbemannten Plattformen gleichermaßen auszuspielen: Unbemannte Plattformen können in großer Zahl und für gefährliche oder ermüdende Aufgaben eingesetzt werden, während menschliche Piloten die Kontrolle über das Missionsgeschehen behalten und einen rechtlich und ethisch vertretbaren Waffeneinsatz sicherstellen.

Während die Entwicklungsprogramme FCAS und NGAD eine sehr klare Vorstellung davon vermitteln, welche Flugzeugtypen und -muster in Zukunft zusammenarbeiten werden, lassen die Programme die Frage offen, wie genau die Kooperation zwischen den bemannten und unbemannten Plattformen aussehen soll (Copp & Weisgerber, 2021; Gertler, 2020). Agieren die unbemannten Plattformen weitgehend unabhängig und beziehen den menschlichen Piloten nur sporadisch in Entscheidungen mit ein? Arbeiten bemannte und unbemannte Systeme auf Augenhöhe wie menschliche Teams? Oder werden die unbemannten Systeme den bemannten Systemen hierarchisch unterstellt?

Ein vielversprechender Kooperationsansatz besteht darin, *Flights*² aus bemannten und unbemannten Plattformen zu bilden, wobei die bemannten Plattformen die Rolle der *Flight Leads* (Gruppenführer) übernehmen, während die unbemannten Plattformen als *Wingmen*

¹ FPV (First-Person-View): Operateure steuern die Drohnen anhand eines Kamerabildes, das in der Drohne aufgenommen wird und zum Operateur übertragen wird (z.B. auf eine VR-Brille). Die Steuerung erfolgt i.d.R. manuell.

² Eine Flight ist eine kleine taktische Einheit innerhalb der Strukturen der Luftwaffe, Marine- oder Heeresfliegern.

agieren (Lockheed Martin, 2017). Die menschlichen Piloten kommandieren dabei die unbemannten Plattformen, die – ähnlich wie menschliche Wingmen – einen gewissen Handlungsspielraum bei der Umsetzung dieser Befehle haben. Dieser Ansatz stellt sicher, dass die menschlichen Piloten die Kontrolle über die unbemannten Systeme behalten, wirft jedoch eine zentrale Frage auf: Wie muss die Führung der unbemannten Plattformen gestaltet sein, damit ein Pilot neben seiner eigenen Plattform mehrere unbemannte Luftfahrzeuge effektiv steuern und koordinieren kann? Diese zentrale Fragestellung steht im Fokus der vorliegenden Arbeit.

Ziel dieser Arbeit

Das Ziel dieser Arbeit ist die Entwicklung eines Führungs- und Interaktionskonzepts, das es einem Piloten ermöglicht, aus dem Cockpit eines Kampfflugzeugs mehrere unbemannte Luftfahrzeuge effektiv zu kommandieren. Dabei sollen drei Aspekte schwerpunktmäßig betrachtet werden:

1. Welche Rollen und Aufgaben übernehmen Mensch und Automation im Rahmen des Führungskonzeptes?
2. Wie muss die Nutzerinteraktion gestaltet sein, um eine schnelle und verantwortbare Führung der unbemannten Systeme zu ermöglichen?
3. Welche technischen Funktionen ergeben sich daraus und wie können diese technischen Funktionen umgesetzt werden?

Aufbau der Arbeit

Kapitel 2 gibt einen Überblick über den **Hintergrund** der Arbeit, indem es sowohl das zukünftige Einsatzumfeld von Kampfflugzeugen im Kontext von Manned-Unmanned Teaming als auch grundlegende Aspekte der Mensch-Maschine-Kooperation beleuchtet. Kapitel 3 beschreibt die **Konzeption des Mensch-Maschine-Systems**. Zunächst werden die Anforderungen an das Mensch-Maschine-System ermittelt, dann werden auf Basis dieser Anforderungen die Rollen und Aufgaben von Mensch und Automation festgelegt und anschließend werden die Elemente des Mensch-Maschine-Systems detailliert beschrieben. Das Kapitel 4 behandelt die **Implementierung und Integration** des Konzepts. Im Zentrum stehen dabei die Gestaltung der Nutzerschnittstelle, die Entwicklung der technischen Funktionen und deren Integration in die Simulationsumgebung. Im Anschluss werden in Kapitel 5 vier **experimentelle Untersuchungen** vorgestellt, die zur Validierung des Funktionsprototypen durchgeführt wurden. Das abschließende Kapitel 6 fasst die wichtigsten **Ergebnisse** zusammen, diskutiert deren **Grenzen** und zeigt den weiteren **Forschungsbedarf** auf.

2. Hintergrund

Dieses Kapitel befasst sich mit relevanten Themenfeldern, die die Basis für die vorliegende Arbeit bilden. Zu Beginn des Kapitels wird in Abschnitt 2.1 das **operationelle Umfeld** beschrieben, das die wesentlichen Anforderungen und Rahmenbedingungen für die Systementwicklung vorgibt. Abschnitt 2.2 behandelt die **Mensch-Maschine-Aufgabenteilung**, die sich mit der optimalen Verteilung von Rollen- und Aufgaben zwischen menschlichen Akteuren und automatisierten Systemen beschäftigt. Darauf aufbauend widmet sich Abschnitt 2.3 der **Mensch-Maschine-Interaktion** und diskutiert die Schnittstellen, mit denen Menschen mit automatisierten Systemen interagieren können. Abschnitt 2.4 behandelt **Architekturen für Agenten**, die zur Bearbeitung von Missionsaufgaben an Bord eines UAVs verwendet werden können.

2.1 Einsatzszenarien bemannt-unbemannter Teams

Das operationelle Umfeld spielt eine entscheidende Rolle bei der Auslegung des Führungs- und Interaktionskonzeptes, da es wesentliche Anforderungen und Rahmenbedingungen vorgibt, die bei der Systementwicklung zu berücksichtigen sind. In diesem Abschnitt wird daher das Einsatzumfeld von bemannt-unbemannten Teams in zukünftigen Kampfflugzeugmissionen beschrieben.

Zu diesem Zweck werden nachfolgend drei Missionsszenarien für Manned-Unmanned Teaming vorgestellt, die in Anlehnung an die Missionsvignetten der ETAP Global System Study³ entworfen wurden. Diese drei Missionsvignetten verdeutlichen dabei die unterschiedlichen Missionsaufgaben, Kräftekonfigurationen und Randbedingungen, die sich aus verschiedenen Konfliktstufen ergeben: Die *Missionsvignette A* beschreibt ein Szenario, in dem die verbündeten Kräfte die Luftherrschaft besitzen und unbemannte Luftfahrzeuge primär zur Erweiterung der Sensor- und Effektorkapazitäten sowie aufgrund besserer Stealth-Eigenschaften eingesetzt werden. *Missionsvignette B* stellt ein Szenario dar, in dem die gegnerischen Kräfte über moderate Flugabwehr- und Luft-Luft-Fähigkeiten verfügen, was eine koordinierte Zusammenarbeit zwischen bemannten und unbemannten Plattformen erfordert. *Missionsvignette C* bildet schließlich ein Szenario ab, in dem der Gegner über ein großes Luftkriegspotenzial verfügt, sodass umfangreiche Ressourcen und eine präzise Abstimmung erforderlich sind, um die Missionsziele erfolgreich zu erreichen.

2.1.1 Missionsvignette A: Terrorismusbekämpfung

Das Missionsziel der MV-A besteht darin, einen Präventivschlag gegen ein terroristisches Ausbildungslager in einem Drittstaat durchzuführen. Der Angriff soll möglichst verdeckt erfolgen, d. h. nachts und unter Umgehung aller bekannten Radarpositionen. Die Position der Basis wurde durch Aufklärungssatelliten genau genug bestimmt. Eine Verifizierung der zu bekämpfenden Ziele ist zwingend erforderlich und muss unmittelbar vor dem Einsatz des

³ ETAP GSS Phase 3, Report R2.1 – Operational Assumptions and System Data Base (2015)

Effektors vor Ort erfolgen. Das Erfolgskriterium der Mission ist die vollständige Zerstörung des Ausbildungslagers.

Der Missionsablauf gestaltet sich wie folgt: Zu Beginn überqueren die verbündeten Kräfte (sog. *BlueForce*) die FLOT⁴ und dringen in das Gebiet des Drittstaates ein. Um unentdeckt zu bleiben, erfolgt der Flug im Tiefflug, wobei bekannte SAMs⁵ umgangen werden. Anschließend wird das Ausbildungscamp aufgeklärt, und die Ziele im Camp werden innerhalb eines möglichst kurzen Zeitraums bekämpft. Nach der Bestätigung der erfolgreichen Bekämpfung kehren die Kräfte ebenfalls im Tiefflug zum Heimatflughafen zurück. Eine Entdeckung durch den Drittstaat soll während der gesamten Mission vermieden werden, da dies das Eingreifen einer Alarmrotte auslösen könnte. Abbildung 1 A zeigt eine Übersicht über die Mission.

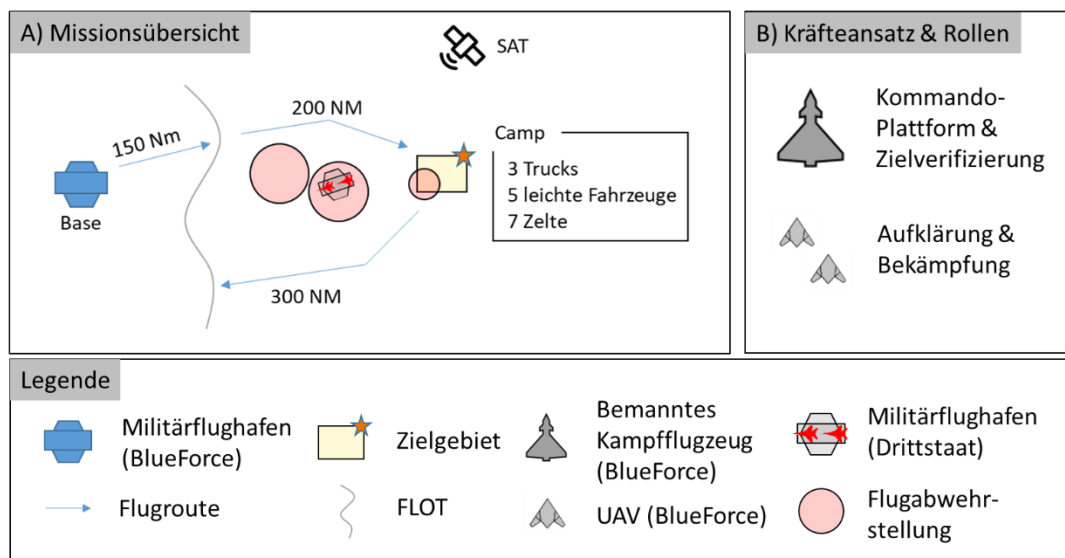


Abbildung 1. Missionsvignette A; A) Missionsübersicht B) Kräfteansatz & Rollen

Der Kräfteansatz zur Durchführung der Mission besteht aus einem bemannten und zwei unbemannten Kampfflugzeugen. Die bemannte Plattform fungiert als abgesetzte Kommando-Plattform für die unbemannten Plattformen und verifiziert die Ziele im Ausbildungslager. Die unbemannten Plattformen sind für die Aufklärung und Zielbekämpfung zuständig. Der Kräfteansatz und die Rollen der Plattformen sind in Abbildung 1 B dargestellt.

2.1.2 Missionsvignette B: Bekämpfung eines Launchers

In Missionsvignette B besteht das Missionsziel in der Lokalisierung und Zerstörung einer ballistischen Raketenabschussvorrichtung (mit Hinweis auf eine nukleare Bedrohung). Die Position der Abschussvorrichtung ist nicht exakt bekannt, aber wurde auf ein Zielgebiet von 25 km² eingegrenzt. Um eine Relokation oder gar einen Einsatz des Raketenwerfers zu verhindern, ist es erforderlich, diesen so schnell wie möglich zu bekämpfen. Das Erfolgskriterium der Mission ist die erfolgreiche Zerstörung der Abschussvorrichtung – diese muss anhand einer Nachaufklärung sichergestellt werden. Aufgrund des hohen

⁴ Die FLOT (Forward Line of Own Troops) bezeichnet die Frontlinie der eigenen Kräfte, die am nächsten zum Gegner oder zum Konfliktgebiet verläuft.

⁵ SAM (Surface-to-Air Missile) wird als Bezeichnung für Flugabwehrstellungen verwendet.

Gefahrenpotenzials bei Verfehlung des Missionsziels werden begrenzte Kollateralschäden und Verluste von unbemannten Plattformen in Kauf genommen.

Der Missionsablauf ist wie folgt: Die BlueForce startet vom Heimatflughafen und fliegt in Richtung der FLOT. Aufgrund der großen Distanz tanken die Flugzeuge vor dem Überqueren der FLOT noch einmal auf. Die Kräfte überqueren die FLOT und fliegen weiter in Richtung Zielgebiet. Im feindlichen Luftraum ist mit Bedrohungen in Form von auftauchenden Flugabwehrstellungen (sogenannte *Popup-SAMs*) und Kampfflugzeugen zu rechnen, die neutralisiert werden müssen, um das Zielgebiet zu erreichen und dort operieren zu können. Am Zielgebiet angekommen, durchsuchen die Kräfte das Gelände, um den Launcher zu finden. Sobald der Launcher entdeckt wird, erfolgt dessen Bekämpfung, gefolgt von einer Nachaufklärung, um die erfolgreiche Zerstörung zu bestätigen. Anschließend zieht sich die BlueForce zügig zum Heimatflughafen zurück. Eine Übersicht über die Missionsvignette B ist in Abbildung 2 A abgebildet.

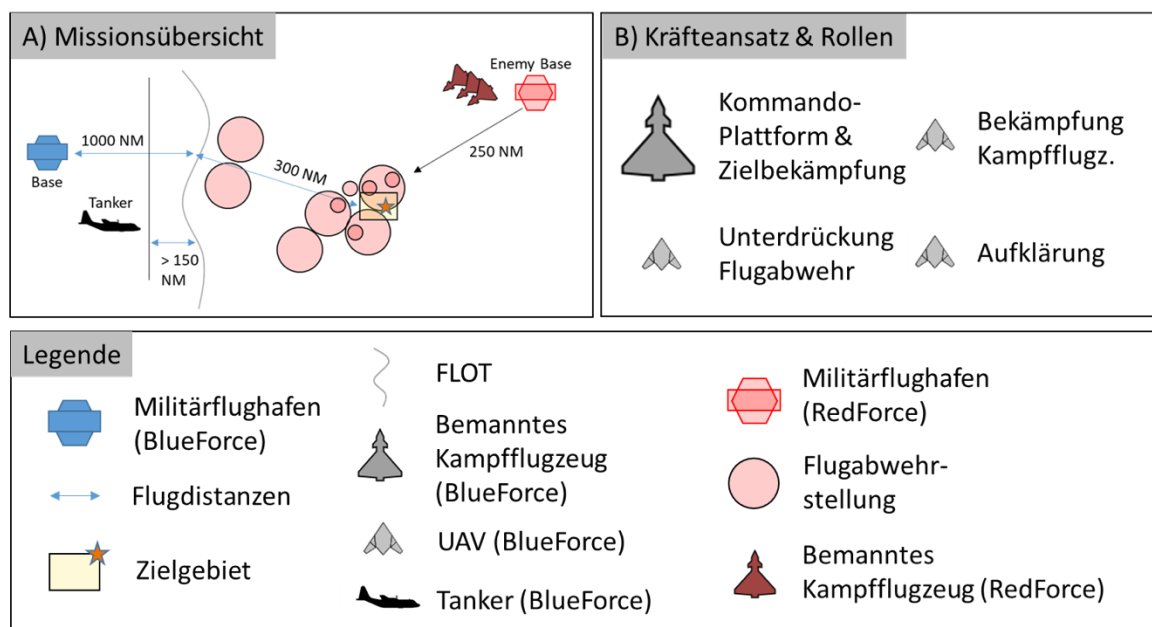


Abbildung 2. Missionsvignette B; A) Missionsübersicht B) Kräfteansatz & Rollen

Die Mission wird mit einem Streitkräfteansatz bestehend aus einem bemannten und drei unbemannten Kampfflugzeugen durchgeführt. Das bemannte Kampfflugzeug fungiert als Kommando-Plattform und ist für die Bekämpfung des Launchers zuständig. Die unbemannten Plattformen sind mit unterschiedlichen Sensoren und Effektoren ausgestattet und werden in der Mission für die Bekämpfung von Flugabwehrstellungen und Kampfflugzeugen sowie für die Aufklärung des Launchers eingesetzt. Der Kräfteansatz und die Rollen der verschiedenen Plattformen sind in Abbildung 2 B dargestellt.

2.1.3 Missionsvignette C: Zerstörung eines C2-Kommandoposten

Das Missionsziel dieser Missionsvignette besteht in der Zerstörung eines C2-Kommandopostens⁶ der gegnerischen Luftabwehr. Zu diesem Zweck muss die feindliche Luftverteidigung überwunden werden und der Kommandoposten mit bunkerbrechenden Waffen bekämpft werden. Die Position des Kommandopostens ist bekannt, aber der Angriff kann durch Störsignale im elektronischen Spektrum (GPS-Spoofing, etc.) beeinträchtigt werden. Das Erfolgskriterium der Mission ist die erfolgreiche Bekämpfung des Kommandopostens. Kollateralschäden und Verluste werden aufgrund der Priorität der Mission akzeptiert.

Der Missionsablauf gestaltet sich wie folgt: Die BlueForce startet von einem verbündeten Militärflughafen und nimmt Kurs auf die FLOT. Dort steht ein Tankflugzeug bereit, um die Kampfflugzeuge aufzutanken. Zudem kreist an der FLOT ein AWACS⁷, das aus der Luft Radardaten über feindliche Luftfahrzeuge bereitstellt. Im Feindgebiet muss die BlueForce mit zahlreichen Popup-SAMs und Formationen feindlicher Kampfflugzeuge rechnen, bestehend aus bemannten und unbemannten Plattformen. Diese Bedrohungen sind zu überwinden, um den Kommandoposten zu erreichen und ihn mit bunkerbrechenden Waffen zu zerstören. Nach der erfolgreichen Zerstörung des Kommandopostens erfolgt der Rückzug, wobei auch beim Verlassen des Missionsgebiets eine hohe Bedrohungslage besteht. In Abbildung 3 A ist die Mission schematisch dargestellt.

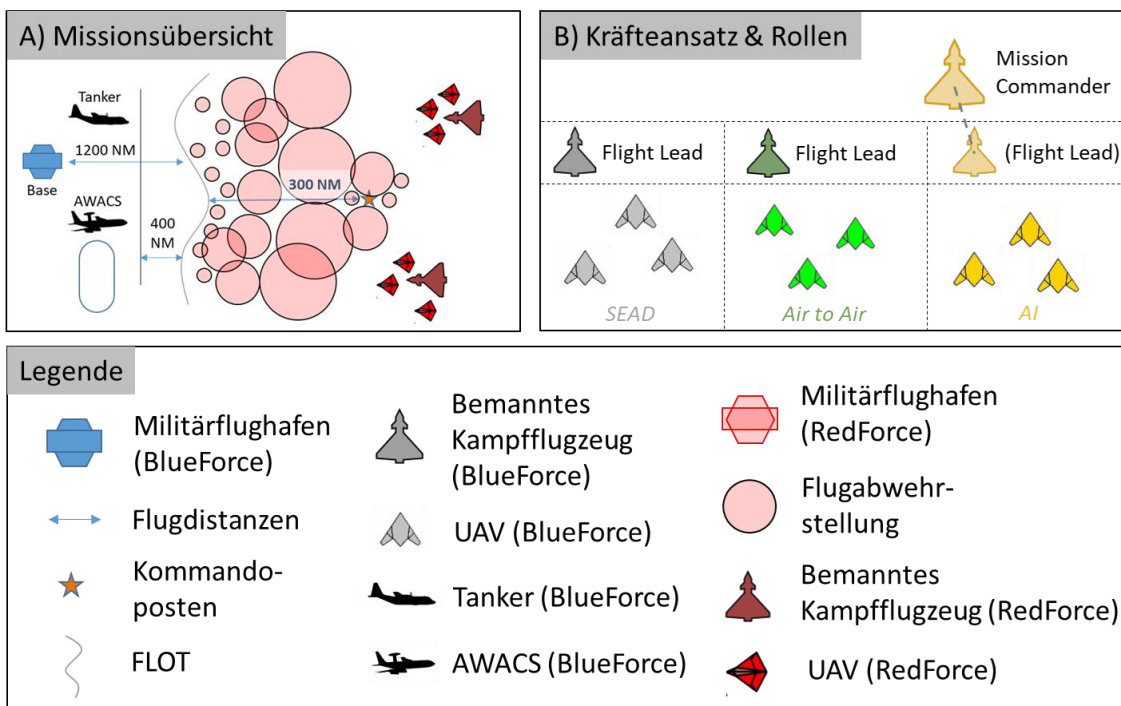


Abbildung 3. Missionsvignette C; A) Missionsübersicht B) Kräfteansatz & Rollen

⁶ C2 (Command & Control) ist für die Planung und Koordination von militärischen Operationen zuständig.

⁷ Ein AWACS (Airborne Early Warning and Control System) ist ein fliegendes Radarsystem.

Die Durchführung der zuvor vorgestellten Mission erfordert eine COMAO⁸ mit einer Vielzahl von bemannten und unbemannten Plattformen, die in mehreren *Flight Packages* organisiert sind. Ein Flight Package ist eine kleiner Flugverband, der jeweils aus einem bemannten und mehreren unbemannten Kampfflugzeugen besteht. In der Mission haben die Flight Packages unterschiedliche Missionsaufgaben. So ist beispielsweise das *Air-to-Air* Package in erster Linie für die Bekämpfung von feindlichen Luftfahrzeugen zuständig, das *SEAD*⁹ Package ist für die Bekämpfung von Flugabwehrstellungen verantwortlich und das *AI*¹⁰ Package führt den Angriff auf den Kommandoposten durch. Die bemannten Kampfflugzeuge sind im Rahmen der COMAO für verschiedene Führungsaufgaben vorgesehen. In jedem Package übernimmt eine bemannte Plattform die Rolle des Flight Leads, der für die Führung der anderen Gruppenmitglieder verantwortlich ist. Die Koordination zwischen den verschiedenen Packages wird vom *Mission Commander* übernommen. Diese Rolle übernimmt der Pilot des Kampfflugzeugs im AI Package – der Pilot ist also in einer Doppelrolle als Flight Lead und Mission Commander im Einsatz. Eine Übersicht über den Kräfteansatz und die Rollen der COMAO ist in Abbildung 3 B dargestellt.

⁸ COMAO: Composite Air Operation (Deutsch: Verbundene Luftoperation).

⁹ SEAD: Suppression of Enemy Air Defences (Deutsch: Unterdrückung der Flugabwehr).

¹⁰ AI: Air Interdiction (Deutsch: Bekämpfung von Bodenzielen).

2.2 Mensch-Maschine-Aufgabenteilung

Eine Rollen- und Aufgabenverteilung, die die jeweiligen Stärken von Mensch und Automation optimal nutzt, ist der Schlüssel zu einer effektiven Zusammenarbeit. Aus diesem Grund befasst sich dieser Abschnitt mit der Verteilung von Rollen und Aufgaben in Mensch-Maschine-Systemen.

Der Abschnitt beginnt mit einer Beschreibung der frühen **funktionalen Ansätze zur Aufgabenteilung** (Unterabschnitt 2.2.1). Unterabschnitt 2.2.2 behandelt das Konzept der **Automationsgrade**, welche verschiedene Abstufungen der Mensch-Maschine-Aufgabenteilung unter der Prämisse einer menschlichen Führung definieren. Abschließend wird in Unterabschnitt 2.2.3 die **Arbeitsprozess- und Arbeitssystemmethode** vorgestellt, die eine flexible Beschreibung und Gestaltung der Aufgaben- und Rollenverteilung zwischen Mensch und Automation ermöglicht.

2.2.1 Funktionale Aufgabenallokation

Bei funktionalen Allokationsansätzen basiert die Zusammenarbeit zwischen Mensch und Automation auf einer strikten Aufgabenteilung: Die Aufgabe wird in Teilaufgaben zerlegt und entsprechend der erforderlichen Kompetenzen entweder dem Menschen oder der Automation zugeordnet.

Ein prominenter Ansatz, um die Aufgabenteilung zwischen Mensch und Automation anhand der jeweiligen Kompetenzen zu definieren, ist der *MABA-MABA*¹¹ Ansatz oder auch *Fitts Liste* (1951). Diese Liste beschreibt, in welchen Fähigkeiten der Mensch der Automation überlegen ist und umgekehrt, und kann somit als Leitlinie dienen, ob eine Aufgabe besser dem Menschen oder der Automation zugewiesen werden sollte. In Abbildung 4 sind die von Fitts beschriebenen Fähigkeiten dargestellt (frei übersetzt).

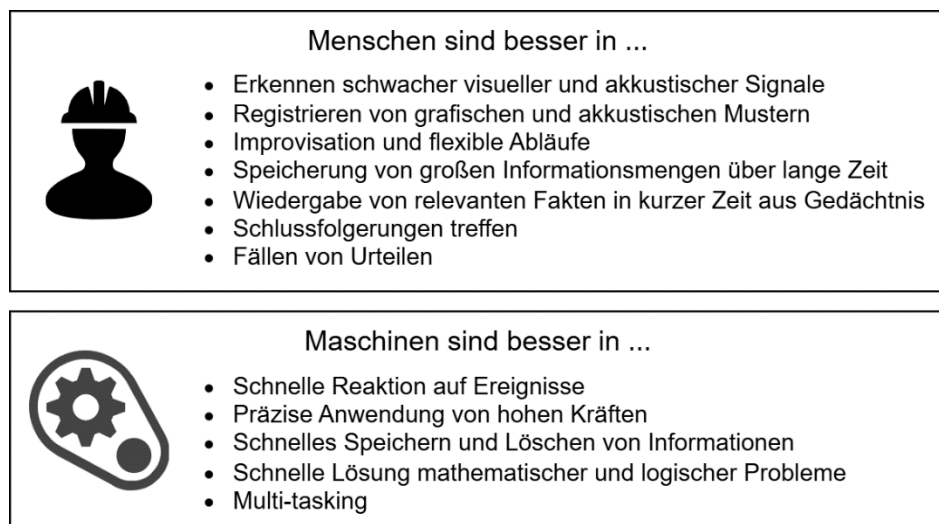


Abbildung 4. Beschreibung der Fähigkeiten von Mensch und Automation nach Fitts (1951).

¹¹ Abgekürzt für Men-are-better-at/Machines-are-better-at.

Bei näherer Betrachtung der MABA-MABA Liste wird schnell deutlich, dass dieser Ansatz den technologischen Fortschritt nicht ausreichend berücksichtigt, da die Automatisierung dem Menschen heute in vielen Punkten überlegen ist, die damals dem Menschen zugeschrieben wurden.

Ein anderer Ansatz zur Definition der Aufgabenteilung ist die Entscheidungsmatrix zur Funktionallokation (Pulliam et al., 1983). Im Prinzip basiert die Entscheidungsmatrix auf dem Grundgedanken der Fitts-Liste, dass die Fähigkeiten von Mensch und Automation bei der Aufgabenverteilung eine entscheidende Rolle spielen sollten, allerdings ist die Entscheidungsmatrix hinsichtlich der konkreten Aufgabenzuordnung deutlich differenzierter. Der Prozess der Aufgabenallokation beginnt bei der Entscheidungsmatrixmethode mit einer (qualitativen) Bewertung der Leistung von Mensch und Automatisierung bei der Bearbeitung der Aufgabe. Anschließend wird die Aufgabe auf Grundlage dieser Bewertung in ein Diagramm eingeordnet, das in verschiedene Regionen unterteilt ist (Abbildung 5).

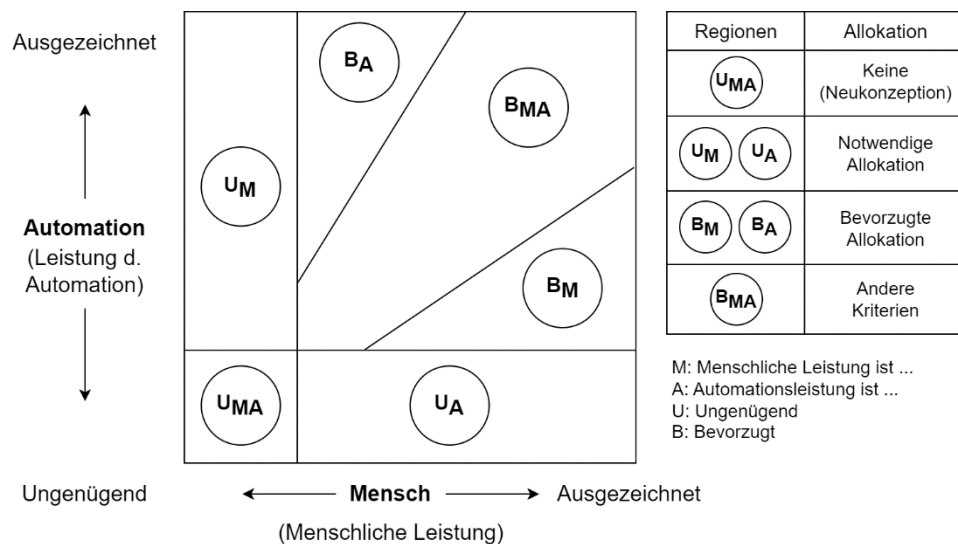


Abbildung 5. Entscheidungsmatrix zur Funktionsallokation (vgl. Pulliam et al. (1983) & Joe et al. (2015)).

Die Methode unterscheidet das weitere Vorgehen zur Aufgabenallokation nun abhängig davon, in welche Region eine Aufgabe eingeordnet wurde. Die mit U bezeichneten Regionen stellen Regionen dar, in denen die Leistung eines Akteurs (Mensch oder Automation) unzureichend ist. Bei Aufgaben, die in die Region U_A fallen, ist die Leistung der Automation zu gering, sodass die Aufgaben dem Menschen zugewiesen werden sollten. Umgekehrt verhält es sich in der Region U_M , in der die Aufgabe folglich der Automation zugewiesen werden sollte. Einen Sonderfall stellt in diesem Kontext die Region U_{MA} dar. Aufgaben, die in diese Region fallen, sollten nach Möglichkeit anders konzipiert werden, da kein Akteur in der Lage ist, die Aufgabe zufriedenstellend zu erfüllen. Die mit B gekennzeichneten Regionen stehen für Aufgaben, die von beiden Akteuren zufriedenstellend erledigt werden können. In der Region B_M ist die erwartete Leistung des Menschen höher und sollte daher dem Menschen zugeordnet werden, sofern keine anderen Argumente gegen diese Zuordnung sprechen. Umgekehrt sollten Aufgaben in der Region B_A bevorzugt von der Automation ausgeführt werden. In der Region B_{MA} sind die Leistungsunterschiede zwischen den Akteuren gering,

sodass die Zuordnung der Aufgaben in dieser Region anhand anderer Kriterien wie beispielsweise Kosten erfolgen kann.

Ein wesentlicher Nachteil funktionaler Allokationsansätze liegt in der impliziten Annahme, dass bisher von Menschen ausgeführte Aufgaben ohne Anpassungen von einem automatisierten System übernommen werden können. In der Regel führt der Einsatz von Automation jedoch zu einer grundlegend anderen Arbeitsweise, da der Mensch mit dem automatisierten System interagieren muss und dadurch mit neuen Aufgaben konfrontiert wird, wie beispielsweise der Eingabe von Daten oder der Überwachung des Systems (Dekker & Woods, 2002; Roth et al., 2019).

2.2.2 Automationsgrade

Einen anderen Ansatz zur Definition einer Mensch-Maschine-Aufgabenteilung stellen die *Automationsgrade* (eng. Levels of Automation) dar. Das Konzept der Automationsgrade basiert auf der Prämisse, dass der Mensch gegenüber dem automatisierten System eine Führungsrolle einnimmt und definiert für dieses Führungsverhältnis verschiedene Stufen, die sich darin unterscheiden, wie viel Entscheidungs- und Handlungsautorität der Mensch an die Automation überträgt. Das anfängliche Konzept geht auf eine Definition von Sheridan & Verplank (1978) zurück, nach der die verschiedenen Formen der Zusammenarbeit anhand von zehn Stufen beschrieben werden (Tabelle 1).

Tabelle 1. Die initiale Definition der Automationsgrade (frei übersetzt nach Sheridan & Verplank (1978)).

Level	Beschreibung der Interaktion
1 (Niedrig)	Der Mensch erledigt alle Aufgaben und übergibt diese an den Computer zur Umsetzung.
2	Der Computer ermittelt alle Handlungsoptionen.
3	Der Computer ermittelt alle Handlungsoptionen und schlägt eine vor.
4	Der Computer wählt eine Handlungsoption vor und der Mensch entscheidet, ob er diese umsetzen möchte.
5	Der Computer wählt eine Option aus und führt sie nach Zustimmung des Menschen aus.
6	Der Computer wählt eine Option aus und führt diese aus, wenn der Mensch nicht widerspricht.
7	Der Computer erledigt die Aufgabe und informiert den Menschen über sein Vorgehen.
8	Der Computer erledigt die Aufgabe und informiert den Menschen auf Nachfrage über sein Vorgehen.
9	Der Computer erledigt die Aufgabe und informiert den Menschen über sein Vorgehen, wenn er es für notwendig hält.
10 (Hoch)	Der Computer erledigt die Aufgabe, wenn er entscheidet, dass sie erledigt werden muss, und informiert den Menschen, wenn er es für notwendig hält.

Eine Einschränkung dieser Definition besteht darin, dass sich die Skala ausschließlich auf Automatisierung bezieht, die den Menschen bei der Auswahl von Entscheidungen oder Handlungen unterstützt und dass diese somit nicht zur Beschreibung von Automation verwendet werden kann, die für andere Funktionen eingesetzt wird (z. B. zur Informationsanalyse). Aus diesem Grund wurde eine Reihe von Modellen entwickelt, die das Konzept der Automationsgrade verallgemeinern, indem sie zwischen den verschiedenen Funktionen unterscheiden, für die Automatisierung eingesetzt werden kann (Endsley & Kaber, 1999; Parasuraman et al., 2000).

Ein Ansatz zur Verallgemeinerung des Konzeptes der Automationsgrade stellt das Modell zur Charakterisierung von Arten und Ebenen der Interaktion mit Automation (Parasuraman et al., 2000) dar, das zur Beschreibung von automatisierten Systemen entwickelt wurde. Die Kernidee des Modells ist, dass die menschliche Informationsverarbeitung aus vier Phasen besteht und jede davon durch Automation unterstützt werden kann. Dementsprechend werden automatisierte Systeme im Modell beschrieben, indem der Automationsgrad des Systems entlang dieser vier Phasen definiert wird. In Abbildung 6 ist die Anwendung des Modells exemplarisch anhand eines automatisierten Systems zur Suche und Bekämpfung eines Zielobjektes dargestellt.

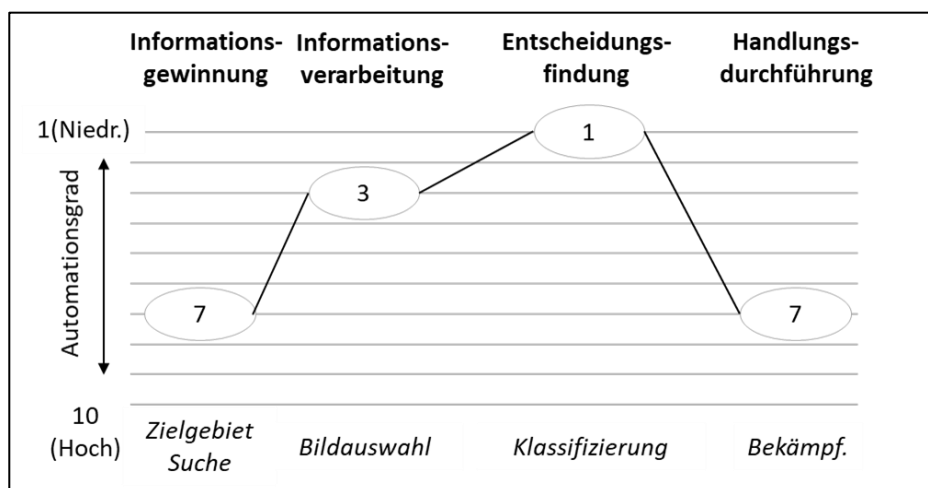


Abbildung 6. Anwendung des Interaktionsmodells im militärischen Kontext. (vgl. Parasuraman et al. (2000) sowie Legras & Coppin (2007)).

Die weitere Ausdifferenzierung der Automationsgrade anhand verschiedener Phasen der Informationsverarbeitung ermöglicht eine präzisere Beschreibung der Aufgabenteilung in Mensch-Maschine-Systemen und legt damit den Grundstein für Richtlinien zur Gestaltung der Aufgabenteilung.¹²

Eine entscheidende Limitation des Ansatzes, die Mensch-Maschine-Aufgabenteilung mithilfe von Automationsgraden zu beschreiben, ist die Prämisse, dass der Mensch die Führung über das automatisierte System übernimmt. Diese Prämisse führt zu einer

¹² Endsley & Kaber (1999) beschreiben Systemtypen durch ihre Automationsgrade in den verschiedenen Phasen der Informationsverarbeitung und identifizieren die Effekte der verschiedenen Systemtypen. Parasuraman et al. (2000) definieren Ober- und Untergrenzen von Automationsgraden in den verschiedenen Phasen.

eindeutigen Rollenverteilung zwischen Mensch und Automation: Die Automation wird für die Bearbeitung von klar definierten Aufgaben eingesetzt und der Mensch überwacht die Automation in der Durchführung dieser Aufgaben. Diese rigide Rollenverteilung wird durch innovative Automationsansätze allerdings zunehmend in Frage gestellt, da diese Ansätze auch andere Gestaltungsoptionen eröffnen, in denen die Automation eine kollaborative Rolle einnimmt (Rieth & Hagemann, 2022; Schulte & Donath, 2018). Eine dieser Gestaltungsoptionen besteht darin, dass Menschen und automatisierte Systeme kooperativ an einer Aufgabe arbeiten und ihre Aufgabenteilung flexibel an die Erfordernisse der Situation und ihre jeweiligen Fähigkeiten anpassen (Roth et al., 2019; Schmitt & Schulte, 2016). Alternativ könnte die Automation als Assistenzsystem konzipiert werden, das weitgehend im Hintergrund agiert und den Menschen bei Problemen unterstützt oder ihn auf Fehler aufmerksam macht (Brand & Schulte, 2021; Schwerd & Schulte, 2021). Um diese teamorientierten Kooperationsformen angemessen entwickeln und beschreiben zu können, sind andere Modelle notwendig, die eine flexiblere Definition der Rollenverteilung in Mensch-Maschine-Systemen erlauben.

2.2.3 Arbeitsprozess- und Arbeitssystemmethode

Die Arbeitsprozess- und Arbeitssystemmethode ist eine Systems-Engineering Methode, mit der Mensch-Maschine-Systeme beschrieben und entworfen werden können (Schulte & Donath, 2018). Hierfür wird zunächst der vom System durchzuführende Arbeitsprozess beschrieben und Anforderungen sowie Randbedingungen für das geplante Mensch-Maschine-System ermittelt. Im zweiten Schritt erfolgt die Instanziierung des zuvor definierten Arbeitsprozesses durch ein Arbeitssystem. Dies bedeutet, dass ein physikalisches System mit menschlichen und automatisierten Akteuren definiert wird, welches den Arbeitsprozess durchführt.

Arbeitsprozess und Arbeitsprozessanalyse

Ein *Arbeitsprozess* ist definiert als das „zielgerichtete Zusammenwirken von Mensch und automatisierten Systemen zur Erreichung eines gemeinsamen Arbeitsziels“ (Onken & Schulte, 2010; Schulte et al., 2016). Ein solcher Arbeitsprozess hat verschiedene Eingangs- und Ausgangsgrößen, wie in Abbildung 7 dargestellt. Diese Größen werden nachfolgend erläutert.

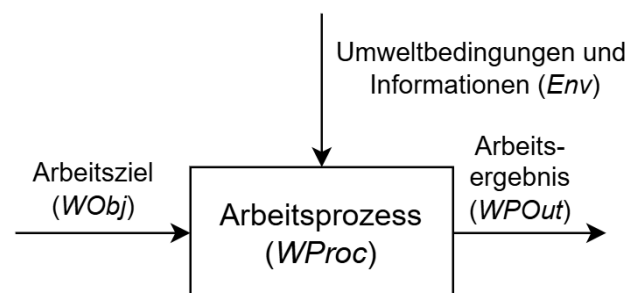


Abbildung 7. Eingangs- und Ausgangsgrößen eines Arbeitsprozesses nach Onken & Schulte (2010).

Das Arbeitsziel (W_{Obj}) definiert den Zweck des Arbeitsprozesses. Da das Handeln des Arbeitsprozesses auf die Erreichung des Arbeitsziels ausgerichtet ist, stellt dieses auch die wichtigste Eingangsgröße für den Arbeitsprozess dar. Der Arbeitsprozess ist in eine

Arbeitsumgebung eingebettet, deren Umweltbedingungen den Arbeitsprozess beeinflussen (*Env*). Durch sein Handeln erzeugt der Arbeitsprozess ein Arbeitsergebnis (WP_{Out}), mit dem er die Arbeitsumgebung verändert, also beispielsweise Aktionen an einem Arbeitsobjekt (*WO*) durchführt oder Informationen für andere Arbeitsprozesse generiert.

Im Rahmen der *Arbeitsprozessanalyse* kann diese Beschreibungssprache eines Arbeitsprozesses verwendet werden, um Anforderungen für die Entwicklung eines Mensch-Maschine-Systems abzuleiten. Hierzu wird folgende Vorgehensweise angewandt (Schulte et al., 2016):

- Festlegung des Arbeitsziels, das mit dem System erreicht werden soll.
- Identifizierung relevanter anderer Arbeitsprozesse, die mit dem System wechselwirken.
- Entwurf eines Netzwerks aus allen Arbeitsprozessen.

Abbildung 8 zeigt exemplarisch eine mögliche Netzwerkdarstellung, die aus der Anwendung dieser Vorgehensweise resultieren könnte. Diese Netzwerkdarstellung kann zur Identifizierung von Anforderungen genutzt werden, da aus dieser hervorgeht, für welche Zwecke das System eingesetzt wird, mit welchen Umgebungsfaktoren das System umgehen können muss und welches Ergebnis der Arbeitsprozess haben soll.

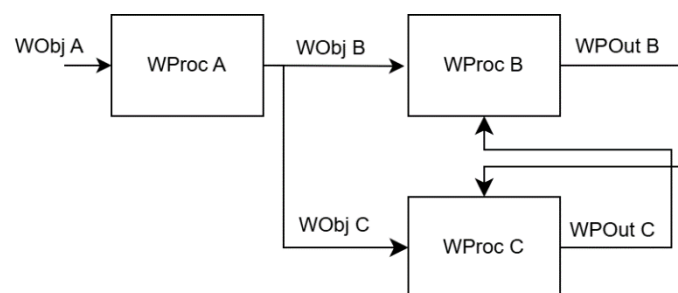


Abbildung 8. Netzwerk aus Arbeitsprozessen (vgl. Schulte & Donath (2018)).

In der Arbeitsprozessanalyse wird die beschriebene Vorgehensweise für jede Lebenszyklusphase (bzw. jeden Anwendungsfall) des Mensch-Maschine-Systems angewendet. Dies ist dadurch begründet, dass jede Phase mit individuellen Arbeitszielen, Umgebungsfaktoren und gewünschten Arbeitsergebnissen einhergehen kann, woraus sich andere Systemanforderungen ergeben können.

Arbeitssystem und Arbeitssystemdesign

Der Begriff *Arbeitssystem* bezeichnet ein Mensch-Maschine-System, das einen Arbeitsprozess durchführt. Ein solches Arbeitssystem kann mit der Modellierungssprache nach Schulte & Donath (2018) beschrieben werden.

Ein zentrales Element dieser Modellierungssprache ist die Einteilung der Akteure in zwei Rollen, die als *Worker* und *Tools* bezeichnet werden. Einen Worker zeichnet aus, dass er das Arbeitsziel kennt, versteht und eigeninitiativ verfolgt. Tools hingegen erhalten Aufgaben von einem Worker und führen diese aus, allerdings kennen sie das Arbeitsziel nicht und sie handeln auch ausschließlich in Folge von Anweisungen eines Workers.

Im Arbeitssystem werden drei verschiedene Arten von Akteuren unterschieden, welche unterschiedliche Rollen einnehmen können: *Menschliche Nutzer, konventionelle Automation und kognitive Agenten* (Abbildung 9).

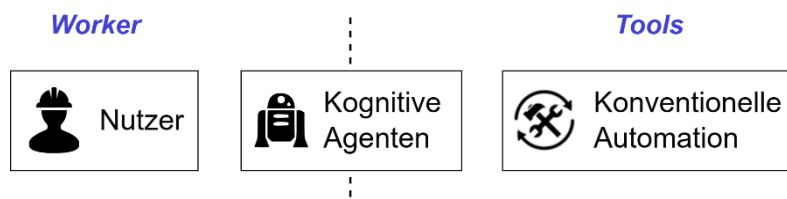




Abbildung 9. Akteure im Arbeitssystem und mögliche Rollen.

Die menschlichen Nutzer nehmen im Arbeitssystem immer die Rolle eines Workers ein, da ohne ihre Initiative das Arbeitsziel nicht erreicht werden kann. Als konventionelle Automation werden Automatisierungsfunktionen bezeichnet, die bestimmte Teilaufgaben ohne den Einsatz höherwertiger kognitiver Funktionen (z. B. Planung) bearbeiten. Da diese Form der Automation das Arbeitsziel nicht versteht und auch nicht eigenständig verfolgen kann, nimmt diese immer die Rolle eines Tools ein. Flexibler ist die Rollenverteilung bei den kognitiven Agenten. Diese Agenten sind in der Lage, höherwertige kognitive Aufgaben zu übernehmen, die bisher ausschließlich von Menschen durchgeführt werden konnten. Damit können kognitive Agenten sowohl die Rolle des Tools, als auch die des Workers einnehmen.

Ein weiteres Element der Arbeitssprache ist die Unterscheidung zwischen verschiedenen Arten von Akteurs-Beziehungen in zwei Grundarten (Tabelle 2).

Tabelle 2. Arten von Akteurs-Beziehungen nach Schulte et al. (2016).

Symbol	Bezeichnung	Bedeutung
	Hierarchische Beziehung	Ein Akteur erteilt Anweisungen an einen unterstellten Akteur.
	Heterarchische Beziehung	Die beiden Akteure arbeiten auf Augenhöhe zusammen.

Aus der Kombination der verschiedenen Rollen der Akteure und ihrer Beziehungen zueinander ergeben sich sogenannte *Akteursbeziehungs-Tupel*¹³. Abbildung 10 zeigt zwei mögliche Ausprägungen dieser Tupel.

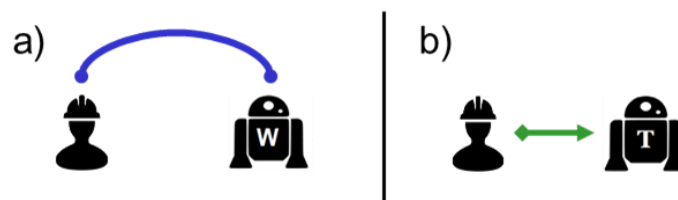


Abbildung 10. A) Heterarchische Beziehung zwischen Nutzer und Worker-Agent. B) Hierarchische Beziehung zwischen Nutzer und Tool-Agent.

¹³ Schulte et al. (2016) verwenden hierfür den englischen Begriff „Actor-Relationship-Actor Tupel“.

Weil jedes dieser Tupel charakteristische Eigenschaften im Sinne der Mensch-Maschine-Zusammenarbeit mit sich bringt, betrachten Schulte et al. (2016) die Tupel als Grundlage für die Definition von Entwurfsmustern für Mensch-Maschine-Systeme.

Zwei Beispiele für ein solche Entwurfsmuster sind das *Assistenzmuster* und das *Delegationsmuster* (Abbildung 11).

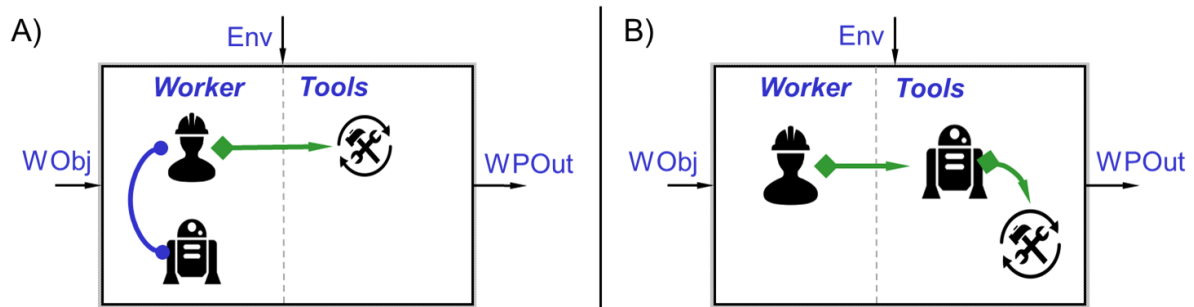


Abbildung 11. A) Assistenzmuster B) Delegationsmuster.

Das Assistenzmuster zielt darauf ab, die Performance des menschlichen Nutzers zu maximieren, indem es die Auswirkungen menschlicher Fehlentscheidungen korrigiert (Theißing & Schulte, 2016), dem Nutzer in Phasen hoher Beanspruchung Aufgaben abnimmt (Brand & Schulte, 2017) oder die Aufmerksamkeit des Nutzers lenkt (Schwerd & Schulte, 2024).

Das Delegationsmuster ist dagegen dadurch charakterisiert, dass der menschliche Nutzer einen Teil seiner Aufgaben an einen kognitiven (Tool-)Agenten abgibt, womit primär eine Reduktion der Beanspruchung oder eine Erhöhung der Führungsspanne erreicht werden kann (Schulte & Donath, 2020).





Um die Trennschärfe zwischen verschiedenen Ausprägungen einer Delegationsbeziehung zu verbessern, definiert Miller (2016) verschiedene Sub-Muster, die die Delegation anhand von sechs Eigenschaften genauer beschreiben. In dieser Arbeit wird eine dieser Eigenschaften zur weiteren Charakterisierung der Delegationsbeziehung verwendet, die sogenannte *Delegationsform*.¹⁴ Die Delegationsform beschreibt, wie der delegierende Akteur seine Anweisungen formuliert, beispielsweise durch die Vorgabe von Zielen.

Im Rahmen dieser Arbeit werden vier Delegationsformen unterschieden: Zielbasiert, Aufgabenbasiert, Handlungsbasiert und Richtlinienbasiert. Diese verschiedenen Delegationsformen sind nicht exklusiv, das heißt eine Delegationsbeziehung kann gleichzeitig verschiedene Delegationsformen beinhalten.¹⁵ Tabelle 3 erläutert die verschiedenen Delegationsformen und führt ihre symbolische Darstellung in der Arbeitssprache ein.

¹⁴ Miller verwendet für diese Eigenschaft den Begriff „Delegation Method“. In dieser Arbeit wird stattdessen der Begriff „Delegationsform“ verwendet, da „Delegationsmethode“ im weiteren Verlauf der Arbeit in einem anderen Kontext verwendet wird.

¹⁵ Wechselwirkungen, die durch die gleichzeitige Anwendung mehrerer Delegationsformen entstehen, sollten im Systementwurf berücksichtigt werden.

Tabelle 3. Beschreibung und Symbolik der Ausprägungen der Delegationsform.

Symbolik	Bezeichnung	Beschreibung
	Zielbasiert (Goal-Based)	Anweisungen erfolgen durch die Vorgabe von zu erreichenden Zuständen.
	Aufgabenbasiert (Task-Based)	Anweisungen erfolgen durch die Vergabe von Aufgaben oder Aufgabensequenzen.
	Handlungs- basiert (Action-Based)	Anweisungen erfolgen in Form von elementaren Handlungen oder Handlungsabläufen.
	Richtlinien- basiert (Policy-Based)	Anweisungen erfolgen durch die Vorgabe von Richtlinien oder Einzelwerten. Da Richtlinien nur den Lösungsraum einschränken und keine Richtung vorgeben, können diese nur ergänzend zu anderen Delegationsformen eingesetzt werden.

Ergänzend zur von Miller (2016) vorgeschlagenen Delegationsform wird in dieser Arbeit eine weitere Eigenschaft eingeführt, um eine präzisere Differenzierung verschiedener Arten der Delegationsbeziehung zu ermöglichen: der *Feedback-Modus*. Dieser beschreibt, in welcher Form der delegierende Akteur Rückmeldungen zu seinen erteilten Anweisungen erhält. Dabei lassen sich zwei Ausprägungen unterscheiden:

- Implizit: Der delegierende Akteur erhält keine direkte Rückmeldung zu seinen Anweisungen. Stattdessen muss er den Erfolg oder Misserfolg aus den Handlungen des delegierten Akteurs oder deren Auswirkungen ableiten.
- Explizit: Der delegierte Akteur gibt aktiv Rückmeldungen zu den erhaltenen Anweisungen, etwa in Form von Zustimmung, Kritik oder Verbesserungsvorschlägen.

In der Arbeitssprache werden die beiden Ausprägungen des Feedback-Modus symbolisch unterschieden, indem beim expliziten Feedback-Modus der Delegationspfeil durch einen dünnen Pfeil in Gegenrichtung ergänzt wird (Abbildung 12).

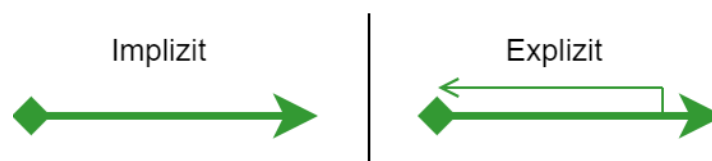


Abbildung 12. Symbolische Unterscheidung der beiden Ausprägungen des Feedback-Modus.

2.3 Mensch-Maschine-Interaktion

Neben einer angemessenen Aufgabenteilung erfordert die Zusammenarbeit zwischen Mensch und Automation auch effektive Kommunikationsmittel. Aus diesem Grund widmet sich dieser Abschnitt der Mensch-Maschine-Interaktion.

Zunächst werden in [2.3.1](#) die **Interaktionsmodalitäten** vorgestellt, die verschiedene Arten der Informationsübertragung beschreiben. Anschließend werden in [2.3.2](#) konkrete **Interaktionsmethoden für Kampfflugzeugcockpits** präsentiert, die zeigen, wie der Informationsaustausch in diesem speziellen Kontext umgesetzt werden kann. Abschließend wird in [2.3.3](#) beschrieben, welche **Arten von Feedback** für eine effektive Zusammenarbeit zwischen Mensch und Automation notwendig sind.

2.3.1 Interaktionsmodalitäten

Die Interaktionsmodalitäten beschreiben verschiedene Wege der Kommunikation zwischen einem Nutzer und einem automatisierten System. Aus Nutzersicht umfassen diese Modalitäten sowohl die *Sinnesmodalitäten*, mit denen der Nutzer Informationen vom System wahrnehmen kann, als auch *Aktionsmodalitäten*, mit denen der Nutzer Informationen an das System übermitteln kann (Abbildung 13).

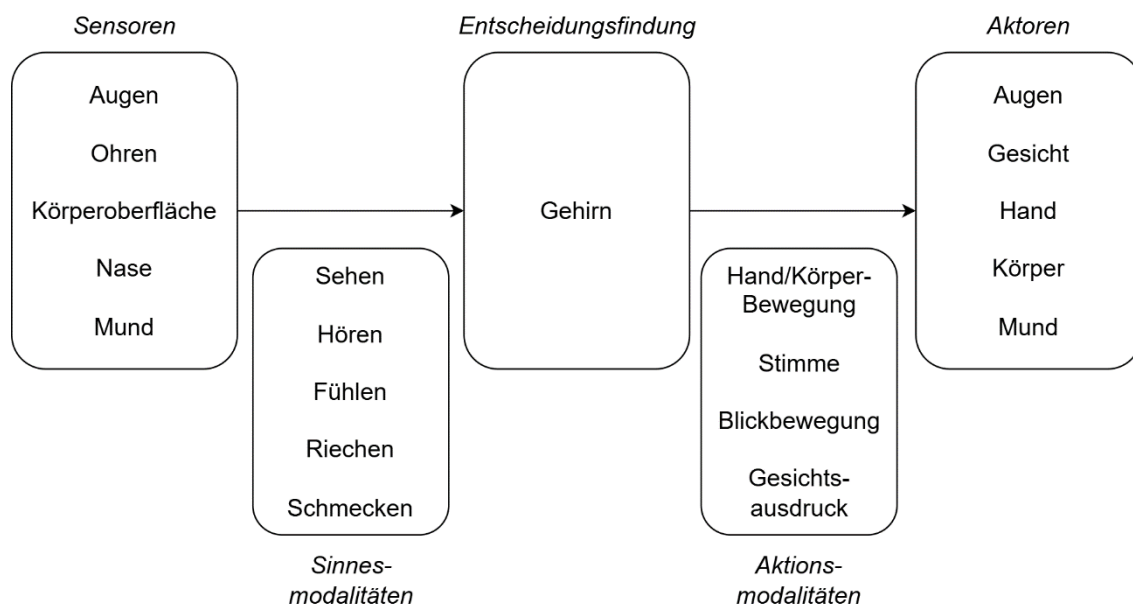


Abbildung 13. Interaktionsmodalitäten des Menschen (vgl. Sharma et al. (1998)).

2.3.1.1 Sinnesmodalitäten

Der Mensch verfügt über fünf verschiedene Sinnesmodalitäten - Sehen, Hören, Tasten, Riechen und Schmecken - die im Kontext der Mensch-Maschine-Interaktion einen unterschiedlichen Stellenwert einnehmen (Dix, 2004; Sharma et al., 1998). In der Mensch-Maschine-Interaktion sind Sehen und Hören die wichtigsten Modalitäten, da die meisten Schnittstellen visuelles und/oder auditives Feedback zur Informationsübertragung nutzen (Akamatsu et al., 1995; Oviatt et al., 2017, Kapitel 7). Der Tastsinn spielt in der Mensch-Maschine-Interaktion meist nur eine untergeordnete Rolle, allerdings nimmt dessen

Bedeutung zur Informationsübertragung beispielsweise durch Vibrationsfeedback von Touchscreens oder Force-Feedback von Steuerelementen zu (Artigas et al., 2016; Poupyrev & Maruyama, 2003). Geruchs- und Geschmackssinn werden in der Mensch-Maschine-Interaktion bislang nur in einzelnen Forschungsanwendungen eingesetzt, was insbesondere an der aufwändigen technischen Umsetzung liegen dürfte (Neves & Câmara, 2020; Obrist et al., 2016).

2.3.1.2 Aktionsmodalitäten

Während die Sinnesmodalitäten stark von den menschlichen Wahrnehmungsfähigkeiten geprägt sind, hängen die Aktionsmodalitäten sowohl von der Fähigkeit des Menschen ab, seine Absichten über diese Modalität auszudrücken, als auch von der Fähigkeit eines Systems, die daraus resultierenden Aktionen des Menschen zu erkennen (Sharma et al., 1998).

In der Mensch-Maschine-Interaktion ist die manuelle Modalität die am meisten verbreitete Aktionsmodalität. Diese Modalität findet überwiegend in der Form von händischen Eingaben Anwendung, beispielsweise in der Bedienung von Tastatur, Maus, Trackballs oder Touchscreens (Akamatsu et al., 1995; Alden et al., 1972). Die manuelle Aktionsmodalität umfasst auch Gesten und Körperbewegungen (Schelle & Stütz, 2018) sowie Eingaben mit den Füßen (Pearson & Weiser, 1986).

Eine weitere wesentliche Aktionsmodalität ist die Stimme. Diese Modalität wird insbesondere in Form einer Spracheingabe verwendet, bei der der Nutzer durch das Sprechen von Wörtern oder Sätzen mit einem System interagiert (Cohen & Oviatt, 1995). Vereinzelt wird die Stimme auch in anderen Formen zur Interaktion genutzt, z. B. indem die Frequenz der Stimmlage zur kontinuierlichen Steuerung einer Größe verwendet wird (Peixoto et al., 2013).

Die Blickbewegungsmodalität, die bisher vor allem zur Verhaltensbeobachtung eingesetzt wurde, gewinnt als Eingabemethode zunehmend an Bedeutung. Die Blickbewegung kann sowohl als aktive als auch als passive Modalität in einen Interaktionsprozess integriert werden. Wird die Blickbewegungsmessung als aktive Modalität verwendet, kann der Nutzer durch das Manipulieren der Blickrichtung eine Eingabe in das System tätigen.¹⁶ Da die Augen jedoch auch unbewusst im Kontext der Wahrnehmung eingesetzt werden, wird die Blickbewegung als aktive Modalität in der Regel mit anderen Modalitäten kombiniert (Mardenbegi & Qvarfordt, 2015; Schweigert et al., 2019). Wenn die Blickbewegung als passive Modalität verwendet wird, kann (bzw. sollte) der Nutzer den Blick nicht direkt zur Eingabe verwenden. Stattdessen wird die Blickrichtung genutzt, um zu ermitteln, mit welchen Inhalten sich der Nutzer gerade beschäftigt, um z. B. gezieltes Feedback zu generieren (Buscher et al., 2009) oder die Informationsdarstellung anzupassen (Itti, 2004).

Der Gesichtsausdruck/Die Mimik spielt als Aktionsmodalität im Kontext der Mensch-Maschine-Interaktion bislang eine eher untergeordnete Rolle. Die Mimik wird in vielen Fällen als passive Eingabemodalität eingesetzt, z. B. um die Stimmungslage des Nutzers zu erkennen und auf dieser Basis das Verhalten des Systems zu verändern (Abdat et al., 2011; Hossain et

¹⁶ z. B. zur Selektion von Objekten auf einem Display (Zhai et al., 1999).

al., 2015). Andere Systeme nutzen die Mimik als aktive Eingabemodalität, indem sie bestimmte Gesichtsgesten, wie z.B. die Bewegung der Mundwinkel, mit Systembefehlen verknüpfen (Lyons, 2004; Rabhi et al., 2018).

Eine neuartige Aktionsmodalität könnte durch die Forschung zu *Brain-Computer-Interfaces* (Gehirn-Computer-Schnittstellen) entstehen. Diese Schnittstellen zielen darauf ab, die Gehirnaktivität als Eingabemethode für automatisierte Systeme zu erschließen. Hierzu messen Brain-Computer-Interfaces die neuronalen Aktivitäten des Gehirns mithilfe von *Elektroenzephalografie*, *Nahinfrarotspektroskopie* oder anderen nicht-invasiven Verfahren und analysieren diese auf Muster. Während die Bedeutung der meisten Gehirnsignale noch nicht ausreichend verstanden ist, wurde die Bedeutung einiger Signale so weit entschlüsselt, dass Menschen lernen können, diese Signale zu beeinflussen, um damit Absichten an ein Brain-Computer-Interface zu übermitteln (Nicolas-Alonso & Gomez-Gil, 2012). Als Aktionsmodalität im Kontext der Mensch-Maschine-Interaktion haben Brain-Computer-Interfaces bislang eine sehr geringe Bedeutung, was insbesondere an der geringen Genauigkeit nicht-invasiver Verfahren liegen dürfte (Zabcikova et al., 2022).

2.3.1.3 Wechselwirkungen zwischen Modalitäten

Bei der Entwicklung eines Interaktionskonzeptes für ein Mensch-Maschine-System ist zu beachten, dass die Eingabe- und Ausgabemodalitäten nicht isoliert betrachtet werden können (Markovich et al., 2020). Vielmehr zeigen verschiedene kognitive Modelle, dass die Eingabe- und Ausgabemodalitäten einer Aufgabe eng miteinander verknüpft sind und mit den Modalitäten parallel ablaufender Aufgaben in Wechselwirkung stehen können. Im Folgenden werden zwei etablierte Modelle vorgestellt, die diese Zusammenhänge näher erläutern.

Das Modell der *SCR-Compatibility*¹⁷ besagt, dass eine besondere Kompatibilität zwischen Eingabemodalitäten, Ausgabemodalitäten und verschiedenen Aufgabentypen (bzw. deren Anforderungen an die kognitive Informationsverarbeitung) besteht (Wickens et al., 1981). Das Modell unterscheidet die Eingabemodalitäten visuell und auditiv, die Ausgabemodalitäten manuell und verbal und die Aufgabentypen räumlich und verbal. Eine Aufgabe wird im Modell als räumlich definiert, wenn sie die Orientierung im dreidimensionalen Raum erfordert, während eine Aufgabe als verbal definiert wird, wenn sie die Verwendung von Sprache oder einer anderen Form von symbolischer Kodierung erfordert.¹⁸ Dem Modell zufolge führt eine visuell-manuelle Modalitätenpaarung zu einer höheren Leistung bei Aufgaben, die eine räumliche Informationsverarbeitung erfordern, während eine auditiv-verbale Modalitätenpaarung zu einer höheren Leistung bei Aufgaben führt, die eine verbale Informationsverarbeitung erfordern.

¹⁷ Abkürzung für **Stimulus-Central-Processing-Response** Compatibility.

¹⁸ Nicht jeder Aufgabentyp kann strikt als verbal oder räumlich beschrieben werden. Die beiden Kategorien sind daher als Endpunkte eines kontinuierlichen Spektrums zu verstehen, in das verschiedene Aufgaben eingeordnet werden können.

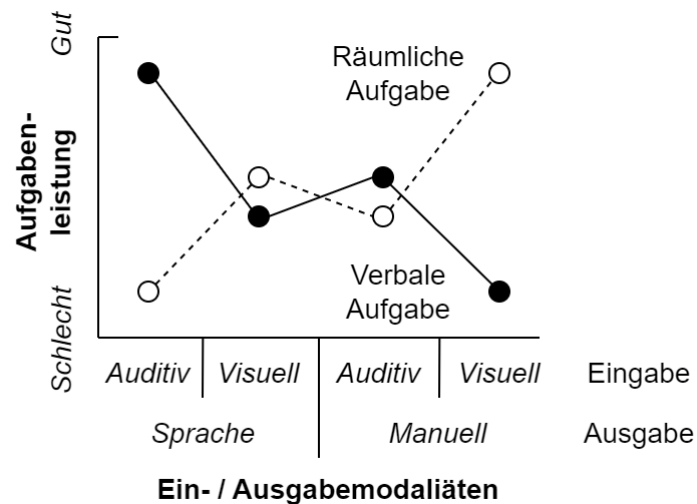


Abbildung 14. Vorhersage der Aufgabenleistung einer Einzelaufgabe in Abhängigkeit von Ein- und Ausgabemodalität nach dem SCR-Kompatibilitätsmodell (Vidulich & Wickens, 1985).

Eine Schlussfolgerung aus dem Modell ist, dass Aufgaben, die in erster Linie auf der Verarbeitung räumlicher Informationen basieren, mit visuell-manueller Interaktion konzipiert werden sollten, während Aufgaben, die in erster Linie die Verarbeitung verbaler Informationen erfordern, mit auditiv-verbaler Interaktion umgesetzt werden sollten (Vidulich & Wickens, 1985).

Die *Multiple-Resource-Theory (MRT)* besagt, dass Menschen nur eine begrenzte Menge an Ressourcen für mentale Prozesse zur Verfügung haben und dass sich diese Ressourcen über verschiedene Kanäle bzw. Modalitäten erstrecken, die relativ unabhängig voneinander genutzt werden können (Basil, 2012; Wickens, 2002). Eine auf dieser Theorie aufbauende Beschreibung verschiedener Ressourcenkanäle bzw. Modalitäten liefert das *Multiple Resource Model* nach Wickens (1984, 2024). Die aktuellste Version dieses Modells ist in Abbildung 15 dargestellt.

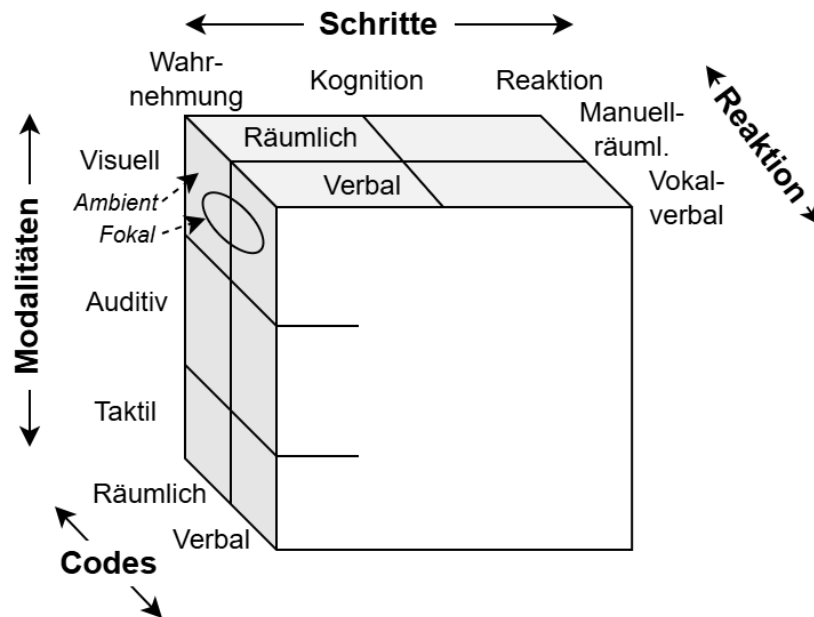


Abbildung 15. 4-Dimensionales Ressourcenmodell nach Wickens (2024).

Das Modell unterscheidet die verschiedenen Möglichkeiten der Wahrnehmung und Verarbeitung von Informationen anhand von vier Dimensionen:

- Wahrnehmungsmodalitäten: Visuell, auditiv und taktil.
- Codes der Informationsverarbeitung: Räumlich und verbal
- Schritte der Informationsverarbeitung: Wahrnehmung, Kognition und Reaktion
- Reaktionsmodalitäten: Manuell-räumlich und vokal-verbal

Basierend auf dieser Definition der Dimensionen besagt das Modell, dass zwei gleichzeitig ausgeführte Aufgaben stärker miteinander interferieren, wenn beide Aufgaben dieselben Ressourcen entlang einer der Modelldimensionen beanspruchen. Dies impliziert, dass in Multitasking-Umgebungen die Interaktionsmodalitäten möglichst so gewählt werden sollten, dass die einzelnen Aufgaben möglichst unterschiedliche Ressourcenkanäle belasten. Ein Beispiel wäre: Wenn eine Aufgabe bereits die visuell-manuelle Modalitätenpaarung nutzt, sollte eine zweite Aufgabe eher mit der auditiv-verbale Modalitätenpaarung realisiert werden.

Auch wenn beide Modelle nur bestimmte Ein- und Ausgabemodalitäten¹⁹ berücksichtigen, können die in den Modellen präsentierten Effekte einen wichtigen Anhaltspunkt für die Auswahl von der individuellen Modalitäten sowie der Modalitätenpaarungen bieten.

¹⁹ Beispielsweise berücksichtigt keines der Modelle die Blickbewegung als Ausgabemodalität.

2.3.2 Interaktionsmethoden für Kampfflugzeuge

In diesem Unterabschnitt werden Interaktionsmethoden vorgestellt, die bereits in Cockpits von Kampfflugzeugen eingesetzt werden oder die in Forschungsprojekten für diesen Einsatz untersucht wurden.

Im Rahmen dieser Arbeit werden Interaktionsmethoden als spezifische Verfahren zur Bedienung eines automatisierten Systems definiert.²⁰ Die Interaktionsmethoden umfassen sowohl Eingabemethoden, mit denen der Nutzer Informationen an das System übermitteln kann, als auch Ausgabemethoden, mit denen das System Informationen an den Nutzer zurückmelden kann.

2.3.2.1 Eingabemethoden

In Kampfflugzeugen gehören physische Tasten und Schalter zu den am meisten verbreiteten Eingabemethoden. So finden sich insbesondere in älteren Kampfflugzeugmustern eine Vielzahl von mechanischen Bedienelementen, die entweder direkt mit bestimmten Systemfunktionen verknüpft sind (z.B. Fahrwerkshebel) oder die abhängig vom Interaktionskontext unterschiedliche Funktionen übernehmen. Diese mechanischen Bedienelemente sind oftmals an Steuerknüppel und Schubhebel angebracht, um die Bedienung während des Fluges zu ermöglichen (Abbildung 16 A). Alternativ werden sie häufig am Rand von Bildschirmen angebracht, um die Zuordnung zu bestimmten Menüoptionen auf dem Bildschirm zu erleichtern (Abbildung 16 B).

Eine weitere verbreitete Eingabemethode zur Systembedienung in Kampfflugzeugcockpits sind *Cursor Control Devices (CCD)*. Hierbei handelt es sich um Eingabegeräte zur präzisen Bewegung eines Cursors auf einem Bildschirm. Im Kampfflugzeugcockpit werden insbesondere vier Arten von CCD eingesetzt: Thumbsticks, Fingersticks, Trackpads und Trackballs (Thomas, 2018; Thomas et al., 2015). Die CCDs werden in der Regel auf oder in der Nähe von Steuerelementen der Flugsteuerung platziert, um eine Systembedienung während des Fliegens zu ermöglichen. Abbildung 16 A zeigt den Fingerstick des Eurofighters, der in den Schubhebel integriert ist.

Viele Kampfflugzeuge der fünften Generation verfügen zudem über eine sprachbasierte Eingabemethode, die in diesem Kontext auch *Direct Voice Input (DVI)* genannt wird. Im Eurofighter Typhoon kann das DVI-System für verschiedene Zwecke genutzt werden, z. B. zur Konfiguration von Menüs, zur Auswahl von Zielen oder zur Eingabe von alphanumerischen Daten (Hartley, 1999).

In modernen Kampfflugzeugen wie der F-35 wurden die mechanischen Bedienelemente durch den Einsatz von Touchscreens (berührungsempfindliche Bildschirme) weitgehend verdrängt (Abbildung 16 C). Die Integration von Touchscreens ermöglicht eine schnelle und intuitive Bedienung des Systems durch den Nutzer, da dieser direkt mit den Bildelementen interagieren kann. Die Bedienung eines Touchscreens während des

²⁰ Während also die Modalitäten die Interaktionskanäle zwischen Mensch und Automation beschreiben, beschreiben die Interaktionsmethoden die konkrete Art und Weise, wie diese Interaktionskanäle im Rahmen der Systembedienung genutzt werden

Fluges kann jedoch eine Herausforderung darstellen, insbesondere wenn G-Kräfte und Vibrationen auftreten. (Thomas et al., 2015; Zwerger, 2021)

Die Blickbewegungsmessung (Eye-Tracking) rückt als Eingabemethode für Kampfflugzeugcockpits zunehmend in den Fokus. So wird die Blickbewegungsmessung in verschiedenen Forschungsarbeiten untersucht, um Aufgaben durchzuführen die ein Auswählen von Objekten erfordern, z. B. als neuartige Variante eines Cursor Control Devices (Hartmann, 2023; Murthy et al., 2020; Shree Dv et al., 2018) oder zur Zielauswahl in Head-up Displays oder Helmet-Mounted Displays (in den Helm integrierte Displays; O’Connell et al. (2012)). Die Blickbewegungsmessung könnte eine schnelle und intuitive Form der Eingabe für Auswahlaufgaben darstellen, allerdings sind die derzeitigen Messsysteme unter bestimmten Bedingungen noch zu ungenau, um einen zuverlässigen Einsatz im gesamten Anwendungsbereich zu gewährleisten (Thomas et al., 2015).

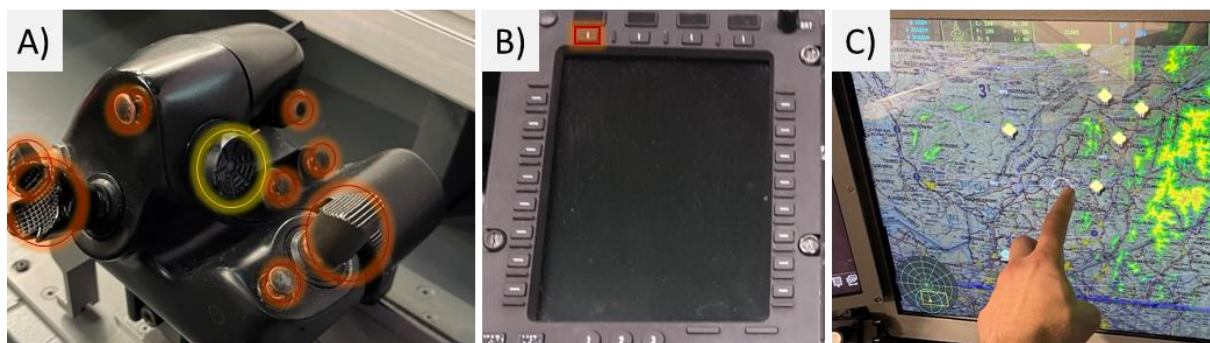


Abbildung 16. Verschiedene Eingabeelemente für Kampfflugzeugcockpits.

A) Bedienelemente am Eurofighter-Schubhebel (Rot: Mechanische Tasten und Schalter; Gelb: Fingerstick als CCD)

B) Mechanische Tasten zur Optionsauswahl am Display-Rand

C) Berührungsempfindliches Display

2.3.2.2 Ausgabemethoden

In modernen Kampfflugzeugcockpits sind *Multi-Function Displays (MFDs)* eine weitverbreitete Ausgabemethode. Ein MFD ist ein Bildschirm im Cockpit, der eine flexible Darstellung von Informationen wie taktische Lage, Systemstatus oder Radarinformationen ermöglicht. Dabei kann die Anzeige von Informationen in der Regel sowohl vom Piloten als auch vom System angepasst werden, je nachdem, welche Informationen im Kontext der Mission zu einem bestimmten Zeitpunkt relevant sind. In vielen Kampfflugzeugmustern werden MFDs mit Knöpfen am Displayrand oder mithilfe eines CCDs bedient, in neueren Mustern werden MFDs auch oft mit Touchscreen-Eingabe kombiniert.

Neben den MFDs kommen in modernen Kampfflugzeugcockpits zwei weitere visuelle Ausgabemethoden zum Einsatz: *Head-up-Displays (HUD)* und *Helmet-Mounted Displays (HMD)*. Diese Methoden nutzen transparente Displays, die in den Helm oder das Cockpit integriert sind, um dem Nutzer Informationen direkt in das Sichtfeld zu projizieren (Jarrett, 2017). Diese Form der Informationsdarstellung hat den Vorteil, dass der Nutzer wichtige Informationen wie Geschwindigkeit und Höhe ablesen und gleichzeitig peripher weitere Informationen wahrnehmen kann. Allerdings ist die Menge der darstellbaren Informationen im Vergleich zu MFDs begrenzt, weshalb die Art der angezeigten Informationen in der Regel kontextabhängig angepasst werden kann.

Akustische Ausgabemethoden sind ebenfalls ein wichtiges Element zur Informationsübermittlung im Kampfflugzeugcockpit. Audiomeldungen werden im Kampfflugzeugcockpit insbesondere in Situationen eingesetzt, die eine sofortige Reaktion des Nutzers erfordern. So werden beispielsweise verschiedene Warntöne verwendet, um Piloten auf unerwartete Ereignisse oder Gefahren aufmerksam zu machen. Warnungen werden auch in Form von standardisierten Sprachnachrichten übermittelt, insbesondere wenn eine präzise Zustandsbeschreibung benötigt wird oder Maßnahmen vorgeschlagen werden sollen (Doll & Folds, 1986; U.S. Department of Defense, 2012).

Haptische Ausgabemethoden²¹ nehmen im Kampfflugzeugcockpit bislang eine eher untergeordnete Rolle ein. In militärischen wie zivilen Luftfahrzeugen wird haptisches Feedback beispielsweise in Form von aktiven Steuerelementen eingesetzt, die den Nutzer mithilfe von Kräften und Vibrationen über Grenzen der Flugenveloppe oder Zustandswechsel des Autopiloten informieren (Gable, 2024; Schmidt-Skipiol & Hecker, 2015; Van Baelen et al., 2021). Im Rahmen von Forschungsarbeiten wurden zudem verschiedene vibro-taktile Ausgabemethoden untersucht, die in erster Linie dazu dienen, räumliche Informationen wie die Orientierung des eigenen Luftfahrzeugs oder die Richtung von Bedrohungen und Zielen zu vermitteln (Van Erp & Self, 2008).

²¹ Der Begriff Haptik umfasst sowohl die passive Wahrnehmung von Druck- und Berührungsempfindungen (Taktilität), als auch die aktive Wahrnehmung in Folge von berührenden Handlungen (Sprenger, 2020).

2.3.3 Feedback-Taxonomie

Das Feedback des automatisierten Systems spielt im Kontext der Mensch-Maschine-Interaktion eine entscheidende Rolle, da es dem Nutzer ein Verständnis von Situation und Systemverhalten vermittelt und ihm ermöglicht, seine Interaktion mit dem System gezielt anzupassen.

Obwohl Inhalt und Form des Feedbacks in der Regel vom Zweck des Mensch-Maschine-Systems und der Rolle des Menschen darin abhängen, lassen sich anwendungsübergreifende Muster erkennen, welche Arten von Feedback integriert werden und welche Auswirkungen davon erwartet werden. Im Folgenden wird daher eine Kategorisierung von Feedback vorgestellt, die verschiedene Arten von Feedback differenziert und deren erwartete Auswirkungen beschreibt. Diese Kategorisierung baut auf der Beschreibung von Feedback-Charakteristika nach Renaud & Cooper (2000) auf und überträgt diese Prinzipien in den Anwendungskontext der UAV-Führung.

Die erste Feedback-Kategorie ist die *Reaktion auf Nutzereingaben*. Ziel dieses Feedbacks ist es, dem Nutzer die Auswirkungen seiner Interaktionen zu verdeutlichen, um ihn in die Lage zu versetzen, die Interaktion entsprechend seinen Absichten zu steuern (Norman, 2013, S. 207). Dieses Feedback kann sich entweder auf die Interaktion selbst beziehen oder auf die Auswirkungen von Anweisungen, die durch die Nutzerinteraktion vermittelt werden. Beispiele hierfür sind im Anwendungskontext:

- Interaktionsbezogenes Feedback: Reaktion eines Touchscreens auf Berührung, Darstellung des Ergebnisses einer Spracherkennung.
- Anweisungsbezogenes Feedback: Verhalten der unbemannten Plattformen in Folge von Anweisungen, Vorhersage von Effekten von Anweisungen.

Eine weitere Feedback-Kategorie ist die *Information über relevante Umwelteinflüsse*. Dieses Feedback hat den Zweck, den Nutzer in Folge von Ereignissen zum Handeln anzuregen und ihm die richtigen Informationen bereitzustellen, damit diese Handlungen möglichst schnell und effektiv umgesetzt werden können. Im Anwendungskontext könnte ein Beispiel für eine Rückmeldung dieser Kategorie eine Meldung sein, in der erläutert wird, dass eine Bedrohung entdeckt wurde und wie sich diese Bedrohung auf die Mission auswirken könnte.

Die dritte Feedback-Kategorie ist die *Information über den Status der Automation*. Dieses Feedback informiert den Nutzer über die Aktionen und Zustände des automatisierten Systems. Diese Form des Feedbacks zielt einerseits darauf ab, das Situationsbewusstsein des Nutzers über das automatisierte System zu erhöhen und dient andererseits dazu, das Vertrauen des Nutzers in das automatisierte System zu kalibrieren. In der Literatur werden Feedback-Maßnahmen, die unter diese Kategorie fallen, auch unter dem Begriff *Transparency* (Transparenz) untersucht, weil die Maßnahmen zur Steigerung der Systemtransparenz beitragen (Chen et al., 2016; Endsley, 2017). Im Kontext der UAV-Führung kann diese Form des Feedbacks beispielsweise in Form einer Selbsterklärung des Vorgehens der unbemannten Systeme umgesetzt werden.

2.4 Agentenarchitekturen

Im operationellen Konzept des Manned-Unmanned Teaming übernehmen die unbemannten Plattformen verschiedenste Missionsaufgaben, die in der Vergangenheit ausschließlich von bemannten Kampfflugzeugen durchgeführt wurden. Um diese Missionsaufgaben durchführen zu können, müssen die unbemannten Plattformen zu einem eigenständigen Handeln in einer komplexen und dynamischen Umgebung befähigt werden.

Die Fähigkeit eines UAV, in komplexen Missionen eigenständig zu agieren, weist deutliche Parallelen zu den Eigenschaften auf, die in der Literatur als charakteristisch für sogenannte autonome²² Agenten beschrieben werden. Diese Gemeinsamkeiten legen nahe, dass Architekturen, die für autonome Agenten konzipiert wurden, auch bei UAVs eingesetzt werden können. Daher wird im Folgenden zunächst eine Definition eines autonomen Agenten vorgestellt, bevor verschiedene Architekturen zur Umsetzung solcher Systeme erläutert werden.

„Ein autonomer Agent ist ein computergestütztes System, das mehrere Ziele hat und in einer unstrukturierten und dynamischen Umgebung völlig eigenständig agiert. Er versucht, seine Ziele zu erreichen, indem er mit Hilfe von Sensoren und Effektoren mit seiner Umgebung interagiert“

Definition eines autonomen Agenten nach Maes (1991).

In der Literatur werden verschiedene Architekturen zur Umsetzung eines autonomen Agenten beschrieben. Diese Architekturen lassen sich anhand der Art der Verhaltensgenerierung in drei Kategorien einteilen: *Deliberativ*, *Reaktiv* und *Hybrid* (Wooldridge & Jennings, 1995).

Deliberative Architekturen zeichnen sich dadurch aus, dass sie über eine interne Zustandsrepräsentation verfügen und Entscheidungen unter Nutzung von Logik treffen, z. B. mittels Inferenz oder Planung. Die Idee dieses Ansatzes ist, dass intelligentes Verhalten durch die Anwendung einer Reihe von logischen Regeln auf den Modellzustand erzeugt werden kann. Ein verbreiteter Ansatz zur Umsetzung eines deliberativen Agenten ist die *Intelligent Resource-bounded Machine Architecture (IRMA)* von Bratman et al. (1988). Diese Architektur verfügt über eine explizite Repräsentation von Annahmen, Zielen und Absichten (Beliefs, Desires & Intentions) und verwendet einen mehrstufigen Prozess zur Generierung und Ausführung von Handlungsplänen aus diesen symbolischen Datenstrukturen (Shen, 2019). Während die Idee, den Zustand abzubilden und Regeln auf ihm anzuwenden, unter verschiedenen Gesichtspunkten attraktiv ist, haben sich rein deliberative Ansätze kaum durchgesetzt, weil die Modellierung realer Probleme zu aufwändig ist oder die Lösungsfindung zu lange dauert (Wooldridge & Jennings, 1995).

Reaktive Architekturen verwenden keine symbolische Zustandsrepräsentation und generieren ihr Verhalten ausschließlich auf der Basis von Sensorinformationen. Die Idee

²² Der Begriff „Autonomie“ ist im Kontext der Mensch-Maschine-Interaktion umstritten, da er sowohl für Eigenständigkeit als auch für Willensfreiheit steht. Da „autonomer Agent“ jedoch ein etablierter Begriff für eigenständige handelnde bzw. robotische Agenten ist, wird dieser Begriff hier verwendet.

hinter diesem Ansatz ist, dass intelligentes Verhalten eine emergente Eigenschaft ist, die auch ohne eine symbolische Modellierung erreicht werden kann (Brooks, 1991). Die einfachste Form einer reaktiven Architektur ist der *Simple Reflex Agent*, der seine Aktionen alleine anhand des wahrgenommenen Zustands und strikter Zustands-Aktions-Regeln auswählt (Russell & Norvig, 2016). Ein umfassenderer Ansatz zur Umsetzung eines reaktiven Agenten ist die *Subsumption Architecture* zur Kontrolle von Robotern (Brooks, 1986). Diese Architektur besteht aus hierarchisch angeordneten Modulen, deren Prozesse parallel ablaufen und die jeweils ein bestimmtes Verhalten erzeugen (z. B. einem Objekt ausweichen). Hierarchisch übergeordnete Prozesse können die Ausgabengrößen der untergeordneten Prozesse gezielt für eigene Zwecke verändern, wodurch die verschiedenen Prozesse in geordneter Form um die Kontrolle des Roboters konkurrieren. Während reaktive Agenten durch Architekturen wie die *Subsumption Architecture* dazu befähigt werden können, „intelligente“ Verhaltensweisen zu generieren, verfügen sie nicht über die Fähigkeit, zielorientiertes Verhalten außerhalb explizit kodierter Prozeduren zu erzeugen (Shen, 2019).

Hybride Architekturen integrieren den deliberativen und den reaktiven Ansatz in einer gemeinsamen Architektur. Die Idee der hybriden Architektur zielt darauf ab, die zielorientierten Planungsfähigkeiten einer deliberativen Architektur mit der schnellen Entscheidungsfindung einer reaktiven Architektur zu vereinen (Wooldridge & Jennings, 1995). Ein Ansatz zur Realisierung einer solchen hybriden Architektur besteht in der Integration von deliberativen und reaktiven Subsystemen in einer gemeinsamen Architektur. Abbildung 17 zeigt ein Beispiel für einen solchen Ansatz (Dennis & Fisher, 2023).

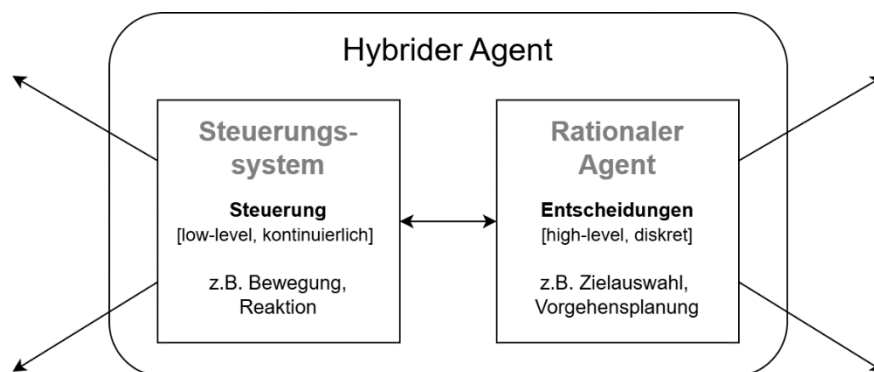


Abbildung 17. Möglicher hybrider Agentenaufbau (vgl. Dennis & Fischer (2023))

In der dargestellten Architektur legt das deliberative Subsystem die Ziele fest und plant das Vorgehen zur Zielerreichung, während das reaktive Subsystem den Vorgehensplan umsetzt und den Agenten entsprechend steuert. Im Gegensatz zu diesem Beispiel ist der Ansatz, eine hybride Architektur aus deliberativen und reaktiven Subsystemen aufzubauen, nicht auf zwei Subsysteme beschränkt. Stattdessen kann die hybride Architektur noch eine Vielzahl weiterer Komponenten enthalten (z. B. zur Pfadplanung oder Bilderkennung). Die hybride Architektur ermöglicht es somit, verschiedenste deliberative und reaktive Verarbeitungsmethoden zum Einsatz zu bringen (Dennis & Fisher, 2023).

3. Konzept assistierte auftragsbasierte Führung

Dieses Kapitel befasst sich mit der Entwicklung des Führungs- und Interaktionskonzeptes. In Abschnitt 3.1 werden zunächst die **Anforderungen an das System** ermittelt, die sich aus den Rahmenbedingungen des Manned-Unmanned-Teaming ableiten. Basierend auf diesen Anforderungen wird in Abschnitt 3.2 das **Mensch-Maschine-System entworfen**. Die darauffolgenden Abschnitte nehmen verschiedene Aspekte des entworfenen Mensch-Maschine-Systems genauer in den Fokus. In Abschnitt 3.3 werden die **Aufträge und deren Delegation** als zentrales Element des Mensch-Maschine-Systems vorgestellt. Die Abschnitte 3.4 und 3.5 erläutern zwei weitere Kernelemente des Konzeptes: den **Delegationsagenten** und den **Plattformsteuerungsagenten**.

3.1 Definition der Anforderungen

In diesem Abschnitt werden die Anforderungen an den Arbeitsplatz eines militärischen Flight Leads identifiziert. Dazu wird die Aufgabe des Flight Lead als Arbeitsprozess beschrieben und anschließend werden mithilfe einer Arbeitsprozessanalyse die Anforderungen an den Prozess systematisch erfasst (siehe 2.2.3).

Im Rahmen der Arbeitsprozessanalyse wird für jeden Anwendungsfall ermittelt, wie der betrachtete Arbeitsprozess mit anderen Prozessen interagiert und welche Ziele, Umweltfaktoren und Arbeitsergebnisse daraus resultieren. Im Falle des militärischen Flight Leads ergeben sich die Anwendungsfälle des Systems aus den Missionen bzw. deren Missionsphasen. Deshalb wird die Arbeitsprozessanalyse im Folgenden anhand einer Beispielmission durchgeführt. Anschließend werden die Anforderungen präsentiert, die durch die wiederholte Anwendung der Methode für verschiedene Missionen ermittelt wurden.

3.1.1 Anwendung der Arbeitsprozessanalyse

Als Beispielmission für die Durchführung der Arbeitsprozessanalyse dient im Folgenden die Missionsvignette C, die die Bekämpfung eines Kommandopostens in einem symmetrischen Konflikt zum Ziel hat (siehe 2.1.3). Die Durchführung einer solchen Mission erfordert den Einsatz mehrerer Flight Packages, die jeweils mehrere unbemannte Plattformen umfassen. Der Mission Commander ist dabei als zentrale Instanz für die Koordination zuständig, indem er Unteraufgaben an die Packages verteilt. Der zugehörige Arbeitsprozess des Mission Commanders wird als *Command and Control Operation* bezeichnet. Betrachtet man in diesem Kontext den Arbeitsprozess eines Flight Packages mit der Rolle *Air Superiority*, könnte das Netzwerk aus Arbeitsprozessen wie folgt aussehen (Abbildung 18).

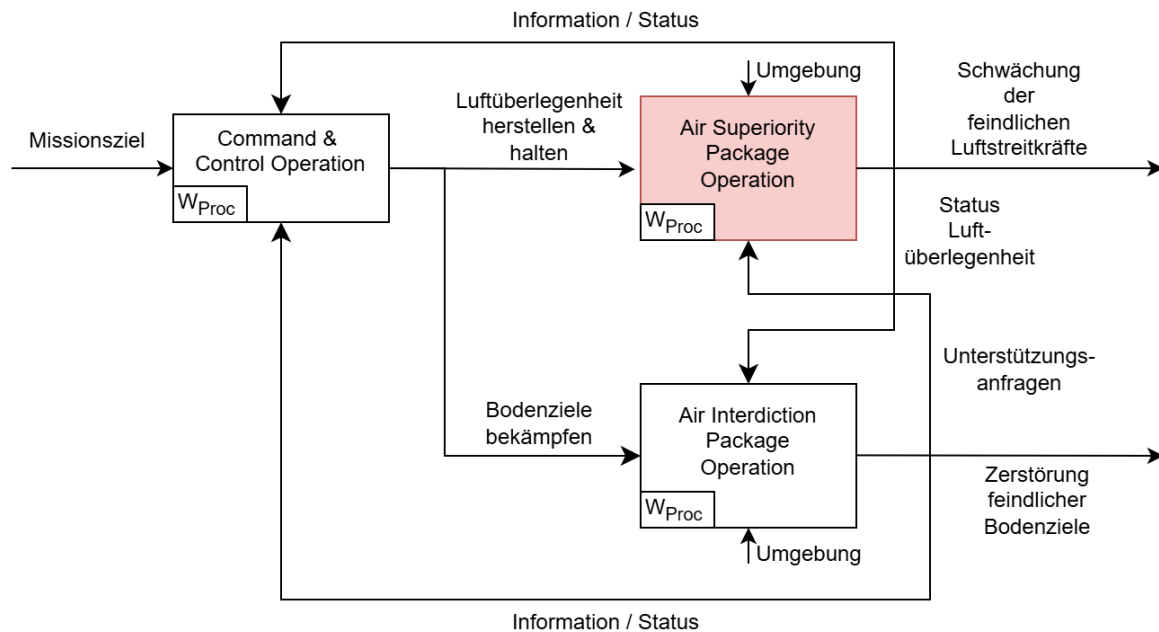


Abbildung 18. Mögliches Netzwerk aus Arbeitsprozessen zur Durchführung einer Mission.

Der Arbeitsprozess *Air Superiority Package Operation* erhält dabei sein Arbeitsziel vom übergeordneten Arbeitsprozess *Command and Control Operation*. Beeinflusst wird der Arbeitsprozess sowohl durch die Umweltfaktoren wie Wetter und feindliche Kräfte, aber auch durch den parallel ablaufenden Arbeitsprozess *Air Interdiction Package Operation*. Als Arbeitsergebnis erzielt der Arbeitsprozess eine Schwächung der feindlichen Luftstreitkräfte sowie Informationen für die Arbeitsprozesse *Command and Control Operation* und *Air Interdiction Package Operation*.

Diese Netzwerkdarstellung bietet einen ersten Überblick darüber, welche Arbeitsziele, Einflussfaktoren und Arbeitsergebnisse im Arbeitsprozess berücksichtigt werden sollten. Ihr hohes Abstraktionsniveau erschwert es jedoch, konkrete Anforderungen für das Mensch-Maschine-System abzuleiten. Um diese Schwierigkeit zu überwinden, wird daher im Folgenden der Detailgrad der Darstellung erhöht und spezifischere Anwendungsfälle innerhalb der Mission betrachtet.

Ein Beispiel für einen konkreten Anwendungsfall in der vorgestellten Mission ist die Missionsphase *Ingress*, deren Ziel das Eindringen in den feindlichen Luftraum ist (Abbildung 19).

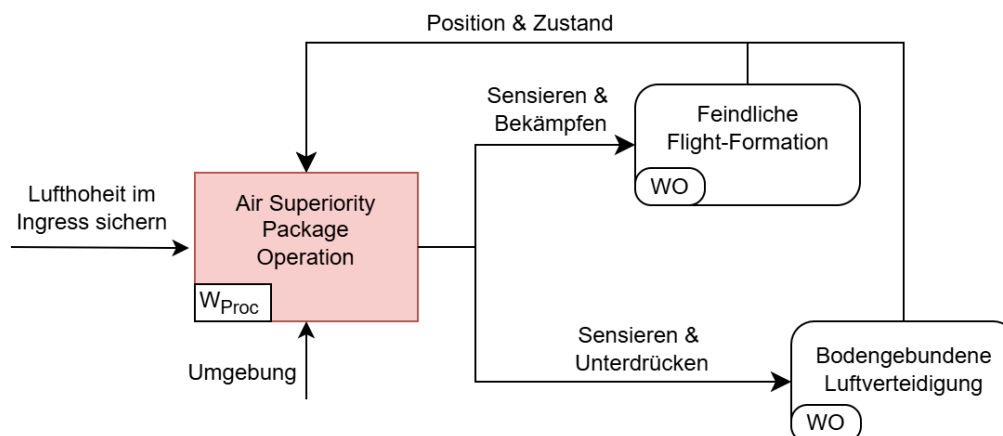


Abbildung 19. Wechselwirkung zwischen Arbeitsprozess „Air Superiority Package Operation“ und Arbeitsobjekten in der Missionsphase Ingress.

Die Netzwerkdarstellung dieser Missionsphase verdeutlicht, dass der Arbeitsprozess in der Lage sein muss, die Sensoren unter verschiedenen Umgebungsbedingungen (Wolken, Sichtweiten, Geländekonturen) effizient zu nutzen. Zudem muss er die Plattform-Effektoren zur Bekämpfung von Luft- und Bodenzielen effektiv einsetzen können.

3.1.2 Anforderungen an das Mensch-Maschine-System

Im Rahmen dieser Arbeit wurde die Arbeitsprozessanalyse für ein bemannt-unbemanntes Flight Package anhand der in 2.1 vorgestellten Missionen durchgeführt. Daraus wurden verschiedene Ausprägungen von Arbeitszielen, Umgebungsfaktoren und Ausgabegrößen identifiziert, die nachfolgend dokumentiert sind. Um ein fundiertes Verständnis über die Missionsabläufe und relevante Einflussfaktoren zu erhalten, wurde dieser Prozess durch Gespräche mit erfahrenen Nutzern bzw. Kampfflugzeugpiloten unterstützt.

Tabelle 4. Arbeitsziele, Umweltbedingungen und Ausgabegrößen, die sich aus den Missionen ergeben.

Arbeitsziele	Umweltbedingungen	Ausgabegrößen
<ul style="list-style-type: none"> • Aufklärung von Bodenzielen • Bekämpfung von Bodenzielen • Nachaufklärung von Bodenzielen • Bekämpfung od. Unterdrückung von Flugabwehrstellungen • Identifizierung von Luftfahrzeugen • Bekämpfung von Kampfflugzeugen • Take-Off • Transit 	<ul style="list-style-type: none"> • Anzahl und Position feindlicher Kräfte • Anzahl verbündeter Flight Packages • Zivilpersonen und zivile Infrastruktur • Wetter • Gelände 	<p>Ansteuerung von</p> <ul style="list-style-type: none"> • Flugmanagementsystem • Sensor-System • Waffensystem • EloKa-System²³ • Selbstschutz-System <p>für jede Plattform des Packages.</p> <p>(Die Art und Anzahl der Plattformen kann variieren)</p>

²³ System für elektronische Kampfführung.

3.2 Systementwurf

Dieser Abschnitt widmet sich dem Entwurf des Mensch-Maschine-Systems. In Unterabschnitt 3.2.1 erfolgt die **Konzeption des Systems** unter Anwendung der Arbeitssystemmethode. Aufbauend darauf werden in Unterabschnitt 3.2.2 die **Rollen und Interaktionen der Akteure** innerhalb des Arbeitssystems detailliert beschrieben. Abschließend wird in Unterabschnitt 3.2.3 die **funktionale Architektur** zur Umsetzung des Arbeitssystems beschrieben.

3.2.1 Kognitives Systemdesign

Nachdem in 3.1 die Anforderungen an den Arbeitsprozess eines Flight Leads definiert wurden, wird im Folgenden das Mensch-Maschine-System entwickelt, das diesen Prozess durchführt. Hierfür wird gemäß Arbeitssystemmethode zunächst mit einem initialen Arbeitssystem begonnen, das ohne den Einsatz kognitiver Agenten auskommt (Schulte et al., 2016).

Initialer Systementwurf

Abbildung 20 zeigt ein initiales Arbeitssystem am Beispiel eines Flight Packages bestehend aus einem bemannten und drei unbemannten Kampfflugzeugen.

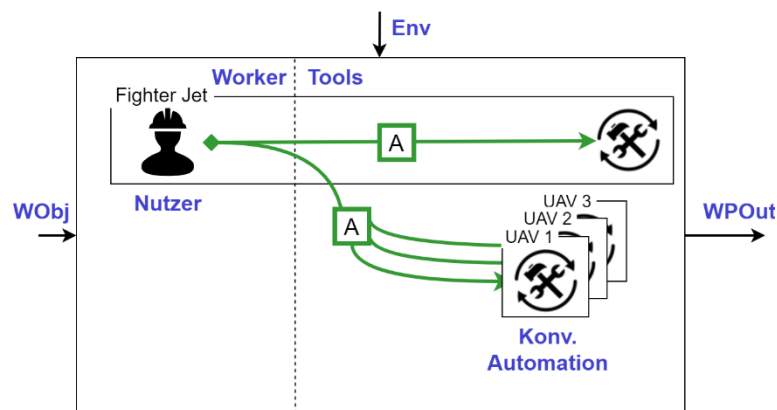


Abbildung 20. Initiales Arbeitssystemdesign ohne kognitive Agenten.

Die Umsetzung eines solchen Arbeitssystementwurfs ist nicht realistisch, da der Nutzer (bzw. Pilot) gleichzeitig vier Luftfahrzeuge führen müsste und dabei lediglich auf konventionelle Plattform-Automation zurückgreifen kann, woraus eine sehr hohe Beanspruchung und/oder eine geringe Missionsleistung resultieren würde.

Auftragsbasierte Führung

Die Führung mehrerer Plattformen kann effizienter gestaltet werden, indem mithilfe des Delegationsmusters bestimmte Aufgaben an einen kognitiven Agenten delegiert werden. Ein nachgewiesenermaßen geeignetes Entwurfsmuster dafür ist die *auftragsbasierte Führung* (Uhrmann & Schulte, 2011; Boehm et al., 2012; Clauss & Schulte, 2014). In der auftragsbasierten Führung delegiert der Nutzer abstrakte Aufträge an einen kognitiven Agenten an Bord des UAVs, der diese Aufträge wiederum versteht und die konventionelle

Plattform-Automation entsprechend kontrolliert.²⁴ Abbildung 21 zeigt den Arbeitssystementwurf unter Nutzung der auftragsbasierten Führung.

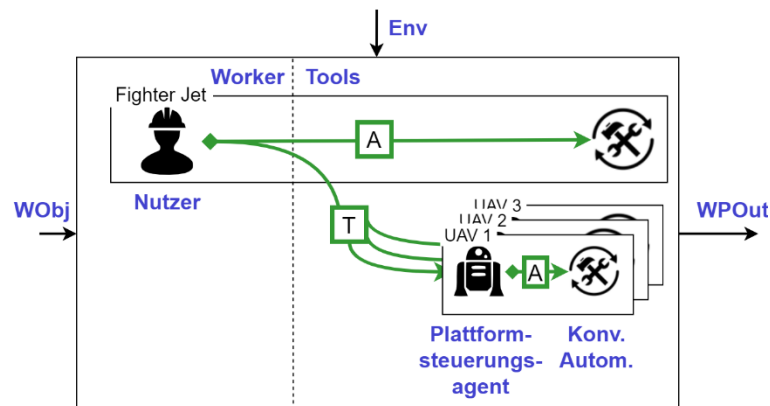


Abbildung 21. Arbeitssystementwurf unter Nutzung der auftragsbasierten Führung.

Durch den Einsatz der auftragsbasierten Führung verschiebt sich die Aufgabe des Nutzers von der direkten Kontrolle der Plattformsysteme zur Erteilung von Aufträgen oder Auftragssequenzen. Diese Änderung der Rollenverteilung ermöglicht es dem Nutzer, mehrere Plattformen gleichzeitig zu führen. Allerdings kann dieser Führungsansatz in dynamischen und komplexen Kampfflugzeugmissionen an seine Grenzen stoßen, da der Nutzer erheblichen Aufwand betreiben muss, um geeignete UAVs für die Auftragsausführung auszuwählen und die verschiedenen Aufträge aufeinander abzustimmen.

Unterstützung der Auftragsverteilung durch Scheduling-Agent

Ein Ansatz, diesen Koordinationsaufwand zu reduzieren, ist die Erweiterung der auftragsbasierten Führung um einen Scheduler, der den Nutzer bei der Verteilung der Aufträge unterstützt (Heilemann & Schulte, 2019). Bei diesem Ansatz definiert der Nutzer den Auftrag und Verteilkriterien im Rahmen der Auftragsdelegation (z. B. eine spezifische Plattform soll den Auftrag durchführen) und der Scheduler erstellt auf Basis dieser Vorgaben einen Missionsplan (Abbildung 22).

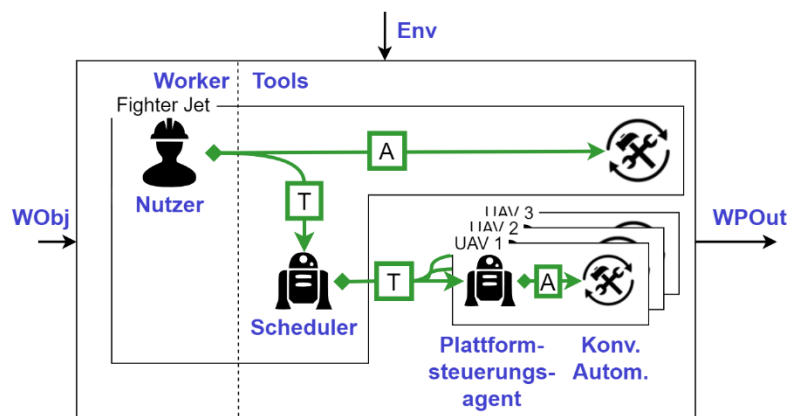


Abbildung 22. Arbeitssystem mit Scheduler zur Aufgabenverteilung.

²⁴ Dieser Agent wird in Anlehnung an seine Funktion im Folgenden als „Plattformsteuerungsagent“ bezeichnet.

Die auftragsbasierte Führung mit Scheduling-Agent wurde in einer Simulatorstudie mit Kampfflugzeugpiloten im Kontext von Air Interdiction Missionen²⁵ untersucht. Dabei zeigte sich, dass der Einsatz eines Scheduling-Agenten die Delegationsleistung im Vergleich zur manuellen Ressourcenplanung deutlich verbessern kann (Heilemann et al., 2021). Allerdings bleibt trotz dieser positiven Ergebnisse fraglich, ob die Erweiterung der auftragsbasierten Führung um einen Scheduling-Agenten ausreicht, um den Anforderungen dynamischerer Missionstypen gerecht zu werden, da die untersuchten Missionen eine verhältnismäßig geringe Dynamik und hohe Planbarkeit aufwiesen und die Ergebnisse somit nur schwer auf dynamischere Missionen übertragbar sind. Zudem sind die Formulierung von Aufträgen und die Definition von Verteilkriterien für den Nutzer weiterhin mit einer hohen Aufgabenbelastung verbunden, was in solchen Missionen zu einer Überforderung führen könnte.

Um auch in hochdynamischen Missionen eine effektive Führung mehrerer unbemannter Systeme aus dem Kampfflugzeugcockpit zu ermöglichen, muss der Nutzer durch eine umfassendere Verlagerung von Missionsmanagementaufgaben an die Automation entlastet werden.

Anpassung der Delegationsform

Eine Möglichkeit zur Verlagerung von Aufgaben ist die Anpassung der Delegationsform. Anstatt Aufträge zu erteilen, könnte der Nutzer den Plattformsteuerungsagenten übergeordnete Ziele vorgeben (Abbildung 23). Diese höhere Führungsebene würde die Beanspruchung des Nutzers reduzieren, da in diesem Fall die Agenten an seiner Stelle entscheiden, wie die übergeordneten Ziele erreicht werden sollen. Allerdings könnte der größere Entscheidungsspielraum der Agenten zu einer schlechteren Nachvollziehbarkeit des UAV-Verhaltens führen, was die Führungsaufgabe des Nutzers erschweren würde.

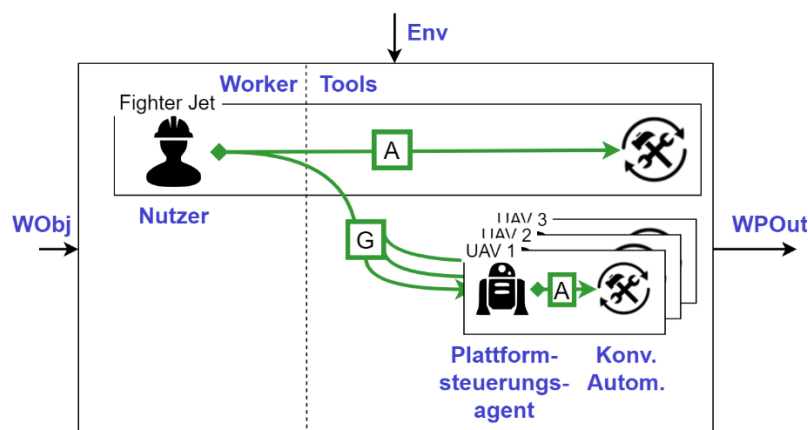


Abbildung 23. Arbeitssystementwurf bei Nutzung einer zielbasierten Delegationsform.

Dieser potenzielle Nachteil eines zielbasierten Führungskonzepts sowie die in mehreren Studien nachgewiesene Eignung eines auftragsbasierten Ansatzes führen zur Entscheidung,

²⁵ Dieser Missionstyp wird durch die Aufklärung und Bekämpfung von Bodenzielen geprägt.

Assistierte auftragsbasierte Führung

Diese Erkenntnisse über die Stärken und Schwächen des Mixed-Initiative Ansatzes führen zum Grundgedanken eines alternativen Ansatzes: Wenn die Unterstützung des Nutzers im Rahmen der UAV-Führung positive Effekte mit sich bringen kann, das Initiativrecht des Agenten jedoch potenzielle Nachteile mit sich bringt, warum dann nicht einen Unterstützungsansatz wählen, bei dem die Initiative zur Auftragserteilung beim Nutzer verbleibt?

Dieser Gedanke bildet den Kern des Entwurfsmusters der *assistierten auftragsbasierten Führung* (Abbildung 25). Das Ziel dieses Entwurfsmusters ist es, den Nutzer so zu unterstützen, dass er sein volles Potenzial ausschöpfen und die unbemannten Systeme möglichst effektiv führen kann, während die Initiative über den UAV-Einsatz vollständig beim Nutzer verbleibt. Zu diesem Zweck wird das Entwurfsmuster der auftragsbasierten Führung um einen *Delegationsagenten* erweitert. Dieser Agent unterstützt den Nutzer dabei, seine Absichten schnell und effektiv in Auftragserteilungen zu überführen. Außerdem überprüft er die erteilten Aufträge auf mögliche Probleme oder Optimierungspotenziale und schlägt dem Nutzer gegebenenfalls Verbesserungen vor.²⁶

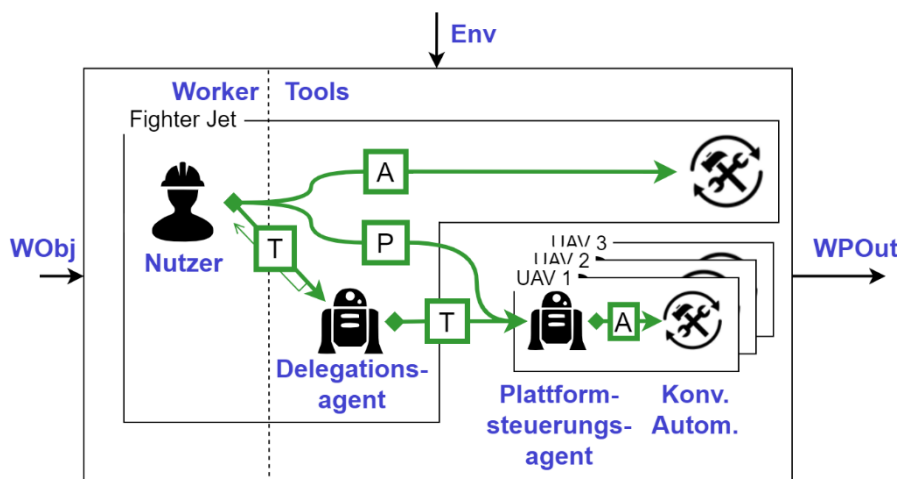


Abbildung 25. Entwurfsmuster „Assistierte auftragsbasierte Führung“.

Aus Gründen der Übersichtlichkeit sind die Beziehungen zwischen Nutzer und Plattformsteuerungsagenten sowie zwischen Delegationsagent und Plattformsteuerungsagenten jeweils nur einmalig dargestellt.

Diese Arbeit untersucht dieses vielversprechende Entwurfsmuster als Lösungsansatz für die Führung mehrerer unbemannter Luftfahrzeuge aus dem Cockpit eines Kampfflugzeugs. In den folgenden Abschnitten des Konzeptkapitels werden die einzelnen Elemente dieses Entwurfsmusters daher ausführlich beschrieben.

²⁶ Dieses Entwurfsmuster weist gewisse Parallelen zur auftragsbasierten Führung mit Scheduling-Agent auf. Es unterscheidet sich jedoch in zwei wesentlichen Punkten: (1) Der Delegationsagent übernimmt neben der Auftragsverteilung zusätzliche unterstützende Tätigkeiten im Rahmen der Auftragsdelegation. (2) Während der Scheduler nur dann auf Probleme hinweist, wenn eine Anweisung nicht umsetzbar ist, analysiert der Delegationsagent jede Anweisung gezielt auf Optimierungspotenziale – auch wenn sie prinzipiell umsetzbar wäre.

3.2.2 Rollen und Beziehungen der Akteure

Im Entwurfsmuster der assistierten auftragsbasierten Führung gibt es vier Arten von Akteuren mit unterschiedlichen Rollen. Eine Übersicht über die Akteure des Entwurfsmusters und deren Rollen ist in Tabelle 5 dargestellt.

Tabelle 5. Akteure im Entwurfsmuster der assistierten auftragsbasierten Führung.

Akteur	Rollenbeschreibung
Nutzer	Der menschliche Nutzer verfolgt die Missionsziele durch die Führung der unbemannten Plattformen des Teams und durch die eigenständige Bearbeitung von Missionsaufgaben.
Delegations-agent	Der Delegationsagent unterstützt den Nutzer bei der Führung der unbemannten Plattformen.
Plattform-steuerungs-agent	Die Plattformsteuerungsagenten kontrollieren die konventionelle Plattform-Automation von UAVs im Sinne der vom Delegationsagenten erhaltenen Anweisungen.
Konventionelle Plattform-Automation.	Die konventionelle Plattform-Automation umfasst verschiedene automatisierte Funktionen, welche dedizierte Aufgaben der Luftfahrzeugsteuerung übernehmen, beispielsweise das Flugmanagementsystem oder das Waffensystem.

Im Entwurfsmuster sind verschiedene Formen von Actor-Relationship-Actor Tupel zu finden (Abbildung 26). Nachfolgend werden diese Tupel und die daraus resultierenden Interaktionen beschrieben.

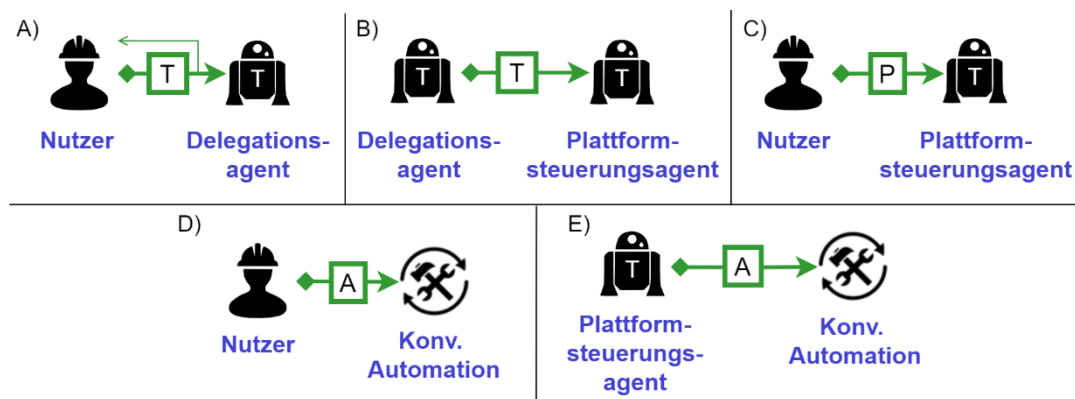


Abbildung 26. Actor-Relationship-Actor Tupel der assistierten auftragsbasierten Führung.

Nutzer – Delegationsagent

Zwischen Nutzer und Delegationsagent besteht eine hierarchische Delegationsbeziehung, im Rahmen derer der Nutzer Aufgaben an den Delegationsagenten überträgt. Eine besondere Eigenschaft dieser Delegationsbeziehung ist der explizite Feedback-Modus, der sich dadurch äußert, dass der Delegationsagent die Anweisungen des Nutzers kritisch hinterfragt. Der Delegationsagent ist als Tool ausgeführt und handelt damit nicht eigeninitiativ im Hinblick auf das übergeordnete Arbeitsziel.

Das Kernelement dieser Beziehung ist die Unterstützung des Nutzers bei der Führung der unbemannten Plattformen. Zu diesem Zweck übermittelt der Nutzer dem Delegationsagenten seine Absichten darüber, welche Missionsaufgaben von den unbemannten Plattformen ausgeführt werden sollen und welche Randbedingungen dabei zu beachten sind. Der Delegationsagent überprüft diese Vorgaben kritisch und unterstützt den Nutzer bei der Definition und Verteilung von Aufgaben an die verschiedenen Plattformen.

Die Rolle des Delegationsagenten ähnelt der eines Beraters: Er unterstützt den Nutzer bei Entscheidungen, übernimmt Aufgaben für ihn und hinterfragt dessen Anweisungen kritisch, um eine effektive Umsetzung der Nutzerabsichten sicherzustellen. Allerdings wird der Agent ausschließlich auf Wunsch des Nutzers tätig.

Delegationsagent - Plattformsteuerungsagent

Zwischen dem Delegationsagent und den Plattformsteuerungsagenten besteht eine hierarchische Beziehung, die auf der Delegation von Aufgaben beruht. Wie der Delegationsagent sind auch die Plattformsteuerungsagenten als Tools konzipiert, sodass sie das Arbeitsziel nicht eigeninitiativ verfolgen.

Im Rahmen dieser Beziehung erteilt der Delegationsagent Aufgaben in Form von klar definierten Aufträgen an die Plattformsteuerungsagenten. Die Plattformsteuerungsagenten sind an die erteilten Aufträge gebunden, verfügen jedoch über einen gewissen Spielraum in deren Umsetzung.

Das Verhältnis zwischen Delegationsagent und Plattformsteuerungsagenten ähnelt dem zwischen militärischen Vorgesetzten und ihren Untergebenen: Wie untergebene Soldaten führen die Plattformsteuerungsagenten die ihnen erteilten Aufträge aus, wobei sie über gewisse Handlungsspielräume verfügen, um angemessen auf die Umgebung reagieren zu können.

Nutzer – Plattformsteuerungsagent

Zusätzlich zur indirekten Beziehung über den Delegationsagenten besteht auch eine direkte hierarchische Verbindung zwischen dem Nutzer und den Plattformsteuerungsagenten.

Innerhalb dieser hierarchischen Beziehung formuliert der Nutzer Verhaltensrichtlinien für den Plattformsteuerungsagenten. Diese Richtlinien legen fest, welche Verhaltensweisen bzw. Handlungen erwünscht, zulässig oder untersagt sind.

Dieses Verhältnis ist vergleichbar mit dem Verhältnis zwischen den obersten militärischen Befehlsinstanzen und den unterstellten Einheiten. So erlassen diese Befehlsinstanzen beispielsweise Einsatzregeln (Rules of Engagement, *ROE*), die von untergeordneten Kommandeuren nicht abgeändert werden dürfen und von allen Einheiten befolgt werden müssen.

Nutzer – Konventionelle Automation (Kampfflugzeug)

Zwischen Nutzer und konventioneller Automation des Kampfflugzeugs besteht eine hierarchische Delegationsbeziehung, welche durch die Delegation von elementaren

Handlungen oder Handlungssequenzen charakterisiert ist. Beispielsweise kann der Nutzer Höhen-, Richtungs- und Geschwindigkeitsvorgaben an den Autopiloten erteilen oder Wegpunkte vorgeben.

Plattformsteuerungsagent – Konventionelle Automation (UAV)

Zwischen den Plattformsteuerungsagenten und deren zugehöriger konventioneller Plattform-Automation besteht ebenfalls eine hierarchische Delegationsbeziehung, die aus der Delegation von Handlungen und Handlungssequenzen besteht.

3.2.3 Funktionale Architektur

Nachdem das Entwurfsmuster der assistierten auftragsbasierten Führung sowie die darin enthaltenen Akteure und ihre Interaktion beschrieben wurde, wird nun die funktionale Architektur konzipiert. Die funktionale Architektur definiert die Funktionen und Informationsflüsse, die für die Umsetzung des Mensch-Maschine-Systems erforderlich sind.

In Abbildung 27 ist die funktionale Architektur dargestellt, mit der im Folgenden das Entwurfsmuster der assistierten auftragsbasierten Führung realisiert wird. Sie besteht aus den Modulen Nutzerschnittstelle, Delegationsagent, Plattformsteuerungsagent und Luftfahrzeug-Automation sowie aus verschiedenen Informationsflüssen zwischen diesen Modulen.²⁷ Die Module und ihre Informationsflüsse werden in den folgenden Abschnitten näher erläutert.

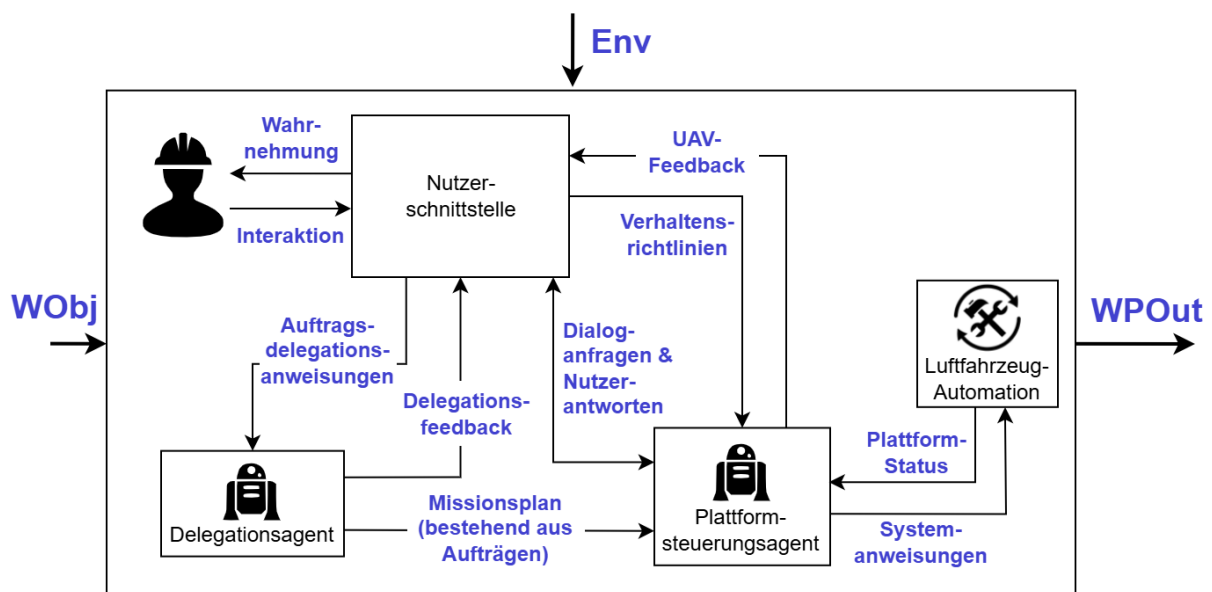


Abbildung 27. Funktionale Architektur der assistierten auftragsbasierten Führung.

²⁷ Im Schaubild wird aus Gründen der Übersichtlichkeit nur ein einzelnes unbemanntes Luftfahrzeug und dessen Automation betrachtet.

3.3 Aufträge als Kommunikationsmedium

In der auftragsbasierten Führung dienen die Aufträge als Beschreibungssprache zwischen Nutzer und Automation. Sie ermöglichen es dem Nutzer, seine Absichten klar zu kommunizieren und schaffen ein gemeinsames Verständnis darüber, welche Erwartungen an die Automation gestellt werden (Miller, Funk, Wu, et al., 2005). Aufgrund dieser zentralen Rolle von Aufträgen beleuchtet dieser Abschnitt die wesentlichen Elemente der auftragsbasierten Führung im Kontext von Manned-Unmanned Teaming. Zu Beginn wird in Unterabschnitt 3.3.1 die **Auftragsdefinition** vorgestellt. Anschließend werden in Unterabschnitt 3.3.2 die verschiedenen **Ausprägungen von Aufträgen** zur Bewältigung von Kampfflugzeugmissionen vorgestellt. Abschließend wird in Unterabschnitt 3.3.3 der **Prozess der Auftragsdelegation** beschrieben.

3.3.1 Auftragsdefinition

Zur Beschreibung militärischer UAV-Aufträge wurde eine Auftragsdefinition mit vier Komponenten entwickelt (vgl. Dudek & Schulte (2022b)). Im Folgenden werden diese Komponenten beschrieben und ihre Bedeutung für die Auftragsdefinition motiviert.

Der *Auftragstyp* differenziert verschiedene Arten von Aufträgen und reflektiert damit die verschiedenen Absichten, die der Nutzer mit der Auftragsdelegation verfolgen kann (Lindner et al., 2019). Die verschiedenen Auftragstypen orientieren sich im Wesentlichen an heutigen militärischen Missionaufgaben wie Aufklärung oder Bekämpfung.

Das *Zielobjekt* definiert, mit welchem physikalischen oder logischen Objekt der Auftrag in Zusammenhang steht.²⁸ Am Beispiel eines Aufklärungsauftrags könnte das Zielobjekt ein zu untersuchendes Fahrzeug oder ein zu durchsuchendes Zielgebiet sein.

Das *Erfolgskriterium* definiert die Bedingungen, unter denen ein Auftrag erfolgreich abgeschlossen ist. Ein solches Erfolgskriterium könnte beispielsweise sein, dass eine bestimmte Anzahl an Aufklärungsbildern aufgenommen wurde.

Die *Parameter* dienen dazu, die Ausführung eines Auftrags genauer zu definieren. Art und Anzahl der Parameter eines Auftrags hängen vom Auftragstyp und dem Zielobjekt ab. Beispielsweise kann ein Nutzer bei der Aufklärung eines Zielgebietes Vorgaben zu Suchstrategien oder Eigenschaften von Aufklärungsbildern machen, während er bei Bekämpfungsaufträgen die Angriffsrichtung vorgeben kann.

²⁸ Eine Beschreibung von logischen Objekten befindet sich im Anhang E.1.

3.3.2 Aufträge für militärische Kampfflugzeugmissionen

Die in 2.1 beschriebenen Missionen sind durch eine hohe Dynamik und ein breites Spektrum an Missionsaufgaben gekennzeichnet. Die Erfüllung dieser Missionsaufgaben erfordert eine Vielzahl verschiedener Auftragsstypen, die abhängig vom militärischen Kontext in verschiedene Kategorien unterteilt werden können:

- Navigation: Diese Kategorie umfasst Auftragsstypen, die der Positionierung von Kräften dienen.
- Air Interdiction: Zu dieser Kategorie gehören Auftragsstypen, die auf die Aufklärung und Bekämpfung von Bodenzielen ausgerichtet sind.
- Offensive Counter Air: Diese Kategorie umfasst Aufträge zur Bekämpfung gegnerischer Luftfahrzeuge.
- Suppression of Enemy Air Defences: Diese Kategorie beinhaltet Aufträge zur Unterdrückung und Bekämpfung der bodengebundenen Luftverteidigung.
- Mehrzweck: Diese Kategorie umfasst Auftragsstypen, die keiner der zuvor genannten Kategorien zugeordnet werden können.

In der folgenden Tabelle 6 werden exemplarisch verschiedene Auftragsstypen sowie die jeweiligen Erfolgskriterien und Parameter dieser Aufträge vorgestellt. Eine vollständige Übersicht über die verschiedenen Aufträge ist in Anhang E.2 zu finden.

Tabelle 6. Auftragsstypen, Erfolgskriterien und Parameter ausgewählter Aufträge.

Kategorie	Auftrags-typ	Beschreibung	Erfolgskriterium	Parameter
Navigation	Loiter	Warteschleifen in einem Gebiet oder in der Nähe eines Punktes	<ul style="list-style-type: none"> • Zeit verstrichen 	<ul style="list-style-type: none"> • Flughöhe
Air Interdiction	Engage	Bekämpfung eines Bodenziels	<ul style="list-style-type: none"> • Payload ausgebracht • Verifizierte Zerstörung 	<ul style="list-style-type: none"> • Abwurfdistanz • Angriffsrichtung • Anzahl Effektoren • Zünder
Offensive Counter Air	Combat Air Patrol	Identifizierung und ggf. Bekämpfung von Luftfahrzeugen	<ul style="list-style-type: none"> • Zeit verstrichen 	<ul style="list-style-type: none"> • Taktik
Suppression of Enemy Air Defences	Target Jamming	Elektronischer Angriff auf Radarstellung	<ul style="list-style-type: none"> • Zeit verstrichen 	<ul style="list-style-type: none"> • Jamming-Typ • Flughöhe

3.3.3 Auftragsdelegation

Bei der Auftragsdelegation handelt es sich um den Prozess zur Erteilung von Aufträgen. In der assistierten auftragsbasierten Führung besteht dieser Prozess aus den Teilschritten *Auftragspezifikation*, *Auftragsverteilung* und einem optionalen *Review*-Schritt (Abbildung 28). Nachfolgend werden die Ziele der verschiedenen Prozessschritte und ihre konzeptionelle Umsetzung im Kontext des Entwurfsmusters näher beschrieben.

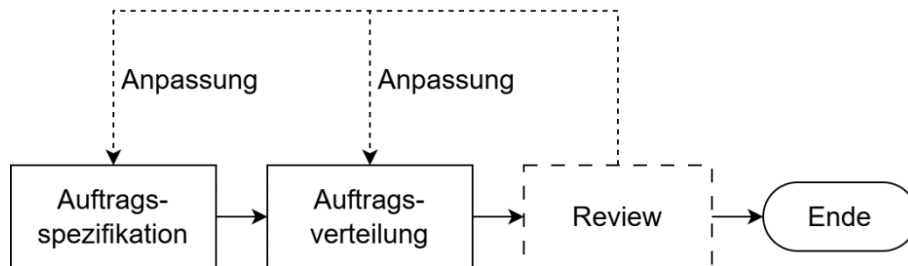


Abbildung 28. Prozessschritte der Auftragsdelegation.

Auftragspezifikation

Im Prozessschritt der Auftragspezifikation werden die Komponenten eines zu erteilenden Auftrags gemäß der in 3.3.1 beschriebenen Auftragsdefinition befüllt, d. h. es werden der Auftragstyp, das Zielobjekt, das Erfolgskriterium und die Parameter des Auftrags definiert. Ein Beispiel für eine Spezifikation eines Auftrags ist in Abbildung 29 dargestellt.

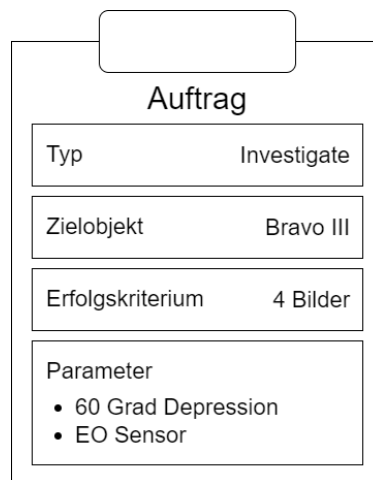


Abbildung 29. Mögliche Auftragspezifikation zur Aufklärung eines Zielobjektes.

Auftragsverteilung

Die Zuweisung von Aufträgen an die Plattformen des Teams wird als Auftragsverteilung bezeichnet. Das Ziel der Auftragsverteilung sollte darin bestehen, die verfügbaren Ressourcen des Teams so einzusetzen, dass eine möglichst optimale Missionsleistung erzielt wird. In der Literatur wird diese Art von Problemen als *Optimal Assignment Problem* bezeichnet (Khamis et al., 2015; Öncan, 2007).

Die Lösung dieses Problems kann durch verschiedene Ansätze erfolgen, die sich entsprechend ihres Organisationsmusters in zentrale oder dezentrale Ansätze klassifizieren

lassen (Horst & Noble, 2010). Zentrale Ansätze zeichnen sich dadurch aus, dass eine zentrale Instanz die Allokation von Aufgaben an einzelne Akteure übernimmt (Brumitt & Stentz, 1998). Demgegenüber erfolgt die Aufgabenallokation in dezentralen Ansätzen ohne zentrale Instanz durch Kommunikation zwischen den Akteuren (Patel et al., 2020).

Im Entwurfsmuster der assistierten auftragsbasierten Führung ist das Organisationsmuster durch die Rolle des Delegationsagenten festgelegt. Dabei kommt ein zentraler Optimierungsansatz zum Einsatz, bei dem ein Scheduler die optimale Zuordnung der Aufträge zu den Plattformen des Teams plant und daraus einen Missionsplan generiert (Abbildung 30).

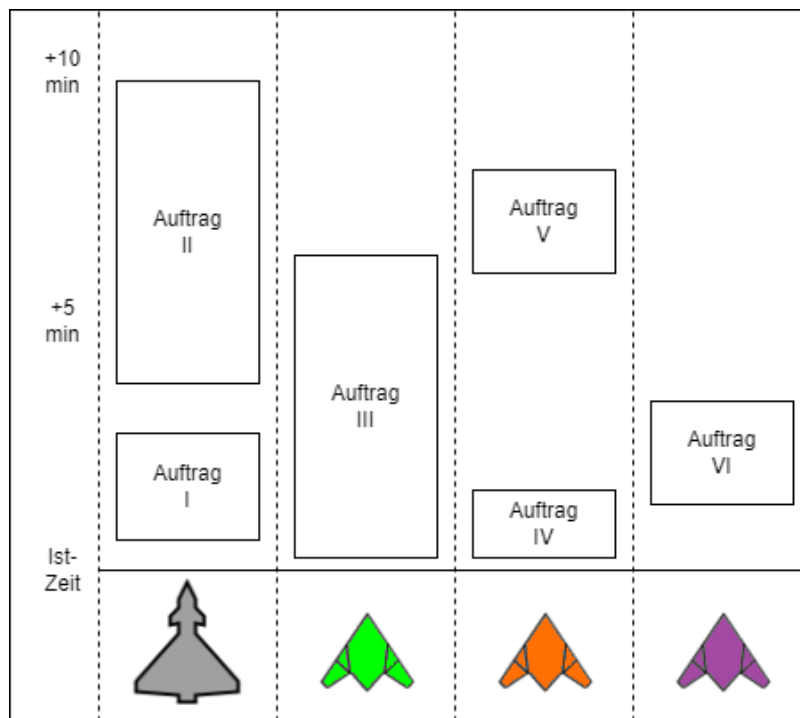


Abbildung 30. Beispiel eines Missionsplanes für eine bemannt-unbemannte Flight.

Dieser Verteilungsansatz wurde gewählt, da zentralisierte Allokationsansätze Vorteile hinsichtlich der Optimalität der Lösung bieten können und die Nachteile zentralisierter Ansätze erst bei einer großen Anzahl von Entitäten zum Tragen kommen (Horst & Noble, 2010). Darüber hinaus schafft diese Form der Zuordnung für den Nutzer Transparenz darüber, von welcher Plattform zu welchem Zeitpunkt Aufträge ausgeführt werden.

Review

Im optionalen Review-Schritt hat der Nutzer die Möglichkeit, den erteilten Auftrag und die damit verbundenen Änderungen am Missionsplan zu überprüfen, bevor der aktualisierte Missionsplan an die unbemannten Plattformen übermittelt wird. Ziel dieses Überprüfungsschrittes ist es, potenziell kritische Fehlentscheidungen zu verhindern, die durch Bedienungsfehler, Systemfehler oder Missverständnisse entstehen können.

Das zugrundeliegende Konzept von Auftragsdelegationen mit Review-Schritt ist, dass alle Änderungen an einer Kopie des Missionsplans, dem sogenannten *Modify-Plan*, vorgenommen werden. Im Review überprüft der Nutzer den Modify-Plan und nimmt gegebenenfalls Anpassungen an der Auftragspezifikation oder -verteilung vor. Der Nutzer kann den Modify-Plan anschließend aktiv schalten oder die Änderungen verwerfen.

Wird ein Delegationsprozess ohne Review durchgeführt, werden die Änderungen auf dem aktiven Missionsplan vorgenommen und führen zu einer sofortigen Anpassung des UAV-Verhaltens.²⁹ Die unterschiedlichen Abläufe zur Delegation eines einzelnen Auftrags mit bzw. ohne Review-Schritt sind in Abbildung 31 dargestellt.

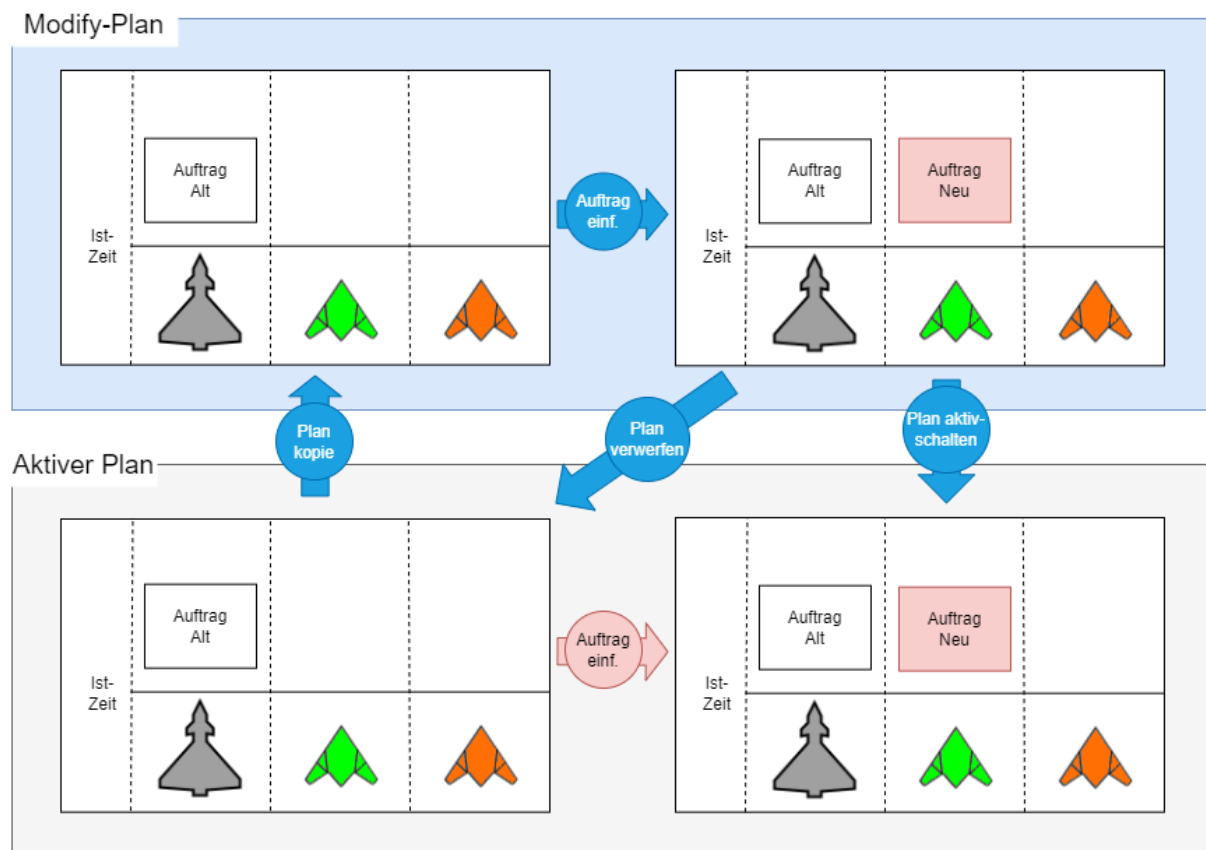


Abbildung 31. Delegation eines Auftrags ohne Review-Schritt (roter Pfeil) und mit Review-Schritt (blaue Pfeile).

²⁹ Auftragsdelegationen mit Review entsprechen somit eher dem Konzept *Management by Consent* während Auftragsdelegationen ohne Review dem Konzept *Management by Exception* entsprechen (Billings, 1997)

3.4 Delegationsagent

Dieser Abschnitt behandelt den Delegationsagenten. Zu Beginn des Abschnittes werden in [3.4.1](#) die **Aufgaben des Agenten** beschrieben. Anschließend wird in [3.4.2](#) die **Architektur** vorgestellt, die zur Umsetzung dieser Aufgaben dient.

3.4.1 Aufgaben

Der Delegationsagent unterstützt den Nutzer durch zwei Fähigkeiten in der Führung der unbemannten Plattformen:

1. Die Bereitstellung einer schnellen und bedarfsgerechten Schnittstelle zur Delegation von Aufträgen.
2. Kritisches Hinterfragen von Auftragsdelegationen.

Nachfolgend werden diese beiden Fähigkeiten genauer beschrieben und motiviert.

Delegationsschnittstelle

Eine Fähigkeit des Delegationsagenten ist die Bereitstellung einer geeigneten Schnittstelle zur Auftragsdelegation. Hierfür bietet der Delegationsagent dem Nutzer verschiedene Interaktionsmethoden zur Auftragsverteilung an, die sogenannten *Delegationsmethoden*. Diese Delegationsmethoden basieren auf drei Kernelementen, die nachfolgend beschrieben und motiviert werden.

Ein Kernelement der Delegationsmethoden ist eine *Vervollständigung*. Das bedeutet, dass der Nutzer im Rahmen der Interaktion mit dem System nicht alle Informationen bereitstellen muss, die für die Auftragsdelegation erforderlich sind. So kann der Nutzer beispielsweise nur einzelne Komponenten eines Auftrags spezifizieren und der Delegationsagent ergänzt die verbleibenden Informationen, um die Auftragspezifikation zu vervollständigen (Abbildung 32).

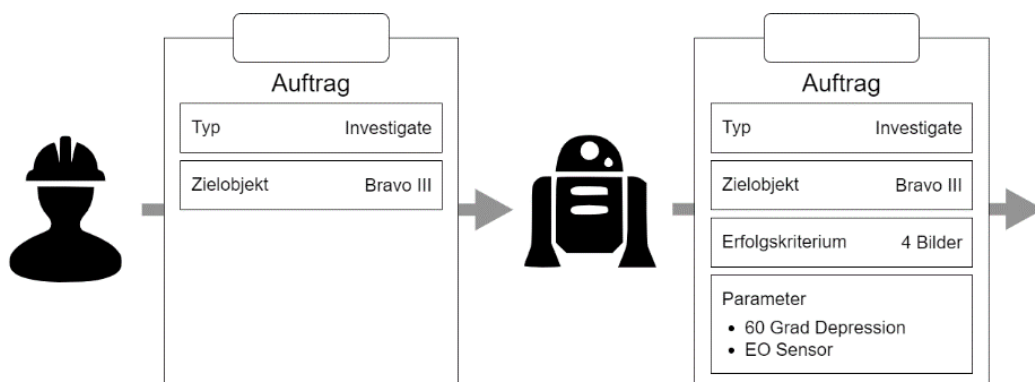


Abbildung 32. Vervollständigung der Auftragspezifikation durch Delegationsagent.

Aufgrund der Vervollständigung muss sich der Nutzer auch nicht um die detaillierte Zuordnung von Aufträgen kümmern. Stattdessen kann der Nutzer dem Delegationsagenten vorgeben, wie ein Auftrag verteilt werden soll und der Delegationsagent erstellt daraufhin einen optimalen Missionsplan, der die Anforderungen des Nutzers erfüllt und zeitliche und physische Ressourcen berücksichtigt (Abbildung 33).

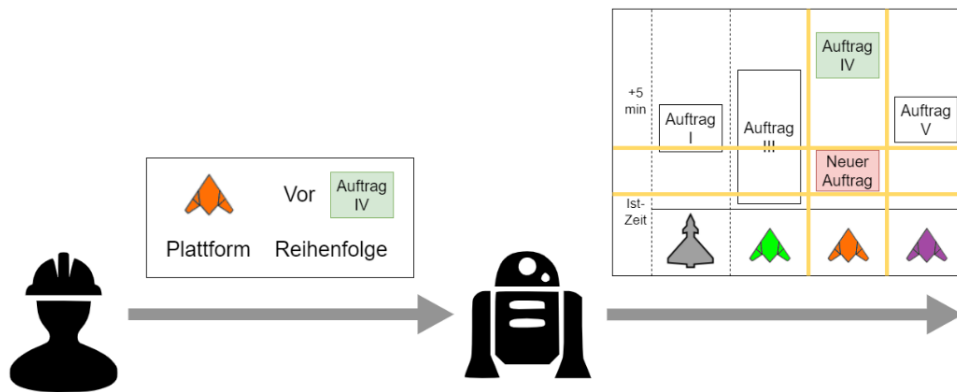


Abbildung 33. Verteilung eines Auftrags durch Delegationsagent.

Zur Verteilung eines Auftrags stehen dem Nutzer verschiedene *Scheduling-Modi* zur Verfügung (Tabelle 7).

Tabelle 7. *Scheduling-Modi zur Auftragsverteilung (vgl. Heilemann & Schulte (2020b))*.

Modus	Beschreibung
Team	Der Agent wählt Plattform und Zeitpunkt der Auftragsdurchführung.
Plattform	Der Nutzer wählt die Plattform und der Agent wählt den Zeitpunkt der Auftragsdurchführung.
Zeitslot	Der Nutzer wählt die Plattform und gibt eine Reihenfolge von Aufträgen vor. Der Agent wählt den exakten Zeitpunkt der Auftragsdurchführung.
TOT ³⁰	Der Nutzer wählt die Plattform und den exakten Zeitpunkt der Auftragsdurchführung.

Ein weiteres Kernelement der Delegationsmethoden ist eine *Adaptierbarkeit*, also eine vom Nutzer dynamisch variierbare Mensch-Maschine-Aufgabenteilung. Eine solche Adaptierbarkeit kann sich positiv auf die Performance des Mensch-Maschine-Systems auswirken (Parasuraman et al., 2005), unter anderem durch eine Reduktion von Workload-Spitzen (Miller & Parasuraman, 2007) und eine verbesserte Kalibrierung des Nutzervertrauens (Miller, Funk, Goldman, et al., 2005). Die Adaptierbarkeit wird in den Delegationsmethoden umgesetzt, indem dem Nutzer verschiedene *Delegationsmodi* angeboten werden. Diese Delegationsmodi unterscheiden sich in den Informationen, die der Nutzer im Rahmen der Auftragspezifikation bereitstellen muss, in den Scheduling-Modi sowie darin, ob die Auftragsdelegation einen optionalen Review-Schritt beinhaltet.³¹

Ein drittes Kernelement der Delegationsmethoden ist *Multimodalität*. Der Begriff „multimodal“ bedeutet in diesem Kontext, dass eine Eingabeschnittstelle auf zwei oder mehr Modalitäten (z. B. Sprache, Berührung) basiert. Dies umfasst sowohl Schnittstellen, bei denen eine Auswahlmöglichkeit zwischen verschiedenen Modalitäten besteht, als auch Schnittstellen, bei denen eine Eingabeinteraktion auf dem kombinierten Einsatz mehrerer

³⁰ In militärischen Missionen wird die Time-Over-Target (TOT) verwendet, um festzulegen, dass eine Handlung oder ein Effekt zu einem exakten Zeitpunkt erfolgen soll.

³¹ Die Delegationsmodi bauen folglich auf dem Konzept der Vervollständigung auf.

Modalitäten beruht (Oviatt et al., 2017). Beide Ausprägungen von Multimodalität können sich positiv auf die Performance auswirken. So kann die Möglichkeit zur Auswahl der Modalität die kognitive Leistungsfähigkeit des Nutzers erhöhen, da dieser die Belastung seiner Ressourcenkanäle flexibel steuern und optimieren kann (Oviatt et al., 2004; Oviatt, 2022) während die Kombination mehrerer Modalitäten zu einer schnelleren und robusteren Interaktion führen kann, weil die Schwächen einer Modalität durch die Stärken einer anderen Modalität kompensiert werden können (Cohen et al., 1989; Oviatt & Cohen, 2000). Die Delegationsmethoden vereinen beide Aspekte der Multimodalität. Einerseits kann eine Methode auf mehreren Modalitäten basieren. Andererseits ermöglichen sie dem Nutzer, durch die Auswahl unterschiedlicher Delegationsmethoden die zugrunde liegende Modalität flexibel zu variieren.

Kritisches Hinterfragen von Auftragsdelegationen

Der Delegationsagent unterstützt den Nutzer, indem er Auftragsdelegationen des Nutzers bewertet und kritisch hinterfragt. Werden bei der Überprüfung eines Auftrags Probleme festgestellt, kritisiert der Delegationsagent den Auftrag und schlägt gegebenenfalls Lösungen zur Behebung der Probleme vor. Diese Fähigkeit wird nachfolgend als *Auftragskritik* bezeichnet.

Die Auftragskritik ist eine Form der automatisierten Entscheidungshilfe, die – wie andere Ansätze – darauf abzielt, den Nutzer bei der Entscheidungsfindung zu unterstützen. Im Gegensatz zu den verbreiteteren, vorschlagsbasierten Ansätzen besteht die Unterstützungsleistung jedoch nicht darin, Lösungsvorschläge bereitzustellen, sondern die vom Nutzer entwickelten Lösungen zu bewerten und gegebenenfalls zu optimieren. Abbildung 34 verdeutlicht den Unterschied zwischen dem kritikbasierten und dem vorschlagsbasierten Ansatz.

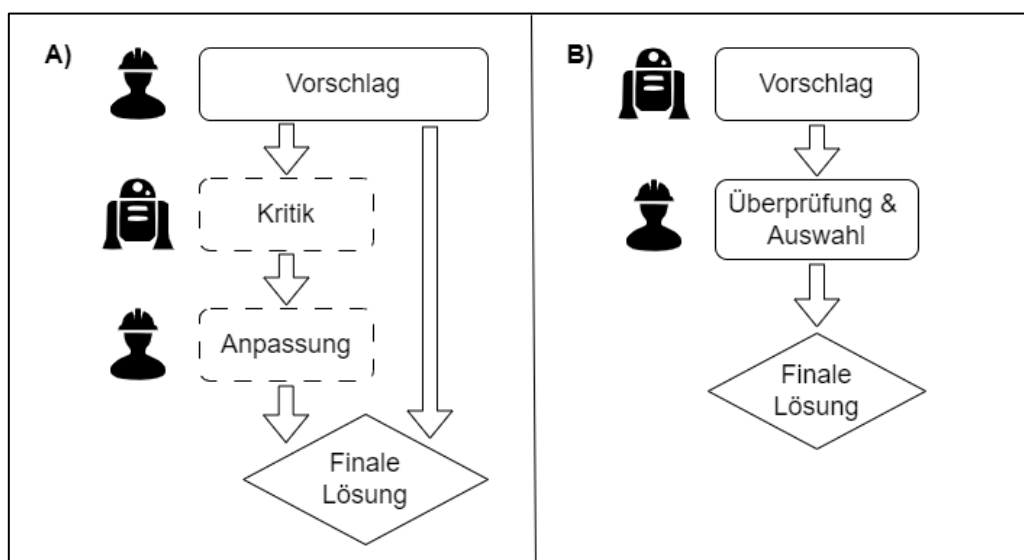


Abbildung 34. Verschiedene Formen der Entscheidungsunterstützung A) Kritikbasiert B) Vorschlagsbasiert.

Ein Argument für den Einsatz solcher kritikbasierter Entscheidungshilfen ist der sogenannte *Decision Bias*. Verschiedene Studien haben gezeigt, dass vorschlagsbasierte Entscheidungshilfen die Leistung verbessern können, wenn die Automation fehlerfrei arbeitet. Allerdings können sie auch zu einer Verschlechterung der Leistung führen, wenn die Automation nicht perfekt arbeitet, da die Entscheidungen der Nutzer durch die Vorschläge der Automation beeinflusst werden können (Metzger & Parasuraman, 2005; Rovira et al., 2007; Reichenbach et al., 2011). Kritikbasierte Entscheidungshilfen könnten diese Voreingenommenheit reduzieren, da der Nutzer eine aktivere Rolle im Entscheidungsprozess einnimmt (Endsley, 2017).

Es sollte jedoch erwähnt werden, dass vorschlagsbasierte Entscheidungshilfen die Entscheidungszeit verkürzen können. Dieser Effekt ist bei kritikbasierten Entscheidungshilfen nicht zu erwarten, da der Nutzer zunächst eigenständig eine Entscheidung treffen muss, bevor eine Unterstützung des Systems erfolgt.

3.4.2 Aufbau

Die Bereitstellung der zuvor beschriebenen Fähigkeiten des Delegationsagenten erfordert die Integration verschiedener Module und Schnittstellen. Abbildung 35 gibt einen Überblick über diese Module und Schnittstellen.

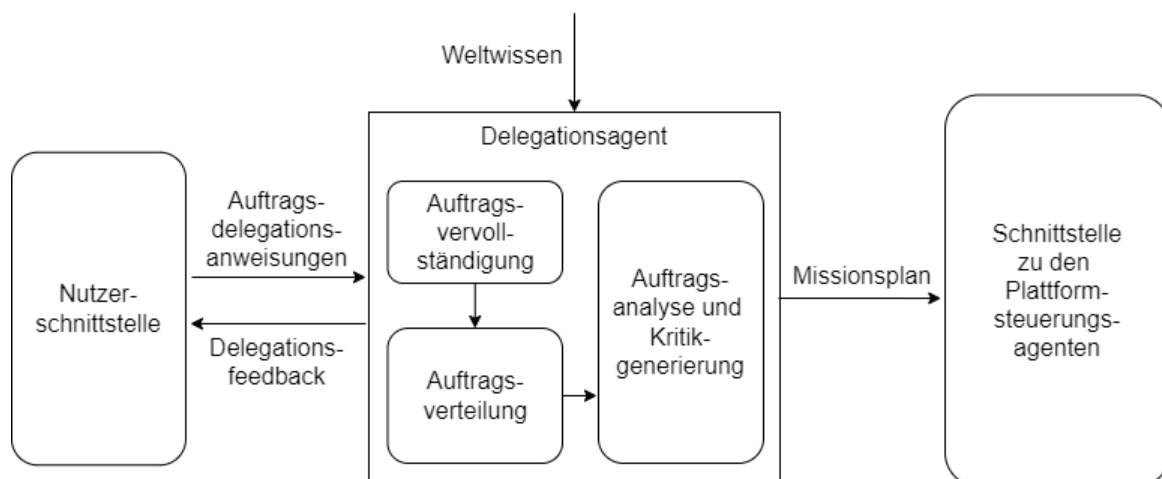


Abbildung 35. Module und Schnittstellen des Delegationsagenten.

Im Folgenden werden die Module des Delegationsagenten sowie die über die Schnittstellen ausgetauschten Informationen genauer beschrieben.

Auftragsvervollständigung: Dieses Modul ist in der Lage, unvollständig spezifizierte Aufträge zu vervollständigen. Abhängig vom Typ der zu spezifizierenden Informationen stellt die Auftragsvervollständigung hierfür entweder Standardwerte bereit oder sie bestimmt die Werte anhand der Situation (z. B. durch Berechnung eines Optimums).

Auftragsverteilung: Dieses Modul ist dafür zuständig, Aufträge an die Plattformen des Teams zu allokalieren. Hierfür verwaltet das Modul den Missionsplan, der die Zuordnung von Aufträgen an die Plattformen des Teams definiert und es verfügt über Planungswerkzeuge zur Bearbeitung und Optimierung des Missionsplans.

Auftragsanalyse und Kritikgenerierung: Dieses Modul ist für die kritische Bewertung der vom Nutzer delegierten Aufträge zuständig. Zu diesem Zweck analysiert das Modul jeden Auftrag und überprüft, ob der Auftrag negative Auswirkungen auf das Missionsergebnis haben könnte. Ist dies der Fall, generiert das Modul eine Meldung und schlägt gegebenenfalls Anpassungen des Auftrags vor.

Auftragsdelegationsanweisungen: Der Nutzer erteilt dem Delegationsagenten verschiedene Anweisungen, welche die Delegation von Aufträgen oder die Überarbeitung bereits erteilter Aufträge zum Ziel haben.

Delegationsfeedback: In Folge von Anweisungen erhält der Nutzer Feedback vom Delegationsagenten. Dies umfasst erkannte Nutzereingaben, die Darstellung von Aufträgen und deren Plattformzuordnung sowie die Benachrichtigungen im Rahmen der Auftragskritik.

Weltwissen: In seinen Modulen berücksichtigt der Delegationsagent verschiedene Informationen, die sich aus einer Repräsentation des Zustandes der Umgebung ergeben (*Weltwissen*). Hierunter fallen die taktische Lage sowie der Zustand der Plattformen des Teams.

Missionsplan: Der Delegationsagent stellt den Plattformsteuerungsagenten eine Kopie des von ihm verwalteten Missionsplans zur Verfügung.

3.5 Plattformsteuerungsagent

Dieser Abschnitt befasst sich mit dem Plattformsteuerungsagenten. Hierfür werden in Unterabschnitt 3.5.1 die **Aufgaben des Agenten** beschrieben, während Unterabschnitt 3.5.2 die **Architektur** zur Umsetzung dieser Aufgaben erläutert.

3.5.1 Aufgaben

Der Plattformsteuerungsagent ist für die Kontrolle eines unbemannten Luftfahrzeuges zuständig. In dieser Funktion ist der Agent für die Bearbeitung der vom Nutzer erteilten Aufträge verantwortlich, das heißt der Agent bricht die Aufträge herunter und steuert die Plattform-Automation im Sinne dieser Aufträge an. Dabei hat er sowohl die taktische Lage als auch die ihm zugewiesenen Verhaltensrichtlinien zu berücksichtigen. Die Aufgabe des Agenten wird in Abbildung 36 veranschaulicht.

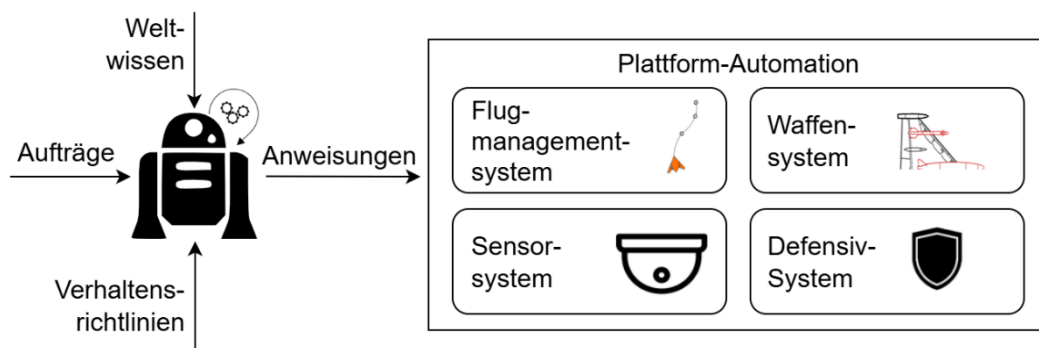


Abbildung 36. Aufgabe des Plattformsteuerungsagenten.

Die vom Nutzer definierten Verhaltensrichtlinien sind dabei als Ergänzung zur auftragsbasierten Führung zu betrachten. Diese Ergänzung ist vorteilhaft, weil bestimmte Anforderungen an das Verhalten der unbemannten Plattformen nicht anhand von Aufträgen definiert werden können, jedoch durch den Nutzer modifizierbar sein sollen (z. B. das Verhalten eines UAVs ohne Auftrag).

3.5.2 Aufbau

Damit der Agent die Plattform-Automation im Sinne der Nutzeranweisungen ansteuern kann, benötigt der Agent eine Funktion zur Verhaltensgenerierung sowie verschiedene Schnittstellen zu Nutzer, Delegationsagent und Plattform. In Abbildung 37 sind der Agent und dessen Schnittstellen dargestellt.

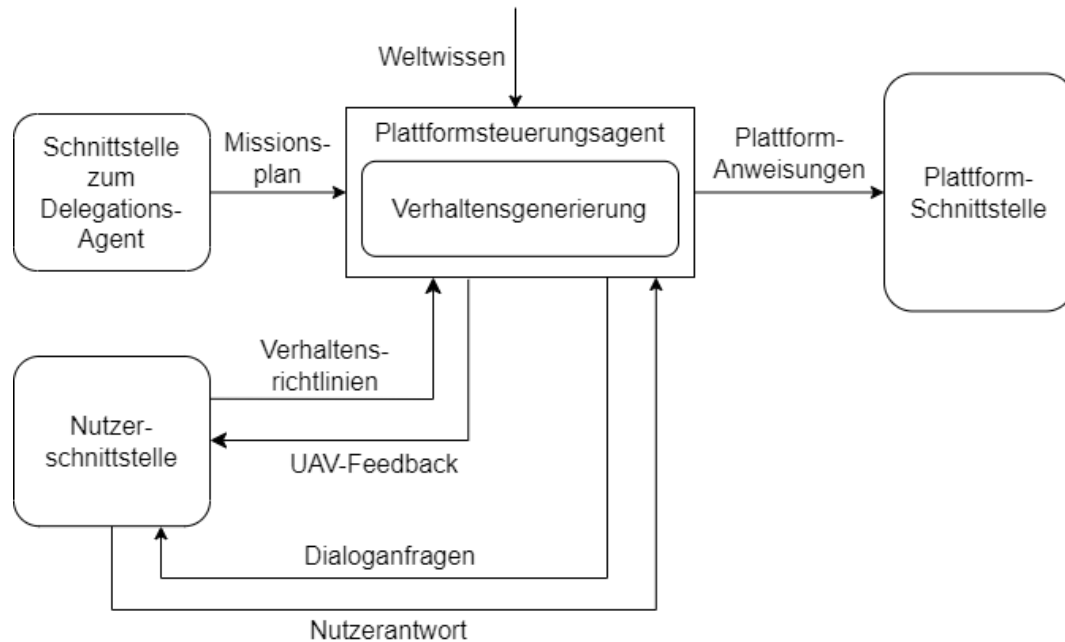


Abbildung 37. Module und Schnittstellen des Plattformsteuerungsagenten.

Nachfolgend wird die Verhaltensgenerierung des Plattformsteuerungsagenten sowie die über die Schnittstellen ausgetauschten Informationen genauer beschrieben.

Verhaltensgenerierung: Dieses Modul erzeugt die Verhaltensweisen, die zur Ausführung der Nutzeraufträge erforderlich sind. Zu diesem Zweck wägt die Verhaltensgenerierung die verschiedenen Ziele des Agenten (z.B. Auftragserfüllung, Selbstschutz) gegeneinander ab, plant die Aktionen des Agenten zur Zielerreichung und steuert die Plattform-Automation entsprechend der gewählten Aktionen an.

Plattform-Anweisungen: Die Ansteuerung der konventionellen Automation erfolgt in Form von Anweisungen an die verschiedenen Plattform-Systeme. Diese Anweisungen umfassen beispielsweise Höhenvorgaben an den Autopiloten, Ausrichtungsvorgaben für das Sensorsystem und Zielvorgaben für das Waffensystem.

Missionsplan: Der Plattformsteuerungsagent erhält vom Delegationsagenten den Missionsplan. Aus diesem Missionsplan entnimmt der Plattformkontrollagent die Aufträge, die seine eigene Plattform betreffen, und arbeitet sie der Reihe nach ab. Das bedeutet, dass die Aktionen des Agenten immer mit dem nächsten noch nicht abgeschlossenen Auftrag im Plan verbunden sind und der Agent keine Aktionen durchführt, die für spätere Aufträge relevant sind. Dies soll einerseits zu einer besseren Nachvollziehbarkeit des Agentenverhaltens führen und andererseits eine Reduktion der Entscheidungscomplexität bewirken.

Verhaltensrichtlinien: Der Plattformsteuerungsagent erhält vom Nutzer verschiedene Verhaltensrichtlinien. Diese Richtlinien definieren das gewünschte Verhalten der Plattform in spezifischen Situationen, z. B. wie sich die Plattform verhalten soll, wenn ihr kein Auftrag zugewiesen wurde. Die Verhaltensrichtlinien gelten auftragsunabhängig und zeitlich unbefristet. Der Nutzer kann die Richtlinien sowohl individuell für einen einzelnen Plattformsteuerungsagenten als auch kollektiv für mehrere Agenten festlegen.

Weltwissen: Zur Verhaltensgenerierung muss der Plattformsteuerungsagent verschiedene Informationen berücksichtigen. Hierzu zählt die taktische Lage und Umweltinformationen wie Wetter und Geländeverläufe.

UAV-Feedback: Der Agent informiert den Nutzer über den Status der Plattform und über den Bearbeitungsfortschritt der ihm zugewiesenen Aufträge. Dazu gehören z. B. Mitteilungen über abgeschlossene Aufträge sowie eine Beschreibung des eigenen Vorgehens.

Dialoganfragen: Durch Dialoganfragen wird der Nutzer dynamisch in die Entscheidungsfindung des Agenten miteinbezogen. Dialoganfragen werden in Fällen eingesetzt, in denen eine Vorgabe des Nutzers wünschenswert ist, aber diese Vorgabe weder durch Delegation von Aufträgen noch durch vordefinierte Verhaltensregeln abgedeckt werden kann. Hierunter fällt beispielsweise eine Entscheidung, ob ein möglicherweise feindliches Luftfahrzeug bekämpft werden soll.

Nutzerantworten: Bei den Nutzerantworten handelt es sich um die Rückmeldung des Nutzers auf Dialogmeldungen.

4. Implementierung und Integration

Dieses Kapitel befasst sich mit der Implementierung und Integration des Führungs- und Interaktionskonzepts. Dazu wird in Abschnitt 4.1 der **Kampfflugzeugsimulator** vorgestellt, in dem das konzipierte Entwurfsmuster integriert und erprobt werden soll. Anschließend werden in Abschnitt 4.2 die **Gestaltungsmaßnahmen der Nutzerschnittstelle** vorgestellt, bevor in Abschnitt 4.3 die **funktionale Umsetzung spezifischer Schnittstellenelemente** beschrieben wird. Daran anschließend behandelt Abschnitt 4.4 die **Umsetzung der Module des Delegationsagenten und des Plattformsteuerungsagenten**. Abschnitt 4.5 rundet das Kapitel mit einer Beschreibung der **Architektur des Gesamtsystems** ab.

4.1 Simulationsumgebung

Als Simulationsumgebung dient der Kampfflugzeugsimulator der Professur für Flugmechanik und Flugführung an der Universität der Bundeswehr München, der gemeinschaftlich von mehreren Mitarbeitern in verschiedenen Forschungsprojekten entwickelt wurde. Der Simulator verfügt über zwei Projektionssysteme zur Darstellung der Außensicht und vier Hardware-Cockpits, wovon zwei Cockpits einen vorderen Cockpitsitz und die anderen zwei einen hinteren Cockpitsitz abbilden. Die Außensichtssysteme und Cockpits können in verschiedenen Konfigurationen betrieben werden, sodass ein breites Missionsspektrum von der Simulation eines einsitzigen Kampfflugzeuges bis hin zur Simulation von zwei zweisitzigen Kampfflugzeugen abgebildet werden kann (Abbildung 38).



Abbildung 38. Kampfflugzeugsimulator der Professur für Flugmechanik und Flugführung.

Die Ausstattung der Cockpits besteht aus drei Touchscreen-Displays, einem Blickbewegungsmesssystem und einem Headset. Die vorderen Cockpits sind zusätzlich mit Bedienelementen zur Flugsteuerung bestehend aus Steuerknüppel, Schubhebel und Ruderpedalen ausgestattet. Abbildung 39 zeigt den Aufbau eines Vordercockpit inklusive der zugehörigen Außensicht. Nachfolgend werden die darin enthaltenen Elemente genauer beschrieben.

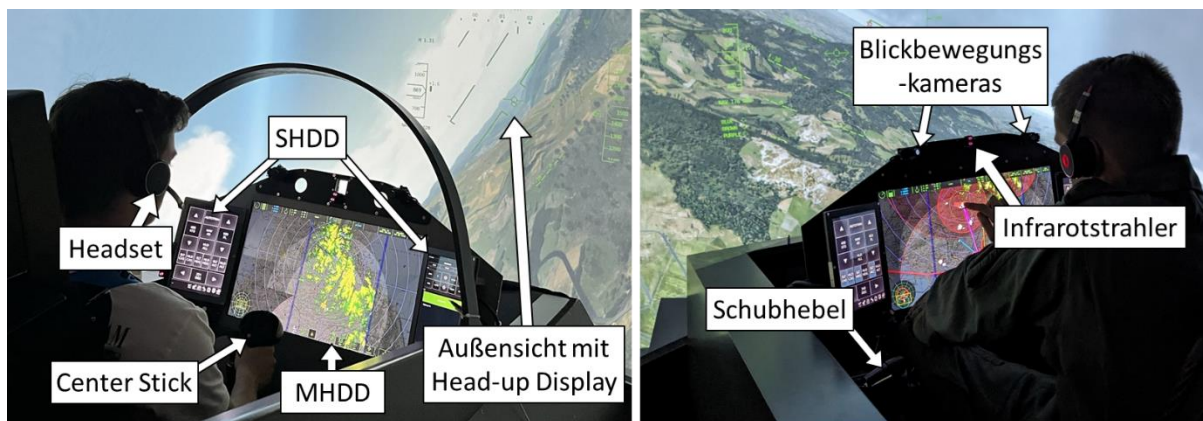


Abbildung 39. Elemente des Cockpits und der Außensicht.

Das *MHDD* (Multifunctional Head-Down Display) dient als zentrales Bedien- und Anzeigeelement und besteht aus einem 24-Zoll Touchscreen-Display. Im Fokus des *MHDD* steht die taktische Karte, mit der positionsbezogene Informationen (z. B. verbündete und feindliche Truppen) in übersichtlicher Form dargestellt werden können.

Die beiden *SHDDs* (Side Head-Down Display) sind neben dem *MHDD* angebracht und bestehen jeweils aus einem 10-Zoll Touchscreen-Display. Auf einem *SHDD* können verschiedene Seiten zur Anzeige und Interaktion dargestellt werden, die mithilfe eines Reiters am unteren Bildschirmrand ausgewählt werden können. In der Initialkonfiguration wird auf dem linken *SHDD* der Missionsplan dargestellt, während das rechte *SHDD* eine Übersicht über die Ressourcen und die aktuellen Aufträge aller Kräfte enthält.

Die Außensicht visualisiert die simulierte Umgebung des bemannten Kampfflugzeugs. In die Außensicht ist zudem ein Head-up Display integriert, auf welchem Informationen zur Flugführung und zur taktischen Lage angezeigt werden. Für die Darstellung der Außenansicht wird ein Projektionssystem der Firma Q4³² eingesetzt, welches über eine sphärische Leinwand mit einem darstellbaren Sichtfeld von 200 x 45 Grad verfügt.

Die Blickbewegungsmessung erfolgt mithilfe eines kamerabasierten Messsystems der Firma SmartEye.³³ Das Messprinzip dieses Systems beruht auf der Analyse zweier optischer Merkmale: des Zentrums der Pupille und der Reflexion von Infrarotlicht an der Kornea, deren relative Position sich mit der Augenbewegung verändert und so die Bestimmung der Blickrichtung ermöglicht (Duchowski, 2007). Zur Umsetzung dieses Messprinzips sind im Cockpit drei Infrarotstrahler und vier Kameras angebracht. Laut Hersteller kann mit dem installierten Messsystem die Blickrichtung mit einer Frequenz von 60 Hz und einer Winkelgenauigkeit von bis zu 0,5 Grad aufgezeichnet werden. Die Umrechnung der Blickrichtung zu einer Pixel-Position auf dem jeweiligen Display erfolgt mithilfe eines Weltmodells, in dem die Position von Kameras, Infrarotstrahlern und Displays verzeichnet ist.

³² Q4 DirectVue (<https://q4services.com/q4-directvue-direct-dome-projection/>)

³³ SmartEye Pro (<https://smarteye.se/>)

Unter den geometrischen Rahmenbedingungen des Cockpits kann die Blickposition auf dem MHDD mit einer Genauigkeit von circa 1 cm ermittelt werden.

Zur Flugsteuerung des bemannten Luftfahrzeugs verfügt das Cockpit über ein HOTAS-System³⁴ sowie über Ruderpedale. Das HOTAS-System der Firma Reiser Systemtechnik ist baugleich mit den im Eurofighter Typhoon verbauten Steuerelementen und verfügt damit über einen mittig angebrachten Steuerknüppel (Center Stick) und einen differenziell ansteuerbaren Schubhebel. Auf beiden Steuerelementen sind verschiedene Knöpfe angebracht, die für die Systembedienung genutzt werden können.

Die Audioausgabe und Tonaufnahme erfolgt über ein Jabra³⁵ Headset. Das Headset wird zum Abspielen von Fluggeräuschen, zur Übermittlung auditiver Systemmeldungen, zur simulierten Funk-Kommunikation zwischen mehreren Nutzern sowie zum Absetzen von Sprachbefehlen verwendet.

³⁴ HOTAS (**H**ands **O**n **T**hrottle **A**nd **S**tick) ist ein Bedienkonzept, bei dem die Hände des Piloten auf Steuerknüppel und Schubhebel verbleiben können. Der Begriff HOTAS wird auch für Eingabeelemente verwendet, die aus diesen beiden Komponenten bestehen.

³⁵ <https://www.jabra.com/de-de/business/office-headsets/jabra-evolve/jabra-evolve-75>

4.2 Gestaltung der Nutzerschnittstelle

Dieser Abschnitt befasst sich mit der Gestaltung der Nutzerschnittstelle. Dabei werden im Rahmen des Unterabschnitts 4.2.1 zunächst die **Interaktionsmethoden zur Erteilung von Aufträgen** vorgestellt, bevor in Unterabschnitt 4.2.2 das **Management von Verhaltensrichtlinien** beschrieben wird, welches die Erteilung von Aufträgen ergänzt. Anschließend befasst sich Unterabschnitt 4.2.3 mit der schnittstellenseitigen Umsetzung der **Auftragskritik**, bevor in Unterabschnitt 4.2.4 das **Feedback von UAVs** beschrieben wird. Der Unterabschnitt 4.2.5 schließt die Gestaltung der Nutzerschnittstelle mit einer Beschreibung der **Nutzerdialoge** ab.

4.2.1 Interaktionsmethoden zur Auftragsdelegation

Die Bereitstellung von schnellen und bedarfsgerechten Interaktionsmethoden zur Auftragsdelegation ist eine der Fähigkeiten des Delegationsagenten. Nachfolgend wird die Implementierung dieser Fähigkeit beschrieben.

Da sich alle betrachteten Interaktionen auf die Delegation von Aufträgen beziehen, werden im Folgenden die Begriffe *Interaktionsmethode* und *Delegationsmethode* synonym verwendet.

Zur Delegation von Aufträgen wurden vier verschiedene Interaktionsmethoden entwickelt. Diese Methoden unterscheiden sich in ihren individuellen Eingabemodalitäten, in den Anteilen von Nutzer und Delegationsagent bei der Auftragspezifikation, in den verfügbaren Scheduling-Modi und darin, ob die Interaktion einen Review-Schritt beinhaltet. Tabelle 8 gibt einen Überblick über diese Eigenschaften der Delegationsmethoden.

Tabelle 8. Überblick über die wesentlichen Eigenschaften der verschiedenen Delegationsmethoden.

Methode	Modalität	Auftragspezifikation				Scheduling Modi	Review
		Auftrags-typ	Ziel-objekt	Erfolgs-kriterium	Para-meter		
Timeline	Touchscreen (Tippen)	Nutzer	Nutzer	Wahl-weise	Wahl-weise	Alle	Ja
Touch-Gesture	Touchscreen (Gesten)	Agent	Nutzer	Agent	Agent	Team (sofort), Plattform (sofort)	Nein
Voice	Sprache & Manuell	Nutzer	Nutzer	Agent	Agent	Team, Plattform, Zeitslot	Ja
Gaze-HOTAS	Blick-bewegung & Manuell	Agent	Nutzer	Agent	Agent	Team, Plattform	Ja

Nachfolgend werden die Delegationsmethoden vorgestellt und deren Gestaltung motiviert.

4.2.1.1 Timeline

Diese Delegationsmethode basiert auf diskreten Touch-Interaktionen mit Schaltflächen in der taktischen Karte und der Missionsplanansicht. Die Methode zielt darauf ab, eine skalierbare

Nutzerkontrolle zu ermöglichen, d.h. dass der Benutzer flexibel entscheiden kann, welche Vorgaben er im Prozess der Aufgabendelegation machen möchte.

Zu diesem Zweck ist die Nutzerinteraktion so gestaltet, dass bestimmte Informationen vom Nutzer zwingend spezifiziert werden müssen, während andere Informationen optional angegeben werden können. Auftragsdelegationen erfordern dabei immer die Spezifikation von Auftragsstyp und Zielobjekt, während das Erfolgskriterium, der Scheduling-Modus und die individuellen Parameter des Auftrags optional vom Nutzer spezifiziert werden können. Die Methode Timeline ist die einzige Methode, welche dem Nutzer den Zugriff auf das Erfolgskriterium und die Auftragsparameter ermöglicht.

Die Verwendung einer Touchscreen-basierten Interaktionsmethode ist für diese Anwendung aus verschiedenen Gründen vorteilhaft. Ein wesentlicher Aspekt ist, dass der Nutzer die Konfigurationsmöglichkeiten auf dem Touchscreen sehen kann und sich diese nicht merken muss (Johnson, 2010, S. 113). So können die verschiedenen Erfolgskriterien und Parameter als Auswahloptionen in Menüs angeordnet werden, sodass der Nutzer eine übersichtliche Darstellung über die verfügbaren Optionen erhält. Zudem bieten Touchscreens eine intuitive Möglichkeit (Holzinger, 2002) zur Auswahl dieser Optionen, da eine Berührung der entsprechenden Schaltflächen ausreicht, um eine Auswahl vorzunehmen. Diese Intuitivität führt auch dazu, dass Touchscreen-Interaktionen mit wenig Training leicht zu bedienen sind, was insbesondere bei selten genutzten Konfigurationsoptionen von Vorteil sein kann (Ng et al., 2013; Tao et al., 2018).

Um eine Auftragsdelegation mit der Delegationsmethode Timeline durchzuführen, muss der Nutzer mehrere Teilschritte ausführen. Diese Teilschritte sind in Abbildung 40 beschrieben und werden nachfolgend genauer erläutert.

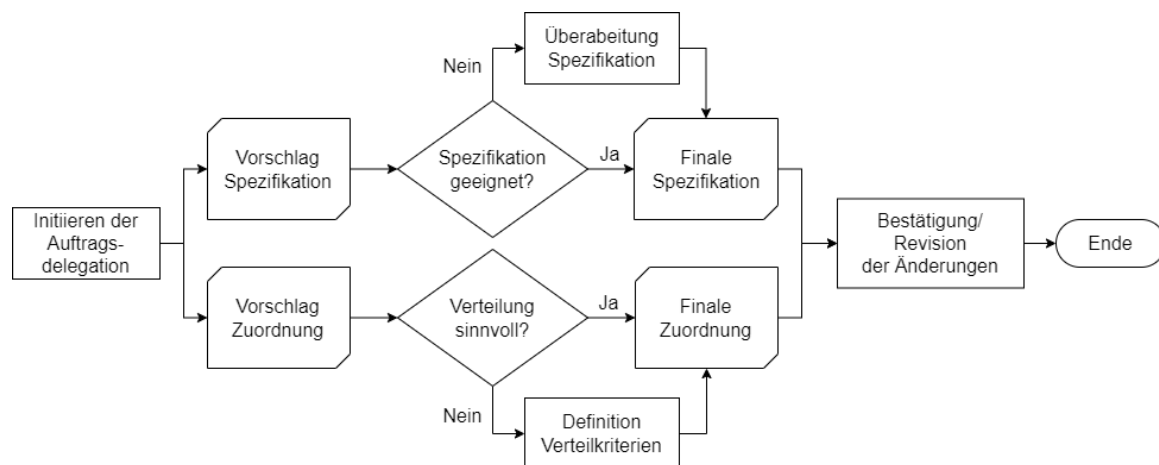


Abbildung 40. Ablauf von Delegation mit der Delegationsmethode Timeline.

Der Nutzer initiiert die Auftragsdelegation, indem er das Zielobjekt des zu erstellenden Auftrags in der taktischen Karte durch Antippen auswählt. Diese Interaktion öffnet ein hierarchisch unterteiltes Radial-Menü (Callahan et al., 1988) am Zielobjekt. Dieses Menü enthält in der ersten Ebene eine Option zum Anlegen eines Auftrags (Abbildung 41 links). Wählt der Nutzer diese Option aus, so werden alle möglichen Auftragsstypen für das gewählte Zielobjekt angezeigt (Abbildung 41 rechts).



Abbildung 41. Radial-Menü nach Auswahl eines Objektes (links) und zweite Ebene nach Auswahl der Option "Task" (rechts).

Nach der Auswahl des gewünschten Auftragsstyps erstellt das System einen ersten Entwurf für eine Auftragspezifikation, bei der das Erfolgskriterium und die Auftragsparameter vom System vordefiniert wurden. Außerdem verteilt das System den Auftrag mit dem Scheduling-Modus „Team“ an die geeignetste Plattform und stellt den resultierenden Missionsplan in der Missionsplanansicht auf dem linken SHDD dar. Die Spezifikation des Auftrags und dessen Zuordnung können anschließend vom Nutzer überarbeitet werden.

Zur Anpassung der Auftragspezifikation steht dem Nutzer ein Fenster am linken unteren Ende des MHDDs zur Verfügung. Dieses Fenster zeigt die möglichen Ausprägungen von Erfolgskriterien und Parametern an, und der Nutzer hat die Möglichkeit jedes Element des Auftrags manuell zu spezifizieren (Abbildung 42).

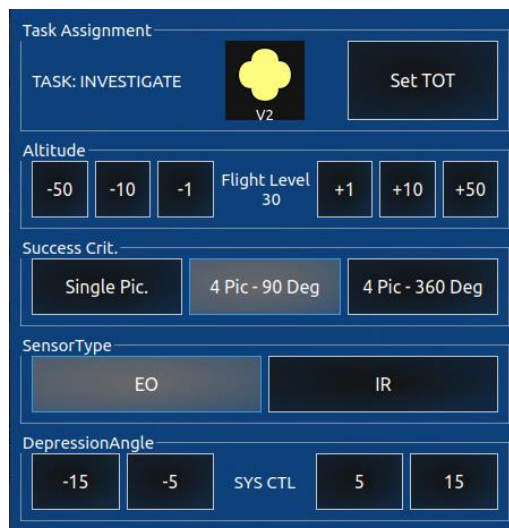


Abbildung 42. Fenster zur manuellen Spezifikation des Aufklärungsauftrags.

Die Zuordnung von Aufträgen kann der Nutzer direkt in der Missionsplanansicht verändern. Hierfür stehen dem Nutzer verschiedene Schaltflächen zur Verfügung, die mit unterschiedlichen Scheduling-Modi verknüpft sind (Abbildung 43). So kann der Nutzer zum Beispiel das Symbol eines Teammitglieds antippen, um diesem den Auftrag zuzuordnen (Scheduling Modus „Plattform“).

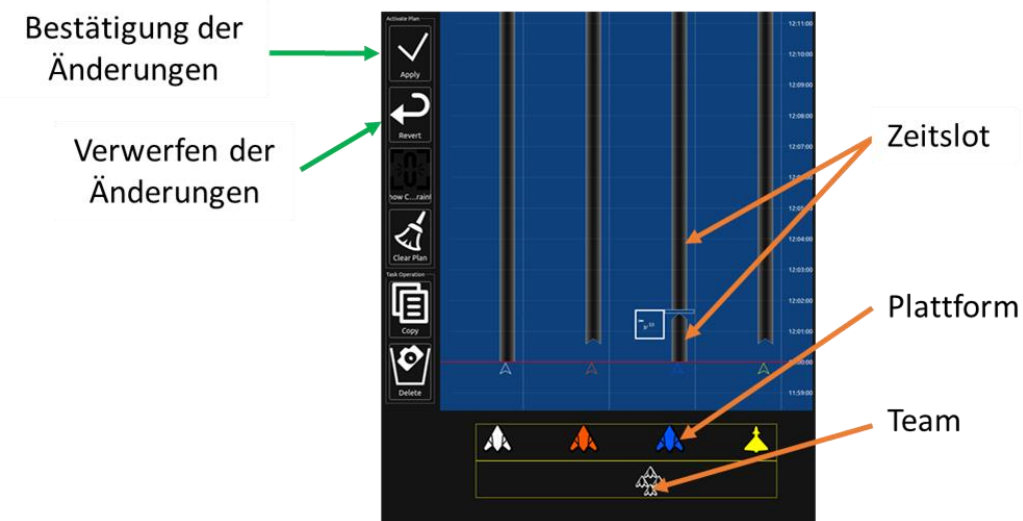


Abbildung 43. Schaltflächen zur Auswahl des Scheduling-Modus (orange) und zur Bestätigung / Ablehnung des Missionsplanes (grün) in Missionsplanungsansicht.

Nachdem der Nutzer den Auftrag und dessen Zuordnung nach seinen Vorstellungen angepasst hat, kann er entweder weitere Aufträge delegieren oder die vorgenommenen Änderungen am Missionsplan bestätigen bzw. rückgängig machen. Dazu stehen dem Nutzer zwei Schaltflächen neben der Missionsplanansicht zur Verfügung (Abbildung 43).

4.2.1.2 Touch-Gesture

Diese Methode basiert auf der Nutzung von Touchscreen-Gesten wie Wischen oder Doppelklicken. Ziel der Methode ist es, dass Nutzerabsichten möglichst schnell in missionswirksames Verhalten der unbemannten Kräfte umgesetzt werden können.

Zu diesem Zweck optimiert die Methode sowohl die Interaktionszeiten für die Auftragsdelegation als auch die Zeit bis zum Beginn der Auftragsbearbeitung. Die Optimierung der Interaktionszeiten erfolgt durch eine Interaktionsgestaltung, die einen minimalen Interaktionsaufwand für den Nutzer zur Folge hat. In diesem Zuge wird ebenfalls auf den nachgelagerten Review-Prozess der Änderungen verzichtet. Der Beginn der Auftragsbearbeitung wird insofern optimiert, als dass Aufträge stets zeitnah durchgeführt werden, d. h. vor bereits bestehenden Aufträgen in den Missionsplan eingefügt werden.

Diese Randbedingungen erklären auch die Wahl von Touchscreen-Gesten als Eingabemodalität für diese Delegationsmethode. So kann der Einsatz von Touchscreen-Gesten zu geringeren Interaktionszeiten gegenüber schaltflächenbasierten Bedienkonzepten führen, insbesondere wenn die Aufmerksamkeit des Nutzers nicht vollständig auf dem zu bedienenden Display liegt (Bragdon et al., 2011). Gleichzeitig können Touchscreen-Gesten, unter der Voraussetzung eines geeigneten Konzeptes, als robust genug für sicherheitskritische Anwendungen betrachtet werden, was insbesondere im Hinblick auf den ausbleibenden Review-Prozess von Bedeutung ist (Breuninger, 2020).

Die Delegationsmethode Touch-Gesture sieht vier verschiedene Modi zur Delegation von Aufträgen vor. Diese Modi sind in Tabelle 9 beschrieben.

Tabelle 9. Delegationsmodi mit der Methode Touch-Gesture.

Name	Beschreibung
Team-Ziel	Bei dieser Form der Auftragsdelegation spezifiziert der Nutzer lediglich das Zielobjekt. Das System wählt anhand von Art und Zustand des Zielobjektes den Auftragstyp aus, und wählt das Erfolgskriterium, die Parameter des Auftrags sowie die Plattform zur Auftragsdurchführung aus. ³⁶
Plattform-Ziel	Der Nutzer spezifiziert das Ziel der Auftragsdelegation und die Plattform, die den Auftrag ausführen soll. Das System vervollständigt die Auftragsspezifikation anhand von Art und Zustand des Zielobjektes.
Plattform-Navigation	Der Nutzer wählt eine Plattform aus und definiert eine Zielposition, zu der die Plattform fliegen soll.
Plattform-CAP	Der Nutzer wählt eine Plattform aus und definiert Position und Ausrichtung eines CAPs. ³⁷

Die verschiedenen Modi der Auftragsdelegation unterscheiden sich hinsichtlich des Ablaufes der Interaktion und der genutzten Gesten. Abbildung 44 stellt die Abläufe der Interaktion sowie die verwendeten Gesten für die verschiedenen Modi dar.

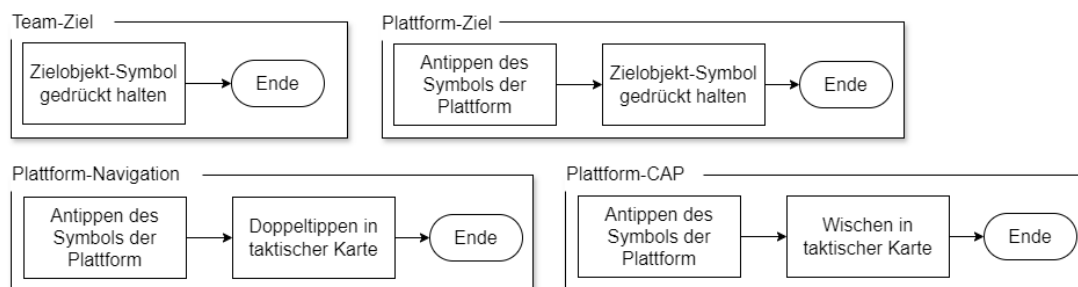


Abbildung 44. Ablauf der verschiedenen Modi der Auftragsdelegation mit der Methode Touch-Gesture

4.2.1.3 Voice

Diese Methode nutzt eine Kombination aus Spracheingaben und Hardwaretasten auf den Steuerelementen zur Auftragsdelegation.

Die Bereitstellung einer sprachbasierten Delegationsmethode bringt verschiedene Vorteile mit sich. So kann die Möglichkeit zur Spracheingabe zu einer Verbesserung der Leistung in Flugaufgaben beitragen, da die Hände des Nutzers währenddessen an den Steuerelementen verbleiben können und der Nutzer weniger Aufmerksamkeit auf die Displayelemente im Cockpit richten muss (Rosenhoover, 1987; Weinstein, 1990). Ein weiterer Vorteil von Spracheingabe ist die Möglichkeit zur Bildung von Macros. Hierunter versteht man die Zusammenfassung mehrerer Interaktionsschritte zu einer einzelnen Interaktion. Der Einsatz solcher Macros ist bei der Spracheingabe praktikabler als bei anderen Modalitäten,

³⁶ Wie die Auswahl von Auftragstyp, Erfolgskriterium und Auftragsparametern umgesetzt wird, wird im Rahmen der Implementierung der Auftragsvervollständigung in 4.4.2 beschrieben.

³⁷ Hierbei wird ein Combat Air Patrol Task angelegt. Im Rahmen dieses Auftrags richtet das UAV seine Sensoren entlang der CAP-Richtung aus und bekämpft ankommende Feindflugzeuge.

weil lediglich ein zusätzlicher Sprachbefehl definiert werden muss, während z. B. bei einer Touchscreen-basierten Interaktion eine sorgfältige Abwägung erforderlich ist, ob die Bereitstellung einer zusätzlichen Schaltfläche für das entsprechende Macro sinnvoll ist (Cohen & Oviatt, 1995; Calhoun & Draper, 2006).

Die Ergänzung der Sprachmodalität um eine manuelle Interaktion zur Aktivierung der Spracheingabe dient dazu, die Interaktion mit dem System robuster zu gestalten. Die manuelle Aktivierung der Spracheingabe verhindert, dass die Spracherkennung kontinuierlich mitläuft und somit sprachliche Äußerungen des Nutzers oder Störgeräusche fälschlicherweise als Auftragserteilung interpretiert werden (Képuska, 2011). Der hier beschriebene Effekt könnte ebenfalls durch die Nutzung von sogenannten *Wake-up-Words* erzielt werden. Hierbei handelt es sich um Wörter, die dem eigentlichen Sprachbefehl vorangestellt werden, um die Sprachsteuerung zu aktivieren. Die Verwendung dieser *Wake-up-Words* hat jedoch gegenüber der manuellen Interaktion den Nachteil, dass die Interaktion länger dauert, da das Aussprechen der Wörter mehr Zeit in Anspruch nimmt als das Drücken einer Taste.

Die Delegationsmethode Voice stellt dem Nutzer drei verschiedene Modi zur Auftragsdelegation bereit. Diese verschiedenen Modi sind in Tabelle 10 beschrieben.

Tabelle 10. Delegationsmodi der Methode Voice.

Name	Beschreibung
Team-Auftragstyp-Ziel	Der Nutzer spezifiziert Auftragstyp und Zielobjekt. Das System wählt das Erfolgskriterium, die Parameter des Auftrags sowie die Plattform und den Zeitpunkt zur Auftragsdurchführung aus.
Plattform-Auftragstyp-Ziel	Der Nutzer spezifiziert die Plattform, den Auftragstyp und das Zielobjekt. Das System wählt das Erfolgskriterium und die Parameter des Auftrags und legt den Zeitpunkt der Auftragsdurchführung fest.
Zeitslot-Auftragstyp-Ziel	Der Nutzer spezifiziert den Auftragstyp, das Zielobjekt sowie Plattform und Zeitslot der Auftragsdurchführung. Das System wählt das Erfolgskriterium und die Parameter des Auftrags und ermittelt den genauen Zeitpunkt der Auftragsdurchführung.

Die Delegation eines Auftrags mit der Delegationsmethode Voice erfolgt durch Betätigen eines Push-To-Talk-Buttons auf dem Schubhebel und der Formulierung des entsprechenden Sprachbefehls durch den Nutzer (Abbildung 45).

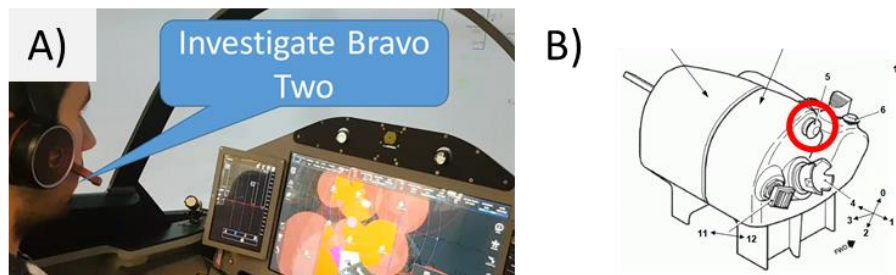


Abbildung 45. Auftragsdelegation mit Delegationsmethode Voice. A) Sprachbefehl B) Push-To-Talk Button.

Die Sprachbefehle zur Auftragsdelegation sind an die *Multi-Service Brevity Codes*³⁸ der NATO angelehnt und folgen einer strikt einzuhaltenden Syntax. Diese Syntax ist abhängig vom Modus der Auftragsdelegation. Tabelle 11 stellt die Syntax der verschiedenen Modi und Beispiele für Sprachbefehle dar. Eine detaillierte Beschreibung des Vokabulars ist in Anhang E.3 zu finden.

Tabelle 11. Syntax und Beispiele von Sprachbefehlen für die verschiedenen Modi.

Modus	Syntax	Beispiel
Team- Auftragstyp-Ziel	<Auftragstyp> <Zielobjekt>	„Investigate bravo two“
Plattform-Auftragstyp-Ziel	<Plattform> <Auftragstyp> <Zielobjekt>	„Blue investigate bravo two“
Zeitslot-Auftragstyp-Ziel	<Plattform> <Auftragstyp> <Zielobjekt> <Präposition> <Referenz-Typ> <Referenz-Objekt>	„Blue investigate bravo two before engage bravo two“

Das System quittiert das Absetzen eines Sprachbefehls durch eine Darstellung der Übersetzung im Heads-up-Display. Im Falle eines gültigen Sprachbefehls initiiert das System die Auftragsdelegation und stellt das Planungsergebnis in der Missionsplanansicht dar. Ausgehend von diesem Punkt kann der Nutzer entweder weitere Aufträge delegieren oder die vorgenommenen Änderungen am Missionsplan bestätigen (Sprachbefehl „Accept“) bzw. rückgängig machen (Sprachbefehl „Decline“).

³⁸ Hierbei handelt es sich um Sprachcodes, mit denen umfangreiche Informationen und Befehle schnell und kompakt übermittelt werden können. Eine Beschreibung dieser Codes findet sich in U.S. Army (2023).

4.2.1.4 Gaze-HOTAS

Diese Methode nutzt eine Kombination aus Blickbewegungsmessung (*Gaze*) und Hardwaretasten auf den Steuerelementen (*HOTAS*) zur Auftragsdelegation.

Verschiedene Argumente sprechen für eine blickbewegungs-basierte Form der Auftragsdelegation. Eines dieser Argumente ist, dass die Blickbewegungsmessung kürzere Interaktionszeiten als andere Interaktionsmodalitäten ermöglichen könnte. So konnte in verschiedenen Studien gezeigt werden, dass bei bildschirmbasierten Interaktionsmethoden (z. B. mittels Touch oder Maus) die eigentliche Interaktion in der Regel von einer vorausseilenden Fixation des Blicks auf die entsprechenden Display-Elemente begleitet ist (Law et al., 2004; Bieg et al., 2010; Oviatt et al., 2017)³⁹. Wenn bei den untersuchten Interaktionsmethoden die Blickbewegung der eigentlichen Interaktion vorausgeht, liegt der Schluss nahe, dass der Einsatz der Blickbewegung im Rahmen der Interaktion zu einer Verkürzung der Interaktionsdauer führen könnte. Ein weiteres Argument für eine Eingabemethode auf Basis von Blickbewegung ist der Interaktionsaufwand. Die Auswahl von Objekten durch Betrachten stellt für den Nutzer eine einfache Interaktionsmöglichkeit dar (Sibert & Jacob, 2000) und kann im Vergleich zu anderen Interaktionsformen zu einer geringeren Ermüdung des Nutzers führen (Zhai et al., 1999). Ein drittes Argument ist, dass die Selektion von Objekten mittels Blickbewegung auch die Auswahl von Objekten außerhalb des Displays zulässt, womit beispielsweise Auftragsdelegationen durch Selektion im Head-up-Display möglich werden.

Eine Herausforderung bei der Verwendung von Blickbewegungen als Eingabemodalität für die Objektselektion besteht darin, festzustellen, ob das Betrachten eines Objekts mit einer Interaktionsabsicht verbunden ist oder nicht (Jacob, 1991; Oviatt et al., 2017). Aus diesem Grund erfolgt in vielen Fällen eine Kombination der Blickbewegungsmessung mit einer zweiten Eingabemodalität, die dafür genutzt wird, den Zeitpunkt der Objektauswahl zu steuern (M. Kumar et al., 2007; Yamato et al., 2000).

Ein solche Kombination findet sich auch in der Delegationsmethode Gaze-HOTAS wieder. Hier wird die Blickbewegungsmessung zur Objektauswahl verwendet, während Hardwaretasten auf den HOTAS-Steuerelementen dazu dienen, die Auswahl von Objekten und das Fortschreiten im Delegationsprozess zu steuern (Abbildung 46).

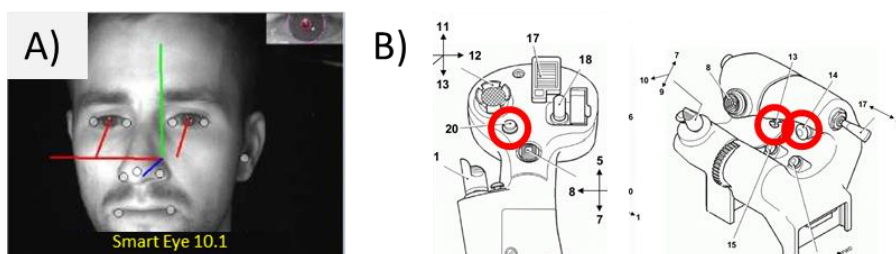


Abbildung 46. Modalitäten und deren Verwendung in Delegationemethode Gaze-HOTAS. A) Blickbewegungsmessung zur Objektauswahl B) Knöpfe auf den Steuerelementen zur Steuerung des Auswahl- und des Delegationsprozesses.

³⁹ Dieser Zusammenhang wird in der Literatur auch als *Target Gaze Behavior* bezeichnet

Die Delegationsmethode Gaze-HOTAS stellt dem Nutzer zwei Modi zur Auftragserteilung bereit. Diese Modi sind in Tabelle 12 beschrieben.

Tabelle 12. Delegationsmodi der Methode Gaze-HOTAS.

Name	Beschreibung
Team-Ziel	Der Nutzer spezifiziert das Zielobjekt. Das System wählt den Auftragsstyp, das Erfolgskriterium, die Parameter des Auftrags sowie die Plattform zur Auftragsdurchführung aus.
Plattform-Ziel	Der Nutzer spezifiziert die Plattform und das Zielobjekt. Das System wählt das Erfolgskriterium und die Parameter des Auftrags und legt den Zeitpunkt der Auftragsdurchführung fest.

Die verschiedenen Modi der Auftragsdelegation folgen dem gleichen Grundablauf. Dieser Ablauf ist in Abbildung 47 beschrieben und wird nachfolgend genauer beschrieben.

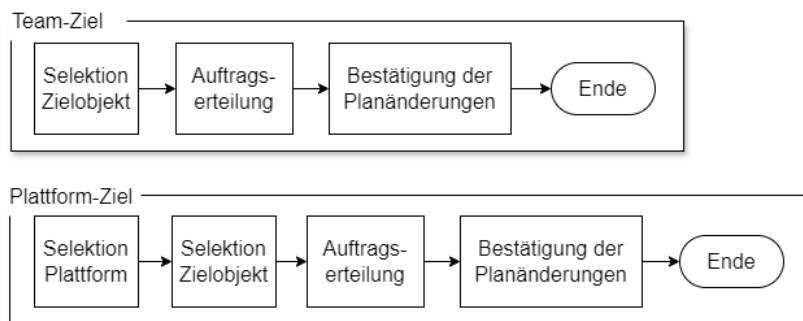


Abbildung 47. Ablauf von Delegationen mit Methode Gaze-HOTAS.

Auftragsdelegationen mit der Methode Gaze-HOTAS beginnen immer mit der Selektion von Objekten. Der Nutzer initiiert einen Selektionsprozess, indem er den Selektionsknopf auf dem Steuerknüppel gedrückt hält (Abbildung 48 A). Der Nutzer kann anschließend Objekte der taktischen Karte auswählen, indem er seinen Blick auf das entsprechende Objekt richtet. Die Selektion eines Objektes wird dem Nutzer durch einen animierten Auswahlmarker visualisiert (Abbildung 48 B). Wurde das gewünschte Objekt vom System ausgewählt, wird die Auswahl durch Loslassen des Selektionsbuttons eingerastet. Die Anzahl der selektierten Objekte hängt vom Delegationsmodus ab. Bei einer Auftragsdelegation mit dem Delegationsmodus „Team-Ziel“ durchläuft der Nutzer den Prozess einmalig und spezifiziert mithilfe der Selektion das Zielobjekt des Auftrags. Im Gegensatz dazu wählt der Nutzer bei einer Delegation mit dem Modus „Plattform-Ziel“ zuerst die ausführende Plattform und anschließend das Zielobjekt aus. Der Nutzer kann die Auswahl von Objekten mithilfe des „Abbruch“-Knopfes jederzeit zurücksetzen (Abbildung 48 C).

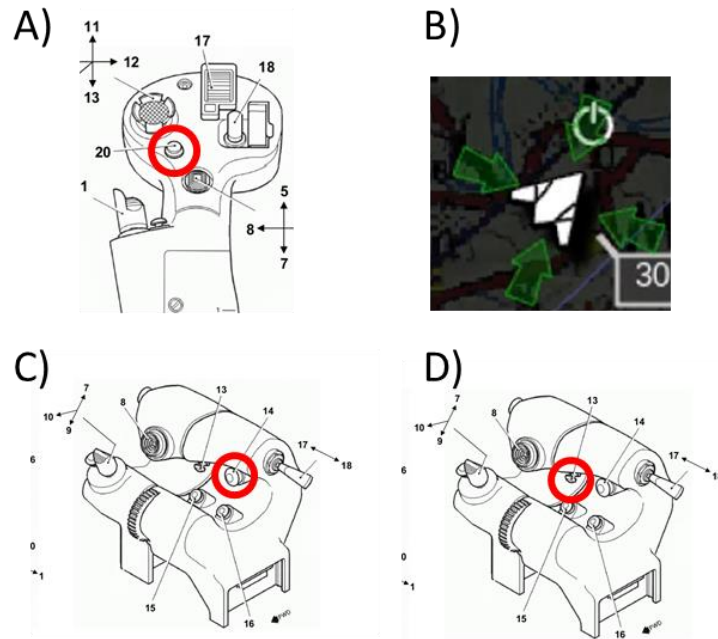


Abbildung 48. A) Selektionsknopf B) Auswahlmarker C) Abbruch-Knopf D) Weiter-Knopf.

Nachdem die Selektion von Objekten erfolgreich abgeschlossen wurde, setzt der Nutzer den Delegationsprozess durch Drücken des „Weiter“-Knopfes auf dem Schubhebel fort (Abbildung 48 D). Das System initiiert in Folge die Auftragsdelegation⁴⁰ und stellt den resultierenden Missionsplan in der Missionsplanansicht dar. Der Nutzer kann diesen Missionsplan durch erneutes Drücken des Fortschrittsknopfes aktivschalten, oder die Änderungen durch Drücken des „Abbruch-Knopfes“ rückgängig machen.

⁴⁰ Analog zur Methode Touch-Gesture ergänzt das System hierzu den Auftragstyp, das Erfolgskriterium und die Parameter des Auftrags. Wie diese Auswahl umgesetzt wird, wird im Rahmen der Implementierung der Auftragsvervollständigung in [4.4.2](#) beschrieben.

4.2.2 Richtlinienmanagement

Im Entwurfsmuster der assistierten auftragsbasierten Führung wird die Auftragsdelegation ergänzt durch die Vorgabe von Verhaltensrichtlinien, mit denen der Nutzer das Verhalten der unbemannten Plattformen auftragsunabhängig und zeitlich unbefristet beeinflussen kann. Nachfolgend wird die Implementierung dieser Verhaltensrichtlinien beschrieben.

Ein Schwerpunkt der Verhaltensrichtlinien ist das sogenannte *Idle-Verhalten*. Das Idle-Verhalten beschreibt das gewünschte Verhalten der Plattformen, wenn keine weiteren Aufträge anstehen. Hierzu wurden zwei primäre Verhaltensweisen definiert, die durch weitere Einstellungen genauer spezifiziert werden können. Eine Beschreibung der Konfigurationsmöglichkeiten des Idle-Verhaltens ist in Tabelle 13 dargestellt.

Tabelle 13. Beschreibung der einstellbaren Idle-Verhaltensweisen.

Bezeichnung	Beschreibung
Position halten	Die Plattform behält die letzte Position bei und führt Warteschleifen durch. Das Verhalten bei Bedrohung durch feindliche Luftfahrzeuge wird durch drei Subtypen definiert: <ul style="list-style-type: none"> • Passiv: Die Plattform reagiert nicht auf Feindflugzeuge. • Offensiv: Die Plattform bekämpft aktiv feindliche Luftfahrzeuge in der Nähe. • Defensiv: Die Plattform zieht sich zum bemannten Kampfflugzeug zurück.
Unterstützung	Die Plattform unterstützt ein Teammitglied. Hierzu begibt es sich in eine Formationsposition um das Teammitglied. Welche weiteren Aktionen ausgeführt werden, kann durch zwei ergänzende Auswahloptionen konfiguriert werden: <ul style="list-style-type: none"> • Jamming: Das Teammitglied wird durch Jamming vor Radarstellungen geschützt. Die Art des Jammings richtet sich nach der größten Bedrohung für das Teammitglied. • Automatische Bekämpfung: Feindliche Luftfahrzeuge, die eine Bedrohung für das Teammitglied darstellen, werden automatisiert bekämpft.

Eine weitere Verhaltensrichtlinie ist das *Defensive-Verhalten*. Diese Verhaltensrichtlinie beschreibt das Verhalten der Plattform, wenn das Raketenwarnsystem einen anfliegenden Lenkflugkörper detektiert. Für diesen Fall wurden zwei Verhaltensweisen definiert:

- Volle Gegenmaßnahmen: Die Plattform reagiert auf die Bedrohung, indem sie alle geeigneten Gegenmaßnahmen ergreift (z. B. Ausstoß von Täuschkörpern und Änderung der Flugbahn). Das Überleben der Plattform hat Vorrang vor der Auftragserfüllung.
- Begrenzte Gegenmaßnahmen: Die Plattform reagiert auf die Bedrohung, indem sie nur solche Gegenmaßnahmen einleitet, die keine negativen Auswirkungen auf die Auftragserfüllung haben (z. B. Täuschkörper).

Die Verwaltung von Verhaltensrichtlinien erfolgt über eine grafische Benutzeroberfläche auf dem rechten SHDD, die sogenannte *Policy Page*. Die Policy Page ermöglicht es dem Benutzer, die Verhaltensrichtlinien der unbemannten Plattformen festzulegen und sie dient gleichzeitig zur Anzeige der erteilten Verhaltensregeln. Die Policy Page ist in Abbildung 49 dargestellt.

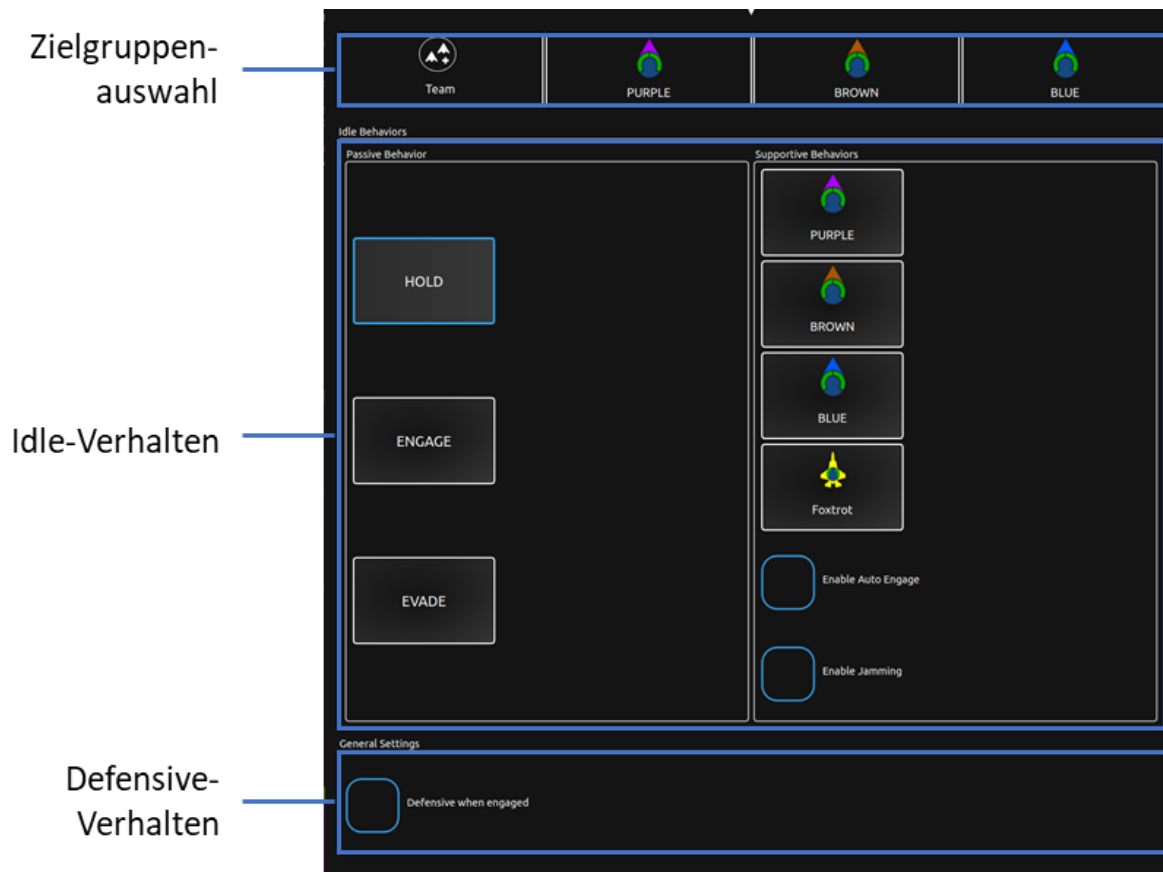


Abbildung 49. Policy Page zur Konfiguration von Verhaltensrichtlinien.

Verhaltensrichtlinien können entweder individuell oder kollektiv verwaltet werden. Hierzu verfügt die Policy Page über einen Auswahlreiter, über den der Benutzer eine einzelne Plattform oder das Team als Zielgruppe festlegen kann. Hat der Nutzer eine Zielgruppe ausgewählt, kann er unterhalb des Auswahlreiters die Verhaltensrichtlinien für diese Zielgruppe festlegen. Die Definition von Verhaltensrichtlinien erfolgt durch die Auswahl der entsprechenden Schaltflächen und Kontrollkästchen, die sich unterhalb des Auswahlreiters befinden. Diese Displayelemente sind in verschiedenen Bereichen gruppiert, um die Zuordnung zu den entsprechenden Verhaltensrichtlinien zu verdeutlichen. Aktive Verhaltensrichtlinien werden dem Nutzer durch eine farbliche Hervorhebung der Schaltflächen bzw. durch ein Häkchen in der entsprechenden Checkbox angezeigt.⁴¹

⁴¹ Ist im Reiter das Team ausgewählt, erfolgt die Anzeige der aktiven Richtlinien nur, wenn für alle Teammitglieder die gleichen Verhaltensrichtlinien gelten.

4.2.3 Auftragskritik

Die Auftragskritik ist eine der beiden Kernfähigkeiten des Delegationsagenten. Diese Fähigkeit besteht darin, dass der Delegationsagent jeden vom Nutzer delegierten Auftrag bewertet und gegebenenfalls ein kritisches Feedback zum Auftrag gibt. Bei der Auftragskritik handelt es sich um eine Reaktion auf Nutzereingaben gemäß der in Unterabschnitt 2.3.3 vorgestellten Feedback-Taxonomie.

Zur Umsetzung der Auftragskritik wurden verschiedene Anwendungsfälle identifiziert, in denen eine Auftragskritik einen positiven Einfluss auf die Missionsleistung haben könnte. Dazu wurden zunächst verschiedene Kategorien von Anwendungsfällen definiert. Diese Kategorien entsprechen den unterschiedlichen Missionsebenen, auf denen Probleme durch Auftragserteilungen entstehen können (Tabelle 14).

Tabelle 14. Kategorisierung von Anwendungsfällen der Auftragskritik.

Kategorie	Beschreibung	Beispiel
Missionsziel	Eine Auftragserteilung trägt nicht zum Erreichen der Missionsziele bei oder steht in Konflikt mit den Missionszielen.	Ein Auftrag zielt auf die Bekämpfung einer Radarstellung ab, die keine Bedrohung darstellt bzw. darstellen wird.
Missionsplan	Die Auftragsdelegation ist in Bezug auf den bestehenden Missionsplan nicht sinnvoll.	Ein delegierter Auftrag erzielt denselben Effekt wie ein bestehender Auftrag im Missionsplan.
Auftrag	Der Auftrag wird bei der gegebenen Spezifikation nicht den gewünschten Effekt erzielen oder es sind Probleme bei der Auftragsdurchführung zu erwarten.	Die Parameter eines Aufklärungsauftrags sind so gewählt, dass das gewünschte Aufklärungsergebnis aufgrund von Wolken oder Hindernissen nicht erzielt wird.

In dieser Arbeit wurde die Auftragskritik exemplarisch für verschiedene Anwendungsfälle auf der Auftragsebene implementiert. Diese Ebene wurde gewählt, da eine zuverlässige Erkennung von Fehlerfällen eine zentrale Voraussetzung für die Untersuchung der Effekte einer Auftragskritik ist. Auf der Auftragsebene ist dies besonders gut möglich, da korrekte und fehlerhafte Entscheidungen dort präziser voneinander abgegrenzt werden können als auf höheren Ebenen. Gleichzeitig nimmt auf höheren Ebenen die Anzahl der zu berücksichtigenden Einflussfaktoren zu, wodurch die zuverlässige Erkennung von Fehlerfällen zusätzlich erschwert wird.⁴²

⁴² Die Implementierung einer Kritikfähigkeit auf der Missionszielebene würde zudem bedeuten, dass der Delegationsagent vom Tool zum Worker wird, da er in diesem Fall das Arbeitsziel versteht und durch seine Kritik aktiv zu dessen Erreichung beiträgt. Diese geänderte Rollenverteilung könnte weitere Auswirkungen auf die Dynamik zwischen Nutzer und Delegationsagent haben, die in dieser Arbeit nicht untersucht wurden. Eine Untersuchung dieses Entwurfsmusters könnte jedoch interessante Erkenntnisse für zukünftige Forschungsarbeiten liefern.

Es wurden mehrere Anwendungsfälle identifiziert, wie ein Auftrag so spezifiziert werden könnte, dass er nicht den gewünschten Effekt erzielen wird oder Probleme bei der Auftragsdurchführung zu erwarten sind. Diese Fälle und die betreffenden Auftragsstypen sind in Tabelle 15 beschrieben.

Tabelle 15. Mögliche Fälle einer ungeeigneten Auftragspezifikation.

Auftragstyp	Anwendungsfall
Engage	<ul style="list-style-type: none"> Parameterset erzielt suboptimales Bekämpfungsergebnis (z. B. Gefährdung neutraler Objekte).
Investigate	<ul style="list-style-type: none"> Auftragshöhe liegt über Schichtbewölkung. Auftragshöhe nicht mit Gelände vereinbar. Sensorreichweite zu gering für gewähltes Parameterset.
DEAD / Area Suppression	<ul style="list-style-type: none"> Auftragshöhe nicht mit Gelände vereinbar. Bewaffnung nicht ausreichend für Auslöseintervall (nur im Fall von präventiver Area Suppression).

Nachdem die verschiedenen Fälle von ungeeigneten Auftragspezifikationen ermittelt wurden, muss der Interventionszeitpunkt festgelegt werden, d. h. der Zeitpunkt, zu dem der Nutzer über die fehlerhafte Auftragspezifikation informiert wird. Theoretisch kommt dafür jeder Zeitpunkt in Frage, der zwischen dem Zeitpunkt der ersten Fehlkonfiguration⁴³, und dem Zeitpunkt, an dem der Problemfall im Rahmen der Auftragsausführung sichtbar wird, liegt. Einen Anhaltspunkt für die Auswahl eines geeigneten Interventionszeitpunktes stellen die Anforderungen an das Verhalten von Assistenzsystemen nach Theißing und Schulte (2016) dar. Diese Anforderungen sehen vor, dass das Eingreifen sowohl so früh wie nötig erfolgen soll, um negative Auswirkungen rechtzeitig zu verhindern, als auch so spät wie möglich, um dem Nutzer genügend Zeit zur eigenständigen Lösung des Problems zu geben. Unter Berücksichtigung dieser Anforderungen wird der Interventionszeitpunkt der Auftragskritik auf das Ende des Delegationsprozesses festgelegt. Dieser Zeitpunkt ist dadurch definiert, dass der problembehaftete Auftrag in den aktiven Plan eingefügt werden soll. Abhängig von der Delegationsmethode tritt dieser Fall entweder ein, wenn der Nutzer den Modify-Plan aktiviert, oder – bei Auftragsdelegationen ohne Review – wenn der Nutzer die Delegation eines Auftrags abgeschlossen hat (Abbildung 50).

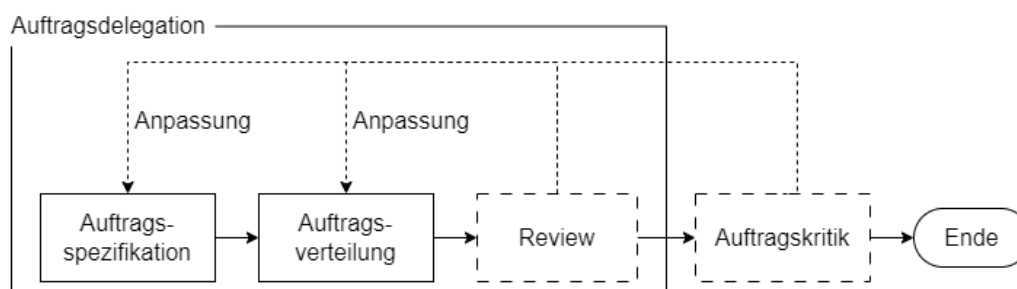


Abbildung 50. Auftragskritik nach Abschluss der Nutzerinteraktion zur Auftragsdelegation.

⁴³ Dadurch gekennzeichnet, dass die Spezifikation des Auftrags erstmalig unpassende Werte annimmt.

Der gewählte Interventionszeitpunkt stellt sicher, dass eine fehlerhafte Auftragsdelegation keine negativen Auswirkungen auf die Mission haben kann, weil der entsprechende Auftrag nicht zur Bearbeitung in den aktiven Plan eingefügt wird. Der Interventionszeitpunkt bietet dem Nutzer außerdem eine gewisse Zeitspanne, um den Problemfall eigenständig zu bemerken und zu beheben. Es ist anzunehmen, dass ein noch späterer Zeitpunkt die eigenständige Problembhebung durch den Nutzer nicht weiter begünstigt, weil Menschen nach Abschluss einer Interaktion dazu neigen, die damit verbundenen Informationen zu vergessen, um Ressourcen für andere Aufgaben freizusetzen (Johnson, 2010, S. 106).

Als Nutzerschnittstelle für die Auftragskritik wird eine diskrete Touchscreen-Interaktion verwendet. Delegiert der Nutzer einen Auftrag mit unpassender Spezifikation, erhält er eine Meldung in einer Benachrichtigungsleiste oberhalb der Missionsplanansicht, die den Nutzer auf den entsprechenden Problemfall hinweist. Für bestimmte Problemfälle bietet das System zudem Lösungsvorschläge an und ermöglicht durch entsprechende Schaltflächen eine einfache Umsetzung dieser Vorschläge (Abbildung 51).

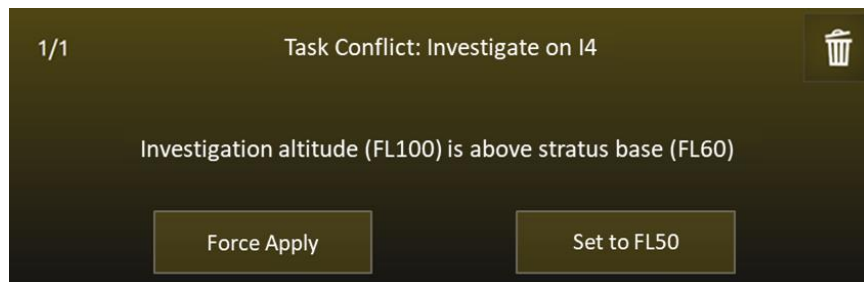


Abbildung 51. Auftragskritik nach Delegation eines Aufklärungsauftrags mit ungeeigneter Höhenvorgabe. Das System schlägt in diesem Fall eine Anpassung der Aufklärungshöhe vor.

Der Nutzer kann auf verschiedene Arten auf die Kritik des Systems reagieren. Falls das System einen Lösungsvorschlag unterbreitet hat, kann der Nutzer den Systemvorschlag durch Drücken der entsprechenden Schaltfläche annehmen. Darüber hinaus hat der Nutzer auch die Möglichkeit, den Auftrag wie ursprünglich spezifiziert durchführen zu lassen, indem er mit einer entsprechenden Schaltfläche das Einfügen in den aktiven Plan erzwingt. Eine weitere Möglichkeit ist, dass der Nutzer manuell den betreffenden Auftrag verändert und anschließend den Missionsplan aktiv schaltet. Dieses Vorgehen kann allerdings eine erneute Auftragskritik nach sich ziehen, falls der Problemfall nicht behoben wurde oder ein anderes Problem vorliegt.

4.2.4 UAV-Feedback

Das UAV-Feedback informiert den Nutzer über den Zustand der Plattform und der ihr zugewiesenen Aufträge. Das Ziel des UAV-Feedbacks ist es, das Situationsbewusstsein des Nutzers hinsichtlich des Missionsverlaufs und des Status der unbemannten Kräfte zu erhöhen, sodass der Nutzer rechtzeitig Maßnahmen ergreifen kann, wenn die Mission nicht planmäßig verläuft. Gemäß der in Unterabschnitt 2.3.3 vorgestellten Feedback-Taxonomie lässt sich das UAV-Feedback in die Kategorien „Informationen über den Status der Automation“ und „Informationen über relevante Umwelteinflüsse“ einteilen.

Um das UAV-Feedback zu implementieren, wurden zunächst die Feedback-Anforderungen des Nutzers ermittelt. In der Literatur werden diesbezüglich insbesondere diejenigen Informationen genannt, die einen Vergleich der Situation mit den Zielen, Plänen und Erwartungen des Nutzers ermöglichen und mit denen der Nutzer die ordnungsgemäße Funktion des automatisierten Systems nachvollziehen kann (Norman, 2013; Wintersberger et al., 2021; Zühlke, 2012). Übertragen auf den Anwendungsfall der Führung unbemannter Luftfahrzeuge sind diesbezüglich folgende Informationen zu nennen:

- Auftragsfortschritt: Beginn, Abbruch und Abschluss von Aufträgen sowie das Erreichen von Zwischenzielen.
- Vorgehensweise des Agenten: Aktionen und Entscheidungen des Agenten.
- Taktische Lage: Bedrohungen für die eigene Plattform und Lageänderungen, die sich negativ auf Aufträge auswirken können.

Um diese Informationen an den Nutzer zu übermitteln, wurden eine Reihe von Einzel-Maßnahmen ergriffen, die mithilfe von verschiedenen Sprachausgaben, Popup-Meldungen und Anzeigen realisiert wurden. Nachfolgend werden diese Maßnahmen und deren Gestaltung vorgestellt und motiviert.

Handlungsfeedback: Das Handlungsfeedback hat den Zweck, die Vorgehensweise der unbemannten Plattformen transparent zu machen, indem es den Nutzer über die gegenwärtig ausgeführte Handlung eines UAV informiert. Die Übermittlung des Handlungsfeedbacks erfolgt mithilfe von verschiedenen Icons, die dauerhaft neben dem Symbol des zugehörigen UAV in der taktischen Karte angezeigt werden (Abbildung 52).



Abbildung 52. Handlungsfeedback-Icons in verschiedenen Situationen.

Die Entscheidung, das Handlungsfeedback mithilfe einer Icon-basierten Anzeige umzusetzen erfolgt aus verschiedenen Gründen. Für die Darstellung der Information auf dem Display spricht, dass Aktionswechsel häufig stattfinden können. Eine Übermittlung anhand von Tönen oder Sprachmeldungen hätte daher eine hohe Belegung des auditiven Kanals zur

Folge. Die Umsetzung als permanente Anzeige soll ermöglichen, dass der Nutzer die Information jederzeit und einfach verfügbar hat, was bei einer Darstellung von Zustandswechseln nicht der Fall wäre. Die Icon-basierte Form der Darstellung wurde gewählt, weil diese eine schnellere und einfachere Informationsaufnahme gegenüber textbasierten Darstellungsformen ermöglicht (Friedrich et al., 2022).

Fortschrittmeldungen: Diese Feedback-Maßnahme informiert den Nutzer über den Fortschritt von Aufträgen. Zu diesem Zweck werden beim Erreichen von bestimmten Meilensteinen der Auftragsdurchführung Sprachausgaben generiert (z. B. bei Abschluss eines Auftrags). Diese Sprachausgaben sind an die Multi-Service Brevity Codes angelehnt und enthalten immer das meldende UAV, den Meilenstein und gegebenenfalls ein Zielobjekt, auf das sich der Meilenstein bezieht. In Tabelle 16 sind beispielhaft einige Fortschrittmeldungen und deren Bedeutung dargestellt.

Tabelle 16. Beispiele für Fortschrittmeldungen und deren Bedeutung.

Meldung	Bedeutung
<i>Blue start investigate on bravo six</i>	UAV „Blue“ beginnt einen Aufklärungsauftrag und fliegt hierfür zum Zielobjekt.
<i>Blue arrived at investigate on bravo six</i>	UAV „Blue“ ist am Zielobjekt eines Aufklärungsauftrags angekommen und beginnt mit der Aufklärung.
<i>Orange complete investigate on bravo six</i>	Das UAV „Orange“ hat den Aufklärungsauftrag am Zielobjekt „Bravo Six“ abgeschlossen.
<i>Blue magnum</i>	UAV „Blue“ hat eine radarsuchende Rakete abgefeuert.

Die Entscheidung, Fortschrittmeldungen mittels Sprachausgabe zu realisieren, wurde vor dem Hintergrund getroffen, dass Fortschrittmeldungen sehr wichtig sind, da sie es dem Nutzer ermöglichen, den Missionserfolg zu verfolgen. Auditive Benachrichtigungsformen können hierfür geeigneter sein, weil diese auch dann wahrgenommen werden können, wenn der Nutzer sich mit anderen Aufgaben befasst (Wickens et al., 2016).

Entscheidungsmeldungen: Diese Meldungen setzen den Nutzer über besondere Entscheidungen des Plattformsteuerungsagenten in Kenntnis. Dies betrifft jene Entscheidungen, bei denen der Agent von seinem üblichen Verhalten abweicht oder die nur aufgrund besonderer Rahmenbedingungen, wie z. B. einer bestimmten taktischen Lage, getroffen werden. Ein Beispiel für eine solche Entscheidung ist, dass ein Luftfahrzeug vom UAV als feindlich klassifiziert wurde, weil es einen *Hostile Act*, d. h. eine feindselige Aktion begangen hat. Die Übermittlung von Entscheidungsmeldungen erfolgt mithilfe von Popup-Meldungen, die in der taktischen Karte am betreffenden UAV angezeigt werden (Abbildung 53).

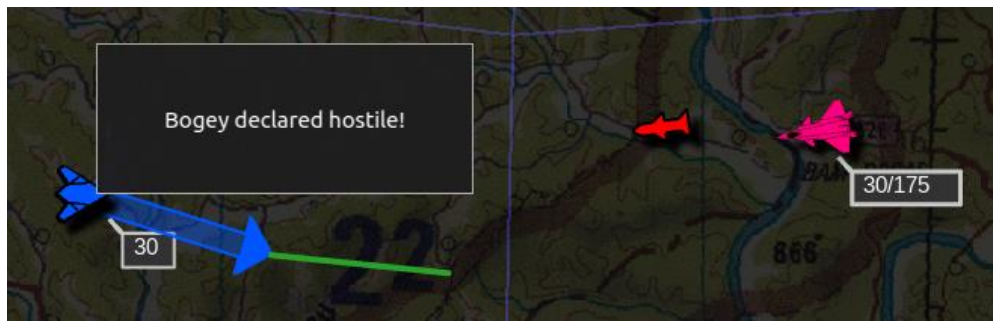


Abbildung 53. Meldung über automatische Klassifizierung bei feindseliger Handlung.

Die Umsetzung als visuelle Meldung am betreffenden UAV wurde gewählt, da der Nutzer die Meldung so einfacher zuordnen kann und anschließend auf der Karte direkt wahrnehmen kann, wie sich das Verhalten des UAVs infolge der Entscheidung ändert.

Problem- und Ereignismeldungen: Diese Meldungen dienen dazu, den Nutzer über Probleme und unerwartete Ereignisse zu informieren. Beispiele für solche Ereignisse sind:

- Ein *Bogey* (unidentifiziertes bzw. potenziell feindliches Luftfahrzeug) wurde entdeckt
- Das Zielobjekt eines Auftrags wird später als geplant erreicht
- Die Auftragsdurchführung dauert länger als geplant
- Ein Auftrag ist aufgrund von Umweltfaktoren nicht durchführbar
- Ein UAV hat keinen Treibstoff/keine Waffen mehr
- Ein UAV wird beschossen

Die Übermittlung von Problem- und Ereignismeldungen erfolgt durch Sprachausgaben und Popupmeldungen in der taktischen Karte (Abbildung 54).

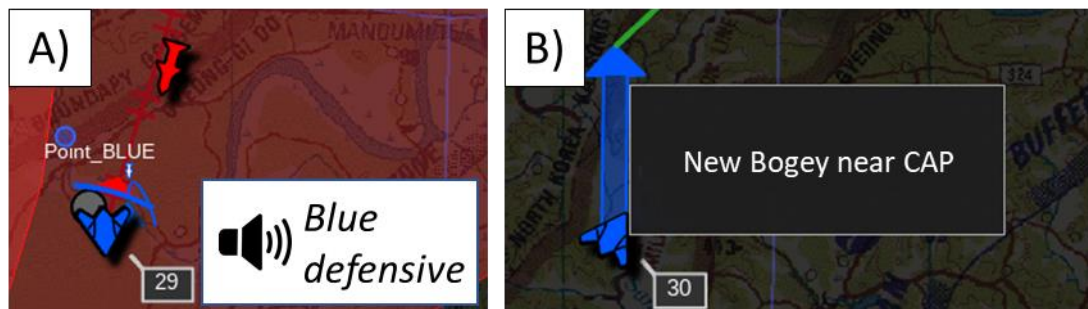


Abbildung 54. Beispiele für Ereignismeldungen.

A) Sprachmeldung bei Beschuss B) Popup-Meldung bei Entdeckung eines Bogeys.

Die Modalität der Meldung wird anhand der Priorität der Meldung variiert. Höher priorisierte Meldungen werden mithilfe von Sprachausgaben übermittelt, während Meldungen mit geringerer Priorität durch Popups übermittelt werden. Diese Aufteilung erfolgt vor dem Hintergrund der besseren Wahrnehmung auditiver Meldungen (vgl. Fortschrittsmeldungen).

4.2.5 Nutzerdialoge

Durch Nutzerdialoge wird der Nutzer dynamisch in die Entscheidungsfindung der Plattformsteuerungsagenten miteinbezogen. Sie kommen zum Einsatz, wenn eine Vorgabe des Nutzers wünschenswert ist, aber diese Vorgabe weder durch Auftragsdelegationen noch durch die Erteilung von Verhaltensrichtlinien abgedeckt werden kann. Dies ist beispielsweise der Fall, wenn der Nutzer in die Lösung eines unerwartet aufgetretenen Problems eingebunden werden soll.

Nutzerdialoge werden mithilfe von Popup-Meldungen geführt, die in unmittelbarer Nähe zum betreffenden UAV angezeigt werden (ähnlich den Entscheidungsmeldungen). Die Nutzerdialoge enthalten eine Textnachricht, die den Grund der Meldung beschreibt und mehrere Schaltflächen, die verschiedene Handlungsoptionen enthalten, zwischen denen der Nutzer auswählen kann, um auf die Meldung zu reagieren (Abbildung 55).

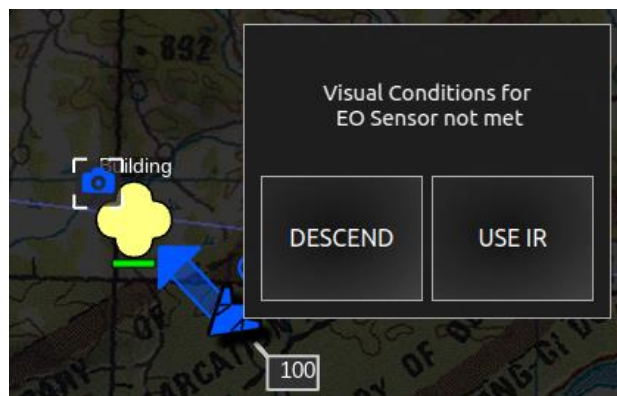


Abbildung 55. Nutzerdialog in Folge von Problemen bei der Aufklärung eines Gebäudes.

Nutzerdialoge sind als nicht-modale Dialoge ausgeführt, d. h. der Nutzer kann mit Elementen außerhalb der Meldung interagieren, ohne die Dialogmeldung abschließen zu müssen. Diese Designentscheidung soll verhindern, dass der Nutzer andere, potenziell wichtigere Arbeitsabläufe, unterbrechen muss, um den Nutzerdialog zu bearbeiten (Fessenden, 2017).

Die Implementierung von Nutzerdialogen erfolgt beispielhaft an mehreren Anwendungsfällen. Im Folgenden werden diese Anwendungsfälle und die jeweiligen Handlungsoptionen des Nutzers beschrieben (Tabelle 17).

Tabelle 17. Implementierte Anwendungsfälle von Nutzerdialogen.

Anwendungsfall	Auswahloptionen des Nutzers
Aufklärung ist aufgrund lokaler Wolkenformationen nicht möglich	<ul style="list-style-type: none"> Reduktion der Aufklärungshöhe (wenn möglich) Wechsel des Sensors
Bogey bedroht Teammitglieder	<ul style="list-style-type: none"> Feindlich klassifizieren Bogey aufklären (durch Visuelle Identifikation)
Bekämpfungsauftrag nicht durchführbar, weil Zielobjekt nicht als feindlich klassifiziert	<ul style="list-style-type: none"> Feindlich klassifizieren Sensorbilder des Ziels betrachten

4.3 Funktionale Umsetzung der Nutzerschnittstelle

In diesem Abschnitt wird die funktionale Umsetzung der Nutzerschnittstelle beschrieben. Der Fokus liegt dabei auf jenen Schnittstellenelementen, die sich von etablierten grafischen und manuellen Bedienkonzepten unterscheiden. Zu diesem Zweck behandelt Unterabschnitt [4.3.1](#) die Implementierung der **Sprachsteuerung** im Rahmen der Delegationsmethode Voice. Daran anschließend wird in Unterabschnitt [4.3.2](#) die **Sprachsynthese** im Kontext des UAV-Feedbacks erläutert. Abschließend folgt in Unterabschnitt [4.3.3](#) die **Auswahl von Objekten mittels Blickbewegung**, die die Grundlage der Delegationsmethode Gaze-HOTAS bildet.

4.3.1 Sprachsteuerung

Die Bereitstellung von Sprachbefehlen für die Delegation von Aufträgen (siehe [4.2.1.3](#)) erfordert zwei technische Funktionen: *Spracherkennung* und *Sprachverarbeitung* (Natural Language Processing). Die Spracherkennung wandelt die Tonsignale, die beim Aussprechen von Sprachbefehlen über das Headset-Mikrofon des Nutzers erfasst werden, in Text um. Die Sprachverarbeitung ordnet dem erkannten Text eine Bedeutung zu, schließt aus dieser Bedeutung auf die Absichten des Benutzers und löst die entsprechenden Systembefehle aus, mit denen diese Absichten verfolgt werden können. Abbildung 56 stellt den Ablauf der Verarbeitung von Sprachbefehlen dar.

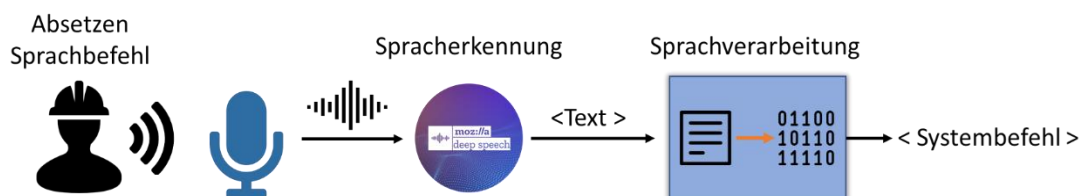


Abbildung 56. Verarbeitungskette von Sprachbefehlen.

Die Spracherkennung wurde mithilfe des Spracherkennungs-Frameworks DeepSpeech⁴⁴ implementiert. DeepSpeech ist ein Open-Source-Projekt der Mozilla Foundation, welches auf der gleichnamigen Forschungsarbeit von Baidu Research (Hannun et al., 2014) basiert und darauf abzielt, ein einfaches und offenes Spracherkennungsframework unter Nutzung von maschinellem Lernen zu entwickeln. Zur Erkennung von Sprache nutzt DeepSpeech ein akustisches Modell auf Basis eines *Rekurrenten Neuronalen Netzes* (RNN) sowie optional ein *N-Gram Language Model* (Jurafsky & Martin, 2024b).

Das RNN analysiert schrittweise die akustischen Merkmale eines Audiosignals und leitet daraus eine Wahrscheinlichkeitsverteilung für jedes erkennbare Zeichen (z. B. Buchstaben des jeweiligen Alphabets) ab. Dabei fließen auch frühere Eingaben in die Generierung dieser Wahrscheinlichkeitsverteilung ein.⁴⁵ Die Transkription ergibt sich schließlich durch das Zusammensetzen der wahrscheinlichsten Zeichen nach dem vollständigen Abtasten des Audiosignals.

⁴⁴ Mozilla DeepSpeech (<https://github.com/mozilla/DeepSpeech>)

⁴⁵ RNN haben die inhärente Eigenschaft, Informationen über die Zeit zu speichern.

Das Language Model beschreibt die Wahrscheinlichkeiten von Wortfolgen auf Basis eines Trainingsdatensatzes und kann verwendet werden, um die Wahrscheinlichkeit eines Wortes in Abhängigkeit von den vorhergehenden Wörtern zu bestimmen.⁴⁶ Dieses Prinzip kann im Rahmen der Spracherkennung zu einer höheren Erkennungsgenauigkeit beitragen, indem die vom RNN generierten Wörter mit dem Language Model abgeglichen und die Transkription mit der höchsten Gesamtwahrscheinlichkeit ausgewählt wird.

Im Rahmen des Projekts DeepSpeech stellt Mozilla vortrainierte Modelle (RNNs und Language Models) für verschiedene Sprachen zur Verfügung. Unter diesen Modellen findet sich auch ein Modell, das für die Erkennung englischer Sprache eingesetzt werden kann und damit prinzipiell auch für die Erkennung der Sprachbefehle zur Auftragsdelegation geeignet ist. Im Rahmen dieser Arbeit wurde allerdings ein eigenständiges Modell entwickelt, welches speziell an den Anwendungsfall der Auftragsdelegation angepasst ist. Dieses angepasste Modell verfügt über ein eingeschränktes Vokabular und eine klar definierte Syntax, womit sich höhere Erkennungsraten erreichen lassen, weil es weniger Möglichkeiten gibt, mit welchen Buchstaben bzw. Wörtern ein Audiosegment in Zusammenhang stehen kann (Arora & Singh, 2012; Qiao et al., 2010). Eine hohe Erkennungsgenauigkeit ist ein wichtiger Faktor bei der Implementierung einer Sprachsteuerung, weil diese zu einer besseren Leistung und einer höheren Nutzerakzeptanz beiträgt (Casali et al., 1990; Kun et al., 2007; Munteanu et al., 2006).

Zur Anpassung von DeepSpeech an Sprachbefehle mit begrenztem Vokabular und klar definierter Syntax stehen in DeepSpeech verschiedene Möglichkeiten zur Verfügung. Im Rahmen dieser Arbeit erfolgte die Anpassung durch zwei Maßnahmen:

- *Fine-Tuning* des RNN: Die Modellparameter des vortrainierten Modells wurden mithilfe eines eigens erstellten Datensatzes mit Sprachkommandos an den Anwendungsfall der Auftragsdelegation angepasst.⁴⁷ Durch diese Feinabstimmung lernt das Modell Wörter zu erkennen, die im englischen Sprachgebrauch selten sind und es wird an die Sprachnuancen deutscher Sprecher gewöhnt.
- Neues Language Model: Der Datensatz mit Sprachkommandos wurde ebenfalls zur Erzeugung eines 3-gram Language Models genutzt. Die Definition eines eigenen Sprachmodells ermöglicht DeepSpeech eine genauere Erkennung der Sprachkommandos, da die Struktur etwas von der üblichen Grammatik englischer Sprache abweicht.

Der eigens erstellte Trainingsdatensatz zum Fine-Tuning des RNNs sowie zur Erzeugung des Language Model erhält in Summe über 3550 Proben von 12 verschiedenen Sprechern. Jede dieser Proben besteht aus einer Aufnahme und der dazugehörigen Übersetzung und wurde in einer Umgebung ohne Hintergrundgeräusche aufgenommen.

⁴⁶ Der Wert „N“ steht dabei für die Anzahl an Wörtern einer Textfolge, ein 3-Gramm Modell schätzt also die Wahrscheinlichkeit eines Wortes basierend auf den zwei Wörtern zuvor.

⁴⁷ Der wesentliche Vorteil dieses Verfahrens besteht darin, dass deutlich weniger Trainingsdaten erforderlich sind, da das zugrundeliegende Modell bereits mit einer großen Datenmenge trainiert wurde.

Die Modelle wurden mit einem weiteren, eigens erstellten Datensatz von 370 Proben getestet, um die Genauigkeit der Spracherkennung zu ermitteln. Der Datensatz deckte verschiedene Bedingungen ab (mit und ohne Störgeräusche) und enthielt Proben von drei Sprechern, die nicht im Trainingsdatensatz vertreten waren. Als Metrik zur Bewertung der Erkennungsgenauigkeit wurde die *Sentence Error Rate (SER)* ermittelt (Jurafsky & Martin, 2024a). Die SER gibt an, wie viel Prozent der erkannten Sätze fehlerhaft sind, d. h. mindestens ein falsches Wort enthalten. Diese Metrik ist für den vorliegenden Anwendungsfall besser geeignet als die häufig verwendete *Word Error Rate (WER)* (McCowan et al., 2004), da es bei einem Sprachbefehl in den meisten Fällen unerheblich ist, ob der Befehl aufgrund eines oder mehrerer Fehler nicht korrekt erkannt wurde. Die Erprobung der Modelle ergab eine SER von zwei Prozent für Proben ohne Störgeräusche und von fünf Prozent für Proben mit Störgeräuschen. Diese Erkennungsrate wird als ausreichend für die vorliegende Anwendung betrachtet.

Für die Implementierung der Sprachverarbeitung kommen eine Vielzahl von verschiedenen Methoden in Frage, mit denen Texte verarbeitet und auf ihre Bedeutung hin analysiert werden können (E. Kumar, 2011). Im vorliegenden Anwendungsfall reduziert sich die Komplexität der Sprachverarbeitung durch die fest vorgegebene Struktur von Sprachbefehlen (siehe [4.2.1.3](#)) deutlich. Daher wurde im Rahmen dieser Arbeit ein einfaches Verfahren⁴⁸ entwickelt, das die Sprachkommandos in zwei Schritten in Systembefehle überführt:

1. Befehls-Matching: Durch die Prüfung von Merkmalen wird der Typ des Sprachbefehls ermittelt.
2. Befehls-Ausführung: Nachdem der Befehlstyp bekannt ist, werden die Elemente des Befehls extrahiert und in Systembefehle übersetzt.

⁴⁸ Da Variationen der Sprachkommandos nicht vorgesehen sind, können alternative Formulierungen und Satzstellungen mit diesem Verfahren nicht erkannt werden.

4.3.2 Sprachsynthese

Für die Sprachausgabe im Rahmen des UAV-Feedbacks wird eine Funktion benötigt, die in der Lage ist, aus Texten Sprache zu synthetisieren. Diese Funktion wird als *Sprachsynthese* oder *Text-To-Speech* (TTS) bezeichnet. Die Sprachsynthese wird heute in der Regel durch ein zweistufiges Verfahren realisiert, das im Grunde eine umgekehrte Version der Spracherkennung darstellt (Jurafsky & Martin, 2024a):

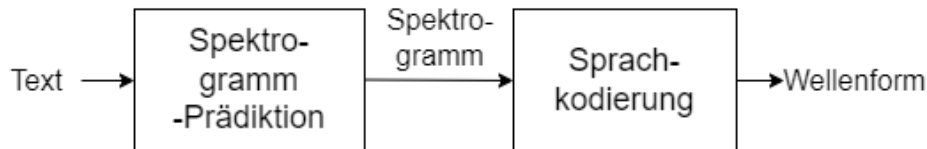


Abbildung 57. Akustikmodell-Vocoder Verfahren zur Sprachsynthese (vgl. Brown, 2021).

In diesem Verfahren wird zunächst der zu synthetisierende Text mithilfe eines akustischen Modells in ein Spektrogramm umgewandelt. Anschließend wird dieses Spektrogramm mithilfe eines *Vocoders* (Sprachkodierers) in die entsprechende Wellenform im Zeitbereich encodiert. In vielen Fällen ist diesen beiden Schritten ein Vorverarbeitungs- bzw. Textnormalisierungsschritt vorgeschaltet. Bei der Textnormalisierung werden alle Begriffe, deren Aussprache von der Schriftweise abweicht, in ihre gesprochene Form umgewandelt. Dies betrifft beispielsweise Abkürzungen, Zahlen oder Ziffern.

Die Implementierung einer Sprachsynthese wurde unter Nutzung der Open-Source Software *Mozilla TTS* realisiert. Mozilla TTS bietet eine End-to-End-Sprachsynthese, die das vorgestellte Akustikmodell-Vocoder-Verfahren mit vorgeschalteter Textnormalisierung nutzt. Mozilla TTS stellt vortrainierte Akustik- und Codierungsmodelle für die Synthese verschiedener Sprachen zur Verfügung, darunter auch mehrere Modelle, die für die Generierung englischer Sprache geeignet sind. Die im Rahmen dieser Arbeit implementierte Sprachsynthese nutzt diese vortrainierten Modelle, um die Sprachmeldungen im Rahmen des UAV-Feedbacks zu generieren.⁴⁹ Die vortrainierten Modelle sind aufgrund der Textnormalisierung von Mozilla TTS in der Lage, bekannte Abkürzungen und Zahlen auszusprechen, jedoch zeigen sie Schwierigkeiten bei der korrekten Synthese von luftfahrt- oder militärluftfahrtbezogenen Begriffen wie "FL 50" (gesprochen „*flight level fifty*“) oder "Bullseye 330 50" (gesprochen „*Bullseye tree tree zero fifty*“). Aus diesem Grund ist der Sprachsynthese mit Mozilla TTS eine zusätzliche, anwendungsspezifische Textnormalisierung vorgeschaltet, die diese Begriffe in ihre gesprochene Form umwandelt.

⁴⁹ Akustikmodell: Tacotron 2; Vocoder: ParallelWaveGAN (<https://github.com/mozilla/TTS/wiki/Released-Models>)

4.3.3 Objektauswahl mittels Blickbewegung

Im Rahmen der Delegationsmethode Gaze-HOTAS kann der Nutzer Objekte auf dem Display auswählen, indem er seinen Blick auf das entsprechende Objekt richtet. Die Realisierung dieser blickbewegungs-basierten Auswahlmöglichkeit erfordert zum einen eine Funktion, die den Blickpunkt des Nutzers auf dem Display ermittelt, und zum anderen eine Funktion, die auf Basis dieser Blickpunkte eine Objektauswahl trifft. Die Funktion zur Bestimmung des Blickpunktes auf dem Display wird bereits durch das zuvor vorgestellte Blickbewegungsmesssystem Smart-Eye Pro abgedeckt (siehe Abschnitt [4.1](#)). Im Folgenden wird daher auf die Selektionsfunktion eingegangen, die die gemessenen Blickpunkte mit Anzeigeobjekten korreliert und eine Objektauswahl trifft.

Ein zentraler Aspekt bei der Implementierung einer solchen Selektionsfunktion ist die Zuordnung von Pixelpositionen zu Anzeigeobjekten. Im Rahmen dieser Arbeit erfolgt die Ermittlung des betrachteten Anzeigeobjektes mithilfe einer speziellen Software-Architektur, die in der verwendeten Simulationsumgebung zum Zweck der Pilotenbeobachtung entwickelt wurde (Mund & Schulte, 2018; Schwerd & Schulte, 2020). Diese Architektur nutzt das Event-Handling-Verfahren, das in GUI-Anwendungen eingesetzt wird, um Touchscreen- und Mausinteraktionen zu verarbeiten (Sells & Griffiths, 2007, S. 110). Dieses Verfahren wird eingesetzt, um dynamisch Informationen darüber zu generieren, welche Anzeigeobjekte der Nutzer betrachtet hat und welche Informationen damit verknüpft sind. Dieser Prozess der Informationsgenerierung läuft wie folgt ab:

- Jeder Blick des Nutzers löst ein spezifisches Blickbewegungsevent aus, welches die Pixelkoordinaten enthält.
- Der Event-Handler ermittelt, welches Anzeigeelement an der entsprechenden Stelle auf dem Display dargestellt ist.
- Das betrachtete Anzeigeelement wird in eine Beobachtungsliste eingetragen, die den Namen des Objektes und die damit verknüpften Informationen enthält.

Da das Messsystem die Pixelposition des Blicks nur mit einer bestimmten Genauigkeit ermitteln kann, wird dieser Prozess zusätzlich für jede Pixelposition im Umkreis von 15 Pixeln durchlaufen. Das Ergebnis ist eine Beobachtungsliste mit den Anzeigeelementen für alle Event-Positionen im Sampling-Bereich. Diese Beobachtungsliste wird anschließend normiert, d. h. duplizierte Einträge werden zusammengefasst und mit einer Wahrscheinlichkeit versehen. Eine Beobachtung wird dabei umso wahrscheinlicher eingeschätzt, je mehr Fläche diese im Sampling-Bereich einnimmt und je näher diese Fläche am geschätzten Blickpunkt des Nutzers liegt. Abbildung 58 veranschaulicht den vorgestellten Prozess zur Erzeugung einer Beobachtungsliste.

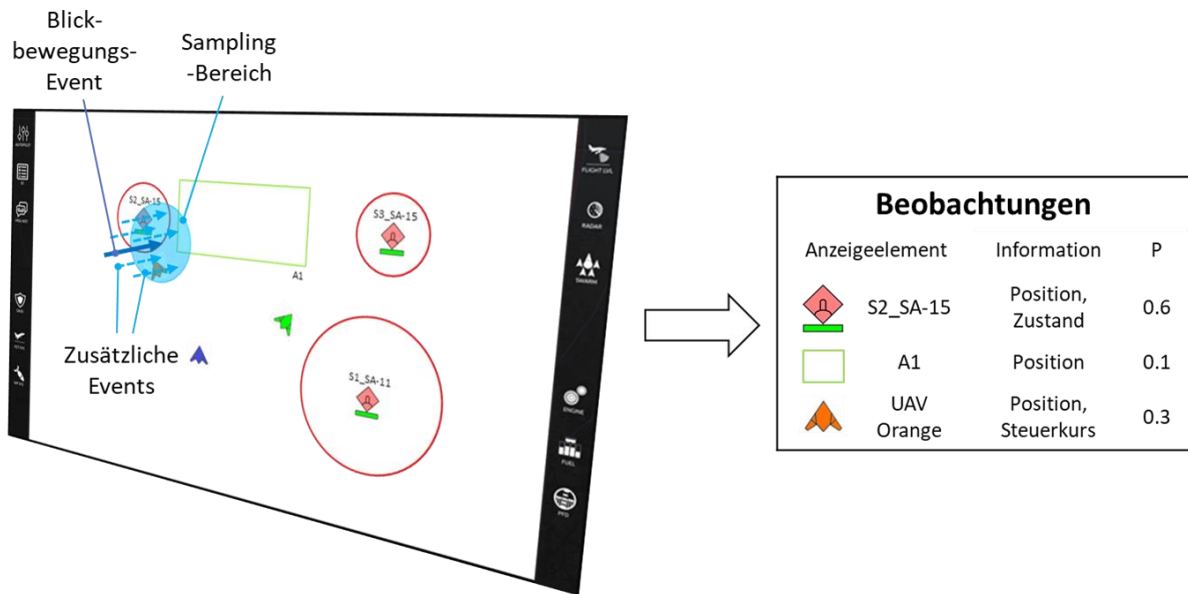


Abbildung 58. Erzeugung einer Beobachtungsliste durch Sampling um Blickposition.

Aufbauend auf dieser Architektur wurde eine Selektionsfunktion entwickelt, die die von der Architektur erzeugte Beobachtungsliste verarbeitet. Der einfachste Ansatz zur Umsetzung einer Selektionsfunktion wäre, dass immer das Objekt ausgewählt wird, dessen Wahrscheinlichkeit in der Beobachtungsliste am höchsten ist. Die Messung der Blickbewegung unterliegt jedoch hochfrequenten Störungen, sodass bei diesem Ansatz die Gefahr besteht, dass die Auswahl zwischen mehreren Objekten hin- und herwechselt, obwohl der Blick des Nutzers an derselben Stelle verweilt. Ein weiterer Nachteil dieses Ansatzes ist, dass in der Beobachtungsliste alle Anzeigeelemente erfasst werden, während für die Selektion eines Zielobjektes nur diejenigen Beobachtungen relevant sind, die direkt mit logischen oder physischen Objekten der Simulation verknüpft sind (z. B. das UAV-Symbol).

Aus diesen Gründen wurden daher zwei verschiedene Filterverfahren in die Selektionsfunktion integriert:

- Relevanz-Filter: Diese Komponente filtert Beobachtungen heraus, die für die Selektion irrelevant sind.
- Zeitverhalten-Filter: Diese Komponente stabilisiert die Objektauswahl, indem die wahrscheinlichsten Beobachtungen über einen Zeitraum von fünf Sekunden gesammelt werden und die Anzeigeelemente nach der Häufigkeit ihrer Betrachtung sortiert werden.

Nach diesen Filterverfahren wählt die Selektionsfunktion das meistbetrachtete Objekt aus. Dieses Vorgehen führt zu einer robusten Auswahlfunktion, die jedoch aufgrund des Zeitfilters eine höhere Interaktionsdauer aufweist als der initiale Ansatz ohne Filter.

4.4 Umsetzung der Agentenfunktionen

Dieser Abschnitt beschreibt die funktionale Umsetzung der Module des Delegationsagenten und des Plattformsteuerungsagenten. Die ersten drei Unterabschnitte befassen sich mit den Komponenten des Delegationsagenten. Unterabschnitt [4.4.1](#) behandelt die **Missionsplanung** als Grundlage für die Auftragsverteilung. Daran anschließend wird in [4.4.2](#) die Implementierung der **Auftragsvervollständigung** beschrieben. Unterabschnitt [4.4.3](#) widmet sich der **Analyse und Bewertung von Aufträgen**, die für die Fähigkeit der Auftragskritik zentral ist. Abschließend befasst sich Unterabschnitt [4.4.4](#) mit der Implementierung der **Verhaltensgenerierung**, die als zentrales Modul des Plattformsteuerungsagenten geeignete Verhaltensweisen für das jeweilige UAV erzeugt.

4.4.1 **Missionsplanung**

Durch den Einsatz einer Delegationsmethode kann der Nutzer Aufträge erteilen, ohne eine feste Zuordnung der Aufträge an die unbemannten Plattformen des Teams zu treffen. Stattdessen gibt der Nutzer durch die Verwendung einer Methode explizit oder implizit Kriterien vor, wie der jeweilige Auftrag verteilt werden soll.⁵⁰ Der Delegationsagent übernimmt anschließend die Verteilung von Aufträgen, indem er einen Missionsplan erstellt, der die vom Nutzer erteilten Aufträge enthält und deren Zuordnung zu Plattformen und Missionszeitpunkten definiert. Diese Fähigkeit bedingt eine technische Funktion zur Erzeugung von Missionsplänen.

Im Rahmen der Forschungsarbeiten von Felix Heilemann und Siegfried Maier wurde eine solche technische Funktion entwickelt und in verschiedenen Mensch-Maschine-Experimenten erprobt (Heilemann et al., 2021; Heilemann & Schulte, 2019; Maier & Schulte, 2021). Aus diesen Arbeiten ging eine Implementierung in Form einer eigenständigen Applikation hervor, die Planungsanfragen entgegennimmt und daraus Missionspläne generiert. Da somit bereits eine erprobte Lösung vorlag, wurde diese Implementierung in der vorliegenden Arbeit übernommen und in die Simulationsarchitektur eingebunden. Die Implementierung dieser Funktionalität war daher nicht Bestandteil dieser Arbeit, weshalb im Folgenden lediglich die Kernaspekte der verwendeten Missionsplanungsfunktionen der genannten Forschenden zusammengefasst werden. Eine umfassende Beschreibung des Missionsplaners ist in Heilemann (2022) zu finden.

⁵⁰ Diese Verteilungskriterien bzw. Scheduling-Modi sind in [3.4.1](#) beschrieben.

Der Missionsplaner nutzt einen Optimierungsansatz, bei dem die Missionsplanerstellung als Flexibles Job-Shop-Scheduling-Problem (FJSP) beschrieben wird. Die folgende Tabelle gibt einen Überblick über die typischen Bestandteile eines FJSP und veranschaulicht, wie das reale Problem damit modelliert wurde (vgl. Russel & Norvig (2016, p. 402); Xie et al. (2019)).

Tabelle 18. Elemente des Job-Shop-Scheduling-Problems und Modellierung des Problems zur Missionsplanung.

Element	Beschreibung	Modellierung der Anwendung
Jobs	Bei den Jobs handelt es sich um Aufgaben variabler Länge, die von einer Maschine bearbeitet werden können.	Die zu verteilenden Aufträge werden als Jobs modelliert.
Maschinen	Maschinen können eingesetzt werden, um Jobs zu bearbeiten.	Zur Bearbeitung der Aufträge können bemannte oder unbemannte Plattformen eingesetzt werden.
Kostenfunktion	Definiert die Kosten einer Lösung. Das Ziel des Job-Shop-Scheduling-Problems ist es, die Kostenfunktion zu minimieren.	Die Missionsdauer dient als Zielfunktion. Somit strebt der Missionsplaner einen möglichst kurzen Missionsplan an.
Randbedingungen	Definiert Anforderungen an valide Lösungsergebnisse, z. B. zwei Jobs müssen in bestimmter Reihenfolge stattfinden.	Anforderungen an den Missionsplan werden als Randbedingungen beschrieben, z. B. wenn ein Nutzer spezifiziert, dass ein Auftrag von einem bestimmten UAV ausgeführt werden soll.
Transitions-Matrix	Gibt die (minimalen) Übergangszeiten zwischen zwei aufeinanderfolgenden Jobs für alle Jobsequenzen an.	Die Flugzeit vom Zielobjekt eines Auftrags zum Zielobjekt eines anderen Auftrags wird als Transitionszeit modelliert. ⁵¹
Ressourcen	Jobs können mit dem Verbrauch von Ressourcen einer Maschine einhergehen. Geeignete Lösungen des Problems erfordern somit, dass die Maschinen ausreichend Ressourcen für die zugewiesenen Aufgaben haben.	Die Bewaffnung von UAVs wird als Ressource beschrieben. Somit werden Lösungen ausgeschlossen, bei denen keine ausreichende Bewaffnung gewährleistet ist.

Zur Lösung des Scheduling-Problems nutzt der Missionsplaner die Optimierungs-Bibliothek IBM® CPLEX.⁵²

⁵¹ Zur Berechnung der Flugzeiten wird die Distanz der Route zwischen den Objekten ermittelt und eine Fluggeschwindigkeit von 450 Knoten angenommen.

⁵² <https://www.ibm.com/de-de/products/ilog-cplex-optimization-studio>

4.4.2 Auftragsvervollständigung

Wie in 3.4.1 beschrieben, verfügt der Delegationsagent über eine Fähigkeit zur Vervollständigung von Aufträgen. Diese Fähigkeit ermöglicht es dem Benutzer, bei der Verwendung einer Delegationsmethode nicht alle Informationen angeben zu müssen, die für die vollständige Spezifikation eines Auftrags erforderlich sind. Nachfolgend wird die Implementierung dieser Fähigkeit beschrieben.

Die Vervollständigung von Aufträgen erfolgt durch einen mehrstufigen Prozess mit mehreren optionalen Schritten. In diesem Prozess wird zunächst auf Basis des Zielobjekts der Auftragstyp ermittelt und anschließend werden auf Basis von Zielobjekt und Auftragstyp das Erfolgskriterium und die Parameter ermittelt. Falls eine Information vom Nutzer spezifiziert wurde, entfällt der entsprechende Prozessschritt (Abbildung 59).

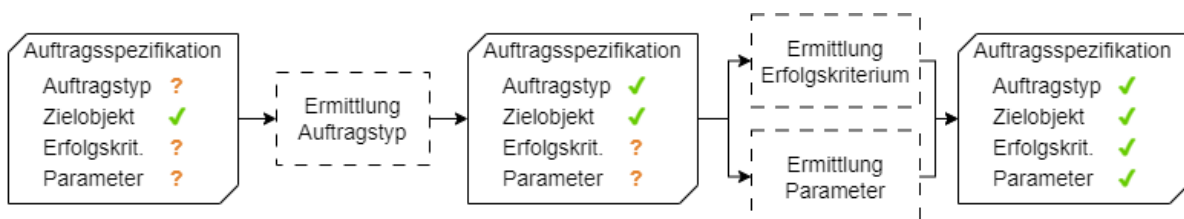


Abbildung 59. Prozess zur Vervollständigung eines Auftrags.

Die Ermittlung des Auftragstyps anhand des Zielobjekts ist nicht eindeutig, da die meisten Zielobjekte mit mehreren Auftragstypen in Verbindung stehen können. Daher wird der Auftragstyp im Rahmen der Auftragsvervollständigung auf Grundlage einer festen Zuordnung ausgewählt, die sich an den Eigenschaften des Objekts orientiert. So ist beispielsweise davon auszugehen, dass Nutzer bei Gebäuden einen anderen Auftragstyp erteilen wollen als bei Flugzeugen. Auf Basis dieser Überlegungen wurden die Eigenschaften verschiedener Objektarten identifiziert, die Einfluss darauf haben, welcher Auftragstyp für ein bestimmtes Objekt naheliegend ist. Anschließend wurde für jede Kombination von Objekteigenschaften ein Auftragstyp zugeordnet. In Tabelle 19 ist diese Zuordnung am Beispiel von Gebäuden und Flugzeugen dargestellt. Eine vollständige Liste der Einflussfaktoren für die verschiedenen Objekttypen ist in Anhang E.4.1 zu finden.

Tabelle 19. Ermittlung des Auftragstyps anhand der Eigenschaften des Zielobjektes.

Einflussfaktoren			Auftragstyp
Objekttyp	Zugehörigkeit	Status	
Gebäude & Fahrzeuge	Unbekannt	Unbekannt	Investigate
	Verbündet & Neutral	Alle	Investigate
	Feindlich	Funktional	Engage
		Bekämpft	Battle Damage Assessment
Flugzeuge	Unbekannt	Unbekannt	Vis-Ident
	Feindlich	Funktional	Commit
	Verbündet	Funktional	Escort-Jamming

Das Erfolgskriterium wird im Rahmen der Auftragsvervollständigung nach einem ähnlichen Prinzip ermittelt. Dabei erfolgt die Auswahl durch ein direktes Mapping vom Auftragstyp auf das Erfolgskriterium. Die Zuordnung der Auftragsarten zu den jeweiligen Erfolgskriterien ist in Anhang E.4.2 dargestellt.

Die Auftragsparameter werden von der Auftragsvervollständigung auf unterschiedliche Weise ermittelt, je nachdem, um welchen Parameter es sich handelt. Einige Parameter werden durch Standardwerte definiert, während andere Parameter situationsabhängig festgelegt werden, z. B. unter Berücksichtigung der taktischen Lage oder der vom Piloten vorgenommenen Einstellungen. Tabelle 20 stellt exemplarisch einige Auftragsparameter und deren Ermittlungsprinzip dar.

Tabelle 20. Verschiedene Arten der Ermittlung von Parametern.

Auftragstyp	Parameter-Bezeichnung	Prinzip der Ermittlung
Engage	Anzahl Effektoren	Standardwert (Ein Effektor)
	Angriffsrichtung	Berechnung der optimalen Richtung sowie des optimalen Zünders unter Berücksichtigung feindlicher und verbündeter Bodentruppen.
	Zünder	
Investigate	Flughöhe	Übernahme der im MHDD eingestellten Kartenhöhe (begrenzt durch Sensorreichweite)
	Sensortyp	Standardwert (Elektro-Optisch)

4.4.3 Auftragsanalyse und -bewertung

Im Rahmen der Auftragskritik soll der Delegationsagent unzweckmäßige Auftragserteilungen identifizieren und Lösungsvorschläge zur Behebung der Probleme präsentieren.⁵³ Um diese Fähigkeit zu implementieren, ist eine technische Funktion erforderlich, die eine Überprüfung der Aufträge durchführt und im Falle einer ungeeigneten Auftragspezifikation Anpassungen der Spezifikation vorschlagen kann.

⁵³ Wie in 4.2.3 ausgeführt, beschränkt sich diese Arbeit auf den Anwendungsfall der ungeeigneten Auftragspezifikationen.

Die Erkennung von Problemfällen und die Generierung von Lösungsvorschlägen erfolgt für jeden Problemfall individuell, wobei sich der Implementierungsaufwand zur Erkennung und Lösungsgenerierung zwischen den verschiedenen Problemfällen zum Teil erheblich unterscheidet. Die folgende Tabelle 21 beschreibt für jeden Problemfall die implementierten Verfahren zur Erkennung und Generierung von Lösungen.

Tabelle 21. Implementiertes Vorgehen zur Erkennung und Lösung von Problemfällen.

Problemfall	Vorgehen zur Erkennung	Vorgehen zur Behebung
Suboptimale Angriffsparameter	<ul style="list-style-type: none"> • Effektabschätzung des Angriffsergebnis der Nutzerparameter • Ermittlung des optimalen Angriffsergebnis durch Simulation verschiedener Parametersets und Abgleich mit Nutzerlösung 	<ul style="list-style-type: none"> • Vorschlag des optimalen Parametersets
Auftragshöhe oberhalb von Schichtwolken	<ul style="list-style-type: none"> • Abgleich von Auftragshöhe und Wolkenbasis 	<ul style="list-style-type: none"> • Ermittlung des höchsten Geländepunktes im Aufklärungsbereich • Falls Wolkenbasis höher als Gelände liegt, erfolgt Vorschlag zur Reduktion der Auftragshöhe • Andererseits Meldung, dass Auftrag nicht durchführbar ist
Auftragshöhe nicht mit Gelände vereinbar	<ul style="list-style-type: none"> • Abgleich von Auftragshöhe und höchstem Geländepunkt im Aufklärungsbereich 	<ul style="list-style-type: none"> • Vorschlag zur Erhöhung der Auftragshöhe (Im Falle eines Investigate-Auftrags nur, falls die Bewölkung dies zulässt)
Sensorreichweite zu gering für Parameter	<ul style="list-style-type: none"> • Berechnung der erforderlichen Reichweite abhängig von Depressionswinkel⁵⁴ und Höhe • Abgleich mit Sensorreichweite 	<ul style="list-style-type: none"> • Vorschlag zur Verringerung der Aufklärungshöhe, falls das Gelände dies zulässt • Andernfalls Vorschlag eines höheren Depressionswinkels
Bewaffnung nicht ausreichend für Auslöseintervall	<ul style="list-style-type: none"> • Berechnung des erwarteten Verbrauchs von Effektoren • Abgleich mit Bordbewaffnung 	<ul style="list-style-type: none"> • Vorschlag zur Erhöhung des Auslöseintervalls

⁵⁴ Dieser Auftragsparameter ist in Anhang E.2 genauer beschrieben.

4.4.4 Verhaltensgenerierung

Wie in [3.5](#) beschrieben, benötigt der Plattformsteuerungsagent eine Funktion zur Generierung des UAV-Verhaltens auf Basis der zugewiesenen Aufträge und Verhaltensrichtlinien sowie der taktischen Lage.

Die Realisierung dieser technischen Funktion kann grundsätzlich durch eine Vielzahl von Verarbeitungsmethoden erfolgen. Allerdings sind bei der Auswahl der Methoden im vorliegenden Anwendungskontext zwei Anforderungen zu berücksichtigen, die die Auswahlmöglichkeiten einschränken:

- **Transparenz:** Damit der Nutzer seine Führungsrolle effektiv ausüben kann, muss das Verhalten des Agenten für ihn transparent sein. Dabei benötigt der Nutzer insbesondere Informationen auf der Entscheidungsebene (z. B. über die Ziele und Aktionen des Agenten), während Informationen auf der Ausführungsebene (z. B. über die genaue Flugroute) weniger bedeutsam sind (Olson & Wuennenberg, 2001). Bei der Auswahl geeigneter Methoden ist daher zu beachten, dass sie die Bereitstellung dieser Informationen ermöglichen.
- **Variabilität:** Militärische Kampfflugzeugmissionen zeichnen sich durch vielfältige Missionsaufgaben und eine hohe Dynamik aus. Daher kommen nur solche Methoden in Betracht, die in der Lage sind, in einer großen Bandbreite von Situationen ein angemessenes Verhalten zu erzeugen.

Entsprechend diesen Anforderungen erfolgte die Realisierung der Verhaltensgenerierung durch eine modulare Architektur mit hierarchischer Struktur. In den Modulen der Architektur werden verschiedene deliberative und reaktive Verarbeitungsmethoden eingesetzt (hybride Architektur).⁵⁵ Eine Übersicht über die Architektur ist in [Abbildung 60](#) dargestellt. Nachfolgend werden die Module der Architektur kurz vorgestellt.

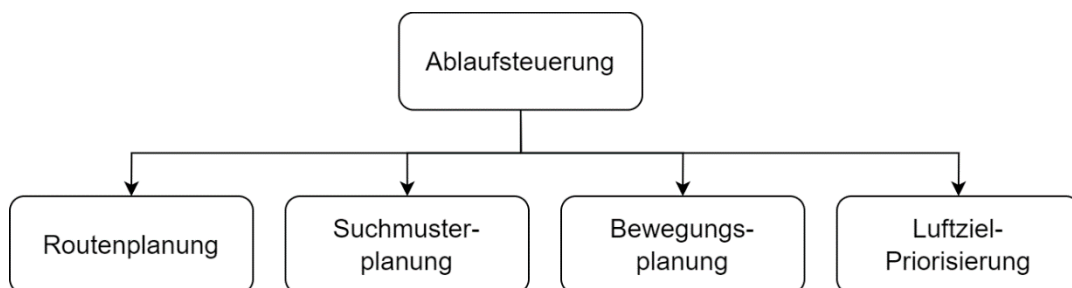


Abbildung 60. Modulare Architektur der Verhaltensgenerierung.

In der oberen Ebene der Architektur befindet sich ein Modul zur Ablaufsteuerung. Dieses Modul spielt eine wesentliche Rolle bei der Verhaltensgenerierung, da es die grundlegenden Entscheidungen in Bezug auf das Agentenverhalten trifft und deren Umsetzung steuert. In der unteren Ebene der Architektur befinden sich spezialisierte Module mit klar abgegrenztem

⁵⁵ Für eine detailliertere Beschreibung deliberativer, reaktiver und hybrider Architekturen sei auf [Abschnitt 2.4](#) verwiesen.

Funktionsumfang, die von der Ablaufsteuerung bei Bedarf eingesetzt werden können. Diese sind:

- Routenplanung: Dieses Modul ist in der Lage, Flugrouten zu generieren, die hinsichtlich der Flugdistanz und der Bedrohung durch Flugabwehrsysteme optimiert sind.
- Suchmusterplanung: Die Suchmusterplanung generiert Flugrouten für die Aufklärung von Zielgebieten.
- Bewegungsplanung: Dieses Modul koordiniert Flugbewegungen, um bewegten Objekten zu folgen, z.B. im Formationsflug.
- Luftziel-Priorisierung: Dieses Modul entscheidet über die Bekämpfungsreihenfolge feindlicher Luftfahrzeuge im Luftkampf.

Im Folgenden konzentriert sich diese Arbeit auf die Implementierung der Ablaufsteuerung, da sie innerhalb dieser Arbeit eine zentrale Rolle einnimmt und für die betrachtete Anwendung ein vergleichsweise neues und bislang wenig erforschtes Thema darstellt. Die Implementierung der Module der unteren Ebene wird hingegen nicht weiter behandelt, da diese Themenfelder vergleichsweise gut erforscht sind und bereits in zahlreichen Forschungsarbeiten aufgegriffen wurden.⁵⁶

4.4.4.1 Methoden zur Ablaufsteuerung

Die Implementierung der Ablaufsteuerung kann auf unterschiedliche Weise erfolgen. In einfachen Fällen kann eine Ablaufsteuerung bereits mit den Standardwerkzeugen einer Programmiersprache realisiert werden. Dieser Ansatz stößt jedoch mit zunehmender Komplexität des zu erzeugenden Systemverhaltens an seine Grenzen, da prozeduraler Code schnell unübersichtlich werden kann und die Verwaltung von Zuständen fehleranfällig wird, wenn eine Vielzahl von Bedingungen die Zustände beeinflussen. Aus diesen Gründen werden umfangreiche Ablaufsteuerungen häufig mit Hilfe von speziellen Methoden realisiert, die eine flexible und modulare Definition von Abläufen ermöglichen. Im Folgenden werden drei verbreitete Methoden zur Ablaufsteuerung vorgestellt, die in der Fachliteratur vielfach aufgegriffen und diskutiert werden (Colledanchise & Ögren, 2017; Yannakakis & Togelius, 2018; Young, 2014).

Eine *Finite-State Machine* (FSM) beschreibt das Verhalten eines Systems mithilfe von Zuständen, Zustandswechseln, Ereignissen und Aktionen (Yannakakis & Togelius, 2018). Eine Finite-State-Maschine befindet sich immer in genau einem Zustand, der mit der Ausführung bestimmter Aktionen verbunden ist. Der Wechsel eines Zustands ist bei einer FSM nicht beliebig möglich, sondern erfolgt in Form von explizit definierten Zustandswechseln, die ausgelöst werden, wenn bestimmte Ereignisse eintreten. Zur Veranschaulichung ist in Abbildung 61 eine (vereinfachte) Finite-State Machine zur Aufklärung eines Zielgebiets dargestellt.

⁵⁶ Exemplarische Forschungsarbeiten aus den jeweiligen Themenfeldern sind: Routenplanung: Sucas et al. (2012); Suchmusterplanung: Kingston et al. (2016); Bewegungsplanung: Justh & Krishnaprasad (2002); Luftziel-Priorisierung: Fu Li et al. (2012)

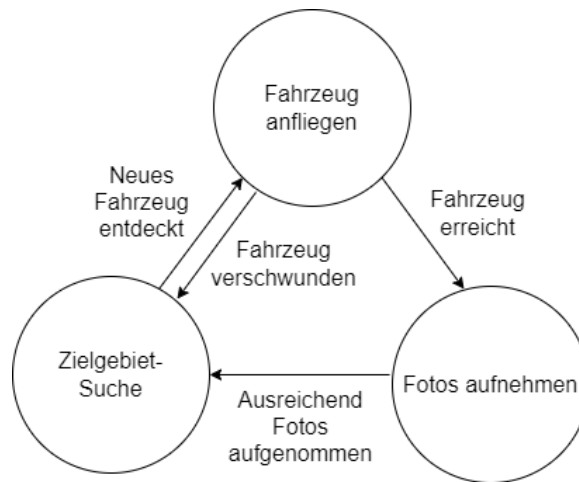


Abbildung 61. Finite-State-Machine zur Aufklärung eines Zielgebiets.
Der Ablauf der Finite-State-Machine wird durch die Übergänge zwischen den Zuständen definiert.

Beim *DecisionTree* wird das Verhalten eines Systems durch eine Baumstruktur definiert (Colledanchise & Ögren, 2017). Die Ablaufsteuerung erfolgt in der Methode des *DecisionTree* durch sequentielles Durchlaufen der Baumstruktur, ausgehend von der Wurzel. Die Verzweigungsknoten der Baumstruktur definieren dabei die Bedingungen, unter denen der nächste Knoten der Baumstruktur ausgewählt wird, während die Endknoten die auszuführenden Aktionen bzw. Entscheidungen des *DecisionTree* darstellen. Somit ergibt sich jede Aktion des Agenten als eine Aneinanderreihung von Bedingungen. In Abbildung 62 wird das Problem der Zielgebietaufklärung mithilfe eines *DecisionTree* beschrieben.

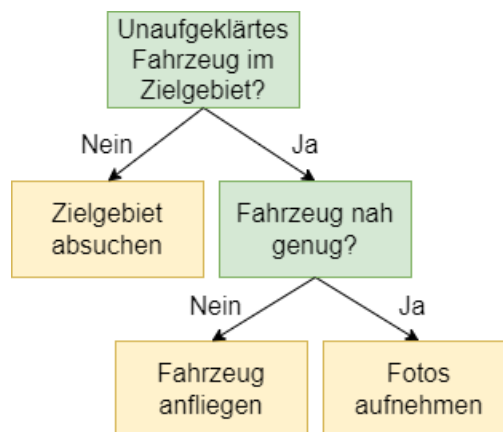


Abbildung 62. DecisionTree zur Aufklärung eines Zielgebiets.
Der Ablauf des DecisionTree wird durch die Bedingungen definiert.

Im *BehaviorTree* wird das Verhalten eines Systems ebenfalls in Form einer Baumstruktur enkodiert. Der *BehaviorTree* kann als eine generalisierte Form des *DecisionTree* angesehen werden, wobei sich die Ablaufsteuerung durch zwei Grundprinzipien vom *DecisionTree* unterscheidet (Colledanchise & Ögren, 2017):

1. Knotenstatus: Jeder Blattknoten meldet bei Ausführung einen Status zurück. Dieser Status beschreibt, ob der Knoten erfolgreich abgeschlossen wurde, erfolglos abgeschlossen wurde oder noch ausgeführt wird.

2. Kontrollflussknoten: Die inneren Knoten des BehaviorTree steuern den Kontrollfluss, indem sie die ihnen zugeordneten Kindknoten in einer bestimmten Reihenfolge ausführen. Die Reihenfolge der Ausführung hängt dabei vom Typ des Kontrollknotens sowie vom zurückgemeldeten Status der Kindknoten ab.

Diese beiden Prinzipien erlauben es, dass Abläufe im BehaviorTree explizit modelliert werden können, während diese im DecisionTree immer durch eine Reihe von zusätzlichen Bedingungen beschrieben werden müssen. Abbildung 63 veranschaulicht diese explizite Modellierung von Abläufen anhand des bekannten Anwendungsfalls der Zielgebietaufklärung.

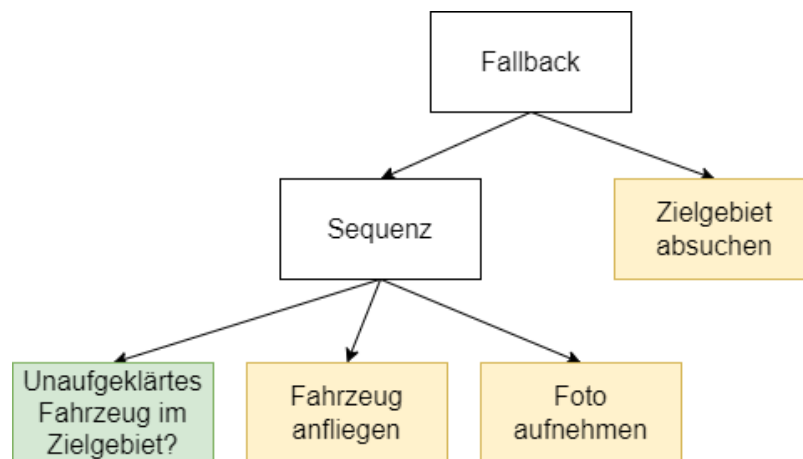


Abbildung 63. BehaviorTree zur Aufklärung eines Zielgebiets.

Der Ablauf des BehaviorTree wird durch den Erfolg/Misserfolg der Blattknoten sowie die Kontrollflussknoten bestimmt.

BehaviorTrees zeichnen sich durch ihre hohe Flexibilität aus, die durch die explizite Modellierung von Abläufen mit Hilfe von Kontrollknoten entsteht. In DecisionTrees und FSM müssen verschiedene Arten von Abläufen (z. B. Wiederholungen, Sequenzen) immer auf dieselbe Art und Weise⁵⁷ beschrieben werden, während in BehaviorTrees einfach ein neuer Knotentyp eingesetzt werden kann, der den entsprechenden Ablauf umsetzt. Dies hat zur Folge, dass BehaviorTrees einfacher erweitert werden können, da die resultierenden Strukturen eine höhere Modularität und bessere Lesbarkeit aufweisen. Die gute Lesbarkeit ermöglicht es auch, Verhaltensweisen mit den Nutzern während der Designphase zu diskutieren und zu verbessern und diese während der Nutzungsphase transparent darzustellen.

Aufgrund der genannten Vorteile wurde die Ablaufsteuerung in dieser Arbeit mithilfe eines BehaviorTrees umgesetzt. Im Folgenden wird hierfür zunächst das zur Umsetzung verwendete Framework erläutert. Anschließend wird beschrieben, wie das zur Durchführung der verschiedenen Missionsaufgaben erforderliche Agentenverhalten unter Nutzung dieses Frameworks umgesetzt wurde.

⁵⁷ In FSMs durch die Modellierung zusätzlicher Zustände und Zustandswechsel und in DecisionTrees durch zusätzliche Bedingungen.

4.4.4.2 Das Framework BehaviorTree.CPP

Die Realisierung des BehaviorTree zur Ablaufsteuerung erfolgte mit Hilfe des Frameworks *BehaviorTree.CPP*.⁵⁸ Dieses Framework umfasst eine umfangreiche Sammlung von Werkzeugen zur Modellierung und Verifizierung von BehaviorTrees sowie eine C++-Bibliothek zur Ausführung der erzeugten Baumstrukturen.

Die Modellierung der Ablaufsteuerung erfolgt im Framework BehaviorTree.CPP auf zwei Ebenen: Einer logischen Ebene und einer programmatischen Ebene. So wird auf der logischen Ebene der Ablauf des Agentenverhaltens in Form der Baumstruktur beschrieben und auf der programmatischen Ebene wird definiert, welche Programmanweisungen hinter den Knoten der Baumstruktur stehen. Abbildung 64 stellt dieses Prinzip in vereinfachter Form dar.

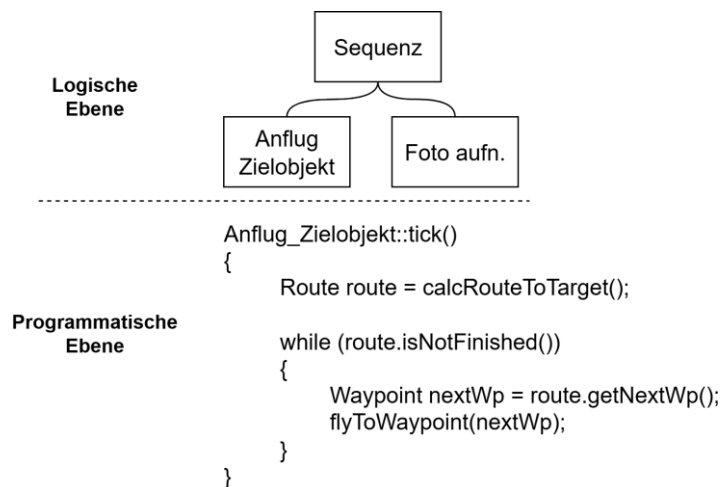


Abbildung 64. Prinzip der Trennung von logischer und programmatischer Ebene.

Logische Ebene

Die Modellierung des Verhaltens auf der logischen Ebene erfolgt mit Hilfe eines grafischen Editors⁵⁹, der ein Auswahlménü für verschiedene Knotentypen bereitstellt und es ermöglicht, die Knoten mithilfe von Drag-and-Drop-Interaktionen zu einer gewünschten Baumstruktur zusammenzusetzen. Das Framework stellt den Nutzern hierzu im Auswahlménü einzelne vordefinierte Knotentypen bereit, die geläufige Anwendungsfälle abbilden können, in der Regel müssen Entwickler jedoch auch eigene Knotentypen definieren, um die spezifischen Anforderungen der jeweiligen Anwendung abzubilden.

Die vom Framework bereitgestellten Knotentypen umfassen primär Kontrollflussknoten, die die jeweiligen Kindknoten nach einem vorgegebenen Schema ausführen können. So stellt das Framework beispielsweise folgende Kontrollflussknoten bereit:

- Sequenz: Dieser Knotentyp führt die Kindknoten nacheinander aus, solange deren Ausführung erfolgreich ist. Ein Fehlschlag führt zum Abbruch der Sequenz.

⁵⁸ Eine Übersicht über dieses Framework ist unter <https://www.behaviortree.dev/> zu finden. Diese Arbeit basiert auf der Version 3.3.0.

⁵⁹ <https://github.com/BehaviorTree/Groot>.

- **Reaktive Sequenz:** Wie die Sequenz führt dieser Knotentyp die Kindknoten nacheinander aus, solange diese erfolgreich sind. Im Gegensatz zur klassischen Sequenz wiederholt er dabei die bereits erfolgreichen Kindknoten.
- **Fallback:** Dieser Knotentyp führt seine Kindknoten der Reihe nach aus, solange die Kindknoten nicht erfolgreich sind. Sobald ein Kindknoten erfolgreich ist, wird der Fallback abgebrochen.
- **Reaktiver Fallback:** Analog zum Fallback führt dieser Knotentyp seine Kindknoten der Reihe nach aus, solange die Kindknoten nicht erfolgreich sind. Im Gegensatz zum klassischen Fallback werden dabei vorherige Kindknoten wiederholt ausgeführt.

Die Integration eigener Knotentypen in die Baumstruktur setzt die vorherige Definition dieser Typen im grafischen Editor voraus. Für diese Definition sind folgende Angaben erforderlich:

- Bezeichnung des Knotens
- Knotenkategorie (Bedingung, Aktion oder Kontrollflusssteuerung)
- Namen von Ein- und Ausgabevariablen des Knotens

Die Implementierung der logischen Ebene ist abgeschlossen, wenn alle vorgesehenen Abläufe durch die Baumstruktur abgebildet sind. Das Framework speichert die Baumstruktur in Form einer XML-Datei, die später zur Ausführung des BehaviorTrees herangezogen werden kann.

Programmatische Ebene

Zur Modellierung der programmatischen Ebene ist im Framework festzulegen, welche Variablen, Objekte und Programmanweisungen hinter den Knotentypen der Baumstruktur stehen. Für alle vom Framework bereitgestellten Knotentypen ist diese Beschreibung bereits vorhanden. Für benutzerdefinierte Knotentypen muss hingegen eine C++-Klasse mit folgenden Elementen definiert werden:

- **Konstruktor:** Benennt den Knotentyp und definiert die Kategorie (Bedingung, Aktion od. Kontrollflusssteuerung).
- **Variablen:** Namen und Datentypen der Ein- und Ausgabevariablen des Knotens.
- **Tick-Funktion:** Programmablauf, der beim Aufruf des Knotens ausgeführt werden soll.
- **Halt-Funktion (Optional):** Programmablauf, der ausgeführt werden soll, wenn die Ausführung eines Knotens beendet werden soll.

Verknüpfung der Ebenen

Die Aufteilung von logischer und programmatischer Ebene macht im Framework eine Verknüpfung zwischen den Ebenen erforderlich, damit während der Abarbeitung der logischen Struktur eindeutig bestimmt werden kann, welche Knotentypen ausgeführt werden sollen. Zu diesem Zweck muss für jeden benutzerdefinierten Knotentyp eine Registrierung vorgenommen werden, bei der der im XML verwendete Knotenname der entsprechenden C++-Klasse zugeordnet wird.

Ausführung der Baumstruktur

Die Ausführung eines BehaviorTrees erfolgt mithilfe einer vom Framework-Nutzer zu implementierenden C++-Applikation, die eine entsprechende Laufzeitbibliothek des Frameworks einbindet. Innerhalb dieser Anwendung wird auf Grundlage der XML-Beschreibung eine lauffähige Instanz des BehaviorTrees erzeugt. Die Ausführung dieser Instanz erfolgt durch den zyklischen Aufruf der Tick-Funktion des Wurzelknotens; dieser Aufruf veranlasst die Laufzeitbibliothek dazu, die Baumstruktur gemäß den Regeln der Modellierungssprache zu durchlaufen.

4.4.4.3 Umsetzung des Agentenverhaltens im Framework

Die Umsetzung des Agentenverhaltens mithilfe des Frameworks BehaviorTree.CPP basierte auf drei Schritten, die in der praktischen Umsetzung im Sinne eines iterativen Vorgehens wiederholt durchlaufen wurden. Im ersten Schritt wurden die zu erzielenden Verhaltensweisen des Agenten definiert. Daraufhin erfolgte im zweiten Schritt die Modellierung dieser Verhaltensweisen auf der logischen Ebene des Frameworks anhand der Regeln der zugrunde liegenden Modellierungssprache. Anschließend wurde im dritten Schritt eine Anwendung implementiert, in der die programmatische Ebene definiert ist und mit der die erzeugten Baumstrukturen ausgeführt werden können. Im Folgenden werden die verschiedenen Schritte näher erläutert.

Definition des Agentenverhaltens

In diesem Schritt galt es zu definieren, welche Verhaltensweisen zur erfolgreichen Bearbeitung der Missionsaufgaben der unbemannten Plattformen zum Einsatz kommen sollen. Dabei war auch zu berücksichtigen, wie sich verschiedene Umgebungsfaktoren auf diese Verhaltensweisen auswirken. So verändert beispielsweise das Auftreten feindlicher Bedrohungen bereits in einfachen Navigationsaufgaben, welche Verhaltensweisen als taktisch sinnvoll gelten.

Die Beschreibung dieser Verhaltensweisen erforderte daher ein umfassendes Verständnis der erwarteten Taktiken und Rahmenbedingungen in zukünftigen Luftoperationen. Dieses Wissen wurde nicht anhand einer spezifischen Methode erworben, sondern im Verlauf der Forschungsarbeit durch eine Vielzahl unterschiedlicher Informationsquellen aufgebaut:

- **Fachliteratur:** Ein Element des Wissenserwerbs war das Studium von Fachliteratur zu Fachbegriffen, physikalischen Grundlagen und Taktiken der Luftkriegsführung sowie zu Einsatzszenarien in aktuellen und zukünftigen Luftoperationen (Boyne, 2002; Shaw, 1985; Wills, 2016). Das Ziel bestand darin, ein Verständnis von den verschiedenen Missionsaufgaben und deren Rahmenbedingungen zu entwickeln.
- **Besuche in fliegenden Verbänden:** Durch das Begleiten von einzelnen Inhalten der taktischen Ausbildung angehender Kampfflugzeugpiloten und die Beobachtung von Trainingsmissionen konnte ein tiefergehendes Verständnis über Taktiken und Hintergründe des Luftkampfes gewonnen werden.
- **Interviews mit Kampfflugzeugpiloten:** Um die physikalischen Grundlagen und daraus abgeleitete taktische Vorgehensweisen im Kontext von Luftkampf, SEAD und

elektronischer Kampfführung besser zu verstehen, wurden mehrere Interviews mit aktiven und ehemaligen Kampfflugzeugpiloten der Luftwaffe geführt.

- Feedback zu modellierten Verhaltensweisen: Im Rahmen des iterativen Vorgehens wurden Verhaltensweisen in verschiedenen Reifestadien (z. B. als abstrakte Beschreibungen oder erste Implementierungen) mit Piloten diskutiert. Dieses Feedback wurde zur gezielten Weiterentwicklung der Verhaltensweisen genutzt.

Mithilfe dieses Wissens konnten die gewünschten Verhaltensweisen des Agenten in Abhängigkeit von Missionsaufgabe und Rahmenbedingungen in abstrakter Form beschrieben werden. Als Anwendungsbeispiel für eine solche Beschreibung dient im Folgenden das Verhalten des Agenten während eines „Combat Air Patrol“-Auftrags zur Bekämpfung feindlicher Luftfahrzeuge. Im Rahmen dieses Auftrags ist das gewünschte Verhalten der unbemannten Plattform wie folgt: Vorrangig sollen feindliche Luftfahrzeuge in der Nähe der Zielposition bekämpft werden. Sind keine feindlich klassifizierte Luftfahrzeuge vorhanden, sollen alternativ nicht klassifizierte Luftfahrzeuge aufgeklärt werden. Falls auch diese nicht vorhanden sind, soll die Zielposition angefliegen werden und vor Ort ein „Racetrack“-Pattern durchgeführt werden. Die folgende Abbildung 65 zeigt exemplarisch, wie dieses Verhalten in abstrahierter Form dargestellt werden kann.

```
START_VERHALTENSBESCHREIBUNG
VERHALTENSALTERNATIVEN:
  PRIORITÄT 1:
    VORAUSSETZUNG: Feindliche Luftfahrzeuge vorhanden
    HANDLUNG: Bekämpfung feindlicher Luftfahrzeuge

  PRIORITÄT 2:
    VORAUSSETZUNG: Nicht-klassifizierte Luftfahrzeuge vorhanden
    HANDLUNG: Aufklärung nicht-klassifizierter Luftfahrzeuge

  PRIORITÄT 3:
    VORAUSSETZUNG: Keine Voraussetzung (immer ausführbar)
    HANDLUNG:
      ABFOLGE:
        Anfliegen Zielposition
        Ausführung Racetrack-Pattern
ENDE_VERHALTENSBESCHREIBUNG
```

Abbildung 65. Abstrakte Beschreibung des gewünschten Agentenverhaltens während eines „Combat Air Patrol“-Auftrags.

Modellierung der Verhaltensweisen

In diesem Schritt wurden die zuvor definierten Verhaltensweisen auf der logischen Ebene des Frameworks modelliert. Aufgrund der hohen Flexibilität, die das Framework durch den Einsatz verschiedener Kontrollflussknoten bietet, bestand die zentrale Herausforderung dabei weniger in der bloßen Umsetzung des Verhaltens, sondern vielmehr in einer möglichst sinnvollen Modellierung, da sich verschiedene Modellierungsansätze beispielsweise in Hinblick auf die Effektivität und Nachvollziehbarkeit des Verhaltens deutlich unterscheiden können. Vor diesem Hintergrund werden im Folgenden zwei zentrale Prinzipien erläutert, die während der Modellierung des Agentenverhaltens angewendet wurden.

Ein Modellierungsprinzip betrifft die Auswahl von Kontrollflussknoten zur Umsetzung der Verhaltensweisen. Gemäß diesem Prinzip wurden Sequenzen immer dann eingesetzt, wenn Abläufe mit einer festen Reihenfolge beschrieben werden sollten, während Fallbacks genutzt wurden, um alternative Verhaltensweisen und deren Priorität zu modellieren.

Ein exemplarischer Anwendungsfall für die Modellierung von festen Abläufen mithilfe von Sequenzen ist das Verhalten des Agenten während eines Auftrags zur Bekämpfung einer Radarstellung („DEAD“-Auftrag). Im Rahmen dieses Auftrags besteht das gewünschte Verhalten des Agenten aus zwei aufeinanderfolgenden Schritten (Anfliegen und Bekämpfen). Abbildung 66 A zeigt, wie dieses Verhalten mithilfe einer Sequenz abgebildet werden kann.

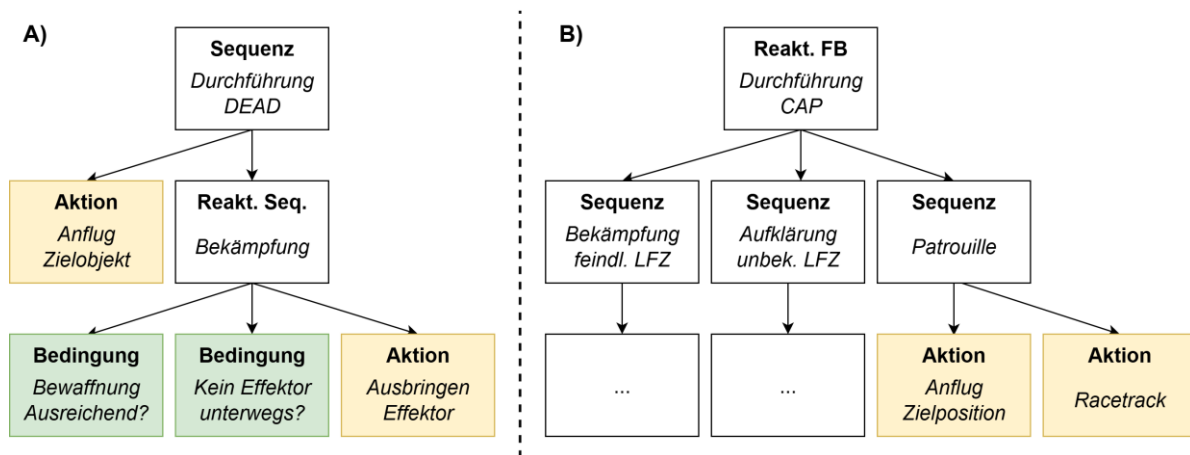


Abbildung 66. Anwendungsbeispiele von Sequenzen und Fallbacks (Verhalten vereinfacht).

A) Sequenz zur Modellierung des Verhaltens im Rahmen der Bekämpfung einer Radarstellung.

B) Fallback zur Modellierung der Prioritäten von Verhaltensweisen während eines „Combat Air Patrol“-Auftrags.

Als Beispiel für die Verwendung von Fallbacks zur Modellierung von alternativen Verhaltensweisen und deren Priorisierung dient der „Combat Air Patrol“-Auftrag, für den das gewünschte Agentenverhalten zuvor beschrieben wurde. Abbildung 66 B zeigt, wie ein (reaktiver) Fallback in diesem Anwendungsfall zur Priorisierung der Verhaltensweisen eingesetzt werden kann.

Das zweite Modellierungsprinzip bestand in der Gliederung der Baumstruktur mithilfe von *Sub-Trees* (Teilbäumen). Dabei handelt es sich um eigenständige Baumstrukturen bestehend aus mehreren Knoten, die wie einzelne Knoten in der (Haupt-)Baumstruktur platziert werden können. So wurden beispielsweise mehrere Knoten, die gemeinsam eine bestimmte Verhaltensweise repräsentieren (z. B. der Ablauf zur Aufklärung eines Bodenziels) zu Sub-Trees zusammengefasst und an verschiedenen Stellen im BehaviorTree eingesetzt (Abbildung 67). Dieser Ansatz ermöglicht es, komplexe Verhaltensweisen im Sinne eines modularen Aufbaus zu kapseln, was die Übersichtlichkeit der Baumstruktur erhöht und die Anpassung einzelner Verhaltensweisen erleichtert.

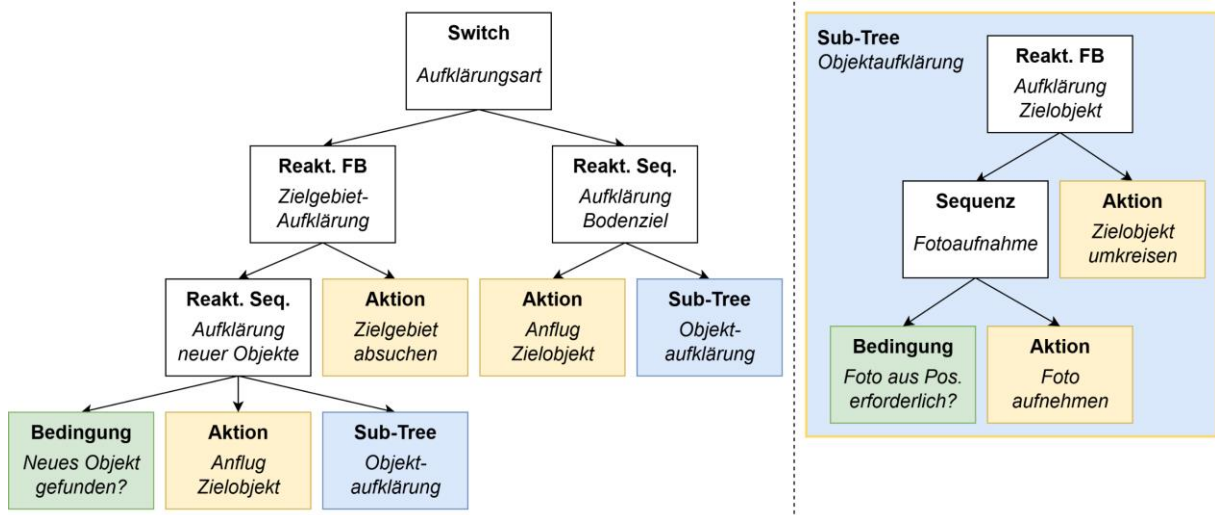


Abbildung 67. Einsatz von Sub-Trees im BehaviorTree. Links: Modellierung des Verhaltens zur Aufklärung mit zweifacher Verwendung eines Sub-Tree zur Objektaufklärung. Rechts: Baumstruktur des Sub-Tree zur Objektaufklärung.

Sub-Trees wurden auch zur Strukturierung des Agentenverhaltens im Rahmen von Aufträgen genutzt, indem die Verhaltensweisen für die einzelnen Auftragsstypen in verschiedenen Sub-Trees untergebracht wurden (Abbildung 68). Dies sollte die Nachvollziehbarkeit des Verhaltens verbessern, da die Verhaltensweisen zur Durchführung eines Auftrags somit ausschließlich innerhalb des entsprechenden Sub-Trees definiert sind und nicht an beliebigen Stellen in der Baumstruktur auftreten können.

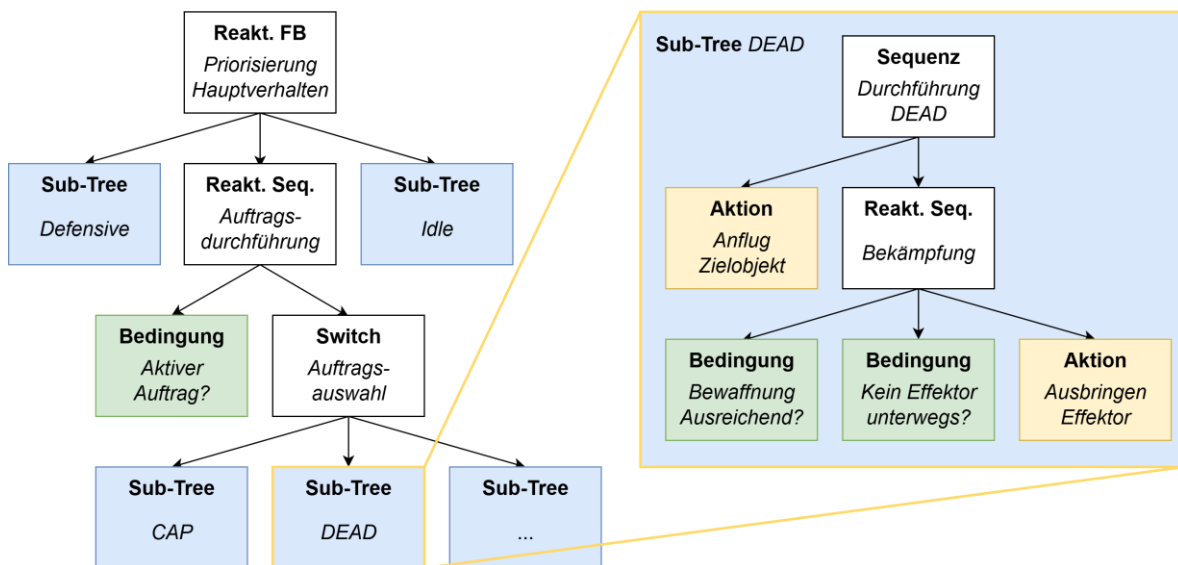


Abbildung 68. Grundstruktur des implementierten BehaviorTree. Die Verhaltensweisen im Rahmen der verschiedenen Aufträge sind in Sub-Trees untergebracht – hier verdeutlicht am Beispiel des „DEAD“-Auftrags.

Implementierung der Laufzeitanwendung

Dieser Umsetzungsschritt bestand in der Implementierung der C++-Applikation zur Ausführung der modellierten Baumstrukturen. Ein wichtiger Bestandteil im Rahmen der Implementierung dieser Anwendung war die Definition und Registrierung der benutzerdefinierten Knotentypen der Baumstruktur. Zu diesem Zweck wurde für jeden dieser Knotentypen eine entsprechende C++-Klasse inklusive der zugehörigen Variablen und

Methoden definiert (Abbildung 69 oben). Dann wurden die jeweiligen Klassen über den Registrierungsmechanismus des Frameworks mit den entsprechenden Knotentypen der Baumstruktur verknüpft (Abbildung 69 unten).

```

Definition Knotentyp → FlyToPosition::FlyToPosition(const std::string& name)
                        : AsyncActionNode(name)
                        {
                        }

Ein-/Ausgabe Variablen-Definition → PortsList FlyToPosition::providedPorts()
                                        {
                                        return { InputPort<Position>("position") };
                                        }

Funktion bei Knoten-Aufruf → NodeStatus FlyToPosition::tick()
                            {
                            // Eingabe holen
                            Position targetPos = getInput("position");

                            // Durchführung, bis der BehaviorTree den Knoten von außen beendet
                            while(!haltRequested())
                            {
                                auto ownPlatform = getOwnPlattform(ownPlatformId);
                                double distance = computeDistance(ownPlatform.position, targetPos);

                                // Wenn nahe genug am Ziel → Erfolg (beendet den Knoten ebenfalls)
                                if(distance < threshold)
                                    return SUCCESS;

                                // Kurs und Geschwindigkeit berechnen
                                double heading = computeHeading(ownPlatform.position, targetPos);
                                Task currentTask = getCurrentTask(ownPlatform.id);
                                double speed = computeDesiredSpeed(distance, currentTask);

                                // Vorgabe an Flugmanagementsystem der Plattform
                                setAutopilotCommand(heading, speed, targetPosition.altitude);

                                // Kurze Pause
                                sleep(smallInterval);
                            }

                            return FAILURE;
                            }

-----

Funktionsaufruf zur Registrierung → void NodeRegistration::registerAllNodes(BehaviorTreeFactory& factory)
                                        {
                                        ///... (Andere Knoten-Registrierungen)

                                        // Ordnet die C++-Klasse "FlyToPosition" dem XML-Knotentyp "FlyToPos" zu
                                        factory.registerNodeType<FlyToPosition>("FlyToPos");

                                        // ...
                                        }

```

Abbildung 69. Implementierung von benutzerdefinierten Knoten im C++-Code (vereinfacht).

Oben: Klassendefinition eines Knoten zum Anfliegen einer Zielposition.

Unten: Verknüpfung von C++-Klassendefinition und Knotentyp im XML-Dokument (Registrierung).

Ein weiterer Bestandteil der Anwendungsimplementierung war das Management der Laufzeit-Instanzen der BehaviorTrees für alle unbemannten Plattformen. Dies wurde umgesetzt, indem in der Anwendung für jede unbemannte Plattform ein unabhängiger Ausführungsthread angelegt wird, der jeweils eine eigenständige BehaviorTree-Instanz

anlegt⁶⁰ und deren regelmäßige Ausführung (und gegebenenfalls Beendigung) sicherstellt. Abbildung 70 stellt das Management der Laufzeit-Instanzen im Rahmen der Applikation exemplarisch dar.

```
void BehaviorTreeManagement::createAndExecuteTrees()
{
    auto teamUAVs = getAllTeamUAVs();

    // Anlegen einer Factory zur Erzeugung von BehaviorTrees (Factory Pattern)
    BehaviorTreeFactory factory;

    // Aufruf der Knotenregistrierungsfunktion
    NodeRegistration::registerAllNodes(factory);

    // Erzeuge einen Thread pro unbemannte Plattform
    std::vector<std::thread> threads;
    for (auto uav : teamUAVs)
    {
        threads.emplace_back([&factory, uav]() {
            // Instanziierung des BehaviorTree aus dem XML-File
            auto tree = factory.createTreeFromFile("behavior_tree.xml");

            // Kontinuierliche Ausführung
            while (treeShouldRun(uav->id))
            {
                tree.tickRoot();
                sleep(tickInterval);
            }
        });
    }
}
```

Abbildung 70. Erzeugung und Ausführung der BehaviorTree-Instanzen für alle unbemannten Teammitglieder (vereinfacht).

Ergebnis der Implementierung

Das Ergebnis der drei Implementierungsschritte ist ein BehaviorTree, der die Ablaufsteuerung der unbemannten Systeme für alle verschiedenen Ausprägungen von Aufträgen sowie für die Idle- und Defensive-Verhaltensweisen übernimmt. Insgesamt wurden hierfür 68 eigene Knotentypen entwickelt, die gemeinsam mit den vom Framework bereitgestellten Knotentypen 346-mal eingesetzt wurden, um den BehaviorTree vollständig abzubilden. Eine genauere Beschreibung der resultierenden Baumstruktur sowie der damit umgesetzten Verhaltensweisen findet sich in Anhang [E.5](#).

⁶⁰ Alle Instanzen des BehaviorTree basieren auf derselben XML-Baumstruktur und zeigen somit in identischen Situationen ein identisches Verhalten.

4.5 Simulator-Integration

Nachdem die Implementierung der Nutzerschnittstelle und der technischen Funktionen beschrieben wurde, wird nachfolgend die Integration dieser Komponenten in die Simulationsumgebung vorgestellt.

Die Simulationsumgebung des Kampfflugzeugsimulators wird in einem verteilten Rechnernetzwerk erzeugt, in dem verschiedene Simulations- und GUI-Anwendungen in einer gemeinsamen Architektur zusammenarbeiten. Diese Architektur wurde im Rahmen der Integrationsarbeiten modifiziert, um die neuen technischen Funktionalitäten zu integrieren und Gestaltungsmaßnahmen an der Nutzerschnittstelle vorzunehmen. In diesem Zuge wurden neue Anwendungen erstellt und bestehende Anwendungen überarbeitet. In Abbildung 71 sind die für diese Arbeit relevanten Teile dieser Systemarchitektur dargestellt. Nicht berücksichtigt sind in dieser Übersicht die Anwendungen, die im Rahmen der Forschung zur Pilotenbeobachtung und -assistenz eingesetzt werden. Nachfolgend werden die in der Übersicht dargestellten Anwendungen und ihr Beitrag zur Simulationsumgebung vorgestellt.

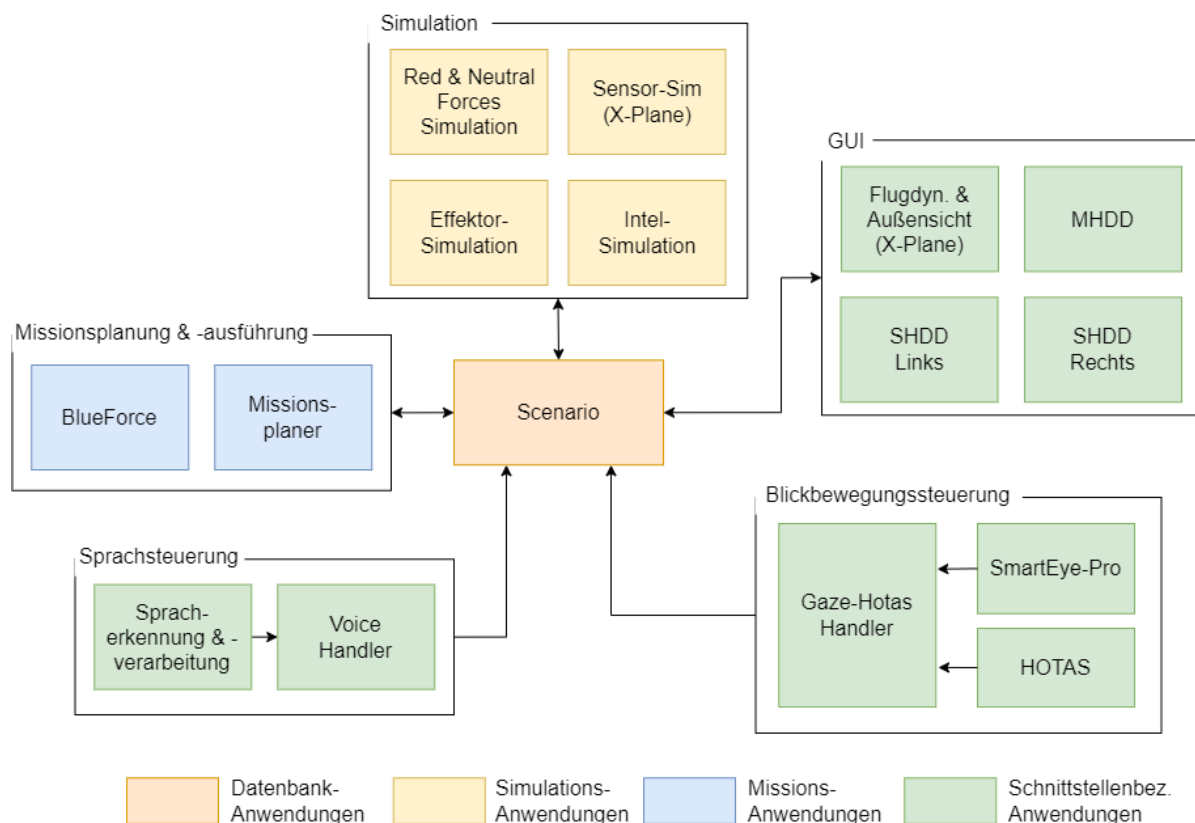


Abbildung 71. Simulator-Architektur mit integrierten Funktionen.

Das zentrale Element der Architektur ist das *Scenario*. Das Scenario fungiert als Datenbank für die Simulationsdaten. Zu diesem Zweck verwaltet das Scenario die Zustände aller Szenarienelemente (*GroundTruth*) sowie die Daten der Nutzerschnittstelle⁶¹ und verteilt diese Daten bei Bedarf an die verschiedenen Applikationen. Die Anpassung der Simulationsdaten erfolgt somit immer über das Scenario, d.h. wenn eine Anwendung Daten

⁶¹ Im MVC-Entwurfsmuster entspricht dies den Funktionen „Model“ und „Controller“.

anpassen möchte (z.B. die Position eines Objektes), sendet sie den Änderungswunsch an das Scenario, das Scenario ändert die Datenstrukturen entsprechend und verteilt die aktualisierten Daten an alle Anwendungen.

Der *Missionsplaner* und die *BlueForce* sind die zentralen Missionssysteme, die zur Planung und Ausführung der verschiedenen Kampfflugzeugmissionen benötigt werden. Der Missionsplaner generiert auf Anfrage von anderen Applikationen Missionspläne. Hierzu nutzt der Planer die zuvor vorgestellte Funktion der Missionsplanung (Abschnitt [4.4.1](#)). Die *BlueForce* ist für die Steuerung und Simulation der unbemannten Kräfte des verbündeten Teams (der *BlueForce*) zuständig. In diesem Rahmen integriert die *BlueForce* für jedes UAV sowohl eine Komponente zur Verhaltensgenerierung ([4.4.4](#)) als auch ein Flugdynamik-Modell, mit dem die Steuervorgaben der Verhaltensgenerierung simuliert werden können.

Zur Integration der Sprachsteuerung dienen zwei Anwendungen. In der *Spracherkennung und -verarbeitung* wird das Audiosignal des Headsets analysiert, um eventuelle Sprachbefehle des Nutzers zu identifizieren.⁶² Der *VoiceHandler* ist anschließend dafür zuständig, erkannte Befehl umzusetzen, also beispielsweise die entsprechenden Missionsplanungsanfragen zu stellen.

Die Integration der Delegationsmethode Gaze-HOTAS erfolgt mithilfe von drei Anwendungen. Die Anwendung *SmartEye Pro* ist für die Blickbewegungsmessung zuständig.⁶³ Die *HOTAS*-Anwendung verarbeitet die Eingaben an den Steuerelementen. Die Daten der Blickbewegungsmessung und der Nutzereingaben werden in der Applikation *Gaze-HOTAS Handler* zusammengeführt, um die Auswahl von Display-Objekten zu steuern.

Zur Darstellung der Displays in Außensicht und Cockpit verfügt die Architektur über vier GUI-Applikationen. Diese sind:

- Die Anwendung *Flugdynamik & Außensicht* ist für die Simulation des bemannten Kampfflugzeugs und die Bildgenerierung von dessen Außensicht zuständig. Die Berechnung der Flugdynamik und die Erzeugung der Außensicht erfolgt in einer Instanz der Flugsimulationssoftware X-Plane-11.
- Die Anwendungen *MHDD* und *SHDD* (links bzw. rechts) erzeugen die grafische Benutzeroberfläche der gleichnamigen Touchscreen-Displays.

Die Architektur umfasst zudem vier Simulationsanwendungen. Jede dieser Anwendungen ist für bestimmte Simulationsfunktionen zuständig:

- Die *Red & Neutral Forces Simulation* ist für die Simulation von feindlichen Kräften (sogen. *RedForce*) und neutralen Entitäten zuständig. In diesem Rahmen modelliert die Anwendung das Verhalten und die Dynamik von zivilen und militärischen Flugzeugen sowie das Verhalten und die Radarphysik von Flugabwehrstellungen.
- Die *Effektor-Simulation* simuliert das Flugverhalten und die Wirkung verschiedener Effektoren wie Luft-Luft- und Luft-Boden-Raketen.

⁶² Gemäß der in [4.3.1](#) vorgestellten technischen Funktion.

⁶³ Diese Anwendung wurde in Abschnitt [4.1](#) näher ausgeführt.

- Die *Sensor-Simulation* dient der Nachbildung der Kamera-Sensoren der unbemannten Kräfte. Im Rahmen dessen stellt diese Anwendung einen Sensor-Stream sowie eine Funktion zur Aufnahme von Fotos aus der Perspektive der verschiedenen Kräfte bereit (Elektro-Optisch und Infrarot).
- Die *Intel-Simulation* simuliert die Lagebildgenerierung der Missionsteilnehmer. In diesem Rahmen aggregiert die Intel-Simulation die Informationen, welche von den einzelnen Entitäten der Blue- bzw. RedForce gesammelt wurden und erstellt aus diesen ein konsolidiertes Lagebild, dessen Genauigkeit sich abhängig von der Menge und Qualität der gesammelten Informationen von der GroundTruth unterscheidet.

5. Validierungsstudien

Dieses Kapitel beschreibt die experimentelle Validierung des entwickelten Systems. Zur Vorbereitung der Validierung wurden zunächst verschiedene Vorstudien durchgeführt, die der Optimierung des Systems und der Weiterentwicklung der Experimentalmethodik dienen. Im Rahmen der eigentlichen Validierung wurden mehrere Untersuchungen durchgeführt, die individuelle Forschungsfragen adressierten und dabei entweder das Mensch-Maschine-System als Ganzes oder einzelne Aspekte des Systems evaluierten. Eine Übersicht über die durchgeführten Untersuchungen ist in Tabelle 22 dargestellt.

Tabelle 22. Untersuchungen im Rahmen der experimentellen Validierung.

Studientyp	Teilnehmer	Untersuchungsgegenstand
Vorstudie	10 Offiziere/ Offiziersanwärter	Vergleich der Delegationsmethoden Timeline und Voice in verschiedenen Missionsphasen (5.1.1)
		Vergleich der Delegationsmethoden Touch-Gesture, Voice & Gaze-HOTAS (5.1.2)
		Vergleich von Modalitätenpaarungen von Nutzerdialogen (5.1.3)
Gesamtsystem- erprobung	10 Kampfflugzeugpiloten	Nutzungsverhalten von Funktionen zur UAV-Führung (5.2)
	6 Kampfflugzeugpiloten	Bewertung des Systems (5.3)
Detailstudie	8 Kampfflugzeugpiloten	Vergleich der Delegationsmethoden Touch-Gesture, Voice & Gaze-HOTAS im realistischen Missionskontext (5.4)
		Vergleich von kritikbasierter (Auftragskritik) und vorschlagsbasierter Entscheidungshilfe (5.5)

5.1 Vorstudien

Die in diesem Abschnitt dargestellten Vorstudien dienten sowohl der Weiterentwicklung der Experimentalmethodik als auch der Erprobung und Optimierung der eingesetzten Systemfunktionen.

Jede der nachfolgend dargestellten Studien wurde mit jeweils zehn Offizieren oder Offiziersanwärtern der Universität der Bundeswehr München durchgeführt.

5.1.1 Vergleich von Delegationsmethoden in verschiedenen Missionsphasen

Diese Vorstudie steht im Zusammenhang mit zwei Veröffentlichungen (Dudek & Schulte, 2022a; Gallersdörfer, 2021).

Studienziel: Untersuchung der Vor- und Nachteile der Delegationsmethoden Timeline und Voice sowie des Verhaltens der Probanden bei der Auswahl einer Methode.

Studiendesign: In der Studie führten die Probanden jeweils drei Manned-Unmanned Teaming Missionen durch, in denen sie Aufträge an drei UAVs delegieren mussten. Jede Mission sollte mit einer anderen Delegationsmethode absolviert werden und bestand aus verschiedenen Missionsphasen mit unterschiedlicher Aufgabenbelastung. Dabei wurden verschiedene Performance-Maße sowie die Methodenpräferenz der Probanden ermittelt.

Ergebnisse: Die Performance und Methodenpräferenz hängen stark von der Missionsphase ab. Die Delegationsmethode Voice hatte in allen Flugphasen eine höhere Performance und Präferenz, während die Delegationsmethode Timeline in der Vorplanungsphase am Boden eine höhere Performance und Präferenz aufwies. Die höchste Performance hatten die Probanden, wenn sie frei zwischen den Methoden wählen konnten, allerdings könnten Trainingseffekte zu diesem Ergebnis beigetragen haben.

Schlussfolgerungen: Die Bereitstellung mehrerer Delegationsmethoden könnte vorteilhaft sein. Es sind weitere Untersuchungen notwendig, um diese These zu bestätigen und Trainingseffekte ausschließen zu können.

5.1.2 Vergleich von Delegationsmethoden für dynamische Auftragserteilung

Diese Vorstudie steht im Zusammenhang mit zwei Veröffentlichungen (Dudek & Schulte, 2023; Kühn, 2022).

Studienziel: Untersuchung der Vor- und Nachteile der Delegationsmethoden Touch-Gesture, Voice und Gaze-HOTAS und Untersuchung der Effekte der Bereitstellung verschiedener Methoden.

Studiendesign: Die Probanden führten vier Missionen durch, in welchen sie dynamisch Aufträge an drei UAVs verteilen mussten. In drei der vier Mission sollten die Probanden eine spezifische Delegationsmethode verwenden, in der vierten Mission konnten sie die Methoden frei wählen. Innerhalb einer Mission wurde die Aufgabenbelastung durch zwei

Nebenaufgaben variiert. Während der Missionsdurchführung wurde die Performance in den Haupt- und Nebenaufgaben sowie die Methodenpräferenz gemessen.

Ergebnisse: Probanden delegierten Aufträge am schnellsten in Missionen mit Touch-Gesture, gefolgt von Missionen mit freier Wahl und Voice. Die Performance in der Nebenaufgabe war am höchsten, wenn die Methode frei gewählt werden konnte. In Missionen, in denen eine Methode vorgegeben wurde, war die Performance in der Nebenaufgabe geringer, wenn die Haupt- und Nebenaufgabe dieselbe Eingabemodalität verwendeten. Bei freier Wahl der Methode verwendeten Probanden überwiegend Voice, gefolgt von Touch-Gesture. Probanden passten ihre Methodenauswahl je nach Situation an und wechselten die Methode, wenn während der Nutzung Probleme auftraten.

Schlussfolgerungen: Durch die Bereitstellung mehrerer Delegationsmethoden konnten die Interaktionszeiten der schnellsten Methode nicht unterboten werden. Allerdings legen das Optimierungsverhalten bei der Methodenauswahl sowie die Ressourcenkonflikte in der Nebenaufgabe nahe, dass die Resilienz durch die Bereitstellung mehrerer Methoden erhöht werden könnte. Um die Effekte der Bereitstellung mehrerer Methoden weiter zu untersuchen, wird vorgeschlagen, eine Studie mit Kampfflugzeugpiloten durchzuführen. In einer solchen Studie sollten Performance-Maße definiert werden, die den Erfolg der Mission beschreiben und nicht die Effizienz einer Interaktion. Piloten handeln schließlich auch mit dem Ziel, die Missionsziele zu erreichen, anstatt Interaktionen zu optimieren.

5.1.3 Vergleich von Modalitäten von Nutzerdialogen

Diese Vorstudie steht im Zusammenhang mit einer Masterarbeit (Hofmann, 2022).

Studienziel: Untersuchung von Interaktionsmethoden für system-initiierte Nutzerdialoge auf Basis zweier Modalitätenpaarungen (visuell-manuell und auditiv-verbal).

Studiendesign: In der Studie führten die Probanden mehrere Missionen durch, in denen sie die Durchführung von Aufträgen durch UAVs überwachen mussten. Während der Missionen mussten die Probanden auf systeminitiierte Nachrichten reagieren, die sich aus der Auftragsdurchführung ergaben. Die erwartete Nutzereingabe eines Dialogs variierte von der Auswahl bestimmter Parameter bis hin zur einfachen Bestätigung (entsprechend verschiedener Automationsgrade). In der Studie wurden die Ein- und Ausgabemodalität sowie die Aufgabenlast variiert und die Performance in den Haupt- und Nebenaufgaben gemessen.

Ergebnisse: Auditiv-verbale Dialoge wurden langsamer bearbeitet als visuell-manuelle Dialoge, waren jedoch auch weniger fehleranfällig. Dialoge, die den Probanden größere Entscheidungsfreiheiten einräumten, waren mit höheren Bearbeitungszeiten assoziiert. Die Performance in der Nebenaufgabe war geringer, wenn Haupt- und Nebenaufgabe dieselbe Modalitätenpaarung verwendeten.

Schlussfolgerungen: Die Auswahl von Modalitäten zur Dialogführung sollte sowohl von der Priorität (d. h. seiner Relevanz für die Mission) als auch von der Dringlichkeit des Dialogs (dem verfügbaren Zeitfenster zur Bearbeitung) abhängig gemacht werden. Bei hoher Priorität sollte tendenziell die auditiv-verbale Modalitätenpaarung bevorzugt werden, während bei hoher Dringlichkeit die visuell-manuelle Modalitätenpaarung bevorzugt werden sollte.

5.2 Gesamtsystemerprobung: Nutzung von Führungsfunktionen

Ziel dieser Studie war es, zu untersuchen, ob ein Nutzer seine Ziele und Absichten mit den zur Verfügung gestellten UAV-Führungsfunktionen realisieren kann. Hierfür sollten Kampfflugzeugpiloten die zur Verfügung gestellten Funktionen in einem realistischen Missionskontext einsetzen, wobei das Nutzungsverhalten auf Muster und Besonderheiten hin analysiert wurde.

Im Folgenden wird die Studie zunächst in zusammengefasster Form vorgestellt. Eine Detailbeschreibung von Studiendesign und Ergebnissen findet sich in den nachfolgenden Unterabschnitten.

Kurzfassung

Studiendesign

Aufgabenstellung: In der Studie führten die Probanden eine simulierte Mission mit zwei Missionszielen durch: der Bekämpfung zweier ballistischer Raketenabschussvorrichtungen und der anschließenden Luftnahunterstützung (Close-Air-Support) von Bodentruppen während einer Bodenoperation. Um diese Ziele zu erreichen, mussten sie verschiedene Missionsaufgaben erfüllen (z. B. Aufklärung und Bekämpfung von Bodenzielen), wofür ihnen drei unbemannte Plattformen und das eigene bemannte Kampfflugzeug zur Verfügung standen. Die Aufgabe der Probanden bestand dabei im Missionsmanagement – also im zielgerichteten Einsatz der verfügbaren Plattformen zur Erreichung der Missionsziele. Im Mittelpunkt stand dabei die Führung der UAVs, wofür den Probanden sämtliche im Rahmen dieser Arbeit entwickelten Führungsfunktionen zur Verfügung standen. Das eigene Kampfflugzeug konnten sie entweder als Kommandoplattform nutzen oder aktiv zur Durchführung von Missionsaufgaben einsetzen.

Datenerhebung und -analyse: Mithilfe eines Video-Debriefing wurde der Einsatz von Funktionen zur UAV-Führung untersucht. Dabei wurde die Mission gemeinsam mit dem Probanden durchlaufen und zu definierten Zeitpunkten wurde der Proband zur Verwendung und Nichtverwendung von Funktionen sowie seiner verfolgten Absicht befragt. Der Versuchsleiter bewertete die Sachmäßigkeit der Nutzung, indem er die analysierten Situationen den Kategorien *Use*, *Misuse*, *Abuse* und *Disuse* zuordnete (vgl. Parasuraman & Riley (1997)).

Studienergebnisse

Angemessenheit der Funktionsnutzung: In 178 dokumentierten Fällen wurden Funktionen korrekt verwendet. Fehlerhafte Nutzung trat in 39 Fällen auf, Nichtverwendung geeigneter Funktionen in 22 Fällen, und Zweckentfremdung in 2 Fällen.

Schwerpunkte des Funktionseinsatzes: Die Probanden führten die UAVs primär durch die Verteilung von Aufträgen, welche unmittelbar für die Erfüllung der Missionsziele notwendig waren oder welche die Erfüllung der Missionsziele unterstützten. Außerdem setzten die Probanden häufig Führungsfunktionen ein, um eine taktisch günstige Positionierung der eigenen Kräfte sicherzustellen.

Falsches mentales Modell begünstigt Probleme: In vielen Fällen trugen Diskrepanzen zwischen Nutzererwartung und Systemverhalten zu fehlerhaftem oder ausbleibendem Funktionseinsatz bei.

Nicht-Bedenken von Funktionen: Ein wichtiger Grund für die Nichtnutzung geeigneter Funktionen war, dass die Probanden diese in bestimmten Situationen nicht bedacht haben. Es bleibt offen, ob ein tiefergehendes Training dies vollständig verhindern könnte oder ob andere Faktoren wie begrenzte Explorierbarkeit oder eine hohe Beanspruchung ebenfalls dazu beitragen, dass Funktionen nicht bedacht werden.

Schlussfolgerungen

Eignung der Führungsfunktionen: Die Probanden nutzten die Funktionen zur UAV-Führung in den meisten Fällen korrekt und wie vorgesehen. Dies deutet darauf hin, dass die bereitgestellten Auftragsstypen, deren Parametrisierung und die Verhaltensrichtlinien geeignete Mittel für die Durchführung einer Manned-Unmanned Teaming Mission darstellen.

Bedeutung der Positionierung: Der häufige Einsatz von Funktionen zur Positionierung von UAVs unterstreicht den Stellenwert der Positionierung der eigenen Kräfte für die Probanden. Daher sollten zukünftige Systementwicklungen besonderes Augenmerk auf die Bereitstellung entsprechender Funktionen legen.

Verbesserungspotenziale: Während des Versuchs traten verschiedene Situationen auf, in denen Funktionen falsch eingesetzt, geeignete Funktionen nicht genutzt, oder in denen eine Funktion zweckentfremdet wurde. Die Erfassung und Kategorisierung dieser Fehlerfälle bietet eine Grundlage für Maßnahmen, die entweder spezifische Fehler beheben oder gemeinsame Ursachen adressieren. Eine zentrale Ursache vieler Fehler ist eine Diskrepanz zwischen Nutzererwartung und Systemverhalten. Diese könnte durch drei Maßnahmen reduziert werden: (1) Optimierung des Systemverhaltens in taktischen Aspekten, (2) stärkere Nutzerkontrolle durch zusätzliche Verhaltensrichtlinien und (3) erhöhte Systemtransparenz.

Weiterer Forschungsbedarf: Zukünftige Studien sollten untersuchen, wie das Nicht-Bedenken geeigneter Funktionen minimiert werden kann und welche Faktoren dabei eine Rolle spielen.

5.2.1 Studiendesign

Dieser Unterabschnitt beschreibt das methodische Vorgehen der Studie.

5.2.1.1 Aufgabenstellung

In der Studie absolvierten die Probanden eine simulierte Mission mit zwei aufeinanderfolgenden Missionszielen (Abbildung 72). Das erste Missionsziel bestand in der verdeckten Bekämpfung zweier ballistischer Raketenabschussvorrichtungen. Das zweite Missionsziel war die Luftnahunterstützung (Close-Air-Support; CAS) von Bodentruppen, die eine Bodenoperation durchführten. Zur Erfüllung dieser Missionsziele mussten die Probanden verschiedenste Missionsaufgaben erfüllen, z. B. die Aufklärung und Bekämpfung von Bodenzielen, die Unterdrückung gegnerischer Flugabwehrstellungen und die Bekämpfung feindlicher Kampfflugzeuge. Hierfür standen den Probanden drei unbemannte Luftfahrzeuge sowie das eigene bemannte Kampfflugzeug zur Verfügung. Eine detailliertere Beschreibung von Missionszielen, Kräfteinsatz und Missionsablauf findet sich im Anhang E.6.1.

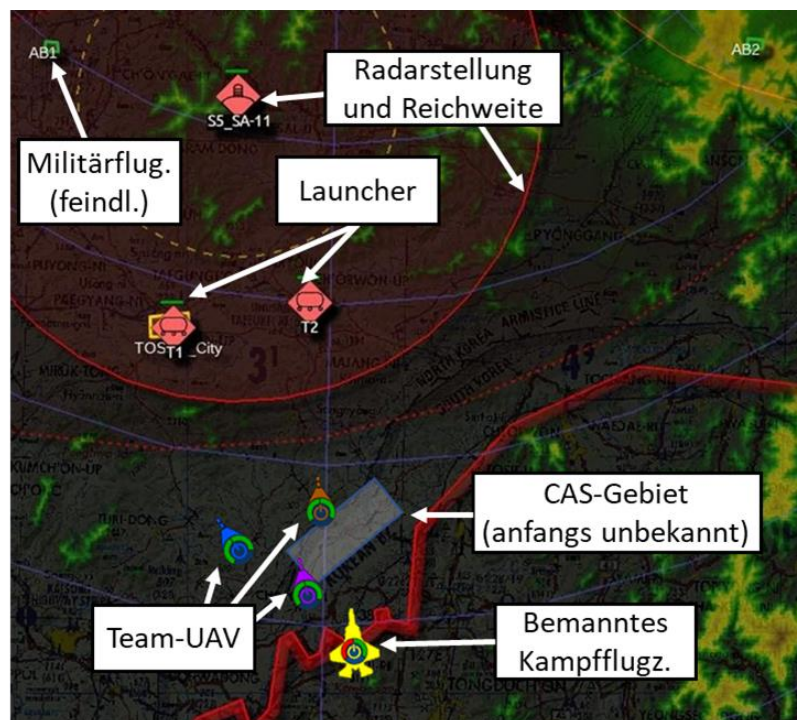


Abbildung 72. Missionsüberblick.

Die Aufgabe der Probanden bestand dabei im Missionsmanagement – also darin, die verfügbaren Plattformen, einschließlich des eigenen Kampfflugzeugs, zielgerichtet zur Erreichung der Missionsziele einzusetzen. Im Mittelpunkt dieser Aufgabe stand die Führung der UAVs, die durch gezielte Auftragsvergabe (z. B. zur Aufklärung oder Zielbekämpfung) in die Mission eingebunden werden konnten. Hierfür standen sämtliche in dieser Arbeit vorgestellten Führungsfunktionen zur Verfügung. Das eigene Kampfflugzeug konnten die Probanden entweder als reine Kommandoplattform oder aktiv zur Durchführung von Missionsaufgaben einsetzen.

5.2.1.2 Datenerhebung und -analyse

Im Anschluss an die Missionsdurchführung wurde der Simulator in einen Debriefing-Modus versetzt, in dem der Zustand der Display-Elemente zu einem beliebigen Zeitpunkt der Mission reproduziert werden konnte. In diesem Debriefing-Modus wurde die gesamte Mission gemeinsam mit dem Probanden durchlaufen. Dabei wurde der Einsatz der UAV-Führungsfunktionen analysiert, indem die Wiedergabe der Mission zu bestimmten Zeitpunkten unterbrochen wurde und der Proband zu seinem Nutzungsverhalten befragt wurde. Dabei wurden die vier Konstrukte *Use*, *Misuse*, *Abuse* und *Disuse* erfasst, um die angemessene Nutzung der Funktionen zu beschreiben. Wie diese Konstrukte im Rahmen des Versuches definiert wurden, ist in Tabelle 23 beschrieben.

Tabelle 23. Befragungszeitpunkte und analysierte Konstrukte (vgl. Parasuraman & Riley (1997)).

Konstrukt	Definition des Konstruktes
Use	Der Proband setzte eine Funktion zur UAV-Führung korrekt ein. Die Befragung ergab, dass sie im Sinne seiner Absicht und entsprechend ihrer vorgesehenen Nutzung verwendet wurde.
Misuse	Der Proband setzte eine Funktion zur UAV-Führung ein. Die Befragung zeigte, dass die Funktion fehlerhaft verwendet wurde.
Abuse	Der Proband setzte eine Funktion zur UAV-Führung ein. Die Befragung ergab, dass sie zweckentfremdet wurde, um andere Ziele zu verfolgen als diejenigen, für die sie ursprünglich konzipiert war.
Disuse	Der Proband verzichtete auf den Einsatz einer Funktion, die in der vorliegenden Situation als angemessen eingeschätzt wurde. Dies kann bedeuten, dass eine potenziell hilfreiche Funktion ungenutzt blieb oder dass eine weniger geeignete verwendet wurde. Die Einschätzung der Eignung basierte auf der subjektiven Bewertung durch den Versuchsleiter.

5.2.1.3 Probanden

Der Versuch wurde mit zehn Kampfflugzeugpiloten der deutschen Luftwaffe durchgeführt. Alle Piloten waren männlich und zwischen 32 und 59 Jahre alt ($M_{Alter} = 38,2$ Jahre). Die Piloten verfügten über 100 bis 1956 Stunden Erfahrung im Führen von Kampfflugzeugen ($M_h = 785$ h; $\sigma = 478$ h). Tabelle 24 gibt einen Überblick über die Teilnehmer des Versuchs.

Tabelle 24. Demografische Daten der Probanden.

Kennziffer ⁶⁴	Alter	Flugstunden Kampfflugzeug	Flugstunden sonstige
1	32	500	260
2	34	100	2300
3	36	690	260
4	36	900	270
5	36	1000	-

⁶⁴ Die vergebenen Kennziffern dienen der konkreten Identifikation einer Person und werden im Rahmen aller Versuchsbeschreibungen konsistent verwendet. Die Kennziffern stehen aus Datenschutzgründen nicht in Verbindung mit den bei Ergebnissen verwendeten Nummern.

6	37	350	1470
7	37	800	250
8	37	1000	500
9	38	550	1550
10	59	1956	1200

5.2.1.4 Training

Der beschriebene Versuch fand im Rahmen einer Experimentalkampagne mit fünf Crews bestehend aus zwei Personen statt. Jeder Proband durchlief im Rahmen der einwöchigen Teilnahme zunächst ein zweitägiges Training und anschließend verschiedene Versuche mit unterschiedlichen Schwerpunkten. Daher hatten die Probanden zu Beginn des Versuches einen leicht verschiedenen Trainingsstand durch zusätzliche Erfahrungen aus anderen Versuchen. Durch die Randbedingungen der Experimentalkampagne ergab sich ferner, dass zwar alle Delegationsmethoden erklärt und geübt wurden, in den meisten Trainingseinheiten wurden jedoch überwiegend die Delegationsmethoden Timeline und Touch-Gesture verwendet, wodurch die Probanden in den Delegationsmethoden Voice und Gaze-HOTAS weniger geübt waren. Aufbauend auf diesem Trainingsstand absolvierten die Probanden eine 30-minütige Trainingsmission, in der alle Aspekte der Experimentalmission enthalten waren.

5.2.1.5 Versuchsaufbau

Die Versuchsdurchführung erfolgte im Kampfflugzeugsimulator, dessen Aufbau und Gestaltung sich im Wesentlichen an der Beschreibung in Kapitel 4 orientierte.

Aufgrund der Rahmenbedingungen der Experimentalkampagne wies der verwendete Versuchsaufbau jedoch zwei Abweichungen von der in dieser Arbeit beschriebenen Implementierung auf. Zum einen kamen andere Symbole zur Darstellung der eigenen unbemannten Kräfte zum Einsatz, die zusätzliche Statusinformationen enthielten. Zum anderen standen den Probanden weitere Führungsfunktionen zur Verfügung, die aus einer parallel durchgeführten Forschungsarbeit innerhalb der Experimentalkampagne resultierten. Der Einfluss dieser beiden Unterschiede auf die Ergebnisse des Versuches wird rückblickend als gering eingeschätzt.

5.2.2 Ergebnisse

Die Funktionen zur Führung von UAVs wurden größtenteils gemäß ihrer ursprünglichen Konzeption verwendet. Es gab jedoch auch Situationen, in denen die Funktionen nicht genutzt, fehlerhaft genutzt wurden oder in seltenen Fällen auch zweckentfremdet wurden. Nachfolgend werden die Ergebnisse zum Funktionseinsatz und die wesentlichen Trends der Nutzung vorgestellt.

5.2.2.1 Angemessene Verwendung von Funktionen

Es gab sehr viele Einzelsituationen, in denen die Probanden eine Funktion so verwendeten, wie sie konzipiert war. Um das Debriefing in einem angemessenen Zeitrahmen durchführen zu können, wurden aufeinanderfolgende Verwendungen von einer Funktionalität zusammengefasst.

In Summe wurden im Debriefing 178 Situationen dokumentiert, in denen Funktionen zur UAV-Führung eingesetzt wurden, wie sie konzipiert waren und in denen der Einsatz ohne Fehler erfolgte. Dabei wurde erwartungsgemäß häufig auf Funktionen zurückgegriffen, welche in direktem Zusammenhang mit der Erledigung der Missionsziele standen oder welche die Erreichung dieser Ziele unterstützten. Interessanterweise nutzten die Probanden jedoch auch sehr häufig Funktionen, um die eigenen Kräfte zu positionieren. Eine Übersicht über die wichtigsten Gründe zur Funktionsnutzung ist in Tabelle 25 zu finden.

Tabelle 25. Beschreibung und Häufigkeiten der Nutzung verschiedener Funktionen.

Anzahl	Beschreibung
73 Fälle	<p>Missionsrelevante Auftragserteilung</p> <p>In diesen Situationen delegierten die Probanden Aufträge, welche direkt für die Erfüllung der Missionsziele notwendig waren. Am häufigsten wurden hierfür Bekämpfungsaufträge erteilt, gefolgt von Aufklärungsaufträgen.</p>
56 Fälle	<p>Positionierung von Teammitgliedern</p> <p>Die Probanden verwendeten häufig Funktionen, um eine geeignete Positionierung der eigenen Kräfte sicherzustellen. Am häufigsten verwendeten die Probanden die Funktion Plattform-Navigation (Touch-Gesture), um UAVs zu einem Punkt zu kommandieren, aber auch die Supportive Idle-Verhaltensweisen wurden häufig eingesetzt.</p> <p>Es gab verschiedene Gründe, warum Probanden Einfluss auf die Positionierung haben wollten. Zum Beispiel reagierten die Probanden auf Bedrohungen, indem sie ihre Kräfte aus der Bedrohung kommandierten. Daneben setzten die Probanden die Positionierung vorausschauend ein, um für zukünftige Situationen optimal vorbereitet zu sein (z. B. um kurze Flugwege zu antizipierter Aufgabe zu erreichen).</p>
49 Fälle	<p>Missionsunterstützende Auftragsdelegation</p> <p>In diesen Fällen delegierten die Probanden Aufträge, welche die Erfüllung der Missionsziele unterstützten. Dabei handelte es sich überwiegend um CAP-Aufträge zur Bekämpfung feindlicher Luftfahrzeuge sowie um DEAD Aufträge zur Bekämpfung von Flugabwehrstellungen.</p>

Zur Auftragsdelegation verwendeten die Probanden überwiegend die Methoden Timeline und Touch-Gesture. Die anderen Delegationsmethoden wurden nur vereinzelt verwendet.

5.2.2.2 Fehlerhafte Verwendung von Funktionen

Im Debriefing wurden 39 Situationen dokumentiert, in denen der Einsatz einer Funktion nicht geeignet war, um die Absicht der Probanden zu verfolgen. Viele dieser Fehlerfälle wiesen Ähnlichkeiten zueinander auf. Eine Übersicht über die häufigsten Fehlerfälle ist in Tabelle 26 zu finden.

Tabelle 26. Beschreibung und Häufigkeiten von Fehlerfällen beim Einsatz von Führungsfunktionen.

Anzahl	Beschreibung
16 Fälle	<p>Ungeeignete Standardparameter</p> <p>Die Probanden übernahmen eine ungeeignete initiale Auftragsparametrisierung. In allen Fällen wurde zum Problem, dass die Standardhöhe für Aufklärungsaufträge oberhalb der Wolkenbasis lag und die UAVs somit keine geeigneten Bilder schießen konnten. In einzelnen Fällen bemerkten die Probanden den Fehler frühzeitig, in anderen Fällen verzögerte sich die Aufklärung dadurch merklich.</p>
9 Fälle	<p>Suboptimale Auftragsverteilung</p> <p>Der Scheduler wählte bei der Auftragsdelegation eine andere Plattform zur Durchführung des Auftrags aus, als vom Probanden für diese Situation vorgesehen. In fünf dieser Fälle wählte der Scheduler das Kampfflugzeug der Probanden aus und die Probanden wollten den Auftrag nicht selbst durchführen. In anderen Fällen wollten die Probanden aus taktischen Gründen ein anderes UAV als das vom Scheduler ausgewählte einsetzen.</p>
7 Fälle	<p>Bedienfehler</p> <p>Die Probanden bedienten das System versehentlich falsch. Beispielsweise führten sie unabsichtlich Touchscreen-Gesten aus, die zur Erteilung von Aufträgen führten oder sie drückten einen falschen Knopf.</p>
5 Fälle	<p>Falsches Verständnis von Führungsfunktionen</p> <p>In diesen Fällen missinterpretierten die Probanden den Einsatzbereich der angebotenen Führungsfunktionen. So wurde beispielsweise ein Auftrag delegiert, obwohl die Absicht bereits durch einen anderen Auftrag abgedeckt war oder es wurde eine Verhaltensrichtlinie bei einem UAV festgelegt, die in der vorliegenden Situation keinen Effekt hatte.</p>
2 Fälle	<p>Verzögerte Planaktivierung</p> <p>In diesen beiden Fällen veränderten die Probanden den Missionsplan durch Delegation bzw. Löschen von Aufträgen. Diese Änderung hatte zunächst keinen Effekt, weil der Plan erst viel später von den Probanden aktiv geschaltet wurde.</p>

5.2.2.3 Unzweckmäßiger Gebrauch von Funktionen

Die Probanden setzten die Funktionen nur selten anders ein als ursprünglich vorgesehen. Zwei Situationen konnten identifiziert werden, die zwar nicht eindeutig als Zweckentfremdung zu werten sind, aber tendenziell in diese Kategorie fallen:

- Verwendung der Plattform-Navigation (Touch-Gesture) um eine manuelle Aufklärung durchzuführen, da das System in der vorliegenden Situation die Erteilung von Aufklärungsaufträgen nicht ermöglichte.
- Umpositionierung eines UAVs mittels Plattform-Navigation, um die Objektdichte auf dem Display an einer Stelle zu verringern.

5.2.2.4 **Ausbleiben der Funktionsnutzung**

Im Debriefing wurden 22 Fälle dokumentiert, in denen Probanden eine Funktion nicht verwendeten, obwohl die Funktion aus Sicht des Versuchsleiters zur Verfolgung der Absichten geeignet gewesen wäre. Die Nichtnutzung kann sowohl einzelne Situationen als auch längere Zeiträume betreffen, wenn z. B. Probanden über einen längeren Zeitraum hinweg eine bestimmte Funktion nicht verwendet haben. Tabelle 27 stellt die Fälle der Nichtnutzung dar.

Tabelle 27. Häufigkeit und Beschreibung von Fällen der Nichtnutzung.

Anzahl	Beschreibung
10 Fälle	Funktion nicht bedacht In zehn Fällen haben Probanden eine Funktion nicht verwendet, weil sie nicht daran gedacht haben, dass die Funktion für die vorliegende Situation geeignet sein könnte. Dies betraf vor allem die Auftragserteilung mit den Delegationsmethoden Voice und Gaze-HOTAS, jedoch gab es auch Fälle, in denen Probanden eine Funktion der Methode Touch-Gesture nicht bedacht hatten.
4 Fälle	Auftragsbearbeitung dauert zu lange In diesen Fällen verzichteten die Probanden auf die Verteilung von Aufklärungsaufträgen, da sie erwarteten, dass sie durch die manuelle Bedienung der Sensoren schneller Ergebnisse erzielen würden als durch den Einsatz eines UAVs, welches zunächst zum Ziel fliegen müsste, um Bilder gemäß den Auftragsanforderungen aufzunehmen.
4 Fälle	Fehlende Vertrautheit mit Funktion Die Probanden berichteten in diesen Fällen davon, dass sie eine Funktion nicht gut genug verinnerlicht hatten, als dass sie sich sicher gewesen wären, dass die Funktion hilfreich ist.
2 Fälle	Unzureichende Kontrolle In diesen beiden Fällen verwendeten die Probanden eine Funktion zur Auftragserteilung bei der Delegationsmethode Touch-Gesture nicht, weil sie die Kontrolle über die Verteilung des Auftrags nicht an das System abgeben wollten.
2 Fälle	Funktion ungeeignet In diesen beiden Fällen haben die Probanden bestimmte Delegationsmethoden nicht verwendet, weil sie sie für nicht geeignet hielten. In einem Fall verwendete ein Proband die Delegationsmethode Voice nicht, weil bestimmte Aufträge nicht damit erteilt werden konnten. Im anderen Fall verwendete ein Proband Gaze-HOTAS nicht, weil Display Clutter vorlag.

5.2.3 Diskussion

In diesem Unterabschnitt werden die vorgestellten Einzelergebnisse interpretiert und zueinander in Bezug gesetzt.

5.2.3.1 Hohe Bedeutung der Positionierung

Die Funktionen zur Positionierung von UAVs hatten im durchgeführten Versuch einen hohen Stellenwert für die Probanden. Diese Möglichkeit wurde von den Probanden aus verschiedenen Gründen eingesetzt:

- Reaktion auf taktische Lageänderungen
- Vorpositionierung von Teammitgliedern zur Verbesserung der künftigen Handlungsfähigkeit
- Kompensation von unzulänglichem UAV-Verhalten

Auch wenn durch eine Optimierung des UAV-Verhaltens die Eingriffe zur Kompensation von unzulänglichem UAV-Verhalten minimiert werden könnten, wird auch weiterhin der Bedarf bestehen, dass Piloten die Positionierung von Kräften mitbestimmen, um schnell auf taktische Lageänderungen reagieren zu können oder um künftige Situationen vorbereiten zu können. Daraus wird geschlossen, dass sowohl die Plattform-Navigation (Touch-Gesture) als auch die Definition von Idle-Verhaltensweisen wichtige Bestandteile des Führungs- und Interaktionskonzeptes darstellen.

5.2.3.2 Diskrepanz zwischen Erwartung und Systemverhalten

Im Versuch traten verschiedene Situationen auf, in denen das Systemverhalten von den Erwartungen der Probanden abwich. Hierzu zählen:

- Die Auswahl von unpassenden (Standard-)Parametern für Aufklärungsaufträge durch das System.
- Die automatisierte Vergabe eines Auftrags an eine andere Plattform als die vom Nutzer implizit gewünschte.
- Das Unwissen von Probanden, dass eine Funktion nicht nötig ist, weil der gewünschte Effekt durch eine andere Funktion bereits abgedeckt ist.

Diese Abweichungen zwischen mentalem Modell des Nutzers und der tatsächlich zugrunde liegenden Funktion führte zu verschiedenen Effekten. In einigen Fällen sank die Missionsperformance, weil die Korrektur eines Fehlers mit Aufwand verbunden war. In anderen Fällen maximierten die Probanden ihre Kontrolle über das System, indem sie Funktionen wählten, die die Freiheitsgrade der Automation einschränkten.

Um eine Verschlechterung der Missionsleistung zu verhindern, sollten Maßnahmen ergriffen werden, die die Diskrepanz zwischen Nutzererwartung und Systemverhalten minimieren.

Eine Maßnahme wäre es, das taktische Verhalten des Systems zu verbessern. Wenn ein Proband beispielsweise einen Bekämpfungsauftrag gegen eine Flugabwehrstellung erteilte und sich diese in Reichweite und Flugrichtung eines eigenen UAVs befand, so erwartete der

Nutzer, dass das entsprechende UAV auch den Auftrag erhält, weil es taktisch sinnvoll ist. Diese Erwartung wurde vom System nicht immer erfüllt, da der Scheduler primär auf einen kurzen Missionsplan hin optimierte und zusätzlich die Flugrichtung der einzelnen Plattformen nicht berücksichtigte. Durch die Verbesserung des taktischen Verhaltens eines Systems könnte die Diskrepanz zwischen Nutzererwartung und Systemverhalten optimiert werden und auch über Fehlerfälle hinaus eine Verbesserung der Missionsleistung erreicht werden. Dabei ist jedoch zu bedenken, dass ein Systemverhalten, das auf bestimmte Kriterien hin optimiert ist, nicht zwangsläufig als taktisch sinnvoll erachtet wird. So kann beispielsweise ein Nutzer eines Systems ein Verhalten für ungeeignet bewerten oder ein anderes Verhalten erwarten, obwohl es dem optimalen Verhalten entspricht.

Ein anderer Ansatz, der dieses Problem vermeidet, wäre eine Erweiterung der durch den Nutzer spezifizierbaren Verhaltensrichtlinien. Dadurch könnte der Nutzer eine tiefere Kontrolle über bestimmte Systemfunktionen ausüben, ohne dass er dies immer explizit machen muss. Beispielsweise könnte das Problem der falsch gewählten Standardhöhen vermieden werden, indem der Nutzer für jedes UAV eine individuelle Standardhöhe zur Auftragsdurchführung zuweisen kann. Die Erweiterung der Verhaltensrichtlinien sollte außerdem ergänzt werden durch eine systemische Analyse, die die Anwendbarkeit der Verhaltensrichtlinien für jeden Auftrag überprüft und den Nutzer darauf hinweist, wenn davon abweichende Einstellungen besser geeignet wären. Beispielsweise könnte bei erteilten Aufträgen überprüft werden, ob die in der Verhaltensrichtlinie spezifizierte Standardhöhe für den Auftrag geeignet ist.

Anstatt dem Nutzer mehr Kontrolle und damit Verantwortung zu übertragen, könnten alternativ Maßnahmen zur Verbesserung der Transparenz des Systemverhaltens erwogen werden. Beispielsweise könnte das System bei der Erteilung von Aufträgen an UAVs dem Nutzer eine Vorgehensbeschreibung anzeigen, sodass der Nutzer die Effekte der Auftragsdelegation besser einschätzen kann.

5.2.3.3 Nicht-Bedenken von Funktionen

Im Versuch zeigten sich verschiedene Situationen, in denen Probanden eine Funktion nicht verwendeten, weil sie die Funktion nicht bedacht hatten. Sehr ausgeprägt war dieser Effekt bei den Delegationsmethoden. Hier verwendeten die Probanden überwiegend die Delegationsmethode Timeline und in geringerem Maße auch die Methode Touch-Gesture, während die Methoden Voice und Gaze-HOTAS kaum Beachtung fanden. Dies kann wahrscheinlich auf verschiedene Effekte zurückgeführt werden. Ein Grund ist, dass die Delegationsmethoden Voice und Gaze-HOTAS im Training eine geringere Aufmerksamkeit bekommen haben und die Probanden sich daher weniger an diese beiden Methoden zurückerinnerten und auch weniger geübt in deren Anwendung waren. Allerdings zeigten sich auch Situationen, in denen die Probanden die Methode Timeline verwendeten, obwohl die Methode Touch-Gesture geeigneter gewesen wäre, was in diesem Fall nicht auf das Training zurückzuführen sein kann. Für diese Situationen gibt es zwei mögliche Erklärungsansätze. Ein Ansatz ist, dass die Probanden die explorative Methode Timeline bevorzugten, weil man bei dieser Methode durch die Reaktion des Displays durch die einzelnen Interaktionen geleitet wird und nahezu jede Interaktion rückgängig gemacht werden kann. Ein anderer

Erklärungsansatz ist, dass es einen Effekt gibt, der die Probanden bei hoher Beanspruchung daran hindert, die Delegationsmethode zu wechseln, z. B. weil keine Kapazitäten zur Verfügung stehen, um über weitere Lösungsoptionen nachzudenken.

Um aus den beobachteten Effekten und Erklärungsansätzen weitere Schlüsse ableiten zu können, sollten weitere Studien aufgesetzt werden um zu ermitteln, welchen Anteil die Einflussfaktoren Training, Explorierbarkeit der Methoden sowie Beanspruchung auf die Verwendung von Funktionen haben.

5.2.4 Fazit

Mithilfe des Versuchs konnten wichtige Erkenntnisse zum Einsatz des Führungs- und Interaktionskonzept gewonnen werden. In der Regel wendeten die Probanden die Funktionen zur UAV-Führung fehlerfrei und wie konzipiert an, was ein Indiz dafür ist, dass den Probanden mit den bereitgestellten Auftragsstypen, deren Parametrisierungen sowie den Verhaltensrichtlinien die richtigen Mittel zur Missionsdurchführung zur Verfügung standen.

Auch wenn die meisten Funktionsverwendungen einen direkten Beitrag zur Verfolgung der Missionsziele leisteten, setzten die Probanden auch häufig Funktionen ein, um eine geeignete Positionierung ihrer Kräfte sicherzustellen – etwa zur Reaktion auf Lageänderungen oder zur Vorbereitung auf zukünftige Situationen. In einem künftigen Führungs- und Interaktionskonzept sollten solche Funktionen daher vorgesehen werden.

Es wurden verschiedene Fälle identifiziert, in denen der Einsatz von Funktionen nicht angemessen war, weil der Einsatz mit Fehlern verbunden war, weil geeignete Funktionen nicht eingesetzt wurden oder, in seltenen Fällen, weil Funktionen zweckentfremdet wurden. Diese Fälle wurden im Rahmen der Auswertung kategorisiert und es wurden Gemeinsamkeiten zwischen den Fällen identifiziert. Diese Kategorisierung ermöglicht die Entwicklung von Maßnahmen, die entweder spezifische Fehlerfälle adressieren oder gemeinsame Ursachen angehen, die verschiedenen Fehlerfällen zugrunde liegen. Beispielsweise könnte der Fehlerfall der suboptimalen Auftragsverteilung durch den Scheduler behoben werden, indem die Optimierungsfunktion des Schedulers überarbeitet wird, um taktische Randbedingungen besser zu berücksichtigen. Eine Gemeinsamkeit zwischen verschiedenen Fehlerfällen war eine Diskrepanz zwischen Nutzererwartung und Systemverhalten. Um diese Diskrepanz zu verringern kommen verschiedene Maßnahmen wie Optimierung des Systemverhaltens, Erweiterung der Nutzerkontrolle durch Verhaltensrichtlinien und Verbesserung der Transparenz in Frage.

Es wurden verschiedene Fälle dokumentiert, in denen die Probanden die Verwendung von Führungsfunktionen unterließen, obwohl diese geeignet gewesen wären (dies betraf z. B. die Auftragserteilung mittels Voice oder Gaze-HOTAS). Der Hauptgrund dafür war, dass die Probanden nicht darüber nachgedacht haben, dass die jeweilige Funktion geeignet sein könnte. Es ist unklar, ob alle diese Fälle durch ein umfassenderes Training hätten vermieden werden können, ob es eine Rolle spielt, wie explorativ eine Funktion ist, oder ob es noch einen weiteren Effekt gibt, der die Probanden bei hoher Beanspruchung davon abhält, über weitere Funktionen zur Erreichung der eigenen Ziele nachzudenken. Weitere Untersuchungen sollten durchgeführt werden, um die Ursachen für das Nicht-Bedenken von Funktionen zu ermitteln.

5.3 Gesamtsystemerprobung: Bewertung des Systems

Ziel dieser Studie war die (subjektive) Bewertung des entwickelten Führungs- und Interaktionskonzeptes durch Kampfflugzeugpiloten der Luftwaffe. Zu diesem Zweck sollten die Piloten das entwickelte System in einem realistischen Missions- und Aufgabenkontext anwenden, wobei die Stärken und Schwächen der einzelnen Elemente mit Hilfe von Fragebögen erfasst wurden.

Im Folgenden wird die Studie zunächst in zusammengefasster Form vorgestellt. Eine Detailbeschreibung von Studiendesign und Ergebnissen findet sich in den nachfolgenden Unterabschnitten.

Kurzfassung

Studiendesign

Aufgabenstellung: Die Probanden führten eine Mission zur Bekämpfung von zwei ballistischen Raketenabschussvorrichtungen durch. Zur Erreichung dieses Ziels mussten sie verschiedene Missionsaufgaben erfüllen, darunter Aufklärung und Bekämpfung von Bodenzielen, Unterdrückung gegnerischer Flugabwehr und Luftkampf. Hierfür standen ihnen drei UAVs und das eigene bemannte Kampfflugzeug zur Verfügung. Analog zur Studie „Nutzung von Führungsfunktionen“ bestand die Aufgabe der Probanden im Missionsmanagement, mit Schwerpunkt auf der UAV-Führung. Zur Führung der UAVs konnten die Probanden sämtliche im Rahmen dieser Arbeit entwickelten Führungsfunktionen einsetzen. Das bemannte Kampfflugzeug konnten sie entweder als Kommandoplattform nutzen oder aktiv für Missionsaufgaben einsetzen.

Datenerhebung und -analyse: Im Anschluss an die Mission füllten die Probanden mehrere Fragebögen aus, um verschiedene Aspekte des Führungskonzeptes zu bewerten. Im Fokus standen dabei: (1) die Funktionalitäten zur UAV-Führung, (2) die eingesetzten Feedback-Mechanismen sowie (3) das Vertrauen in die UAV-Agenten. Ergänzend dazu umfassten die Fragebögen auch Items zur Modellierungsgüte und zum UAV-Verhalten, da sich diese Aspekte potenziell auf die Wahrnehmung und Bewertung des Führungskonzeptes auswirken können.

Studienergebnisse

Auftragstypen und Parametrisierungen: Die Auftragstypen und deren Parameter wurden von den Probanden positiv bewertet, allerdings könnten weitere Auftragstypen und Parametrisierungen sinnvoll sein.

Delegationsmethoden: Die Interaktionsmethoden zur Auftragsdelegation wurden von den Probanden eher positiv bewertet. Die Probanden schlugen verschiedene Maßnahmen zur Verbesserung der individuellen Methoden sowie deren Interoperabilität vor.

Instantane Auftragserteilung: Die Probanden erachteten es als sehr wichtig, dass Aufträge instantan (d.h. ohne Review) delegiert werden können und Touch-Gesture wurde als geeignete Methode dafür angesehen.

Vorgabe von Verhaltensrichtlinien: Die Zuweisung von Idle-Verhaltensweisen wurde positiv bewertet und es wurden weitere Verhaltensweisen vorgeschlagen.

Handlungsfeedback: Das Handlungsfeedback wurde von den Probanden positiv bewertet.

Auditive + Visuelle Meldungen: Auditive Meldungen wurden positiv bewertet, während visuelle Meldungen in der taktischen Karte lediglich mittelmäßiges Feedback erhielten.

Nutzerdialoge: Die Nutzerdialoge wurden unterschiedlich aufgenommen. Inhalte und Auswahloptionen einiger Dialogmeldungen wurden kritisiert, verbunden mit Vorschlägen zur Verbesserung.

Auftragskritik: Die Probanden wünschten sich eine Kritikfähigkeit in der Auftragsdelegation, wenn das Missionsziel gefährdet ist. Ansonsten sollte die Kritikfähigkeit einstellbar sein.

Modellierungsgüte: Die Modellierung wurde insgesamt positiv bewertet.

UAV-Verhalten: Das Verhalten der UAVs wurde mittelmäßig bewertet, wobei im Speziellen das taktische Verständnis kritisiert wurde.

Vertrauen in UAV-Agenten: Die Probanden waren eher dazu geneigt, den UAVs zu vertrauen.

Schlussfolgerung

Eignung des Gesamtkonzeptes: Die überwiegend positive Bewertung des Systems deutet darauf hin, dass das Führungs- und Interaktionskonzept grundsätzlich geeignet ist, um mehrere UAVs während einer Manned-Unmanned Teaming Mission mit verschiedensten Missionsaufgaben erfolgreich zu führen.

Eignung von Teilaspekten: Die Studienergebnisse zeigen, welche Elemente des Führungs- und Interaktionskonzepts in einem zukünftigen System berücksichtigt werden sollten und welche Elemente nicht geeignet sind oder überarbeitet werden müssen.

5.3.1 Studiendesign

Dieser Unterabschnitt beschreibt das methodische Vorgehen der Studie.

5.3.1.1 Aufgabenstellung

In der Studie führten die Probanden eine simulierte Mission zur Bekämpfung von zwei ballistischen Abschussvorrichtungen durch (Abbildung 73). Um dieses Ziel zu erreichen, mussten die Probanden verschiedene Missionsaufgaben erfüllen, darunter Aufklärung & Bekämpfung von Bodenzielen, Unterdrückung von Flugabwehrstellungen und Luftkampf. Hierfür standen ihnen drei unbemannte Luftfahrzeuge sowie das eigene bemannte Kampfflugzeug zur Verfügung. Weitere Details zur Mission finden sich im Anhang [E.6.2](#).

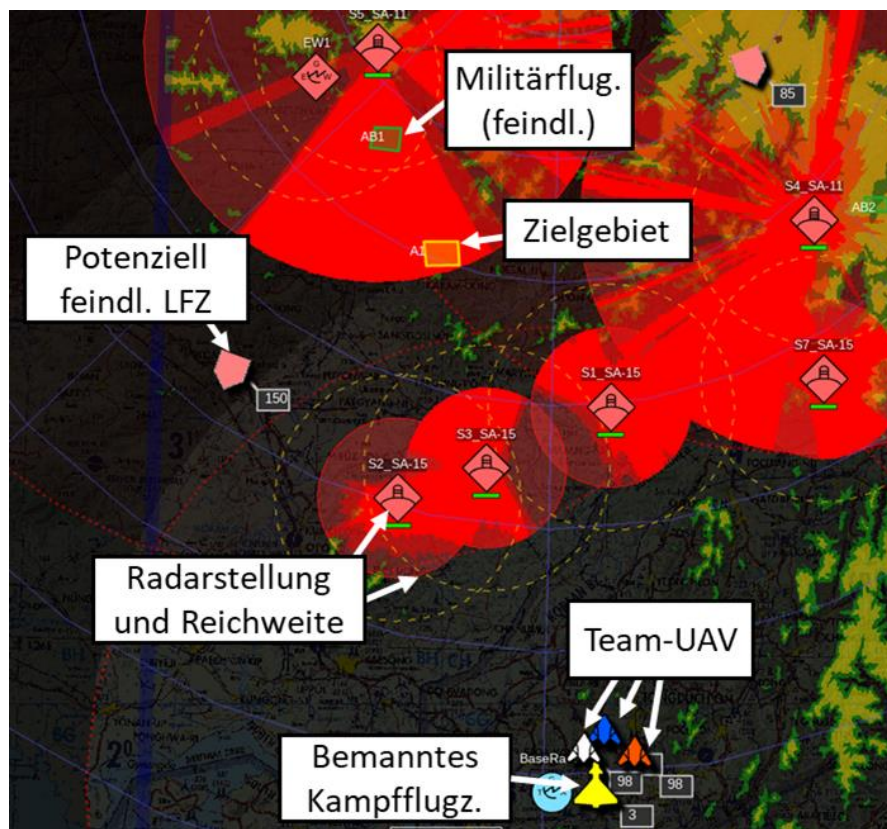


Abbildung 73. Missionsüberblick.

Analog zur Studie „Nutzung von Führungsfunktionen“ (siehe [5.2.1.1](#)) bestand die Aufgabe der Probanden im Missionsmanagement, also im gezielten Einsatz der Teamplattformen.

5.3.1.2 Datenerhebung und -analyse

Im Fokus des Versuchs stand die subjektive Einschätzung der Probanden zum Führungs- und Interaktionskonzept. Zu diesem Zweck füllten die Probanden im Anschluss an den Versuch zwei verschiedene Fragebögen aus.

Der erste Fragebogen erfasste Bewertungen verschiedener Elemente des Führungskonzepts sowie Einschätzungen zu deren Stärken, Schwächen und möglichen Verbesserungsvorschlägen. Zusätzlich wurden Einschätzungen zu alternativen

Gestaltungsoptionen eingeholt. Der Fragebogen war anwendungsspezifisch und enthielt nicht standardisierte Items. Im Fokus des Fragebogens standen zwei Themenbereiche:

- Funktionen zur UAV-Führung (z. B. Interaktionsmethoden der Auftragserteilung)
- Feedback (z. B. Audiomeldungen)

Ergänzend dazu wurden in diesem Fragebogen zwei weitere Aspekte erhoben, die sich zwar nicht unmittelbar auf das Führungskonzept beziehen, jedoch dessen Wahrnehmung und Bewertung beeinflussen können:

- UAV-Verhalten (z. B. Sinnhaftigkeit und Nachvollziehbarkeit)
- Modellierung der Simulationsumgebung (z. B. Realismus der Flugdynamik)

Mit dem zweiten Fragebogen wurde das Vertrauen in die unbemannten Systeme erfasst. Hierfür wurde der standardisierte Fragebogen „Trust in Automation“ (Körber, 2019) eingesetzt.

5.3.1.3 Probanden

Der Versuch wurde mit sechs Kampfflugzeugpiloten der deutschen Luftwaffe durchgeführt. Alle Piloten waren männlich und zwischen 32 und 48 Jahre alt ($M_{Alter} = 41,6$ Jahre). Die Piloten verfügten über 102 bis 2100 Stunden Erfahrung im Führen von Kampfflugzeugen ($M_h = 1281$ h; $\sigma = 808$ h). Tabelle 28 gibt einen Überblick über die Teilnehmer des Versuchs.

Tabelle 28. Demografische Daten der Probanden.

Kennziffer ⁶⁵	Alter	Flugstunden Kampfflugzeug	Flugstunden sonstige
11	32	102	280
12	37	800	260
13	38	600	1400
14	45	2000	600
15	47	2085	180
16	48	2100	50

5.3.1.4 Training

Der Versuch wurde als letzter Teil einer Versuchsreihe durchgeführt, die auch die Studien „Delegationsmethoden“ und „Kritikfähigkeit in der Auftragsdelegation“ umfasste. Die Probanden waren daher bereits mit dem Simulator vertraut. Zur weiteren Vorbereitung absolvierten sie zudem eine Trainingsmission, die alle relevanten Missionsaufgaben umfasste.

5.3.1.5 Versuchsaufbau

Der Versuch wurde im Kampfflugzeugsimulator durchgeführt. Die Konfiguration des Simulators entspricht der Beschreibung in Kapitel 4.

⁶⁵ Die vergebenen Kennziffern dienen der konkreten Identifikation einer Person und werden im Rahmen aller Versuchsbeschreibungen konsistent verwendet. Die Kennziffern stehen aus Datenschutzgründen nicht in Verbindung mit den bei Ergebnissen verwendeten Nummern.

5.3.2 Ergebnisse

In diesem Unterabschnitt werden die Ergebnisse der Befragungen beschrieben.

5.3.2.1 UAV-Führungsfunktionen

Die UAV-Führungsfunktionen wurden von den Probanden insgesamt positiv bewertet (Abbildung 74).

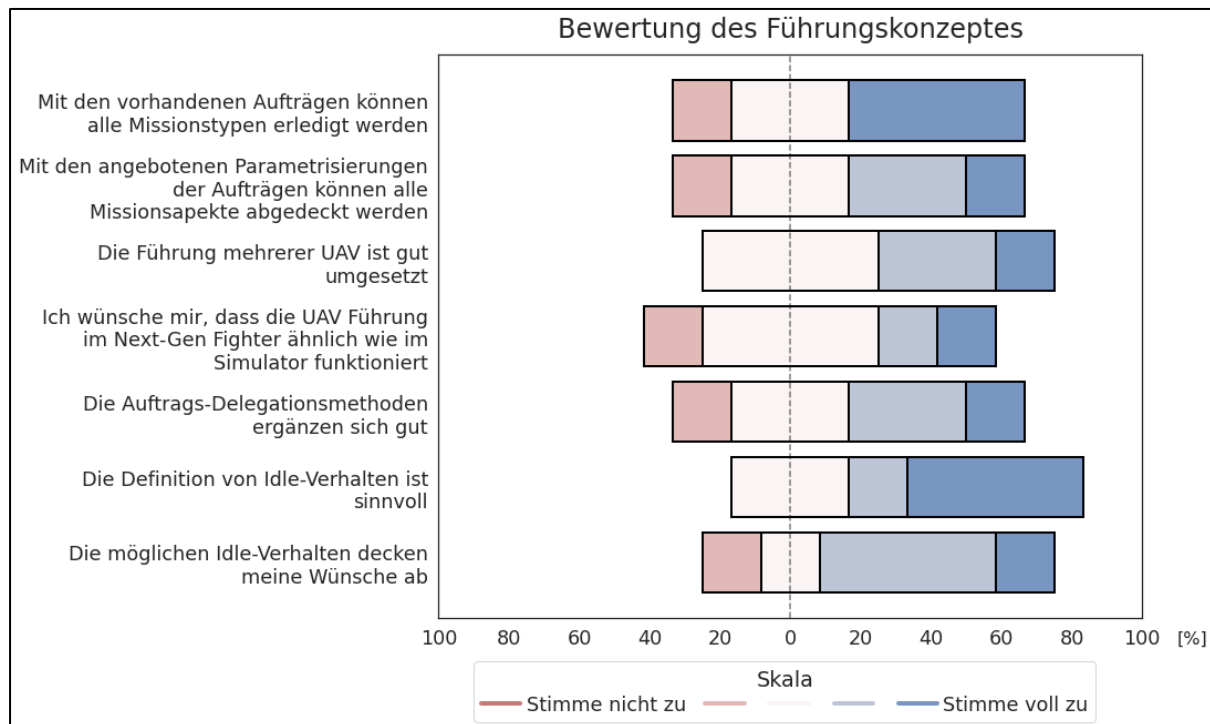


Abbildung 74. Rückmeldungen zu Fragen zu UAV-Führungsfunktionen.

Aufträge und Parametrisierungsoptionen

Die Ergebnisse deuten darauf hin, dass die vorhandenen Aufträge und ihre Parametrisierungen grundsätzlich gut geeignet sind, um die anfallenden Missionsaufgaben zu bearbeiten. Um Nutzern mehr Flexibilität im Umgang mit unterschiedlichen Missionsanforderungen und Randbedingungen zu ermöglichen, könnte es für ein zukünftiges System jedoch sinnvoll sein, zusätzliche Auftragsstypen einzuführen oder bestehende Aufträge um weitere Parameter zu ergänzen. So äußerte ein Proband in einem Kommentar den Wunsch, beim Auftrag „Escort Jamming“ (siehe Anhang [E.2](#)) die Ausrichtung des Störsignals flexibel definieren zu können – etwa mit Blick auf bereits bekannte oder erwartete Radarstellungen.

Delegationsmethoden

Die Delegationsmethoden zur Auftragserteilung wurden eher positiv bewertet. Daraus wird geschlossen, dass die Probanden die Methoden grundsätzlich für geeignet erachten, aber noch Verbesserungspotenziale sehen. In Kommentaren erwähnten die Probanden hierzu:

- Die Interaktionen der Methode Timeline sollten verbessert werden, z. B. durch Drag-and Drop.
- Das Feedback der erkannten Kommandos bei Nutzung der Delegationsmethode Voice soll im MHDD zu sehen sein, außerdem wäre eine visuelle Hervorhebung von Auftragsziel und von ausführendem UAV während der Delegation sinnvoll.
- Beim Gaze-HOTAS soll durch das Angebot eines Zoom-Windows eine einfachere Selektion von Objekten ermöglicht werden.
- Die Delegationsmethoden sollten eine Interoperabilität unterstützen. Dabei sollten die Zustände zwischen den Methoden konsistent sein.

Instantane Auftragsdelegation

Die Probanden erachten es als sehr wichtig, dass eine Delegationsmethode zur instantanen Auftragsdelegation (ohne Review) zur Verfügung steht (Abbildung 75). Diese Ergebnisse unterstreichen die Bedeutung einer kurzen Interaktionsdauer für die Probanden. Im Versuch wurde zur instantanen Auftragsdelegation die Delegationsmethode Touch-Gesture verwendet. Diese wurde von den Probanden als geeignet angesehen.

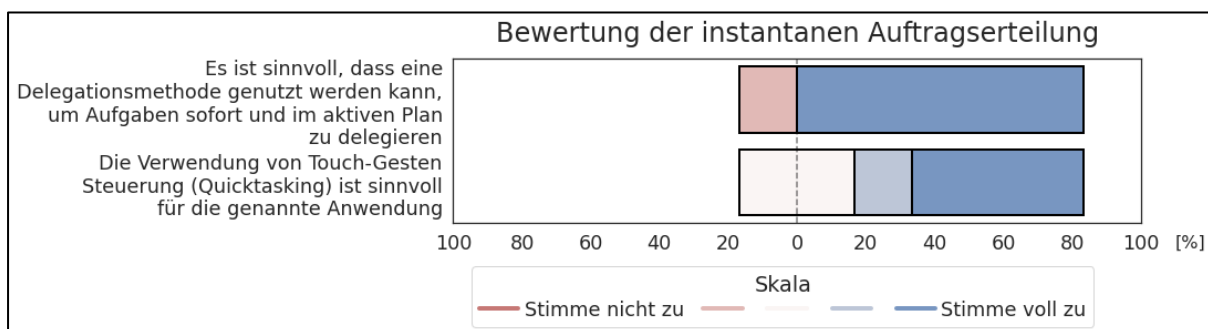


Abbildung 75. Rückmeldung zur Bereitstellung einer Delegationsmethode zur instantanen Auftragsdelegation.

Die instantane Auftragsdelegation wird für verschiedene Anwendungen als vorteilhaft angesehen, insbesondere für die Positionierung von UAVs und zur Bekämpfung von spontan auftauchenden Flugabwehrstellungen (*Popup-SAMs*; Abbildung 76).

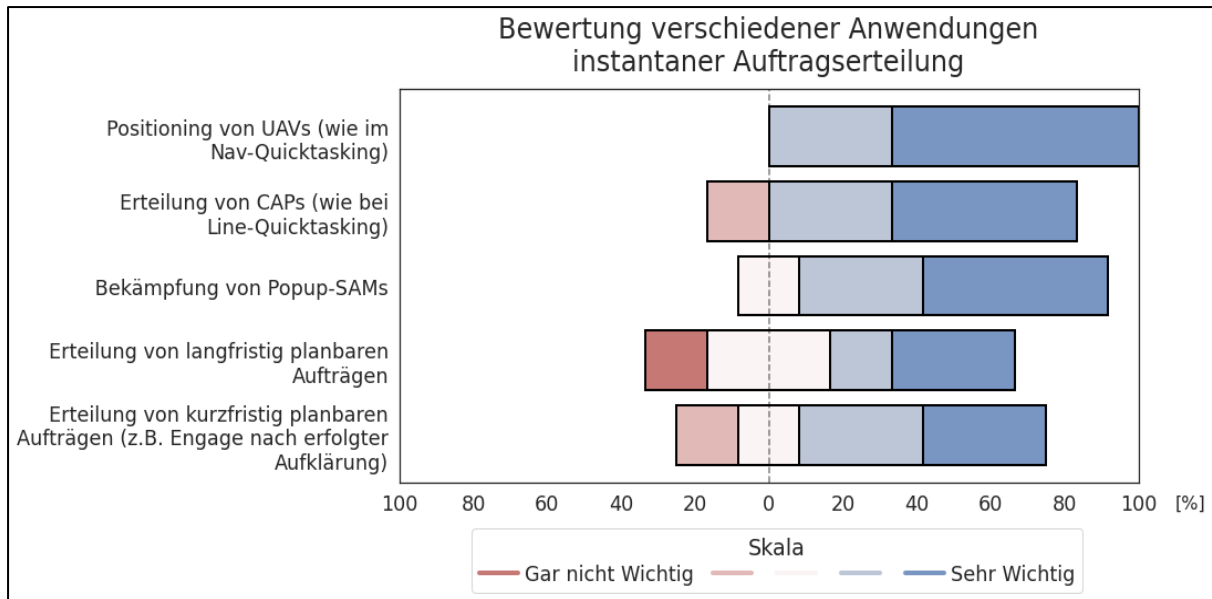


Abbildung 76. Rückmeldungen zu Anwendungsfällen von instantaner Auftragserteilung.

Von vier Probanden wurde ein „Abort“ als weiterer Anwendungsfall für eine instantane Auftragserteilung genannt. Darunter verstehen die Probanden die Möglichkeit, ein UAV aus einer (meist neuen) Bedrohung heraus zu lenken. Grundsätzlich könnte auf solche Situationen auch mit der Plattform-Navigation (Touch-Gesture) reagiert werden, indem ein UAV zu einem Punkt außerhalb der Bedrohung geschickt wird. Eine Funktion zum Abort wäre allerdings expliziter als die Erteilung eines Navigationsauftrags, sodass das UAV beispielsweise mit aggressiveren Flugmanövern darauf reagieren könnte oder anderweitige Maßnahmen ergreifen könnte, die ein Navigationsauftrag nicht rechtfertigen würde.

Vorgabe von Verhaltensrichtlinien

Die Möglichkeit zur Zuweisung von Idle-Verhalten wurde positiv bewertet. Die hierfür zur Verfügung gestellten Verhaltensweisen erhielten überwiegend positives Feedback, allerdings könnten weitere Verhaltensweisen sinnvoll sein. Ein Proband erwähnte in einem Kommentar, dass er die Risikofreudigkeit („Acceptable level of risk“) der UAVs individuell konfigurieren möchte.

5.3.2.2 Feedback

Handlungsfeedback

Das Handlungsfeedback (Rückmeldung der aktuellen Aktion eines UAVs) erhielt von den Probanden positive Bewertungen (Abbildung 77). Sowohl die Effekte des Feedbacks als auch dessen Gestaltung wurden positiv bewertet.

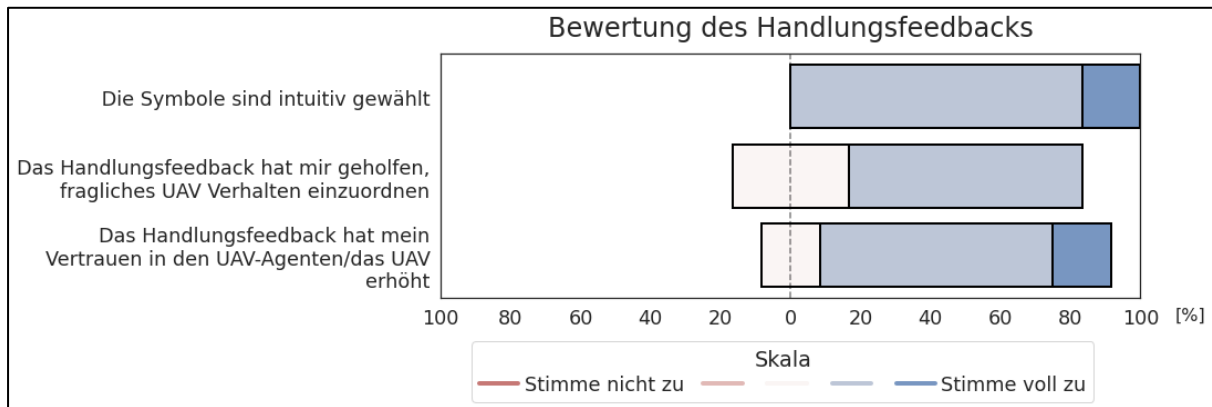


Abbildung 77. Rückmeldungen zum Handlungsfeedback.

Die Probanden hielten die Ebene des Handlungsfeedbacks für sinnvoll (Abbildung 78 A) und sie wünschen sich, dass die Handlungen von UAVs dauerhaft angezeigt werden (wie im Versuch; Abbildung 78 B).

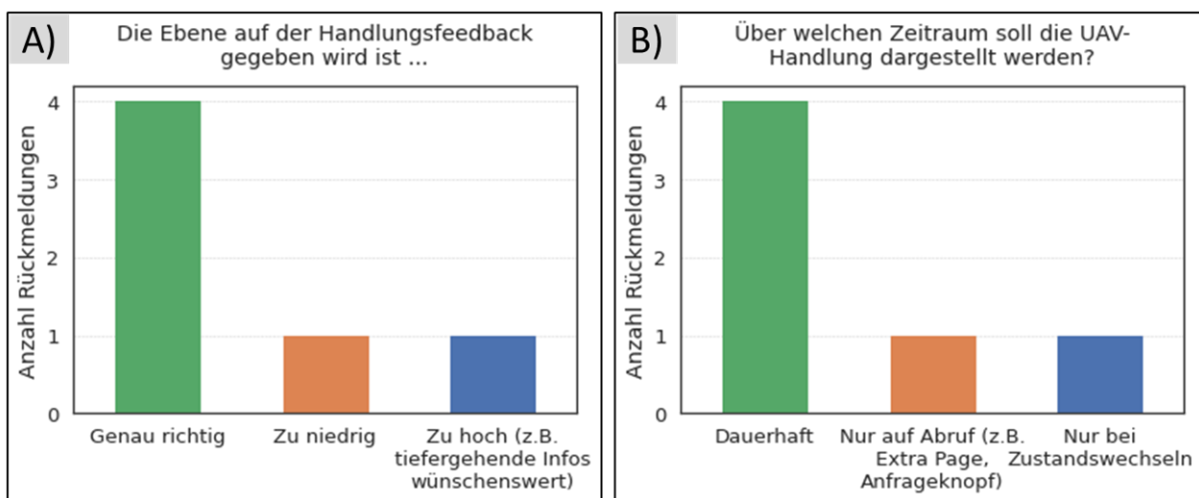


Abbildung 78. A) Rückmeldungen zur Ebene des Handlungsfeedbacks B) Rückmeldungen zur Darstellungsdauer.

In einem Kommentar erwähnte ein Proband, dass er es im Zusammenhang mit dem Handlungsfeedback auch hilfreich fand, dass die UAVs ein Symbol anzeigten, wenn sie gerade keinen Auftrag verfolgten, da er so schnell erfassen konnte, welchem UAV er noch einen Auftrag erteilen konnte.

Visuelle Meldungen

Meldungen in der taktischen Karte wurden von den Probanden mittelmäßig bewertet, wobei sich Unterschiede zwischen den Probanden zeigten (Abbildung 79).

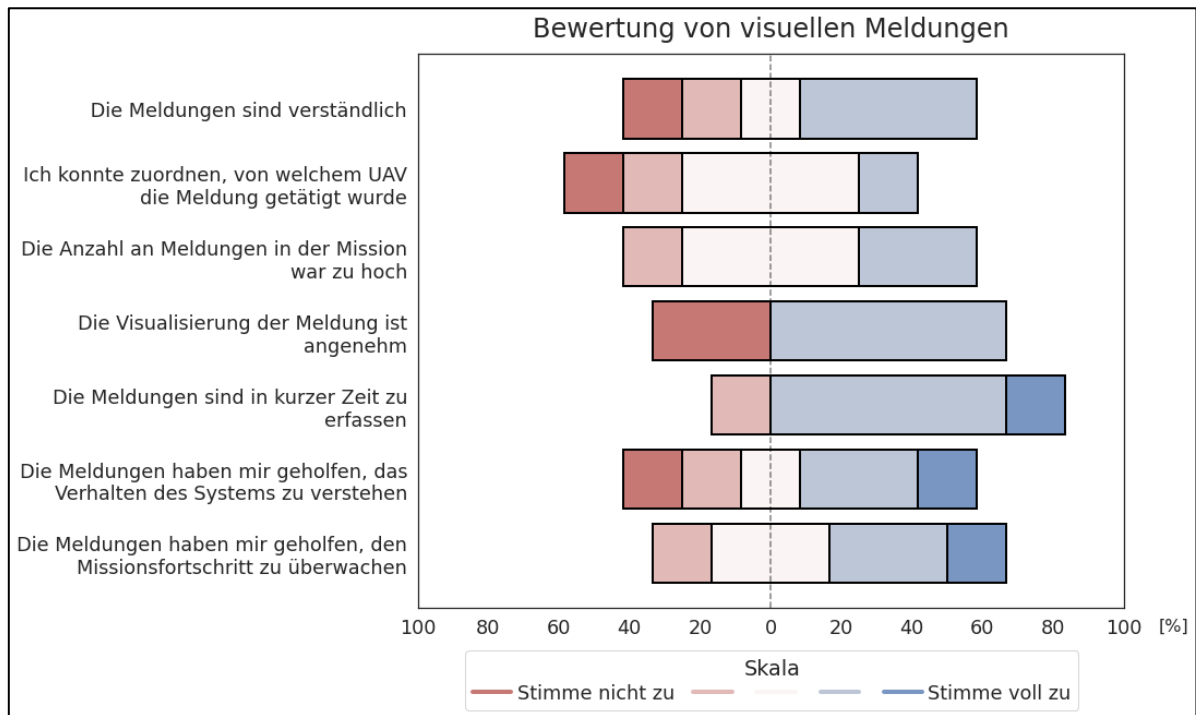


Abbildung 79. Rückmeldungen zu visuellen Benachrichtigungen in der taktischen Karte.

In Kommentaren erwähnten die Probanden die Meldungen zum Hervorheben von Objekten in der taktischen Karte positiv. Ein Proband kritisierte, dass bestimmte Meldungstexte keine Standard-Terminologie (z. B. Brevity Codes) verwendeten.

Audiomeldungen

Audiomeldungen wurden von den Probanden positiv aufgenommen (Abbildung 80), allerdings empfanden einzelne Probanden die Menge an Audiomeldungen als zu hoch.

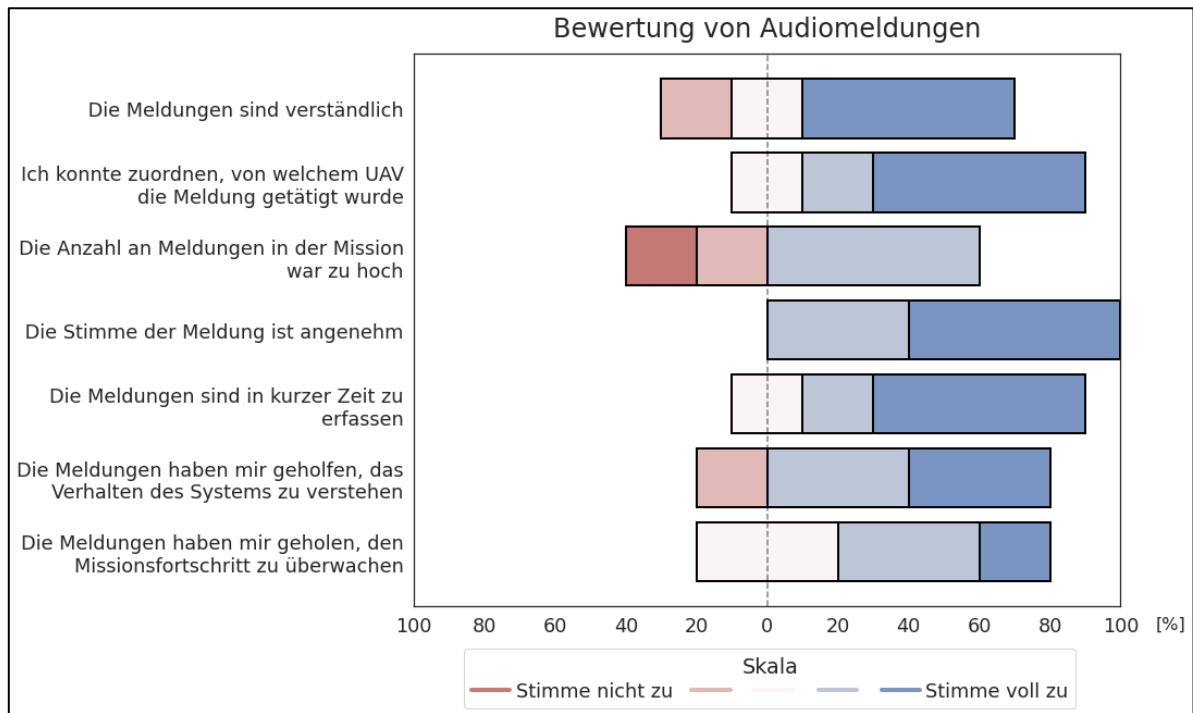


Abbildung 80. Rückmeldungen zu Audiomeldungen.

In Kommentaren wurden die Audiomeldungen positiv erwähnt. Dabei wurde unter anderem erwähnt, dass die Meldungen gut waren, um den Missionsfortschritt zu überwachen. Außerdem wurde positiv hervorgehoben, dass die Audiomeldungen überwiegend auf Standard-Terminologie beruhten. In einzelnen Kommentaren wurden weitere Audiomeldungen für bestimmte Anwendungen gewünscht.

Nutzerdialoge

Nutzerdialoge waren bei den Probanden umstritten (Abbildung 81). Dabei wurde der Inhalt der Dialogmeldung zwar eher positiv bewertet, allerdings stellen die Ergebnisse den Einsatz von Nutzerdialogen grundsätzlich in Frage. Eine große Mehrheit der Probanden wünschte sich eine bessere Kontrolle über den Bearbeitungszeitpunkt von Dialogmeldungen und es gab vereinzelte Rückmeldungen, dass Nutzerdialoge den Missionsfluss stören und dass deren Nutzung vermieden werden soll.

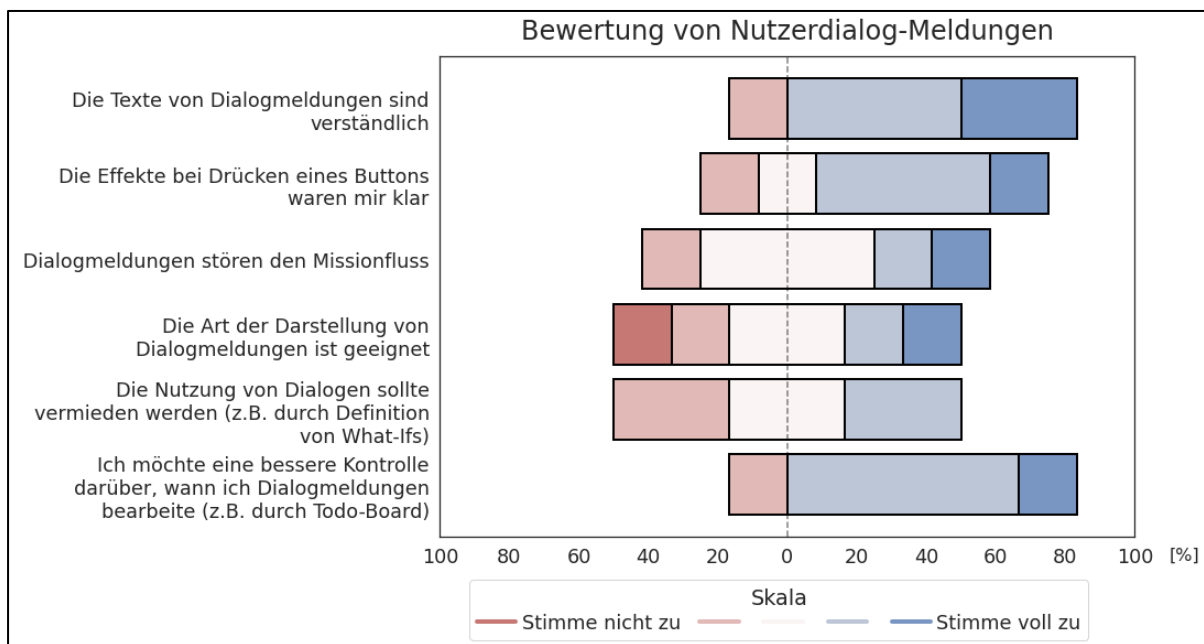


Abbildung 81. Rückmeldungen zu Nutzerdialogen.

In Kommentaren der Probanden war eine spezifische Meldung, welche auf die Anwesenheit von potenziell feindlichen Flugzeugen aufmerksam macht, besonders umstritten. Positiv erwähnt wurde, dass damit auf neue Bedrohungen hingewiesen wurde. Negativ erwähnt wurde, dass die Meldung eine Bedrohung suggeriert, was nicht notwendigerweise der Fall sein muss. Außerdem wurde erwähnt, dass die Meldung am betreffenden Luftfahrzeug erscheinen soll (anstelle des meldenden UAVs) und dass die Handlungsoptionen nicht sinnvoll sind.

Auftragskritik

Die Probanden hatten eine unterschiedliche Einstellung zur Kritik in der Auftragsdelegation, abhängig davon, in welchen Situationen ein System Kritik übt (Abbildung 82). Während die Probanden einer Auftragskritik bei Gefährdung des Missionsziels eher positiv gegenüberstanden, wurde sie etwas kontroverser eingeschätzt, wenn das Missionsziel nicht gefährdet war.

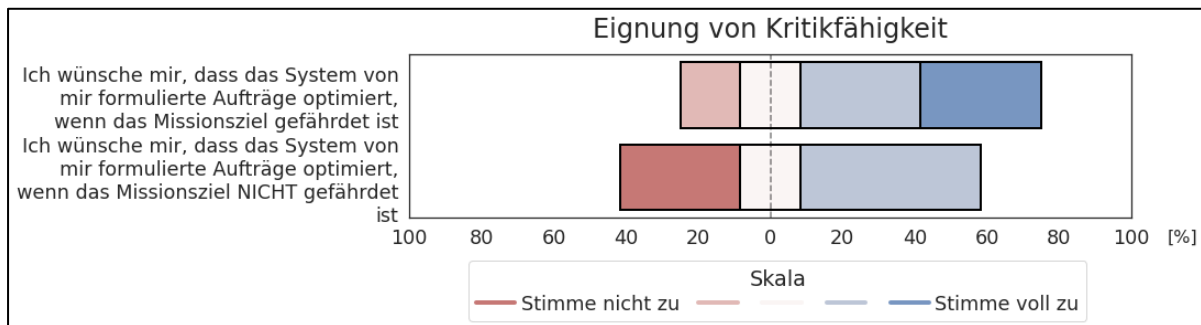


Abbildung 82. Rückmeldungen zur Eignung der Kritikfähigkeit abhängig von der Gefährdung des Missionsziels.

Eine Idee, wie sich diese verschiedenen Präferenzen vereinen lassen, lieferte ein Proband in einem Kommentar: Nutzer sollen einstellen können, ob das System auch ohne Gefährdung des Missionsziels Kritik am Nutzer übt.

5.3.2.3 Modellierungsgüte

Die Probanden bewerteten die Modellierung überwiegend positiv (Abbildung 83). Als sehr positiv bewerteten die Probanden die Modellierung der Sichtbarkeit von Objekten, die Freund-Feind Klassifizierung und die Simulatorinfrastruktur. Verbesserungspotenziale sehen einzelne Probanden in der Szenariendefinition sowie der Modellierung von Flugphysik, Luftkampf und Flugabwehrstellungen.

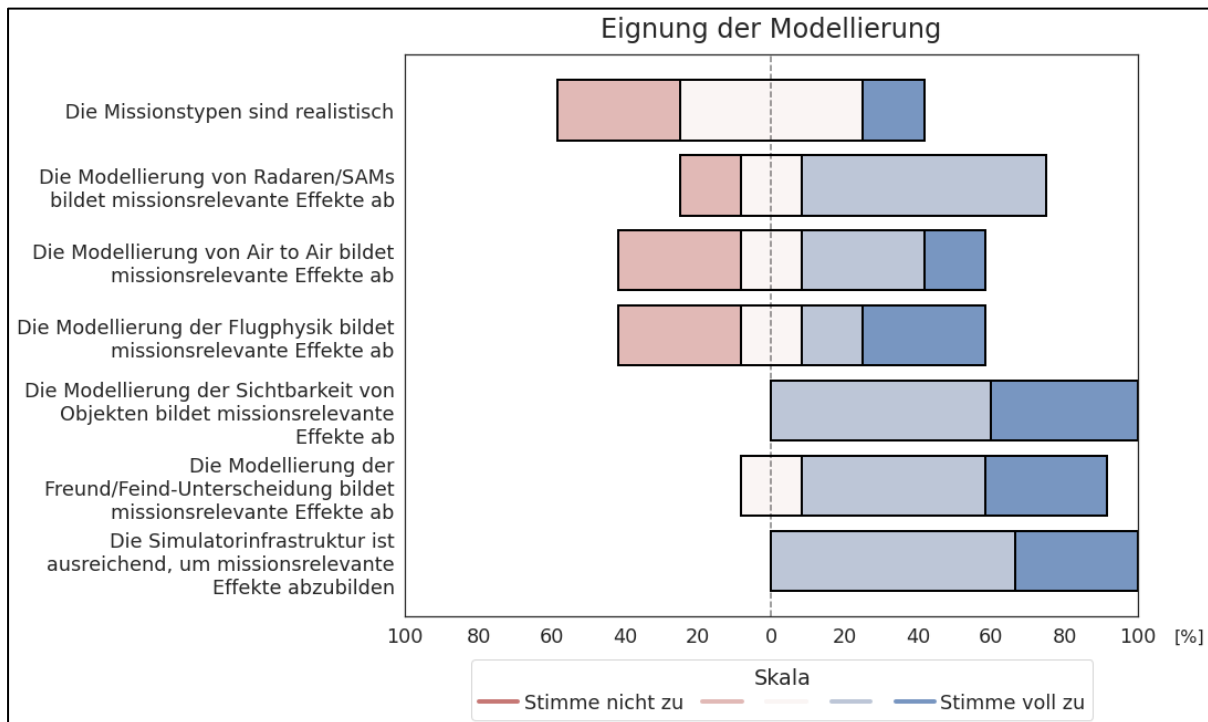


Abbildung 83. Rückmeldungen zu Fragen zur Modellierungsgüte.

In Kommentaren erwähnten die Probanden folgende Punkte:

- Die Modellierung von Flugabwehrstellungen könnte im Hinblick auf Strahlverhalten und Waffeneinsatz verbessert werden.
- Die Simulation enthält zu wenig Unsicherheiten. Beispielsweise sollten Positionen von feindlichen Objekten weniger genau ermittelt werden können.
- Gegnerische Kampfflugzeuge sind zu harmlos.
- Die taktische Lage sollte nicht so genau voraufgeklärt sein.
- Die Kurvenradien des bemannten Fliegers sind etwas zu klein.

5.3.2.4 UAV-Verhalten

Das UAV-Verhalten wurde mittelmäßig bewertet (Abbildung 84). Die Probanden schätzten die Fähigkeiten der UAVs zwar nicht als unzureichend ein, jedoch wurde das taktische Verhalten der UAVs nicht immer als sinnvoll und nachvollziehbar empfunden.

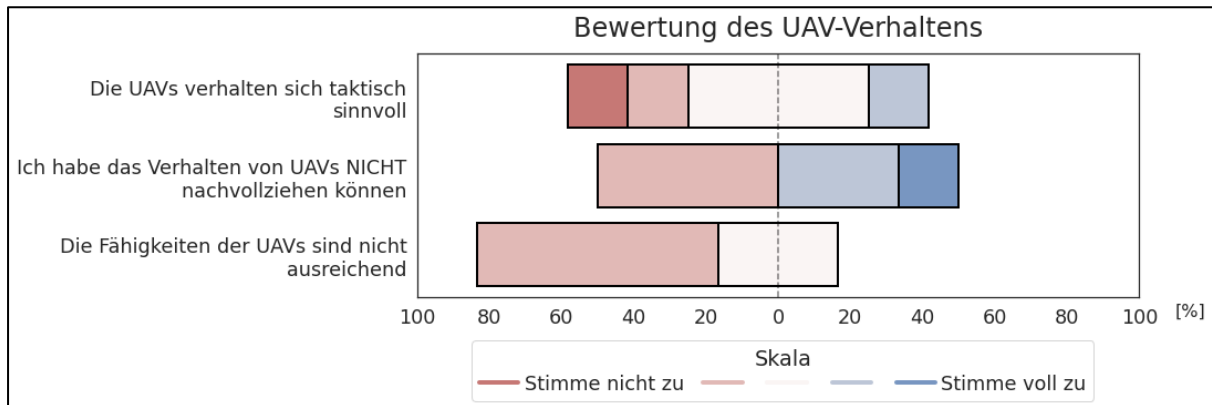


Abbildung 84. Rückmeldungen zum Verhalten von UAVs.

In Kommentaren erwähnte ein Proband hierzu, dass ein UAV die Regel „Survival over Task“ verletzt habe, indem es in eine Flugabwehrstellung geflogen ist, um ein feindliches Luftfahrzeug zu bekämpfen.

5.3.2.5 Vertrauen in das UAV

Die Probanden schätzen die Zuverlässigkeit der UAVs mittelmäßig bis eher positiv ein (Abbildung 85).

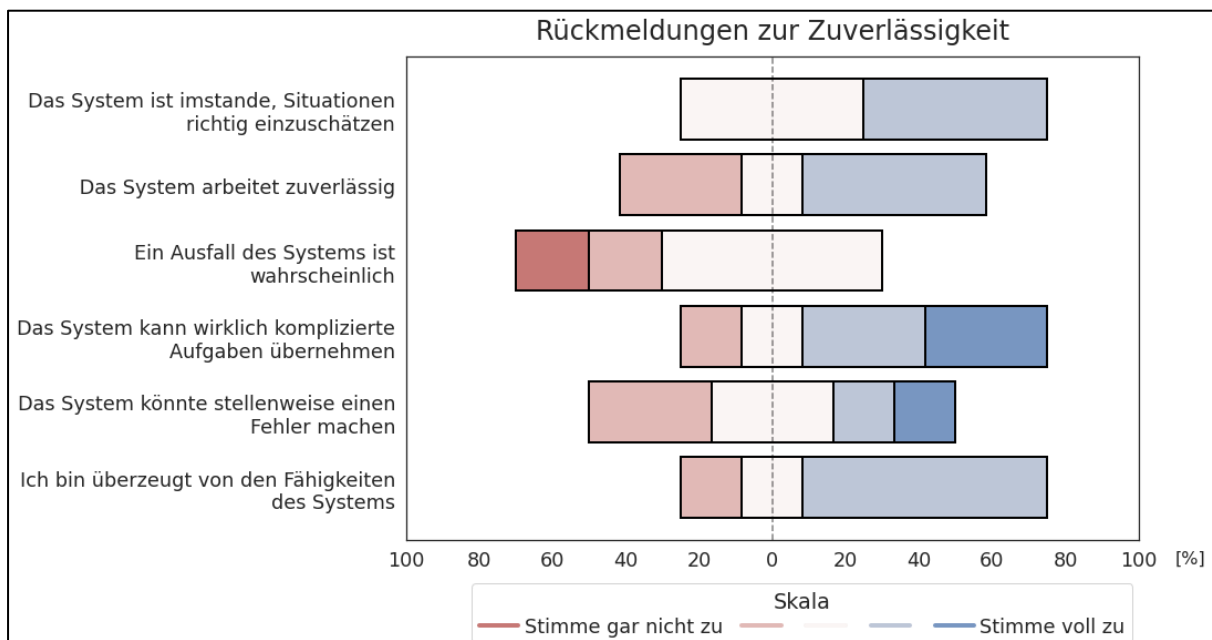


Abbildung 85. Rückmeldung zur Zuverlässigkeit der UAVs.

Die Verständlichkeit des UAV-Verhaltens wurde von den Probanden eher positiv eingeschätzt (Abbildung 86). Die Prädiktion des Verhaltens schien den Probanden etwas schwerer zu fallen als das Nachvollziehen einer Verhaltensweise.

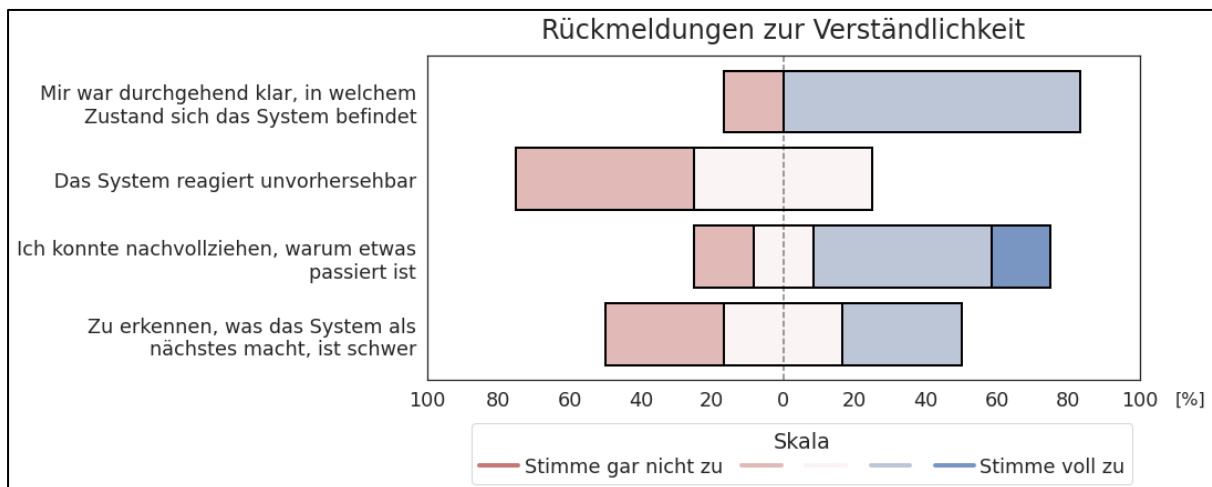


Abbildung 86. Rückmeldungen zur Verständlichkeit des Verhaltens.

Die Probanden waren insgesamt wenig bis gar nicht mit der Zusammenarbeit mit UAVs vertraut (Abbildung 87).

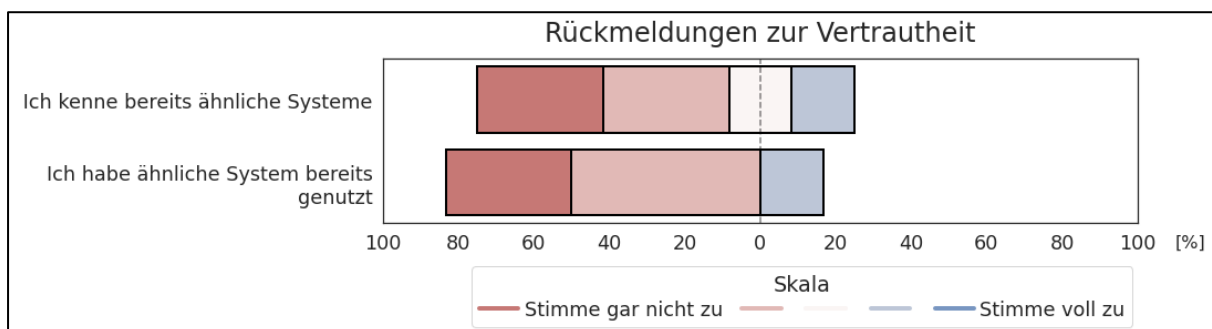


Abbildung 87. Rückmeldungen zur Vertrautheit mit UAVs.

Die Probanden brachten den Entwicklern des Systems viel Vertrauen gegenüber (Abbildung 88).

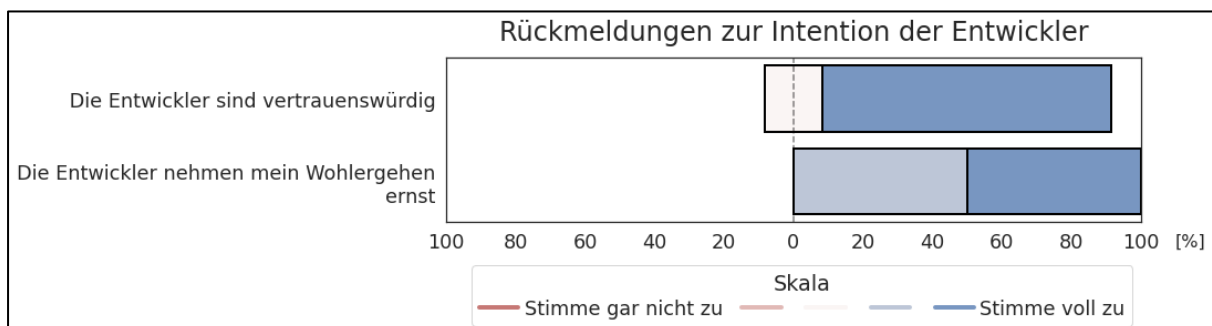


Abbildung 88. Rückmeldung zur Intention der Entwickler.

Die Probanden neigten dazu, automatisierten Systemen zu vertrauen (Abbildung 89). Allerdings gaben diese auch an, neue Systeme zunächst mit Vorsicht zu verwenden.

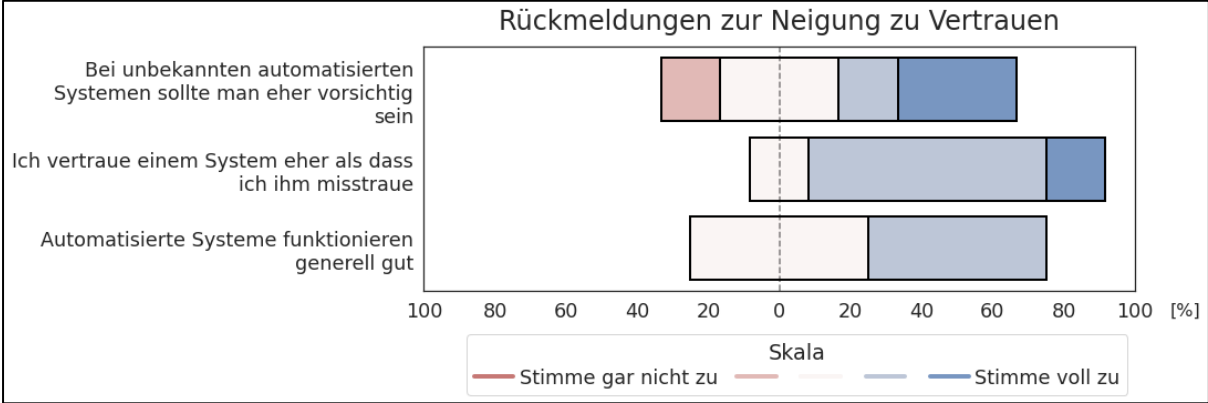


Abbildung 89. Rückmeldungen zur Neigung zu Vertrauen.

In Summe brachten die Probanden den UAVs ein mittelmäßiges bis eher positives Vertrauen entgegen (Abbildung 90).

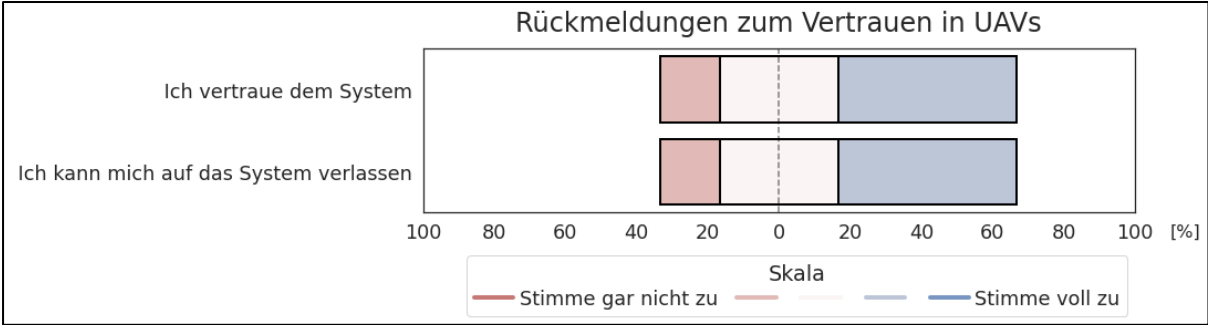


Abbildung 90. Rückmeldungen zum Vertrauen in die UAVs.

5.3.3 Diskussion

In diesem Unterabschnitt werden die zuvor dargestellten Einzelergebnisse interpretiert und zusammengeführt.

5.3.3.1 Eignung von Führungsfunktionen

Die überwiegend positiven Bewertungen der Probanden deuten darauf hin, dass die Funktionalitäten zur UAV-Führung aus Sicht der Probanden geeignet sind, um mehrere UAVs im Rahmen einer Mission mit verschiedensten Missionsaufgaben zu führen. Darüber hinaus liefert der Versuch folgende Erkenntnisse im Hinblick auf Führungsfunktionen:

- Die Auftragsstypen und Parameter sind geeignet für die meisten Missionsaufgaben; zur Steigerung der Flexibilität könnten zusätzliche Optionen sinnvoll sein.
- Die Delegationsmethoden sind geeignet, können aber noch optimiert werden, auch hinsichtlich Interoperabilität.
- Die Probanden wünschen sich eine Möglichkeit zur instantanen Auftragsdelegation. Die hierfür angebotenen Anwendungsfälle sind geeignet, zusätzlich wünschen sich die Probanden eine Funktion zur Reaktion auf Bedrohungen.
- Die Option zur Vorgabe von Idle-Verhaltensweisen sowie die dafür vorgeschlagenen Verhaltensweisen sind sinnvoll. Weitere Verhaltensrichtlinien sind gewünscht.

5.3.3.2 Eignung des Feedbacks

In Summe wurde das Feedback von den Probanden positiv bewertet, jedoch gab es Unterschiede in der Bewertung der verschiedenen Feedback-Maßnahmen. Diese Differenzen lassen sich sowohl durch Unterschiede in der übertragenen Information als auch durch Unterschiede in der Gestaltung erklären. So wurde beispielsweise das Handlungsfeedback positiv bewertet, weil Probanden sowohl die übertragene Information (Aktion des UAVs), als auch die dafür verwendete Darstellungsform (dauerhaftes Symbol neben UAV) für geeignet hielten. Der gleichen Logik folgend wurden auditive Meldungen von den Probanden deutlich besser bewertet als visuelle Meldungen oder Nutzerdialogmeldungen, weil die Probanden einerseits die übermittelte Information für wichtiger erachteten und andererseits Audiomeldungen eher der gewohnten Standard-Terminologie folgten. Im Hinblick auf Feedback liefert der Versuch folgende Erkenntnisse:

- Das Feedback der aktuellen Handlung eines UAVs ist sinnvoll und die gewählte Darstellung ist geeignet.
- Ein Feedback zur Erklärung der nächsten Schritte könnte die Vorhersagbarkeit des UAV-Verhaltens verbessern.
- Systemmeldungen sollten so standardisiert wie möglich sein (z. B. durch Brevity Codes).
- Der Bearbeitungszeitpunkt von Nutzerdialogen sollten planbar sein oder deren Nutzung sollte soweit möglich vermieden werden.
- Nutzer sollten festlegen können, ob sie eine Kritikfähigkeit in der Auftragsdelegation nur bei Gefährdung des Missionsziels oder auch in anderen Situationen wünschen.

5.3.4 Fazit

In Summe bewerteten die Probanden das Führungs- und Interaktionskonzept positiv. Dies deutet daraufhin, dass das Führungs- und Interaktionskonzept grundsätzlich geeignet ist, um mehrere UAVs während einer Manned-Unmanned Teaming Mission mit verschiedensten Missionsaufgaben erfolgreich zu leiten. Diese Einschätzung teilte auch ein Proband in seiner Zusammenfassung:

„Das Konzept geht in eine gute und richtige Richtung, jetzt kommt es auf das Feintuning an“

Die Studie liefert außerdem Erkenntnisse darüber, welche Elemente des Führungs- und Interaktionskonzepts aus Nutzersicht in einem zukünftigen System berücksichtigt werden sollten, welche ungeeignet sind und welche überarbeitet werden müssen. Die zentralen Erkenntnisse sind:

- Die Auftragsstypen und Parameter decken mögliche Missionsaufgaben überwiegend, aber nicht vollständig ab.
- Die Delegationsmethoden sind geeignet, allerdings könnten die Methoden hinsichtlich Einfachheit und Feedback der Interaktion optimiert werden. Außerdem sollte eine Interoperabilität der Methoden ermöglicht werden.
- Die Option zur instantanen Auftragserteilung ist von großer Relevanz für verschiedenste Anwendungen in denen schnelles Handeln erforderlich ist.
- Die Führung mittels Verhaltensrichtlinien ist eine sinnvolle Ergänzung zur auftragsbasierten Führung.
- Ein Feedback der aktuellen UAV-Handlung ist sinnvoll, zur Verbesserung der Vorhersagbarkeit könnten auch künftige Handlungsschritte dargestellt werden.
- Systemmeldungen sollten möglichst standardisiert sein (z. B. durch Brevity Codes).
- Der Bearbeitungszeitpunkt von Nutzerdialogen sollte planbar sein.
- Eine Kritikfähigkeit in der Auftragsdelegation ist sinnvoll. Diese Fähigkeit sollte jedoch für Fälle, in denen das Missionsziel nicht gefährdet ist, einstellbar sein.

5.4 Detailstudie: Delegationsmethoden

Ziel dieser Studie war es, die Vor- und Nachteile der verschiedenen Delegationsmethoden sowie die Effekte der Bereitstellung mehrerer Methoden zu untersuchen.

Um einen aussagekräftigen Vergleich der verschiedenen Methoden zu ermöglichen, wurde ein Versuchsaufbau gewählt, der sicherstellt, dass die spezifischen Aufgaben der Probanden mit jeder Delegationsmethode bewältigt werden können. Der Versuchsaufbau zeichnet sich außerdem durch eine hohe Dynamik aus, da in diesen Situationen ein schnelles und effektives Handeln von Piloten gefragt ist, wodurch die Stärken und Schwächen der Methoden besser hervortreten sollten.

Im Folgenden wird die Studie zunächst in zusammengefasster Form vorgestellt. Eine Detailbeschreibung von Studiendesign und Ergebnissen findet sich in den nachfolgenden Unterabschnitten.⁶⁶

Kurzfassung

Studiendesign

Aufgabenstellung: In der Studie führten die Probanden 12 Missionen durch. In jeder Mission bestand die Aufgabe der Probanden darin, feindliche Fahrzeuge zu zerstören, bevor diese einen Fluss überqueren konnten. Um dies zu erreichen, konnten die Probanden Bekämpfungsaufträge an ihre UAVs erteilen. Zur Erteilung von Aufträgen standen den Probanden die Delegationsmethoden Touch-Gesture, Voice und Gaze-HOTAS mit den Scheduling-Modi „Team“ und „Plattform“ zur Verfügung.

Variation von Delegationsmethode und Aufgabenbelastung: Die Studie verwendete ein 4x3 Versuchsdesign mit Delegationsmethode und Aufgabenbelastung als unabhängige Variablen. Die Delegationsmethode bestand aus vier Versuchsbedingungen. In drei dieser Bedingungen mussten die Probanden eine spezifische Delegationsmethode verwenden und in der vierten Bedingung konnten die Probanden frei zwischen den Methoden wählen. Die Aufgabenbelastung bestand aus drei Bedingungen. In der ersten Bedingung führten die Probanden ausschließlich die Hauptaufgabe durch. In den beiden anderen Bedingungen sollten sie zusätzlich eine auditiv-verbale bzw. eine visuell-manuelle Nebenaufgabe bearbeiten.

Datenerhebung und -analyse: In jeder Mission wurde die Performance in den Haupt- und Nebenaufgaben gemessen. In Missionen mit freier Wahl wurde zusätzlich die Verwendungshäufigkeit der Methoden erfasst. Nach jeder Mission wurde mit einem NASA TLX die Beanspruchung der Probanden bestimmt. Fragebögen erfassten zudem die Erfahrungen mit den individuellen Methoden sowie mit der Bereitstellung mehrerer Methoden.

⁶⁶ Die Studie steht im Zusammenhang mit einer Veröffentlichung (Dudek & Schulte, 2024b).

Studienergebnisse

Hauptaufgabenperformance: In Summe wurde die höchste Hauptaufgabenperformance mit Voice erzielt, gefolgt von Touch-Gesture und Gaze-HOTAS. Die Performance mit freier Wahl lag zwischen den Werten für spezifische Methoden.

Nebenaufgabenperformance: Die Performance in der Nebenaufgabe variierte für verschiedene Kombinationen von Delegationsmethode und Nebenaufgabe. Diese Variation könnte auf Ressourcenkonflikte zurückzuführen sein, da die Performance in der Nebenaufgabe immer dann geringer war, wenn die Hauptaufgabe dieselbe Eingabemodalität wie die Nebenaufgabe verwendete.

Verwendungshäufigkeit der Methoden: Bei freier Methodenwahl verwendeten die Probanden überwiegend Voice, gefolgt von Gaze-HOTAS und Touch-Gesture. Das Nutzungsverhalten der Probanden war individuell, sechs Probanden verwendeten überwiegend Voice, ein Proband ausschließlich Voice und ein Proband überwiegend Gaze-HOTAS.

Subjektive Bewertung der Methoden: Voice erhielt die beste Bewertung im Hinblick auf Zuverlässigkeit, Schnelligkeit und Beanspruchung. Touch-Gesture erhielt mittelmäßiges Feedback und Gaze-HOTAS leicht negatives Feedback.

Störfaktoren der Methoden: Ein entscheidender Störfaktor der Auftragsdelegation war eine hohe Objektdichte auf dem Display (Display Clutter), was insbesondere die Methoden Touch-Gesture und Gaze-HOTAS betroffen hat. Nebenaufgaben wurden dagegen nur dann als Störfaktoren angesehen, wenn die Eingabemodalität mit der Modalität der Delegationsmethode übereinstimmte.

Bereitstellung mehrerer Methoden: Die Performance in den Haupt- und Nebenaufgaben zeigte keine eindeutigen Vorteile durch die Bereitstellung mehrerer Methoden. Allerdings zeigten die Versuche, dass die Probanden die Auswahlmöglichkeit zwischen verschiedenen Methoden intensiv nutzten und die Methode gezielt wechselten, wenn Störfaktoren die Anwendung einer Methode behinderten.

Schlussfolgerung

Bereitstellung mehrerer Methoden: Diese Ergebnisse legen nahe, dass die Bereitstellung mehrerer Delegationsmethoden die Resilienz des Systems erhöhen könnte. Ein konkreter Vorschlag ist die Bereitstellung von zwei sehr unterschiedlichen Delegationsmethoden, weil damit einerseits die Anfälligkeit gegenüber Störfaktoren verringert werden kann und andererseits Kapazitäten für die Optimierung der einzelnen Methoden erhalten bleiben.

Weiterer Forschungsbedarf: In zukünftigen Studien sollte untersucht werden, ob die Bereitstellung mehrerer Delegationsmethoden auch dann vorteilhaft ist, wenn Probanden einer Vielzahl von Missionsaufgaben gegenüberstehen, anderen Nebenaufgaben nachgehen, und die einzelnen Methoden gegenüber spezifischen Störfaktoren robuster sind. Ferner sollte der zuvor genannte Lösungsvorschlag zur Bereitstellung zweier Methoden genauer untersucht werden.

5.4.1 Studiendesign

Dieser Unterabschnitt beschreibt das methodische Vorgehen der Studie.

5.4.1.1 Aufgabenstellung

Im Rahmen der Studie führten die Probanden zwölf Missionen durch, in denen sie Aufträge an drei UAVs delegieren mussten. Ziel der Mission war es, feindliche Fahrzeuge an der Überquerung eines Flusses zu hindern. Um dies zu erreichen, mussten die Probanden die Fahrzeuge mithilfe ihrer UAVs zerstören.

Feindliche Fahrzeuge erschienen in immer kürzer werdenden Zeitintervallen an verschiedenen Positionen entlang des Flusses. Dabei gab es drei verschiedene Arten von Fahrzeugen: Panzer, BTRs (Amphibienfahrzeuge) und Humvees. Alle Fahrzeuge versuchten, auf die andere Seite des Flusses zu kommen. Hierfür bewegten sich Panzer und Humvees auf eine von zwei Brücken zu, während BTRs versuchten, die Flüsse zu durchqueren. Die Probanden sollten nur Panzer und BTRs bekämpfen; Humvees sollten nicht bekämpft werden. Sobald ein Fahrzeug den Fluss überquert hatte, sollte es nicht mehr bekämpft werden. Dies wurde visualisiert, indem das Symbol zunächst ausgraute und anschließend nach fünf Sekunden verschwand. Eine Übersicht über die Mission ist in Abbildung 91 dargestellt.

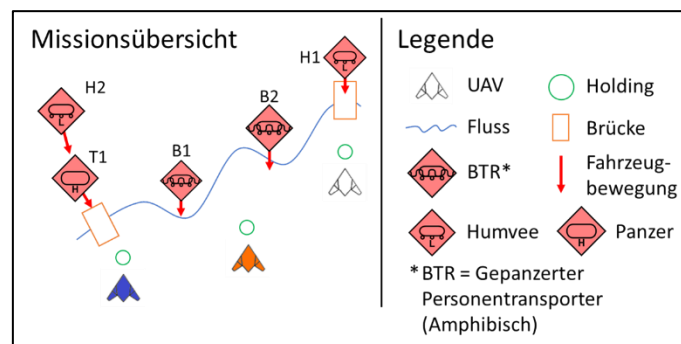


Abbildung 91. Missionsübersicht.

Um die Fahrzeuge zu zerstören, konnten die Probanden ihren UAVs Bekämpfungsaufträge erteilen. Hierfür hatten Sie die Delegationsmethoden Touch-Gesture, Voice und Gaze-HOTAS zur Verfügung und sie konnten sich entscheiden, ob Aufträge mittels Team-Modus oder mittels Plattform-Modus verteilt werden sollten.⁶⁷

Die Bekämpfung von Fahrzeugen erforderte unterschiedliche Waffen, abhängig vom Fahrzeugtyp. Aufgrund begrenzter Verfügbarkeit der Waffen musste die Auswahl des UAVs für jedes Fahrzeug sorgfältig erfolgen. Bei Delegation eines Auftrags mittels Team-Modus wählte das System ein UAV mit ausreichender Bewaffnung aus, allerdings ohne eine gleichmäßige Waffennutzung sicherzustellen. Durch die Delegation von Aufträgen mittels Plattform-Modus konnten die Probanden eine gleichmäßige Waffennutzung erreichen, allerdings war damit tendenziell eine längere Interaktion verbunden und der Auftrag konnte nicht delegiert werden, falls das gewählte UAV nicht mehr über die Bewaffnung verfügte. Die

⁶⁷ Die Scheduling-Modi sind in [3.4.1](#) beschrieben.

Probanden sollten somit beide Scheduling-Modi zielgerichtet einsetzen, um einerseits eine schnelle Bekämpfung zu erreichen und andererseits Munitionsengpässe zu vermeiden.

5.4.1.2 Variation von Delegationsmethode und Aufgabenbelastung

Die Studie nutzte ein 4x3-Versuchsdesign, bei dem die Delegationsmethode und die Aufgabenbelastung als unabhängige Variablen fungierten. Jeder Proband absolvierte eine Mission unter jeder Versuchsbedingung.

Hinsichtlich der Delegationsmethode wurden vier Bedingungen untersucht. In drei Bedingungen mussten die Probanden eine bestimmte Delegationsmethode verwenden, während sie in der vierten Bedingung frei zwischen den Delegationsmethoden wählen konnten.

Die Aufgabenbelastung bestand aus drei Bedingungen. In einer Bedingung bestand die Aufgabenbelastung ausschließlich aus der Hauptaufgabe. In den beiden anderen Bedingungen wurden Nebenaufgaben eingeführt, die zusätzliche kognitive Anforderungen an bestimmte Aufmerksamkeitsressourcen stellten:

- **Auditiv-verbale Aufgabe:** Die Probanden mussten dem Funkverkehr zuhören und Funksprüche, die an das Rufzeichen „Foxtrott“ gerichtet waren, mit dem Wort "Check" bestätigen. Funksprüche für andere Rufzeichen sollten ignoriert werden.
- **Visuell-manuelle Aufgabe:** Die Probanden sollten einen Bereich auf der taktischen Karte überwachen und eine Taste drücken, sobald ein rotes Dreieck in diesem Bereich erschien. Auf andersfarbige Dreiecke sollte nicht reagiert werden.

Die Stimuli der Nebenaufgaben wurden alle zwölf Sekunden ausgelöst, und sowohl die auditiv-verbale als auch die visuell-manuellen Stimuli waren für eine Dauer von vier Sekunden präsent. Innerhalb jeder Mission mit einer Nebenaufgabe gab es zwölf richtige und zwölf falsche Stimuli in einer pseudo-randomisierten Reihenfolge.

Die Reihenfolge der Versuchsbedingungen wurde zwischen den Probanden variiert, wobei die Delegationsmethode immer erst dann gewechselt wurde, wenn alle drei Aufgabenbelastungen mit dieser Methode abgeschlossen wurden (Abbildung 92).

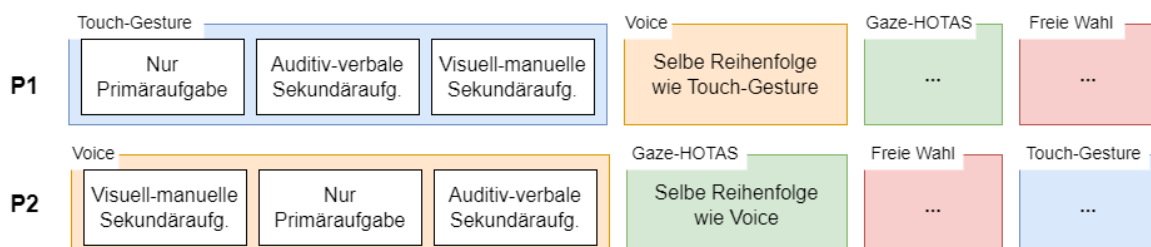


Abbildung 92. Reihenfolge der Versuchsbedingungen.

5.4.1.3 Datenerhebung und -analyse

Die Daten, die im Rahmen der Studie gesammelt wurden, setzen sich aus Messungen zusammen, die während der Mission durchgeführt wurden, sowie aus Fragebögen, die nach jeder Mission ausgefüllt wurden.

Primäraufgaben-Performance

Die Performance in der Primäraufgabe wird durch zwei Messgrößen operationalisiert:

- Zielbekämpfungsquote: Das Verhältnis bekämpfter Ziele (Panzer und BTRs) zu allen Zielen.
- Anzahl bekämpfter Falschziele: Die Anzahl der bekämpften Nicht-Ziele (Humvees).

Beide Messgrößen wurden automatisiert erfasst und verarbeitet.

Nebenaufgaben-Performance

Die Performance in der Nebenaufgabe wird durch drei Messgrößen operationalisiert:

- Anzahl richtig-positiver Reaktionen: Die Anzahl der Reaktionen auf korrekte Stimuli. Reaktionen, die mehr als eine Sekunde nach Ende des Reizes erfolgten, wurden nicht berücksichtigt.
- Anzahl falsch-positiver Reaktionen: Die Anzahl der Reaktionen auf falsche Stimuli.
- Richtig-positive Reaktionszeit: Dauer bis zur Reaktion auf einen korrekten Stimulus. Bei der visuell-manuellen Aufgabe wurde diese Dauer automatisiert erfasst. Bei der auditiv-verbale Aufgabe wurden die Sendezeiten der Funksprüche automatisiert erfasst. Der Zeitpunkt der („Check“) Rückmeldung wurde durch den Versuchsleiter manuell per Knopfdruck erfasst und automatisiert weiterverarbeitet.

Sofern nicht anders beschrieben, wurden die Messgrößen automatisiert erfasst und verarbeitet.

Verwendungshäufigkeit der Delegationsmethoden.

Die Bestimmung dieser Metrik erfolgte in den Versuchsbedingungen, in denen die Probanden frei zwischen den Delegationsmethoden wählen konnten. Hier wurde automatisiert erfasst, wie häufig die Probanden die jeweiligen Methoden verwendeten.

Fragebögen

Am Ende jeder Mission wurde ein NASA TLX (Hart & Staveland, 1988) durchgeführt. Nach Abschluss eines Missionssatzes mit einer Delegationsmethode wurde ein Fragebogen durchgeführt, um die Erfahrungen des Probanden mit der jeweiligen Delegationsmethode zu erfassen. Ebenso wurde nach allen Missionen mit freier Wahl ein Fragebogen durchgeführt, um die Auswirkungen der Wahlfreiheit sowie die Einflussfaktoren bei der Wahl der Methode zu ermitteln. Beide Fragebögen wurden eigens für die Studie konzipiert und waren nicht standardisiert.

5.4.1.4 Probanden

Der Versuch wurde mit acht Kampfflugzeugpiloten der deutschen Luftwaffe durchgeführt. Alle Piloten waren männlich und zwischen 32 und 48 Jahre alt ($M_{Alter} = 40.0$ Jahre). Die Piloten verfügten über 102 bis 2100 Stunden Erfahrung im Führen von Kampfflugzeugen ($M_h = 1196$ h; $\sigma = 782$ h). Ein Pilot verfügte über Erfahrung in der Führung von Drohnen (150 h). Tabelle 29 gibt einen Überblick über die Teilnehmer des Versuchs.

Tabelle 29. Demografische Daten der Probanden.

Kennziffer ⁶⁸	Alter	Flugstunden Kampfflugzeug	Flugstunden sonstige
11	32	102	280
12	37	800	260
13	38	600	1400
14	45	2000	600
15	47	2085	180
16	48	2100	50
17	36	1250	250
18	37	630	1500

5.4.1.5 Training

Das Training bestand aus zwei Teilen und dauerte insgesamt 3 Stunden. Im ersten Teil erhielten die Probanden eine Einführung in das System und führten eine Trainingsmission durch, bei der alle Delegationsmethoden angewendet wurden. Im zweiten Teil des Trainings führten die Probanden vier Trainingsmissionen durch (eine mit jeder Delegationsmethoden-Bedingung). In den Missionen mussten die Probanden zwei Konvoys verteidigen, indem sie Bekämpfungsaufgaben an zwei UAVs delegieren. Die Trainingsmissionen enthielten alle relevanten Aspekte der Experimentalmissionen. Im Gegensatz zu den Experimentalmissionen sollten in der Trainingsmission auch Humvees bekämpft werden und beide Nebenaufgaben kamen in einer Mission vor. In beiden Nebenaufgaben gab es vier richtige und vier falsche Stimuli. Nach jeder Trainingsmission erhielten die Probanden Feedback über ihre Performance in den primären und sekundären Aufgaben sowie Hinweise zur Verbesserung ihrer Performance.

⁶⁸ Die vergebenen Kennziffern dienen der konkreten Identifikation einer Person und werden im Rahmen aller Versuchsbeschreibungen konsistent verwendet. Die Kennziffern stehen aus Datenschutzgründen nicht in Verbindung mit den bei Ergebnissen verwendeten Nummern.

5.4.1.6 Versuchsaufbau

Die Versuchsdurchführung erfolgte im Kampfflugzeugsimulator, dessen Konfiguration im Wesentlichen der Beschreibung in Kapitel 4 entspricht.

Abweichend von dieser Beschreibung wurden verschiedene Fähigkeiten des Simulators angepasst, um die Versuchsbedingungen zu kontrollieren oder eine bessere Vergleichbarkeit zu gewährleisten. Hierzu zählen:

- Deaktivieren verschiedener Führungsfunktionalitäten
 - Delegationsmethode Timeline vollständig deaktiviert.
 - Delegationsmethode Voice: Keine Delegation mittels Timeslot-Modus.
 - Delegationsmethode Touch-Gesture: Keine Positionierung mittels Plattform-Navigation oder Plattform-CAPs.
- Anpassen der Delegationsmethode Touch-Gesture: Erteilte Aufträge landen nicht direkt im aktiven Plan, sondern der Plan muss wie bei der Methode Timeline bestätigt werden.
- Wenn ein UAV keine Aufträge mehr offen hatte, kehrte es an eine Warteposition zurück, die für jedes UAV einzigartig war.
- Das bemannte Kampfflugzeug wurde so konfiguriert, dass es automatisiert eine Warteschleife abflog, sodass Probanden ihr eigenes Flugzeug nicht führen mussten.

Die Displays wurden so konfiguriert, dass auf dem MHDD die taktische Karte angezeigt wurde, auf dem linken SHDD wurde der Missionsplan dargestellt und auf dem rechten SHDD wurde die Bewaffnung der Teammitglieder angezeigt (Abbildung 93). Vor dem Start jeder Mission durften die Probanden die taktische Karte so konfigurieren, dass alle relevanten Missionsobjekte gut sichtbar waren.



Abbildung 93. Simulator in Experimentalkonfiguration.

5.4.2 Ergebnisse

Dieser Unterabschnitt stellt die Ergebnisse der Studie dar.

5.4.2.1 Hauptaufgaben-Performance

In den Versuchsbedingungen, in denen die Probanden bestimmte Delegationsmethoden anwenden mussten, wurde die höchste Zielbekämpfungsquote mit Touch-Gesture erreicht (Abbildung 94; $M = 85,5\%$), gefolgt von Voice ($M = 84,8\%$) und Gaze-HOTAS ($M = 81,7\%$). Unter den Bedingungen der freien Wahl lag die Zielbekämpfungsquote dazwischen ($M = 83,9\%$). Bei der Interpretation der Grafik ist zu beachten, dass der Mittelwert dargestellt ist, da er aufgrund der diskreten Verteilung der Datenpunkte und der vergleichsweise geringen Stichprobengröße (24 pro Bedingung) als das aussagekräftigere Maß erschien.

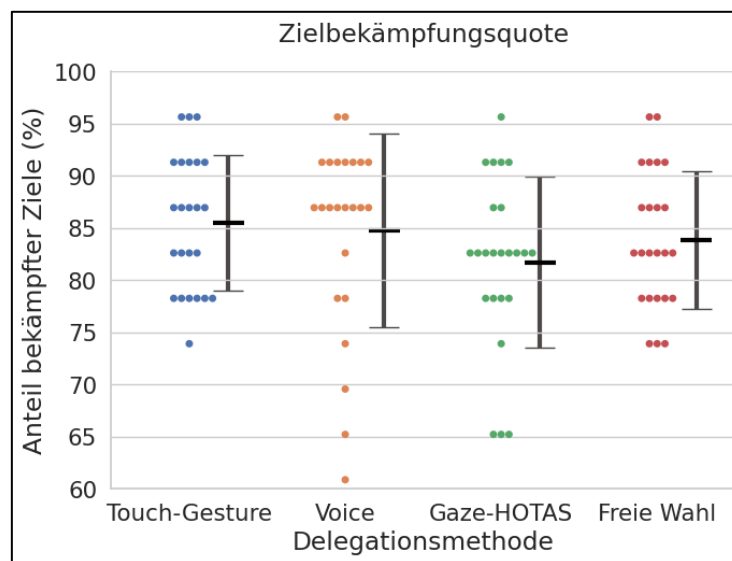


Abbildung 94. Zielbekämpfungsquote nach Methode. Die Stichprobengröße von jeweils 24 Proben ergibt sich aus drei Missionen pro Methode und Proband (keine Differenzierung nach Aufgabenbelastung).

Die Zielbekämpfungsquote variierte unter den Bedingungen Voice und Gaze-HOTAS stärker als unter den Bedingungen Touch-Gesture und freie Wahl. Eine weitere Differenzierung der Zielbekämpfungsquote nach Aufgabenbelastung liefert keine tiefgehenden Erkenntnisse.

Die Probanden griffen nur selten Falschziele/Humvees an (18 von 576 möglichen Falschzielen; 6 mögliche Falschziele pro Mission bei 12 Missionen pro Proband), wobei sich eine bemerkenswerte Konzentration in der ersten Mission zeigte (6 von 48 möglichen Falschzielen). Dies könnte darauf zurückzuführen sein, dass Probanden die Vorgaben des Briefings nicht korrekt rekapitulierten. Wenn Probanden in der ersten Mission Falschziele bekämpften, wurden sie darauf hingewiesen, die Einsatzregeln zu befolgen. Lässt man die erste Mission in der Betrachtung außen vor, so wurden Falschziele vor allem in Bedingungen mit Touch-Gesture und Gaze-HOTAS bekämpft (Tabelle 30; 132 mögliche Falschziele für jede Bedingung in den verbleibenden Missionen). Weniger Falschziele wurden bei freier Methodenwahl bekämpft und keine Falschziele mit Voice.

Tabelle 30. Anzahl bekämpfter Falschziele nach Methode (ohne erste Mission).

Touch-Gesture	Voice	Gaze-HOTAS	Freie Wahl
5	0	5	2

5.4.2.2 Nebenaufgaben-Performance

Die Anzahl der richtig-positiven Reaktionen variierte für verschiedene Kombinationen von Delegationsmethode und Nebenaufgabe (Abbildung 95). Vergleicht man die richtig-positiven Reaktionen für jede Methode, so schneidet Voice bei der visuell-manuellen Aufgabe besser, bei der auditiv-verbale Aufgabe jedoch schlechter ab als die beiden anderen Methoden. Dieser Unterschied könnte auf einen Ressourcenkonflikt zurückzuführen sein, da Voice die gleiche Eingabemodalität wie die auditiv-verbale Aufgabe verwendet, während Touch-Gesture und Gaze-HOTAS die gleiche Modalität wie die visuell-manuelle Aufgabe verwenden.

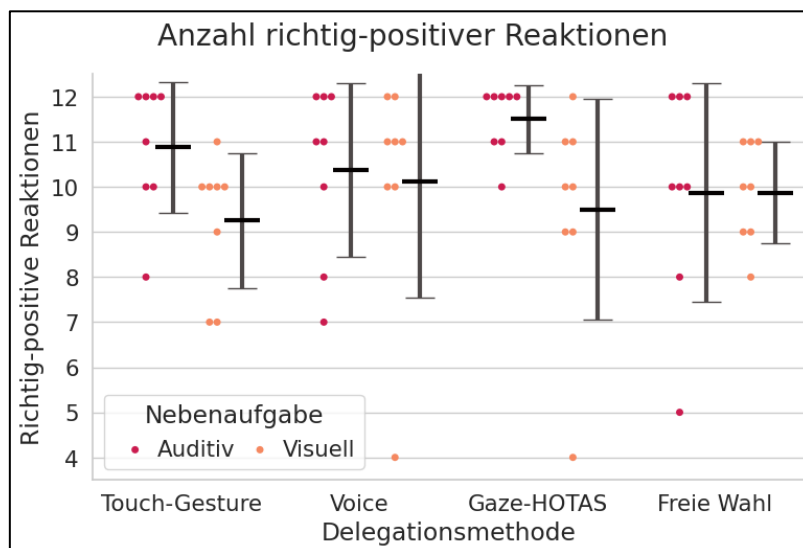


Abbildung 95. Anzahl richtig-positiver Reaktionen nach Methode und Nebenaufgabe. Die Stichprobengröße beträgt für alle Bedingungen acht Proben: Je eine Mission für jede Kombination aus Methode und Aufgabenbelastung pro Proband.

Die Probanden reagierten bei beiden Nebenaufgaben nicht auf falsche Stimuli.

Die richtig-positive Reaktionszeit in der auditiv-verbale Aufgabe war niedriger, wenn die Probanden Touch-Gesture (Abbildung 96; MD = 2,34 s) oder Gaze-HOTAS (MD = 2,43 s) verwendeten, verglichen mit Voice (MD = 2,57 s) und Free Choice (MD = 2,59 s).

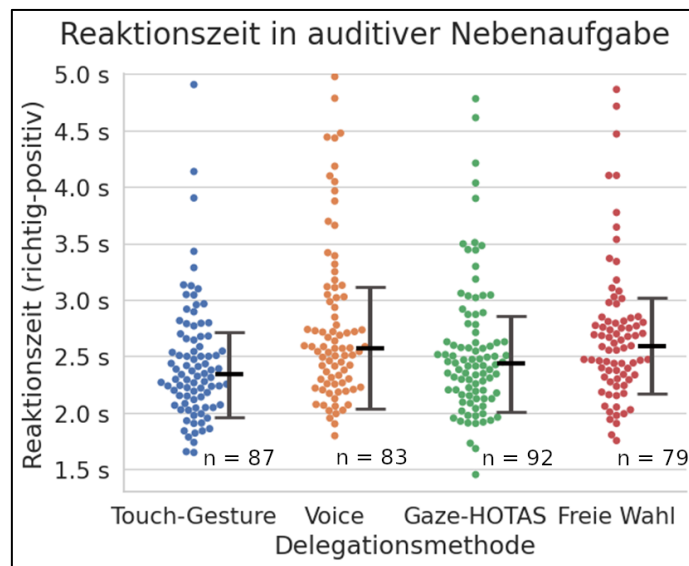


Abbildung 96. Reaktionszeit in der auditiv-verbale Nebenaufgabe nach Methode. Für jede Methode lag die theoretisch erreichbare Stichprobengröße bei 96 (12 korrekte Stimuli pro Mission, eine Mission je Proband).

Die richtig-positive Reaktionszeit bei der visuell-manuellen Aufgabe war bei allen Delegationenethoden ähnlich. Die höchsten Reaktionszeiten wurden in der Touch-Gesture Missionen (MD = 1,36 s) und die niedrigsten in Missionen mit freier Wahl (MD = 1,25 s) gemessen. Diese Unterschiede sind nicht signifikant.

5.4.2.3 Verwendungshäufigkeit der Delegationenethoden

In den Missionen mit freier Methodenwahl nutzten die Probanden vor allem Voice (75,9 %), gefolgt von Gaze-HOTAS (21,7 %) und Touch-Gesture (2,4 %). Die Häufigkeit der Verwendung von Delegationenethoden variierte bei unterschiedlichen Aufgabenbelastungen (Tabelle 31).

Tabelle 31. Verwendungshäufigkeit der Methoden nach Aufgabenbelastung.

Aufgabenbelastung	Verwendungshäufigkeit		
	Touch-Gesture	Voice	Gaze-HOTAS
Tasking	1.9 %	63.9 %	34.3 %
Tasking + Auditive	0.9 %	80.9 %	18.2 %
Tasking + Visual	4.8 %	84.1 %	11.1 %

Die Probanden zeigten Unterschiede in der Häufigkeit der Nutzung der einzelnen Methoden. Sechs von acht Probanden nutzten überwiegend Voice, ein Proband ausschließlich Voice und ein Proband überwiegend Gaze-HOTAS. Wenn die Probanden die Möglichkeit

hatten, mehrere Delegationsmethoden frei zu verwenden, nutzten sie eine Zeit lang ausschließlich eine Methode, bevor sie zu einer anderen wechselten (Abbildung 97).

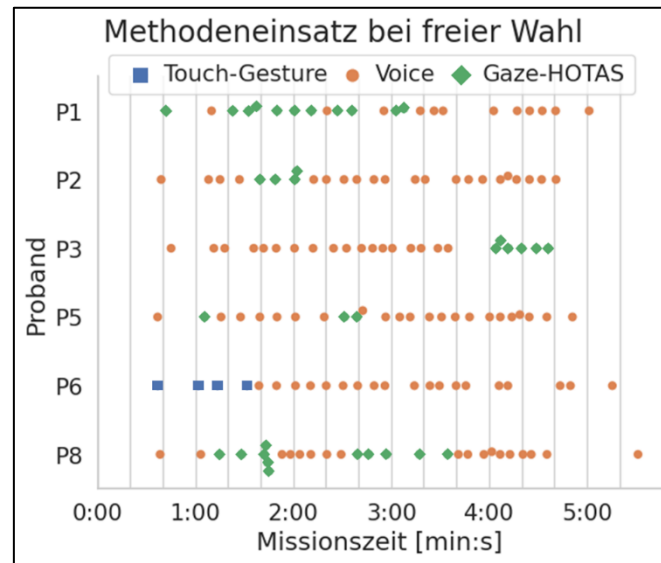


Abbildung 97. Verwendung von Delegationsmethoden in Missionen mit freier Wahl ohne Nebenaufgabe (P4 und P7 ausgeschlossen, da sie nur Gaze-HOTAS bzw. Voice verwendeten).

Die Wechsel zwischen den Methoden wurden mithilfe von Replay-Daten genauer analysiert. Es zeigte sich ein Muster: Wenn Objekte dicht nebeneinander dargestellt wurden (Display Clutter) und die Probanden dadurch Schwierigkeiten hatten, die Objekte auszuwählen, wechselten sie von Gaze-HOTAS zu Voice. Es wurden keine ähnlichen Gründe für den Wechsel von Voice zu Gaze-HOTAS festgestellt. Allerdings fällt auf, dass dieser Wechsel seltener vorkommt, wenn ein hoher Zeitdruck herrscht.

5.4.2.4 Beanspruchung

Wird die subjektive Beanspruchung nach der Delegationsmethode unterschieden, so wurde die geringste Beanspruchung unter den Versuchsbedingungen der freien Wahl (MD = 49,6) berichtet, gefolgt von Touch-Gesture (MD = 50,8), Voice (MD = 53,8) und Gaze-HOTAS (MD = 56,3). Die Unterschiede zwischen den Methoden sind signifikant ($p \cong 0,022$; repeated measures ANOVA).

Eine weitere Differenzierung nach Aufgabenbelastung zeigt keinen klaren Trend. Es existiert keine Korrelation zwischen subjektiver Beanspruchung und Zielbekämpfungquote.

5.4.2.5 Bewertung der Methoden

Die Probanden bewerteten Voice als die zuverlässigste und schnellste Delegationsmethode (Abbildung 98). Sie gaben auch an, dass ihre Nutzung nur eine geringe Beanspruchung mit sich bringt. Touch-Gesture erhielt insgesamt ein mittelmäßiges Feedback und Gaze-HOTAS erhielt leicht negatives Feedback, wobei insbesondere die Zuverlässigkeit und die Beanspruchung kritisiert wurden.

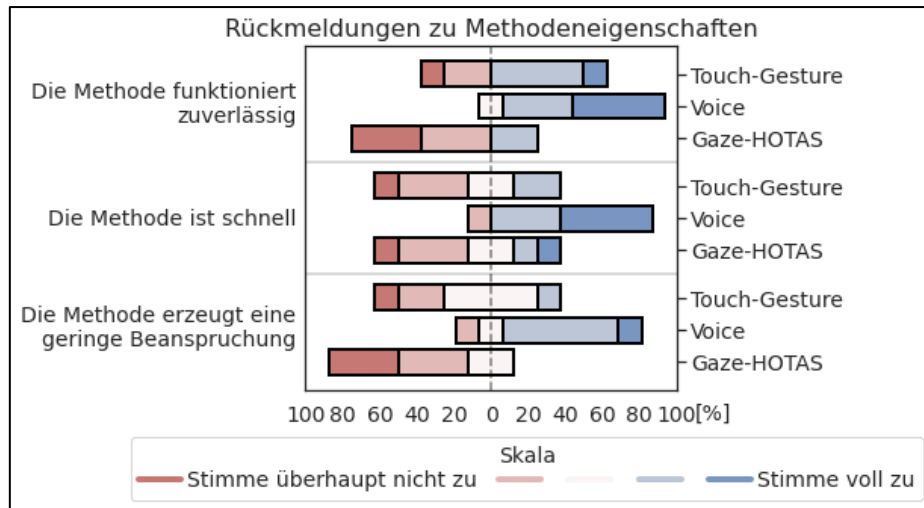


Abbildung 98. Rückmeldungen zu Eigenschaften der Methoden.

Die Störfaktoren bei der Verwendung einer Methode sind gemäß Aussage der Probanden zwischen den Methoden verschieden (Abbildung 99). In Summe wurde Display Clutter als der Hauptstörfaktor angesehen. Alle Probanden gaben an, dass Display Clutter die Delegation mit Touch-Gesture und Gaze-HOTAS störte, und drei von acht Probanden gaben an, dass Clutter Voice störte. Es scheint, als ob Display Clutter bei der Interaktion mit Objekten als störender empfunden wird als bei der Wahrnehmung von Informationen. Ein weiterer nennenswerter Effekt ist, dass Nebenaufgaben als besonders störend empfunden wurden, wenn Haupt- und Nebenaufgabe dieselbe Eingabemodalität verwendeten. Dies wird darin erkennbar, dass die auditiv-verbale Aufgabe vor allem die Delegation mit Voice stört, während die visuell-manuelle Aufgabe vor allem die Delegation mit Touch-Gesture und Gaze-HOTAS stört.

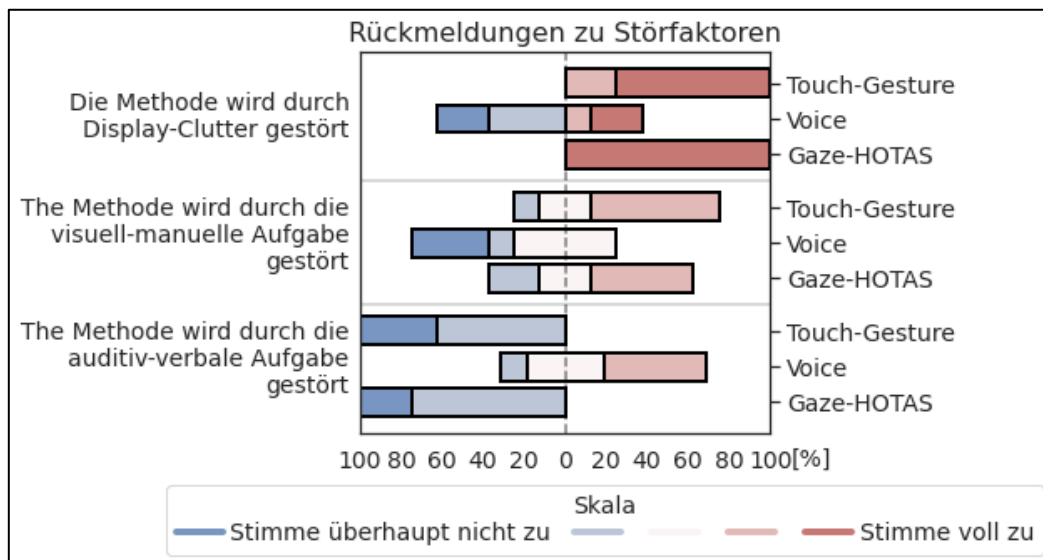


Abbildung 99. Rückmeldung zu Störfaktoren (invertierte Farbgebung beachten).

5.4.2.6 Einflussfaktoren der Methodenwahl und Effekte der Wahlfreiheit

Die Probanden betrachteten die Zuverlässigkeit als wichtigste Eigenschaft für die Verwendungshäufigkeit einer Delegationsmethode, gefolgt von der Geschwindigkeit. Die Beanspruchung, die durch die Anwendung der Methode entsteht, war nur von untergeordneter Bedeutung.

In Bezug auf die Auswahlstrategie bei der Methodenwahl gaben fünf Probanden an, hauptsächlich eine Methode zu verwenden und die anderen Methoden als Backup zu nutzen, falls die bevorzugte Methode nicht gut funktioniert. Zwei Probanden gaben an, ausschließlich eine Methode zu verwenden, während ein Proband berichtete, dass er sich in jeder Situation neu für eine Delegationsmethode entscheidet.

Die unmittelbare Entscheidung für eine Methode basierte nach Angaben der Probanden in erster Linie auf der Menge an Objekten um das Zielobjekt (Display Clutter), gefolgt von der Nebenaufgabe (Abbildung 100). Der Scheduling-Modus hatte trotz unterschiedlicher Interaktionsabläufe nur einen geringen Einfluss auf die Wahl einer Methode.

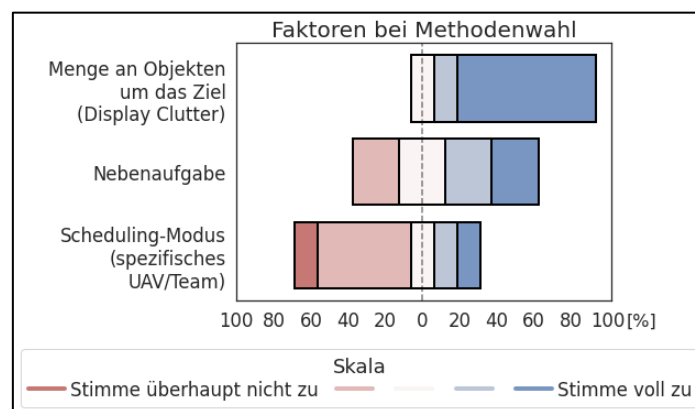


Abbildung 100. Berücksichtigte Faktoren bei der Auswahl einer Methode.

Die Probanden schätzten die Auswirkungen der Bereitstellung mehrerer Delegationsmethoden als positiv ein (Abbildung 101).

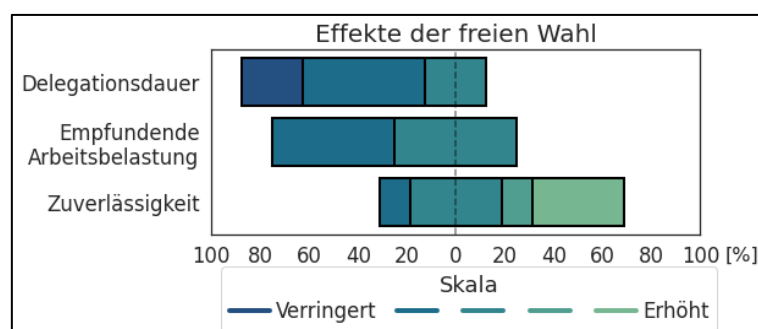


Abbildung 101. Auswirkungen der Bereitstellung mehrerer Methoden.

5.4.3 Diskussion

In diesem Unterabschnitt werden die vorgestellten Einzelergebnisse interpretiert und zueinander in Bezug gesetzt.

5.4.3.1 Methodenvergleich und Nutzung der Methoden

Vergleicht man die Delegationsmethoden hinsichtlich ihrer Gesamtperformance, so erzielte Voice die besten Ergebnisse, gefolgt von Touch-Gesture. Gaze-HOTAS lieferte die schlechteste Performance. Dies deckt sich mit der subjektiven Bewertung der Methoden.

In Einklang mit diesen Ergebnissen nutzten die Probanden in Missionen mit freier Methodenwahl hauptsächlich Voice. Interessanterweise folgte an zweiter Stelle nicht Touch-Gesture, sondern Gaze-HOTAS. Der Grund dafür könnte sein, dass, obwohl die durchschnittlichen Interaktionszeiten mit Gaze-HOTAS länger sind als mit den anderen beiden Methoden, diese Methode unter idealen Bedingungen die kürzesten Interaktionszeiten bietet und die Probanden daher Gaze-HOTAS verwenden, wenn geeignete Bedingungen vorliegen (Tabelle 32).

Tabelle 32. Median der Delegationszeiten nach Methode bei Betrachtung aller Daten und bei Betrachtung des schnellsten Zehntels jedes Probanden.

Datensatz	Interaktionszeit der Auftragsdelegation (Median)		
	<i>Touch-Gesture</i>	<i>Voice</i>	<i>Gaze-HOTAS</i>
Schnellstes Zehntel	4.3 s	5.8 s	3.8 s
Vollständig	8.9 s	9.1 s	10.4 s

5.4.3.2 Hohe Bedeutung von Display Clutter

Display Clutter ist wahrscheinlich der Hauptfaktor, der zur unterschiedlichen Performance und Bewertung der Methoden beiträgt. Display Clutter wurde von den Probanden als besonders störend empfunden, und die Probanden gaben an, dass Touch-Gesture und Gaze-HOTAS gut funktionieren, wenn sich keine anderen Objekte in der Nähe des Zielobjekts befanden. Daher sollten Maßnahmen ergriffen werden, um die Zuverlässigkeit dieser beiden Delegationsmethoden in Clutter-Situationen zu verbessern, bevor diese in einem Kampfflugzeugcockpit zur Anwendung kommen. Da sich Display Clutter in erster Linie auf die Interaktion auszuwirken scheint, müssen diese Maßnahmen nicht unbedingt die Informationsdichte verringern (wie dies z. B. durch Layer-Management erreicht wird), sondern können darauf abzielen, die Interaktion selbst robuster zu machen (z. B. durch einen zweiten Auswahlschritt bei hoher Objektdichte). Bei Gaze-HOTAS könnte eine Verbesserung auch durch eine höhere Genauigkeit des Blickbewegungsmesssystems erreicht werden, da die Messungen bei einigen Probanden eine Abweichung von mehr als zwei Zentimetern auf dem Bildschirm aufweisen konnten, insbesondere wenn diese auf den unteren Teil des Bildschirms blickten.⁶⁹

⁶⁹ Wenn der untere Teil des Displays betrachtet wird, sind die Augenlider nur wenig geöffnet. Dies wirkt sich negativ auf die Genauigkeit der Blickbewegungsmessung aus.

5.4.3.3 Bereitstellung mehrerer Methoden

Die Performance in den Haupt- und Nebenaufgaben gibt keine Indizien dafür, dass die Bereitstellung mehrerer Delegationsmethoden von Vorteil ist. Zwar zeigt sich bei freier Wahl der Methode eine geringere Streuung der Hauptaufgaben-Performance, aber der Median dieser Werte liegt zwischen den Medianen bei festgelegter Methode. In der Sekundäraufgabe war die Performance bei freier Methodenwahl ähnlich wie die Performance mit Voice.⁷⁰

Dennoch gibt es einige Aspekte, die für die Bereitstellung mehrerer Methoden sprechen:

- **Verwendung:** In den Missionen mit freier Wahl wurden von den Probanden verschiedene Methoden verwendet, trotz der Unterschiede in der Performance und der Bewertung der verschiedenen Methoden. Dies deutet darauf hin, dass jede Methode spezifische Anwendungsfälle haben kann, in denen sie von den Probanden eingesetzt wird. Darüber hinaus deutet die häufig genannte Auswahlstrategie, eine Methode hauptsächlich und die anderen als Backup zu verwenden, darauf hin, dass die Verfügbarkeit mehrerer Methoden insbesondere dann von Vorteil sein kann, wenn Störfaktoren vorliegen oder bei der Nutzung einer Methode unerwartete Probleme auftreten.
- **Ressourcenkonflikte:** Die Leistung der Probanden in der Nebenaufgabe war geringer, wenn die Nebenaufgabe dieselbe Modalität wie die Hauptaufgabe verwendete. Im Aufgabenumfeld von künftigen Kampfflugzeugpiloten ist Multitasking sehr wahrscheinlich, sodass mit einem Auftreten solcher Ressourcenkonflikte zu rechnen ist. Die Bereitstellung mehrerer Methoden kann den Piloten helfen, ihre mentalen Ressourcen zu managen, da sie die Methode jederzeit wechseln können.
- **Subjektive Bewertung:** Die Probanden erwarten Verbesserungen in allen abgefragten Aspekten, wenn mehrere Methoden bereitgestellt werden. Wenn man bedenkt, dass neben den in dieser Studie gesammelten Erfahrungen auch bisherige Erfahrungen zu dieser Einschätzung beitragen könnten, ist dieser Punkt besonders relevant.

⁷⁰ Was plausibel erscheint, wenn man bedenkt, dass die Probanden hauptsächlich Voice verwendeten.

5.4.4 Fazit

In der Studie erzielten die Probanden mit allen Delegationsmethoden eine gute Performance, wobei die höchste Gesamtperformance mit der Delegationsmethode Voice erzielt wurde. Alle Delegationsmethoden unterlagen Störfaktoren wie Display Clutter und Ressourcenkonflikten, wobei es individuelle Unterschiede zwischen den Methoden gab. Der Störfaktor Display Clutter könnte einen entscheidenden Beitrag zur höheren Gesamtperformance mit Voice geleistet haben, da er auf Voice am wenigsten Einfluss hatte. Es wird erwartet, dass dieser Vorteil von Voice durch eine robustere Interaktion bei den Methoden Touch-Gesture und Gaze-HOTAS zumindest teilweise ausgeglichen werden kann.

Die Ergebnisse deuten darauf hin, dass es von Vorteil sein kann, mehrere Delegationsmethoden bereitzustellen, um die Resilienz des Systems zu erhöhen. Es wurde gezeigt, dass Störfaktoren die Performance von Methoden stark beeinflussen können. Mehrere Delegationsmethoden können hier Abhilfe schaffen, da Nutzer bei Vorliegen eines Störfaktors die Methode wechseln können. Dieses Verhalten hat sich auch im Versuch gezeigt. Die subjektive Einschätzung der Probanden spricht ebenfalls für die Bereitstellung mehrerer Methoden. Es ist jedoch zu bedenken, dass die Entwicklung jeder zusätzlichen Methode personelle Ressourcen erfordert, die auch zur Verbesserung einer Methode verwendet werden könnten. Ein Kompromiss könnte darin bestehen, zwei Methoden zu entwickeln, die unterschiedliche Ressourcenkanäle belasten und von verschiedenen Störfaktoren beeinflusst werden.

Wenn mehrere Methoden bereitgestellt werden, sollten diese nach Aussagen der Probanden auf dem gleichen logischen Ablauf basieren. Dies unterstützt die Bildung eines mentalen Modells und ermöglicht Interoperabilität zwischen den Methoden (z. B. Auftragsdelegation mit einer Methode und Planaktivierung mit der anderen).

In nachfolgenden Studien sollte weiter untersucht werden, ob die Bereitstellung mehrerer Delegationsmethoden auch dann von Vorteil ist, wenn:

1. Das Missionsziel die Delegation von mehreren verschiedenen Auftragsstypen erfordert.
2. Andere Nebenaufgaben hinzukommen (z. B. n-back).
3. Alle Delegationsmethoden robust gegenüber Display Clutter sind.

Diese Studien sollten außerdem eine größere Anzahl von Probanden umfassen, um die statistische Aussagekraft zu erhöhen.

5.5 Detailstudie: Kritikfähigkeit in Auftragsdelegation

Ziel dieser Studie war es, die Auswirkungen der Kritikfähigkeit in der Auftragserteilung zu ermitteln. Zu diesem Zweck wurde ein vergleichender Versuchsaufbau gewählt, bei dem eine kritikbasierte Entscheidungshilfe mit einer vorschlagsbasierten Entscheidungshilfe am Anwendungsfall der Auftragsparametrisierung verglichen wurde. Dabei wurden folgende Effekte erwartet:

1. Bessere Performance mit dem Vorschlagssystem, wenn das System fehlerfrei funktioniert.
2. Besserer Umgang mit Systemfehlern bei Verwendung des Kritiksystems.

Im Folgenden wird die Studie zunächst in zusammengefasster Form vorgestellt. Eine Detailbeschreibung von Studiendesign und Ergebnissen findet sich in den nachfolgenden Unterabschnitten.⁷¹

Kurzfassung

Studiendesign

Aufgabenstellung: In der Studie führten die Probanden zwei Missionen durch. In beiden Missionen bestand die Aufgabe der Probanden darin, Bekämpfungsaufträge zu parametrisieren, um Gruppen von Objekten (sog. Cluster) zu bekämpfen. Diese Aufgabe fand unter Zeitdruck statt, da in regelmäßigen Abständen von 20 Sekunden neue Cluster erschienen. Die Parametrisierung wurde durch ein Entscheidungsunterstützungssystem unterstützt und die Probanden erhielten visuelles Feedback über das erwartete Schadenausmaß einer Bekämpfung.

Variation von Systemtyp und Funktionsfähigkeit: In der Studie wurden zwei unabhängige Variablen manipuliert: Der Systemtyp des Entscheidungsunterstützungssystems sowie die Funktionsfähigkeit dieses Systems. Der Systemtyp variierte zwischen den Missionen: In einer Mission wurden die Probanden vom Kritiksystem und in der anderen Mission vom Vorschlagssystem unterstützt. Die Funktionsfähigkeit wurde innerhalb einer Mission variiert, indem die Systeme bei vordefinierten Zielobjekten Fehler machten.

Datenerhebung und -analyse: Als Performance-Maße wurden die Anzahl bekämpfter Cluster sowie die Anzahl von inkorrekt parametrisierten Aufträgen erfasst. Außerdem wurde die Interaktionsdauer zur Parametrisierung gemessen. Nach jeder Mission wurden mithilfe von Fragebögen die Beanspruchung der Probanden, die Benutzerfreundlichkeit des Systems, das Vertrauen in das System sowie der Umgang mit Systemfehlern durch die Probanden erfasst.

⁷¹ Diese Studie steht im Zusammenhang mit einer Veröffentlichung (Dudek & Schulte, 2024a).

Studienergebnisse

Missionsperformance: Die Performance war mit dem Vorschlagssystem höher als mit dem Kritikersystem. Mit dem Vorschlagssystem bekämpften Probanden mehr Cluster bis zum Missionsende und sie übermittelten weniger inkorrekt parametrisierte Aufträge.

Interaktionsdauer: Mit dem Vorschlagssystem war die Interaktionsdauer geringer als mit dem Kritikersystem. Die Interaktionsdauern waren erhöht, wenn ein Fehlerfall des Systems vorlag, wovon das Kritikersystem stärker betroffen war.

Beanspruchung: Die Beanspruchung lag in beiden Missionen im niedrigen bis mittleren Bereich und war beim Kritikersystem höher als beim Vorschlagssystem.

Benutzerfreundlichkeit: Die Benutzerfreundlichkeit wurde bei beiden Systemen positiv bewertet. Das Vorschlagssystem wurde etwas besser bewertet als das Kritikersystem.

Vertrauen in das System: Beide Systeme wurden hinsichtlich Zuverlässigkeit und Vertrauen moderat bewertet. Das Vorschlagssystem wurde jedoch als etwas zuverlässiger und vertrauenswürdiger als das Kritikersystem eingeschätzt (trotz gleicher Fehlerquote).

Umgang mit Systemfehlern: Fehler im Kritikersystem wurden deutlicher wahrgenommen, allerdings führten diese häufiger zu Verwirrung und wurden als schwerer zu beheben eingeschätzt als Fehler des Vorschlagssystems.

Missbräuchliche Verwendung des Kritikersystems: Es kam mehrfach vor, dass Probanden absichtlich falsche Konfigurationen übermittelten, um eine Kritikmeldung zu erhalten und die darin enthaltene Lösung einstellen zu können. Dies führte im Falle von Systemfehlern zu inkorrekt parametrisierten Aufträgen.

Schlussfolgerung

Vorschlagssystem überlegen: Im Versuch erzielten die Probanden mit dem Vorschlagssystem eine höhere Gesamtperformance. Dieses Ergebnis wurde vermutlich begünstigt durch (1) kürzere Interaktionszeiten, (2) visuelles Feedback, das zu einer Begrenzung des Bias beitrug und (3) die missbräuchliche Nutzung des Kritikersystems durch einige Probanden.

Potenzial des Kritikersystems: Die Ergebnisse deuten an, dass kritikbasierte Systemansätze die Fehlerwahrnehmung verbessern können und daher als Alternative zu vorschlagsbasierten Systemen in Betracht gezogen werden sollten – insbesondere bei seltenen und wichtigen Entscheidungen. Bei häufigen und weniger kritischen Entscheidungen erscheint hingegen der Einsatz vorschlagsbasierter Systeme sinnvoller, um von kurzen Interaktionszeiten zu profitieren. Um die Effektivität kritikbasierter Systeme zu steigern, sollte die Interaktionszeit optimiert werden und Schutzmaßnahmen gegen Missbrauch ergriffen werden.

Weitere Forschung: Zukünftige Studien sollten die Vor- und Nachteile beider Ansätze weiter analysieren und Richtlinien zur Auswahl von Entscheidungshilfen entwickeln. Dabei sollte auch untersucht werden, ob Vorschlagssysteme ohne Feedback einem stärkeren Bias unterliegen und wie sich eine Zweckentfremdung des Kritikersystems verhindern lässt.

5.5.1 Studiendesign

Dieser Unterabschnitt beschreibt das methodische Vorgehen der Studie.

5.5.1.1 Aufgabenstellung

In der Studie führten die Probanden zwei Missionen durch, in denen Sie Bekämpfungsaufträge erteilen und konfigurieren mussten, um Gruppen feindlicher Objekte zu bekämpfen.

Auf der taktischen Karte erschienen in einer Mission alle 20 Sekunden Gruppen von Objekten, sogenannte *Cluster*. Jeder Cluster enthielt ein feindliches *Primärziel*, umgeben von einer Mischung aus feindlichen und neutralen Objekten (Abbildung 102 A). Das Ziel der Probanden bestand darin, alle feindlichen Objekte in einem Cluster zu zerstören, ohne dabei ein neutrales Objekt zu beschädigen. Um dieses Ziel zu erreichen konnten die Probanden Aufträge zur Bekämpfung des Primärziels erteilen. Dabei konnten sie die Angriffsachse (in 15-Grad-Schritten) und den Zünder (Aufschlag- oder Verzögerungszünder) parametrisieren, um die Wirkung des Angriffs zu kontrollieren. Dieser Schritt der Parametrisierung wurde durch ein Entscheidungsunterstützungssystem unterstützt. Zusätzlich erhielten die Probanden ein visuelles Feedback über das erwartete Schadensausmaß (Abbildung 102 B). Den Probanden war es nicht gestattet, Bekämpfungsaufträge auf andere Objekte als das Primärziel zu erteilen. Selbst andere feindliche Objekte durften nicht direkt angegriffen werden, sondern konnten lediglich im Zuge der Bekämpfung des Primärziels mit zerstört werden.

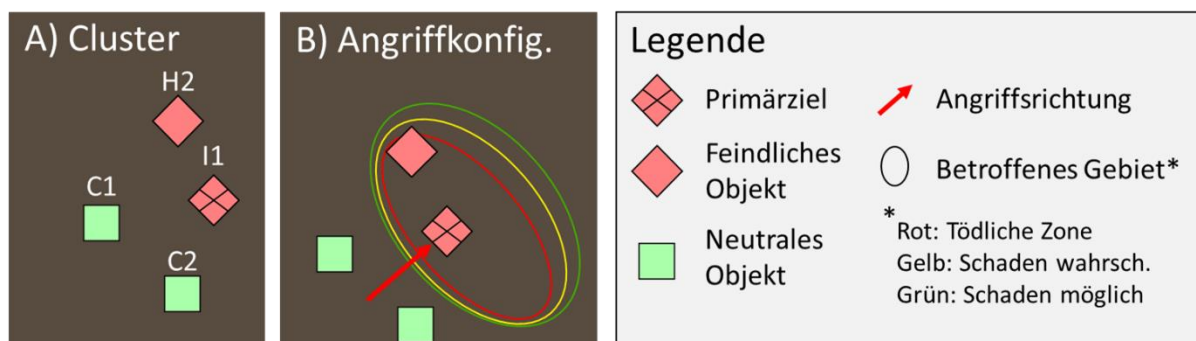


Abbildung 102. A) Beispiel-Cluster B) Geeignete Angriffskonfiguration mit Visualisierung des Bekämpfungseffekts; Rechts: Legende.

Die Missionen enthielten 24 pseudo-zufällig generierte Cluster, die jeweils einem von vier Typen mit unterschiedlichen Konfigurationsschwierigkeiten zuzuordnen waren (Tabelle 33).

Tabelle 33. Beschreibung der Cluster-Typen.

Cluster-Typ	Zusammensetzung	Anteil optimaler Konfigurationen (%)	Anzahl je Mission
Einfach	Nur feindliche Objekte	63 – 93	6
Mittel	Feindliche und neutrale Objekte	10 – 15	6
Schwer		2 – 4	8
Unmöglich		0	4

5.5.1.2 Variation von Systemtyp und Funktionsfähigkeit

In der Studie wurden zwei unabhängige Variablen manipuliert: Systemtyp und Funktionsfähigkeit.

Systemtyp

Die Art der Entscheidungshilfe variierte zwischen den Missionen: in der einen Mission war sie kritikbasiert (sogenanntes Kritiksysteem), in der anderen vorschlagsbasiert (sogenanntes Vorschlagssystem). In Missionen mit dem Kritiksysteem parametrisierten die Probanden jeden Auftrag manuell, ausgehend von einem Standard-Parametersatz. Wenn das System bei einer eingereichten Lösung Verbesserungspotenzial erkannte (Abbildung 103 A), benachrichtigte es den Probanden darüber (Abbildung 103 B). Die Probanden konnten dann wählen, ob sie die Lösung des Systems übernehmen oder ihre eigene Lösung erzwingen wollten. In Missionen mit dem Vorschlagssystem wurde jeder Bekämpfungsauftrag vom System vorkonfiguriert und die Probanden konnten die Parameter nach Belieben anpassen (Abbildung 103 C).

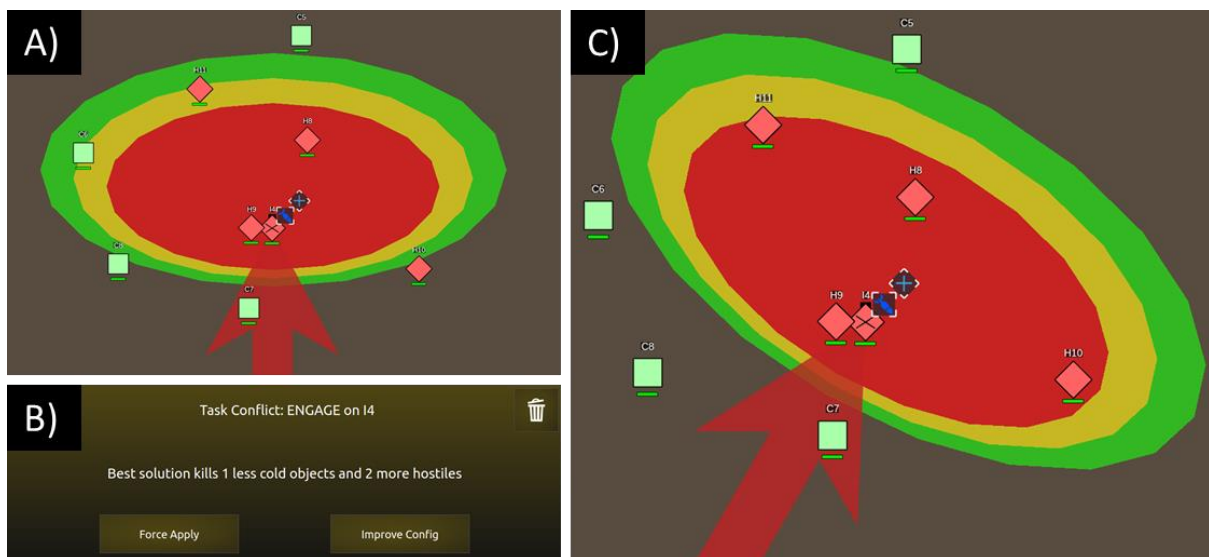


Abbildung 103. A) Suboptimale Lösung B) Kritik-Benachrichtigung C) Vorkonfigurierte Lösung des Vorschlagssystems.

Funktionsfähigkeit

Die Funktionsfähigkeit der Systeme wurde bei jeweils vier der insgesamt 24 Ziele pro Mission (ca. 17 %) absichtlich beeinträchtigt. Tabelle 34 beschreibt das Verhalten beider Systeme abhängig von der Funktionsfähigkeit.

Tabelle 34. Verhalten von Kritik- und Vorschlagssystem abhängig von der Funktionsfähigkeit.

Cluster-Typ	Kritiksystem		Vorschlagssystem	
	Funktionsfähig	Nicht funktionsfähig	Funktionsfähig	Nicht funktionsfähig
Einfach	Das System kritisiert den Piloten im Falle einer sub-optimalen Lösung.	System kritisiert immer die Lösung des Piloten. Die Systemlösung trifft ein feindliches Objekt nicht.	Die vorkonfigurierte Lösung ist optimal.	Die vorkonfigurierte Lösung verschont ein feindliches Objekt.
Mittel & Schwer		Das System kritisiert immer die Lösung des Piloten. Die Systemlösung trifft ein neutrales Objekt.		Die vorkonfigurierte Lösung trifft ein neutrales Objekt.
Unmöglich	Das System kritisiert die Pilotenlösung und weist darauf hin, dass es keine Lösung für das Zielobjekt gibt.	Das System kritisiert die Lösung und meldet, dass es eine bessere Lösung gibt. Die Systemlösung trifft auch neutrale Objekte.	Das System meldet, dass die beste Lösung neutrale Objekte gefährdet und konfiguriert diese vor.	Das System konfiguriert die beste Lösung ohne Vorwarnung vor. Diese Lösung trifft neutrale Objekte.

In Summe gab es für jeden Systemtyp 192 Proben (Cluster). Tabelle 35 zeigt, wie sich diese Proben unter den verschiedenen Versuchsbedingungen aufteilen.

Tabelle 35. Anzahl an Proben (mögliche Konfigurationen) für jedes System.

Cluster-Typen	Funktionsfähigkeit	Probenzahl
Einfach bis Schwer	Funktionsfähig	136
	Nicht funktionsfähig	24
Unmöglich	Funktionsfähig	24
	Nicht funktionsfähig	8

5.5.1.3 Datenerhebung und -analyse

Im Rahmen einer Mission wurden folgende Daten gesammelt:

- Missionsperformance: Die Missionsperformance wird mit zwei Messgrößen operationalisiert.
 - Primärziel-Bekämpfungsquote: Das Verhältnis von bekämpften Primärzielen zu bekämpfbaren Primärzielen (unmögliche Cluster ausgeschlossen).
 - Anzahl falscher Konfigurationen: Anzahl der Aufträge, die so konfiguriert sind, dass neutrale Objekte getroffen werden oder Feinde nicht getroffen werden.
- Interaktionsdauer: Zeit, die für die Parametrisierung eines Auftrags benötigt wird.

Alle Messgrößen wurden automatisiert erfasst und verarbeitet.

Nach jeder Mission mussten die Probanden Fragebögen ausfüllen, in denen die folgenden Punkte adressiert wurden:

- Beanspruchung: Bewertung der Beanspruchung unter Verwendung eines NASA-TLX (Hart & Staveland, 1988).
- Benutzerfreundlichkeit: Auswahl von Fragen eines standardisierten Fragebogens zur Nutzerfreundlichkeit (Brooke, 1996), unter Ausschluss von Fragen, welche für den gegebenen Anwendungsfall nicht passend waren.
- Vertrauen in die Automatisierung: Auswahl von Fragen aus einem standardisierten Fragebogen zu "Trust in Automation" (Körber, 2019) mit Fokus auf die Subskalen "Zuverlässigkeit", "Vertrautheit" und "Vertrauen in Automation".
- Umgang mit Systemfehlern: Bewertung der Fehlererkennung und -behebung mit anwendungsfallspezifischen Fragen.

5.5.1.4 Probanden

Der Versuch wurde mit denselben Probanden durchgeführt, die bereits an der Studie „Delegationsmethoden“ (5.4.1.4) teilgenommen hatten.

5.5.1.5 Training

Da diese Studie mit denselben Probanden wie die Studie „Delegationsmethoden“ durchgeführt wurde, waren die Probanden zum Beginn der Studie bereits mit den wesentlichen Funktionen des Simulators vertraut. Im Rahmen einer Einweisung wurden die Probanden zunächst über die Funktionsweise und Bedienung von Vorschlagssystem und Kritiksystem unterrichtet. Außerdem wurden sie darüber aufgeklärt, dass beide Systeme Fehler verursachen können und sie selbst für die Richtigkeit der ausgeführten Aufgaben verantwortlich sind. Nach der Einweisung absolvierte jeder Proband eine 25-minütige Trainingsmission, die die manuelle Konfiguration von Angriffsaufträgen, das beabsichtigte Herbeiführen von Kritik sowie die Vorkonfiguration von Angriffsaufträgen mit dem Vorschlagssystem umfasste. Im Training kamen keine Systemfehler vor.

5.5.1.6 Versuchsaufbau

Der Versuch wurde im Kampfflugzeugsimulator durchgeführt, dessen Konfiguration im Wesentlichen der Beschreibung in Kapitel 4 entspricht.

In Abweichung davon wurden einzelne Funktionalitäten des Simulators angepasst, um die Versuchsbedingungen zu kontrollieren und eine bessere Vergleichbarkeit zu gewährleisten. Dazu zählen:

- Deaktivierung von nicht verwendeten Delegationsmethoden. Nur die für den Versuch relevante Delegationsmethode Timeline blieb verfügbar.
- Probanden hatten keinen Einfluss darauf, welches UAV die Bekämpfung durchführte.
- Wenn ein UAV alle erteilten Aufträge bearbeitet hatte, positionierte es sich in der Nähe des zuletzt erschienenen Clusters.
- Das bemannte Kampfflugzeug wurde so konfiguriert, dass es automatisiert eine Warteschleife abflog, sodass Probanden ihr eigenes Flugzeug nicht führen mussten.

Die Displays wurden so konfiguriert, dass auf dem MHDD die taktische Karte angezeigt wurde, auf dem linken SHDD wurde der Missionsplan dargestellt und auf dem rechten SHDD wurde die Bewaffnung der Teammitglieder angezeigt (Abbildung 104). Vor dem Start jeder Mission durften die Probanden die taktische Karte so konfigurieren, dass alle relevanten Missionsobjekte gut sichtbar waren.

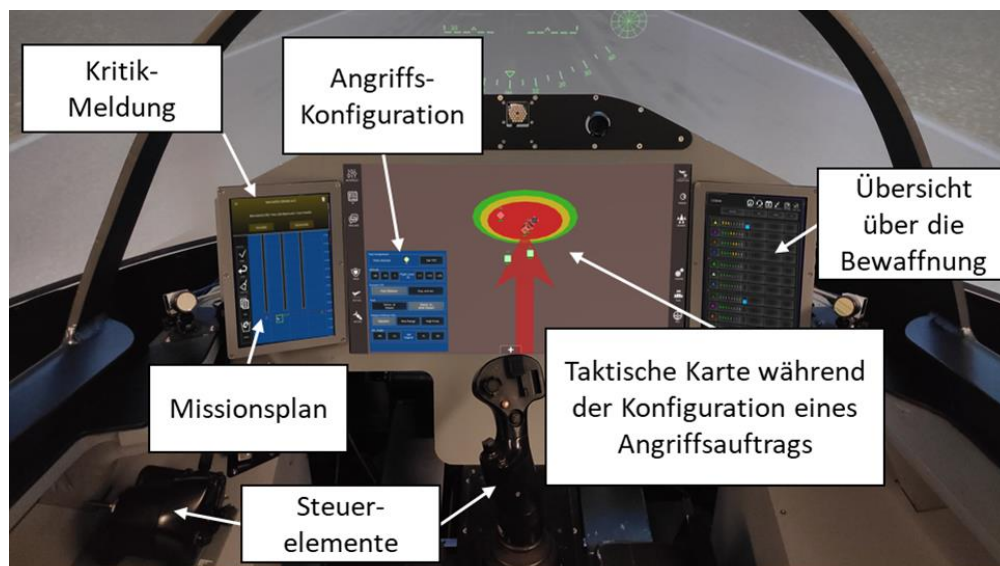


Abbildung 104. Simulatorkonfiguration für die Studie „Kritikfähigkeit in Auftragsdelegation“.

5.5.2 Ergebnisse

Dieser Unterabschnitt stellt die Ergebnisse der Studie dar.

5.5.2.1 Missionsperformance

Primärziel-Bekämpfungsquote

Die Probanden konnten mit dem Vorschlagssystem mehr Ziele bekämpfen, ($M = 93,8\%$; $SD = 4,1\%$) als mit dem Kritiksystem (Abbildung 105; $M = 89,4\%$; $SD = 4,5\%$). Bei der Interpretation der Grafik ist zu beachten, dass der Mittelwert dargestellt ist, da er aufgrund der diskreten Verteilung der Datenpunkte und der vergleichsweise geringen Stichprobengröße (acht Proben je Bedingung) als das aussagekräftigere Maß erschien.

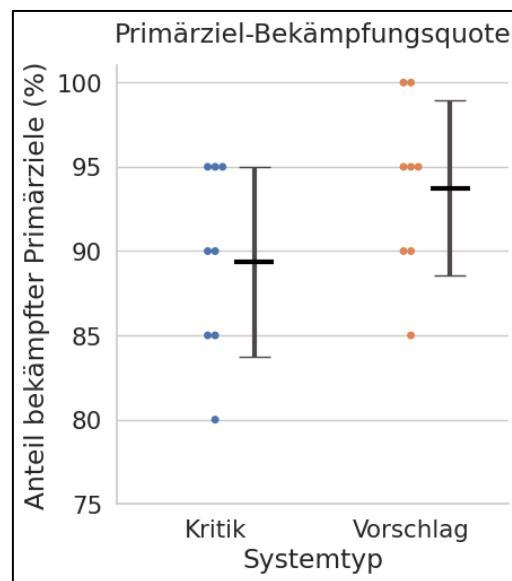


Abbildung 105. Primärziel-Bekämpfungsquote nach Systemtyp.

Anzahl falscher Konfigurationen

Die Probanden übermittelten etwas mehr falsch konfigurierte Aufträge mit dem Kritiksystem (5 von 160 bekämpfbaren Clustern) als mit dem Vorschlagssystem (4 von 160). Die Anzahl falsch konfigurierter Aufträge variierte mit der Funktionsfähigkeit des Systems, wie in Tabelle 36 dargestellt. Wenn die Systemlösung richtig war, machten die Probanden mehr Fehler mit dem Vorschlagssystem, während sie mehr Fehler mit dem Kritiksystem machten, wenn die Systemlösung falsch war.

Tabelle 36. Anzahl falscher Konfigurationen nach Systemtyp und Funktionsfähigkeit.

Funktionsfähigkeit	System-Typ	
	Kritiksystem	Vorschlagssystem
Funktionsfähig	2	3
Nicht funktionsfähig	3	1

Bei der Detail-Analyse dieser Fehler wurden folgende Ursachen festgestellt:

- Wenn das Vorschlagssystem verwendet wurde, war die einzige Fehlerquelle, dass die Probanden die vorkonfigurierte Lösung angepasst und dabei die Auswirkungen der Bekämpfung nicht richtig eingeschätzt haben. In zwei Fällen wählten die Probanden eine ungeeignete Zoomstufe, um alle Objekte um das Ziel herum anzuzeigen, und in zwei weiteren Fällen schätzten die Probanden die Position eines Objekts falsch ein.
- Bei Verwendung des Kritiksystems traten drei Fehler auf, weil die Probanden die Lösung des Systems blind übernommen haben (Das System war in diesen Fällen nicht funktionsfähig). In einem dieser Fälle gab ein Proband zuvor absichtlich eine falsche Konfiguration ein, um eine Lösung vom Kritiksystem zu erhalten. Ein weiterer Fehler trat auf, weil ein Proband eine angemessene Kritik ignorierte, und ein Fehler resultierte aus einem unsachgemäßen Umgang mit der Kritikbenachrichtigung.

5.5.2.2 Interaktionsdauer

Die Probanden parametrisierten Aufträge schneller in Missionen mit dem Vorschlagssystem ($MD_{Vorschlag} = 6,3$ s; $MD_{Kritik} = 10,6$ s). Dies passt zu den Ergebnissen der Primärziel-Bekämpfungsquote, da Probanden mehr Ziele angreifen konnten, wenn sie weniger Zeit für ein einzelnes Ziel aufwenden mussten. Erwartungsgemäß hatten beide Systeme eine längere Interaktionsdauer, wenn die Systemlösung falsch war, wobei das Kritiksystem stärker betroffen war (Abbildung 106). Entgegen den Erwartungen werden die niedrigsten Interaktionsdauern mit dem Kritiksystem (min = 1,1 s) und nicht mit dem Vorschlagssystem (min = 2,2 s) erreicht. Diese Interaktionsdauern wurden erreicht, wenn die Standardkonfiguration bereits optimal war, was darauf hindeutet, dass Probanden mehr Zeit für die Analyse der Auswirkungen einer Lösung aufwandten, wenn diese durch das Vorschlagssystem generiert wurde.

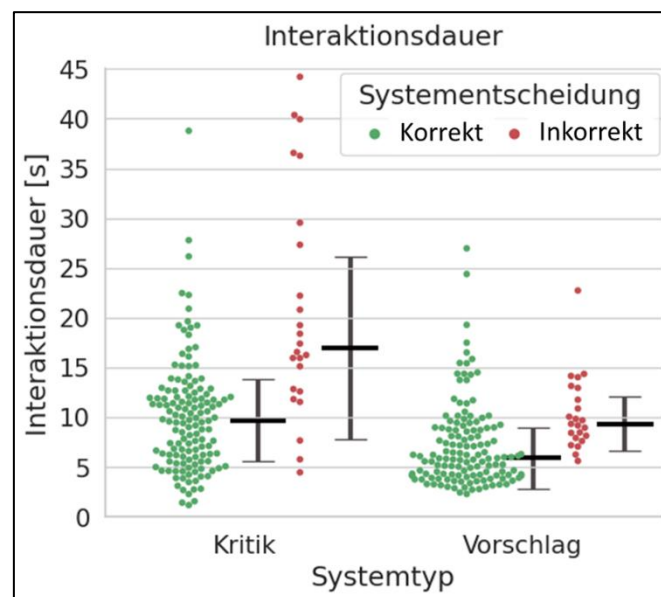


Abbildung 106. Interaktionsdauer nach Systemtyp und Funktionsfähigkeit.

5.5.2.3 Beanspruchung

Die subjektive Beanspruchung lag bei beiden Systemen im niedrigen bis mittleren Bereich (Abbildung 107). Die Probanden hatten eine höhere Beanspruchung bei der Verwendung des Kritiksystems als bei der Verwendung des Vorschlagssystems.

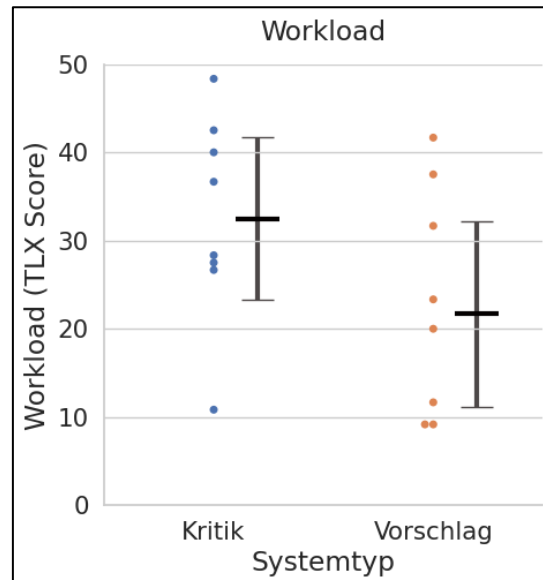


Abbildung 107. Beanspruchung nach Systemtyp.

5.5.2.4 Benutzerfreundlichkeit

Die Probanden bewerteten die Benutzerfreundlichkeit beider Systeme positiv, wobei das Vorschlagssystem etwas besser bewertet wurde (Abbildung 108).

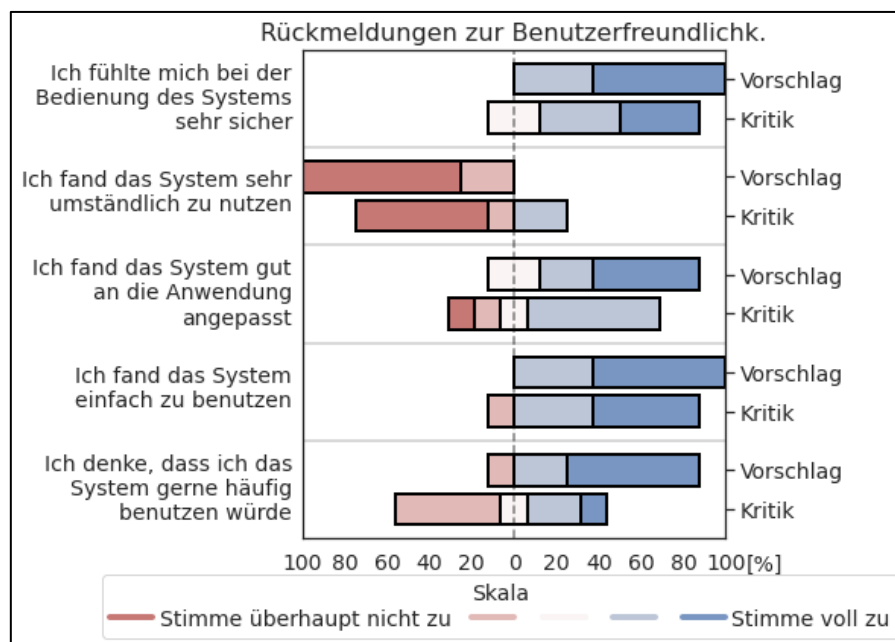


Abbildung 108. Rückmeldungen der Probanden zur Benutzerfreundlichkeit der Systeme.

5.5.2.5 Vertrauen in Automation

Obwohl beide Systeme mit 4 von 24 Zielen eine vergleichsweise hohe Fehlerquote aufwiesen, hielten die Probanden sie dennoch für einigermaßen zuverlässig (Abbildung 109). Dabei wurde das Vorschlagssystem als zuverlässiger eingeschätzt als das Kritiksysteem.

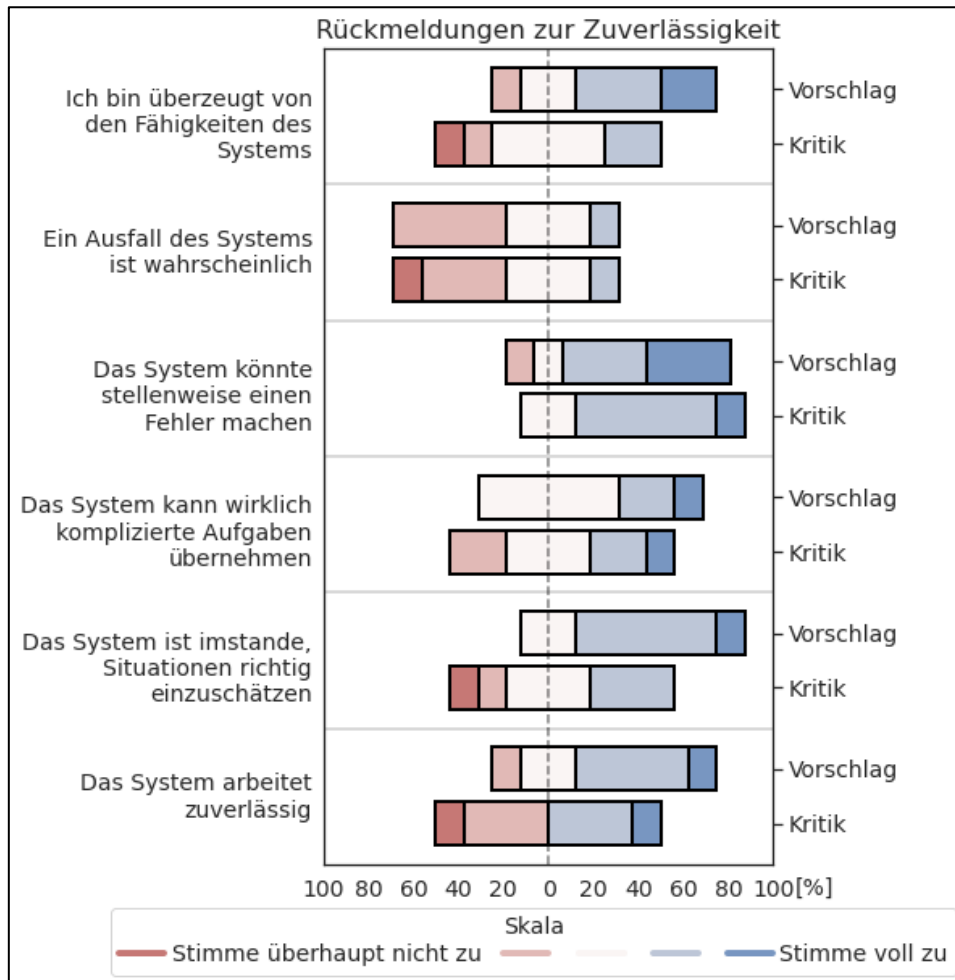


Abbildung 109. Rückmeldungen zur Zuverlässigkeit der Systeme.

Die Probanden waren vor der Studie mit beiden Systemtypen nicht vertraut (Abbildung 110).

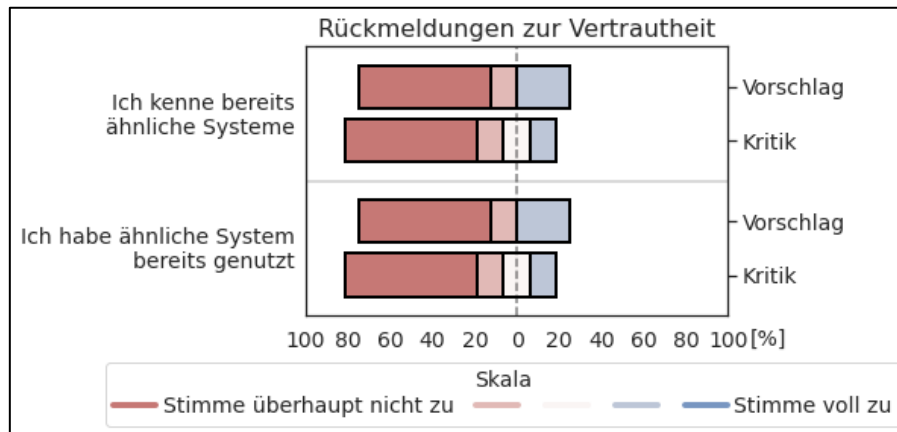


Abbildung 110. Rückmeldungen zur Vertrautheit mit den Systemen.

Insgesamt zeigten die Probanden mäßiges Vertrauen in beide Systeme, mit leichten individuellen Unterschieden (Abbildung 111). Dem Vorschlagssystem wurde etwas mehr Vertrauen entgegengebracht als dem Kritiksistem.

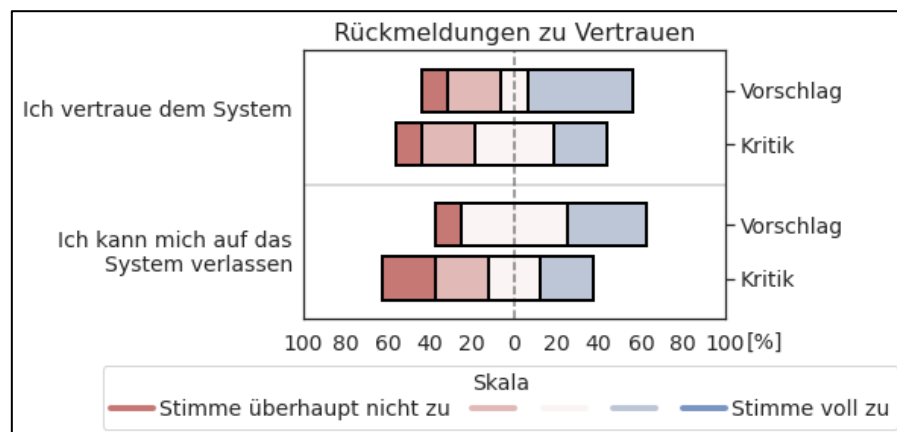


Abbildung 111. Rückmeldung zum Vertrauen in die Systeme.

5.5.2.6 Umgang mit Systemfehlern

Gemäß den Probanden waren Fehler des Kritiksystems offensichtlicher als Fehler des Vorschlagssystems, führten jedoch gleichzeitig zu mehr Verwirrung (Abbildung 112). Auch die Behebung von Fehlern gelang beim Vorschlagssystem nach Einschätzung der Probanden etwas schneller.

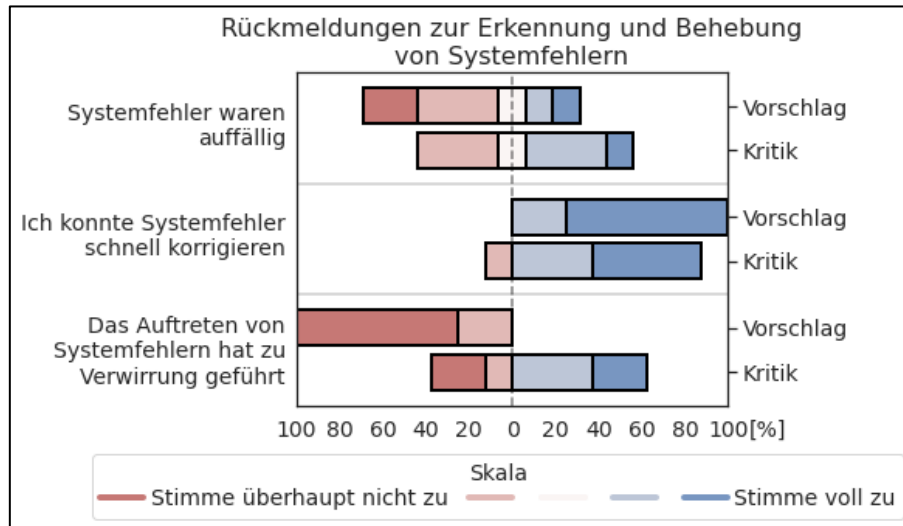


Abbildung 112. Rückmeldungen zur Fehlererkennung und -behebung.

5.5.2.7 Absichtliche Aktivierung des Kritiksystems

Mithilfe einer tiefergehenden Datenanalyse sollten weitere Erkenntnisse gewonnen werden, ob es häufiger vorkam, dass Probanden das Kritiksystem absichtlich auslösten. Das Kritiksystem wurde insgesamt 56 Mal ausgelöst, wobei 41 Aktivierungen durch geplante Fehler des Kritiksystems verursacht wurden (die Probanden waren für diese Aktivierungen also nicht verantwortlich). In den anderen 15 Fällen wurde das Kritiksystem ausgelöst, weil die Probanden eine falsche Konfiguration abgeschickt hatten (basierend auf 160 Primärzielen).

Die Analyse der Replay-Daten zeigt, dass das Kritiksystem in acht Fällen absichtlich ausgelöst wurde. In vier von diesen Fällen bestätigten die Probanden ihre Konfiguration, obwohl deutlich erkennbar war, dass neutrale Objekte von der Bekämpfung betroffen gewesen wären. In den anderen vier Fällen gingen die Probanden einen Schritt weiter und akzeptierten bereits die Standardkonfiguration, obwohl sie falsch war. Neben diesen bewussten Aktivierungen des Kritiksystems wurde das System in sieben Fällen aufgrund von unbeabsichtigten Fehlern in der Konfiguration aktiviert.

5.5.3 Diskussion

In diesem Unterabschnitt werden die vorgestellten Einzelergebnisse interpretiert und zusammengeführt.

5.5.3.1 Höhere Performance mit Vorschlagssystem

Die erste Hypothese, dass Probanden mit dem Vorschlagssystem eine höhere Performance erzielen, wenn beide Systeme richtig funktionieren, kann als richtig angesehen werden.

Die kürzere Interaktionsdauer war vermutlich der Hauptgrund dafür, dass die Probanden mit dem Vorschlagssystem mehr Primärziele bekämpfen konnten. Die Unterschiede in der Interaktionsdauer lassen sich auf eine höhere Anzahl an Interaktionen zurückführen, die bei der Verwendung des Kritiksystems erforderlich sind. Wenn beide Systeme einwandfrei funktionierten, mussten die Probanden beim Vorschlagssystem lediglich die ausgewählten Parameter bestätigen, während beim Kritiksystem zusätzlich ein passendes Parameterset gefunden und eingegeben werden musste. Im Falle von Systemfehlern konnten Benutzer des Vorschlagssystems die Parameter einfach anpassen, während Benutzer des Kritiksystems die vom System vorgeschlagene Lösung übernehmen, bewerten und gegebenenfalls korrigieren mussten, nachdem sie bereits ihre eigene Lösung eingegeben hatten (unter der Annahme, dass sie der Kritik vertrauten). Die einfachere Interaktion mit dem Vorschlagssystem könnte auch der Grund dafür sein, dass die Probanden eine geringere Beanspruchung angaben und die Benutzerfreundlichkeit des Systems positiver bewerteten.

Ein Argument gegen die erste Hypothese ist, dass die Probanden in Bedingungen mit funktionsfähigem System etwas mehr Fehler mit dem Vorschlagssystem machten, als mit dem Kritiksystem. Dieser Aspekt ist jedoch aufgrund der im Vergleich zum Stichprobenumfang geringen Anzahl von Fehlern nicht sehr aussagekräftig (ein zusätzlicher Fehler bei 136 Stichproben für jedes System).⁷²

5.5.3.2 Umgang mit Systemfehlern

Die zweite Hypothese, dass Probanden bei Verwendung des Kritiksystems effektiver mit Fehlern umgehen, wird als nicht richtig angesehen.

Die Probanden übermittelten etwas mehr inkorrekt parametrisierte Aufträge, wenn sie ein nicht funktionierendes Kritiksystem verwendeten, als wenn sie ein nicht funktionierendes Vorschlagssystem verwendeten. Für diese Ergebnisse entgegen der Erwartungen gibt es drei mögliche Erklärungen.

Eine Erklärung liegt in der höheren Interaktionsdauer mit dem Kritiksystem begründet. Diese höhere Interaktionsdauer führte auf Dauer wahrscheinlich zu einem deutlich höheren Zeitdruck, weil in regelmäßigen Abständen neue Ziele auf der Karte erschienen. Da aus Studien bekannt ist, dass die Qualität von Entscheidungen abhängig von der zur Verfügung stehenden Zeit ist (sogenannter *Speed-Accuracy Tradeoff*; Larson & Hawkins (2023)), ist es

⁷² Diese Stichprobenzahl bezieht sich auf die Versuchsbedingungen, in denen das Cluster bekämpfbar war und das System fehlerfrei funktionierte.

wahrscheinlich, dass dieser erhöhte Zeitdruck dazu beigetragen hat, dass Probanden Systemfehler weniger gut wahrnehmen und beheben konnten.

Eine weitere mögliche Erklärung ist, dass die Visualisierung des erwarteten Schadensausmaßes in der taktischen Karte den Probanden geholfen hat, die Auswirkungen eines Parametersatzes einzuschätzen, wodurch der Bias gegenüber Systemlösungen verringert wurde. Gemäß der ursprünglichen Annahme, dass der Decision Bias beim Vorschlagssystem prinzipiell ausgeprägter ist, dürfte von einem solchen Feedback das Vorschlagssystem stärker profitiert haben als das Kritiksystem. Die Bedeutung des visuellen Feedbacks bei der Fehlervermeidung wird durch die Tatsache gestützt, dass einige inkorrekt konfigurierten Aufträge übermittelt wurden, nachdem die Probanden eine Zoomstufe verwendeten, die die Visualisierung nicht vollständig anzeigte. Die Visualisierung könnte deshalb hilfreich gewesen sein, weil sie die Auswirkungen von Entscheidungen unmittelbar und klar darstellt (Renaud & Cooper, 2000).

Zu Fehlern des Kritiksystems hat auch eine Zweckentfremdung des Systems beigetragen. So übermittelten einzelne Probanden absichtlich falsche Konfigurationen, um eine Lösung des Kritiksystems zu erhalten und sie übernahmen diese Systemlösung teils ungeprüft, was im Falle eines Systemfehlers bedeutete, dass die final übermittelte Lösung ebenfalls fehlerhaft war. Neben der Tatsache, dass ein Fehler dokumentiert ist, der auf diese Weise zustande kam, gibt es keinen plausiblen Grund, warum ein Nutzer in solchen Fällen nicht ebenfalls zu Bias neigen sollte. Die absichtliche Verwendung des Kritiksystems könnte über den Bias hinaus problematisch sein, da ein Kritiksystem auch nur als zusätzliche Sicherheitsebene konzipiert werden könnte, wodurch das System möglicherweise nicht alle Fehlerfälle überprüfen kann oder mit geringeren Zuverlässigkeitsanforderungen einhergeht.

Auch wenn die Fehlerquote mit dem Kritiksystem nicht geringer war, so deuten zumindest die subjektiven Einschätzungen darauf hin, dass Fehler eines Kritiksystems eher wahrgenommen werden. Dies zeigt sich beispielsweise darin, dass die Probanden dem Kritiksystem eine geringere Zuverlässigkeit zuschrieben und weniger Vertrauen in das System hatten, obwohl beide Systeme ähnlich zuverlässig waren. Außerdem bewerteten die Probanden die Fehler des Kritiksystems als offensichtlicher.

5.5.4 Fazit

Das Vorschlagssystem war für die Versuchsaufgabe besser geeignet als das Kritiksystem. Die Probanden zeigten eine höhere Performance und hatten eine geringere Beanspruchung, wenn sie das Vorschlagssystem verwendeten. Außerdem bewerteten die Probanden die Benutzerfreundlichkeit positiver.

Dieses Ergebnis lässt sich vermutlich primär auf kürzere Interaktionszeiten mit dem Vorschlagssystem zurückführen. Diese entstanden dadurch, dass die Probanden sowohl bei korrektem Systemverhalten als auch im Falle von Systemfehlern weniger Schritte benötigten, um zur richtigen Lösung zu gelangen.

Entgegen den Erwartungen zeigten die Probanden kein effektiveres Fehlermanagement bei der Nutzung des Kritiksystems. Zwar deuten die subjektiven Einschätzungen der

Probanden auf eine bessere Wahrnehmung von Systemfehlern hin, allerdings war die Anzahl der übermittelten Fehlkonfigurationen mit dem Kritikersystem minimal höher.

Es wurden verschiedene Effekte identifiziert, die wahrscheinlich eine Rolle dabei spielten, dass die Fehlerquote mit dem Kritikersystem nicht geringer ausgefallen ist als mit dem Vorschlagssystem. Ein entscheidender Faktor könnte gewesen sein, dass die höheren Interaktionszeiten auf Dauer zu einem höheren Zeitdruck geführt haben, was sich dann schlussendlich negativ auf die Wahrnehmung und Behebung von Systemfehlern ausgewirkt haben könnte. Auch das visuelle Feedback könnte einen Beitrag dazu geleistet haben, dass Probanden systemische Lösungen effektiver bewerten konnten, wovon, aufgrund des angenommenen höheren Bias, insbesondere das Vorschlagssystem profitiert haben dürfte. Schließlich hat auch die bewusste Aktivierung des Kritikersystems durch einzelne Probanden zu einer höheren Fehlerquote des Kritikersystems beigetragen.

Diese Versuchsergebnisse zeigen, dass im Hinblick auf die Fragestellung, ob Systemfehler mit dem Kritikersystem besser erkannt und behoben werden können, eine differenzierte Betrachtung notwendig ist. So sprechen die subjektiven Einschätzungen der Probanden für eine bessere Fehlerwahrnehmung mit dem Kritikersystem, während die objektiven Fehlerquoten diese These nicht stützen. Allerdings ist davon auszugehen, dass das Kritikersystem besser abgeschnitten hätte, wenn sich die zuvor genannten Effekte nicht negativ auf das Kritikersystem bzw. positiv auf das Vorschlagssystem ausgewirkt hätten. Diese Ergebnisse legen nahe, dass der Kritikansatz tatsächlich zu einer besseren Fehlerwahrnehmung führen könnte, sofern: (1) die höheren Interaktionszeiten nicht letztendlich zu mehr Zeitdruck führen und (2) eine Zweckentfremdung des Kritikersystems verhindert wird.

Eine mögliche Schlussfolgerung aus den Ergebnissen wäre daher, vorschlagsbasierte Systeme vor allem bei häufigen und unwichtigen Entscheidungen einzusetzen, um von kürzeren Interaktionszeiten zu profitieren. Bei seltenen und wichtigen Entscheidungen könnten hingegen kritikbasierte Systeme besser geeignet sein, da sie tendenziell eine bessere Fehlerwahrnehmung ermöglichen.

Um die Unterschiede in den Interaktionszeiten zu verringern, sollten Änderungen an der Ausgestaltung des Kritikersystems erwogen werden. Zwar ist zu erwarten, dass die Interaktion mit einem kritikbasierten System grundsätzlich mehr Zeit in Anspruch nimmt als mit einem vorschlagsbasierten System, dennoch gibt es Möglichkeiten, die Interaktionsdauer zu verkürzen. Ein Ansatz besteht beispielsweise darin, die Kritikmeldungen so zu gestalten, dass die vom Nutzer gewählte Lösung und die Systemlösung direkt nebeneinander dargestellt werden, wobei die erwarteten Vorteile der Systemlösung hervorgehoben werden. Dies könnte zu kürzeren Interaktionszeiten im Fall von Kritik führen, weil Nutzer die Alternativen leichter bewerten können.

Aufgrund der vermuteten positiven Effekte des visuellen Feedbacks wird empfohlen, Feedback-Maßnahmen vorzusehen, welche die Auswirkungen von Entscheidungen klar und unmittelbar darstellen. Allerdings eignet sich nicht jede Anwendung für solche Feedback-

Maßnahmen, da die Folgen einer Entscheidung nicht immer eindeutig vorhergesagt werden können.

Um die Auswirkungen einer Zweckentfremdung des Kritiksystems zu begrenzen, können verschiedene Maßnahmen erwogen werden. Eine Möglichkeit wäre, dass das Kritiksystem dem Nutzer keine Möglichkeit bietet, die Systemlösung direkt zu übernehmen. Wenn das Kritiksystem diese Funktion bereitstellen soll, könnte die oben diskutierte Gegenüberstellung von Benutzer- und Systemlösung hilfreich sein, da sie es den Nutzern erleichtern würde, Systemfehler zu erkennen. Andere Maßnahmen könnten sich darauf konzentrieren, Nutzer von einer Zweckentfremdung des Kritiksystems abzuhalten, zum Beispiel, indem jede Aktivierung des Kritiksystems im Rahmen einer Nachbesprechung kritisch diskutiert wird.

Weitere Untersuchungen sollten sich vorrangig darauf konzentrieren, die Vor- und Nachteile von kritikbasierten und vorschlagsbasierten Entscheidungshilfen genauer zu analysieren und klare Richtlinien dafür zu entwickeln, in welchen Anwendungsfällen welche Form der Entscheidungshilfe geeigneter ist. In diesem Kontext sollte auch untersucht werden, ob der Bias gegenüber einer vorschlagsbasierten Entscheidungshilfe ausgeprägter ist, wenn die Auswirkungen von Entscheidungen nicht durch eine Feedback-Maßnahme verdeutlicht werden. Ferner sollte untersucht werden, welche Faktoren zur Zweckentfremdung des Kritiksystems beitragen und wie dies verhindert werden kann.

6. Zusammenfassung und Ausblick

Dieses Kapitel fasst die wichtigsten Ergebnisse der vorliegenden Forschungsarbeit zusammen und gibt einen Ausblick auf weitere Forschungsfragen, die sich aus der Arbeit ergeben. Zu diesem Zweck werden in Abschnitt 6.1 die **Forschungsbeiträge** dieser Arbeit dargestellt. Anschließend wird die vorliegende Arbeit in Abschnitt 6.2 kritisch reflektiert, um die **Limitationen** dieser Forschungsbeiträge aufzuzeigen. Abschnitt 6.3 schließt das Kapitel ab, indem es den weiteren **Forschungsbedarf** vorstellt, der sich aus den Ergebnissen dieser Arbeit und ihren Limitationen ergibt.

6.1 Beiträge dieser Arbeit

Der Forschungsbeitrag dieser Arbeit liegt in mehreren Themenfeldern im Bereich der Mensch-Maschine-Interaktion. Nachfolgend werden die verschiedenen Beiträge in den jeweiligen Themenfeldern genauer erläutert.

Führungskonzept für Kampfflugzeugmissionen

Die Ergebnisse der Arbeit unterstreichen die Eignung eines auftragsbasierten Führungskonzeptes zur Führung mehrerer unbemannter Plattformen aus dem Kampfflugzeugcockpit. Das Konzept der auftragsbasierten Führung wurde zwar bereits im Kontext von Transporthubschraubereinsätzen erprobt (Uhrmann & Schulte, 2012), allerdings sind Kampfflugzeugmissionen von einer höheren Dynamik und einem größerem Aufgabenspektrum gekennzeichnet. Die im Rahmen dieser Arbeit durchgeführten Untersuchungen haben gezeigt, dass die auftragsbasierte Führung diesen Anforderungen gerecht werden kann, sofern eine adäquate Auftragsdefinition getroffen wird und die Nutzerschnittstelle entsprechend gestaltet ist.

Die im Rahmen dieser Arbeit durchgeführten Untersuchungen haben ferner gezeigt, dass die Führung mittels Verhaltensrichtlinien eine sinnvolle Ergänzung zur auftragsbasierten Führung darstellt, da sie die Nachteile der auftragsbasierten Führung⁷³ kompensiert und die Abbildung von Einsatzregeln ermöglicht.

Definition von UAV-Aufträgen

Ein weiterer Beitrag dieser Arbeit ist in diesem Kontext die systematische Definition von UAV-Aufträgen und die Differenzierung von Auftragsstypen für militärische Kampfflugzeugmissionen. Die im Rahmen dieser Arbeit entwickelte Auftragsdefinition, bestehend aus Auftragsstyp, Zielobjekt, Erfolgskriterium und Parametern, stellt eine gute Grundlage für ein auftragsbasiertes Führungskonzept dar, da sie einerseits eine feste Struktur von Aufträgen vorgibt und andererseits eine beliebig tiefe Ausdifferenzierung von Aufträgen in Form von Parametern ermöglicht.

Auch wenn die im Rahmen dieser Arbeit festgelegten Auftragsstypen und Parameter bei weitem noch nicht alle Missionstypen und Anforderungen zukünftiger

⁷³ Die auftragsbasierte Führung ermöglicht keine dauerhafte Übermittlung von Nutzerabsichten und ermöglicht nicht festzulegen, was ein UAV ohne Auftrag tun soll.

Kampfflugzeugmissionen abdecken können, zeigt diese Arbeit doch, dass diese Art der Auftragsdefinition geeignet ist, um ein breites Spektrum an Missionsaufgaben abzudecken und die festgelegten Auftragsstypen können darüber hinaus als Anregung für die Festlegung von Auftragsstypen in Manned-Unmanned Teaming dienen.

Interaktionsmethoden zur Auftragserteilung

Ein wichtiger Beitrag dieser Arbeit liegt im Bereich der Interaktionsmethoden zur Auftragserteilung (Delegationsmethoden). Im Rahmen dieser Arbeit wurden verschiedene Interaktionsmethoden entwickelt, die eine Auftragsdelegation unter Nutzung diskreter Touchscreen-Interaktionen, Touchscreen-Gesten, Sprachbefehlen sowie einer Kombination aus Blickbewegungen und manuellen Knöpfen ermöglichen.

Die Methoden wurden im Rahmen mehrerer experimenteller Untersuchungen intensiv erprobt. Die Untersuchungen haben gezeigt, dass keine Methode den anderen eindeutig überlegen ist. Vielmehr werden die Methoden unterschiedlich stark von Störfaktoren wie Nebenaufgaben oder Display-Clutter beeinflusst, wodurch je nach Situation eine Methode geeigneter ist als die anderen. Die Untersuchungen haben zudem ergeben, dass die Nutzer von einer Wahlmöglichkeit zwischen verschiedenen Methoden Gebrauch machen, diese Wahlmöglichkeit aber in zeitkritischen Situationen von den Anwendern nicht immer bedacht wird. Die Ergebnisse der Untersuchung legen den Schluss nahe, dass die Bereitstellung von zwei Interaktionsmethoden ein geeigneter Kompromiss zwischen erhöhter Resilienz, der Überschaubarkeit der Interaktionsmöglichkeiten und dem Ressourcenaufwand für die Entwicklung der Methoden sein könnte.

UAV-Feedback

Im Rahmen der Arbeit wurde außerdem das Thema Feedback im Kontext der auftragsbasierten Führung untersucht. Dabei wurden verschiedene Feedbackmaßnahmen implementiert, die zu einer höheren Transparenz des Agentenverhaltens und einem gesteigerten Situationsbewusstsein des Piloten beitragen sollen, indem sie den Nutzer über die Aktionen & Entscheidungen des Agenten sowie über relevante Zustandsänderungen informieren.

Die verschiedenen Feedbackmaßnahmen wurden anschließend im Rahmen einer experimentellen Untersuchung bewertet. Diese Bewertungen liefern verschiedene Hinweise darauf, welche Informationen von den Nutzern im Rahmen eines Feedbacks gewünscht werden und inwieweit verschiedene Gestaltungsformen geeignet sind.

Kritikbasierte Entscheidungsunterstützung

Die Untersuchung eines kritikbasierten Entscheidungsunterstützungssystems ist ein weiterer Forschungsbeitrag dieser Arbeit. Mit der Auftragskritik wurde im Rahmen dieser Arbeit eine kritikbasierte Entscheidungshilfe entwickelt, die die vom Nutzer delegierten Aufträge analysiert und den Nutzer kritisiert, wenn Auftragsparameter sich negativ auf die Missionsleistung auswirken könnten.

Die Auftragskritik wurde anschließend in einer experimentellen Untersuchung mit einem vorschlagsbasierten System verglichen. Im Vergleich erzielte das vorschlagsbasierte System bessere Ergebnisse, was vor allem auf die kürzeren Interaktionszeiten zurückzuführen sein dürfte. Gleichzeitig legen die Versuchsergebnisse nahe, dass Probanden Fehler des Kritiksystems besser wahrnehmen konnten.

Eine mögliche Schlussfolgerung aus diesem Ergebnis wäre, vorschlagsbasierte Systeme vor allem bei häufigen und unwichtigen Entscheidungen einzusetzen, um von kürzeren Interaktionszeiten zu profitieren. Bei seltenen und wichtigen Entscheidungen könnten hingegen kritikbasierte Systeme besser geeignet sein, da sie tendenziell zu einer besseren Fehlerwahrnehmung beitragen können. Im untersuchten Anwendungsfall würde dies beispielsweise bedeuten, auf den unteren⁷⁴ Entscheidungsebenen eher vorschlagsbasierte Systeme zu verwenden, während auf den oberen⁷⁵ Ebenen eher kritikbasierte Systeme zum Einsatz kommen sollten, weil auf den oberen Ebenen Entscheidungen seltener getroffen werden, ihre Tragweite größer ist, und die Automation tendenziell unzuverlässiger arbeitet.

Ein weiterer Beitrag dieser Arbeit besteht in der Aufdeckung einer potenziellen Schwäche kritikbasierter Systeme: Einzelne Probanden zweckentfremdeten das Kritiksystem, indem sie bewusst Fehler bei der Auftragserteilung machten, um anschließend von der Kritik des Systems zu profitieren. Eine solche Zweckentfremdung könnte die Wirksamkeit kritikbasierter Entscheidungshilfen beeinträchtigen, da die Interaktionszeiten in diesem Fall dennoch länger sind als bei einem vorschlagsbasierten System und gleichzeitig keine Verbesserung der Fehlerwahrnehmung zu erwarten ist. Wenn eine kritikbasierte Form der Entscheidungsunterstützung eingesetzt wird, sollten daher Maßnahmen ergriffen werden, um eine solche Zweckentfremdung zu verhindern.

UAV-Verhaltensgenerierung

Abschließend ist der Beitrag dieser Arbeit zur methodischen Umsetzung einer Verhaltensgenerierung für unbemannte Luftfahrzeuge hervorzuheben. So wurde im Rahmen dieser Arbeit die Methode BehaviorTree eingesetzt, um die Verhaltensweisen der unbemannten Plattformen im Rahmen der auftragsbasierten Führung zu erzeugen.

Dies demonstriert zum einen die Eignung der Methode BehaviorTree für den vorliegenden Anwendungsfall. Zum anderen können die im Rahmen der Implementierung getroffenen Designentscheidungen als Referenz für die Entwicklung eines BehaviorTrees zur Steuerung eines UAVs oder für ähnliche Aufgabenumgebungen dienen.

6.2 Kritische Reflexion

Im Folgenden werden die Vorgehensweise und die Designentscheidungen, die im Verlauf dieser Arbeit getroffen wurden, kritisch reflektiert, um aufzuzeigen, inwieweit die Versuchsergebnisse durch diese Designentscheidungen beeinflusst worden sein könnten, und

⁷⁴ z. B. Parameter eines Auftrags

⁷⁵ z. B. Priorisierung von Zielen

um naheliegende Untersuchungsgegenstände vorzustellen, die im Rahmen dieser Arbeit nicht berücksichtigt wurden.

Alternative Konzepte zur UAV-Führung

Ein Themenfeld, das im Rahmen dieser Arbeit nicht umfassend betrachtet wurde, sind alternative Führungskonzepte, die nicht auf der Erteilung von Aufträgen basieren. Im Konzeptkapitel dieser Arbeit wurde festgelegt, dass das Entwurfsmuster der assistierten auftragsbasierten Führung als Lösung für die Anforderungen des Manned-Unmanned Teaming verwendet werden soll.

Obwohl diese Entscheidung aufgrund der positiven Forschungsergebnisse im Bereich der auftragsbasierten Führung begründet war, gibt es möglicherweise andere Führungskonzepte, die insbesondere für die Führung von sehr vielen Entitäten besser geeignet sind. Einige Ideen für alternative Führungskonzepte werden im nachfolgenden Abschnitt [6.3](#) vorgestellt.

Koordination ohne explizite Mechanismen

Die Koordination der Teammitglieder ist ein weiterer Aspekt, der im Rahmen dieser Arbeit nicht hinreichend betrachtet wurde. So wurden im Rahmen dieser Arbeit keine expliziten Koordinationsmechanismen implementiert, um die Zusammenarbeit zwischen bemannten und unbemannten Plattformen sowie zwischen den unbemannten Plattformen zu regeln. Stattdessen erfolgt die Koordination zwischen den Plattformen des Teams implizit: Einerseits dadurch, dass die Teammitglieder die ihnen im Missionsplan zugewiesenen Aufgaben abarbeiten, und andererseits dadurch, dass die Aktionen eines Teammitglieds Auswirkungen auf den Zustand der Welt haben, was schlussendlich auch zu einer Verhaltensänderung der unbemannten Teammitglieder führen kann.

Diese impliziten Mechanismen haben jedoch verschiedene Nachteile. So sind die Aufträge im Missionsplan auf einem hohen Abstraktionsniveau definiert, was den Teammitgliedern Flexibilität bei der Bearbeitung der Aufträge gibt. Diese Flexibilität ist für die Koordination von Nachteil, da sie Unsicherheit über das Verhalten der Teammitglieder schafft. Die Koordination über den Zustand der Welt hat den Nachteil, dass sich die Aktion einer Plattform erst auf den Zustand der Welt auswirken muss, bevor sie sich auf das Verhalten der Teammitglieder auswirkt. Dies kann zu Synchronisationsproblemen führen, z. B. wenn zwei unbemannte Plattformen gleichzeitig die Entscheidung treffen, einen Gegner zu bekämpfen, weil sie nicht über die Entscheidung des jeweiligen Teammitglieds informiert sind.

Explizite Koordinationsmechanismen könnten Abhilfe gegenüber den genannten Schwachstellen schaffen und zu einer deutlich präziseren Abstimmung zwischen den Teammitgliedern führen.

Modellierung

Kritisch zu hinterfragen sind die Entscheidungen, die hinsichtlich der Modellierung der Simulationsumgebung getroffen wurden. So wurden im Rahmen dieser Arbeit verschiedene Modelle entwickelt, um missionsrelevante Effekte, wie z.B. die Dynamik des unbemannten Luftfahrzeuges oder das Verhalten feindlicher Kräfte, abzubilden. Diese Modelle bilden die

verschiedenen Effekte in vereinfachter Form ab, da eine zu detaillierte Modellierung den Entwicklungsaufwand stark erhöht hätte und sich zudem negativ auf die Kontrollierbarkeit der Versuchsbedingungen ausgewirkt hätte. Sollten die getroffenen Vereinfachungen jedoch einen missionsrelevanten Effekt nicht hinreichend genau modelliert haben, könnte dies die Validität der Versuchsergebnisse in Frage stellen.

In diesem Zusammenhang könnten die Einschätzungen der Probanden in den experimentellen Untersuchungen einen Hinweis darauf geben, an welchen Stellen missionsrelevante Effekte möglicherweise nicht ausreichend abgebildet wurden. Diesbezüglich sahen einzelne Probanden Verbesserungspotential bei der Modellierung der Radardarstellungen, der Kampfflugzeuge, der Missionsszenarien und des taktischen Verhaltens. Insgesamt ergab sich aber eine eher positive Bewertung der Modellierung, was auch durch die Aussage eines Probanden bestätigt wurde:

„Ich denke, es ist sehr schwer ein komplexes Szenario in einer Simulation nachzubilden – speziell mit so vielen Aspekten wie hier. In Summe ist das gut gelungen, auf alle Fälle ausreichend“

Stichprobengröße der experimentellen Untersuchungen

Eine Limitation dieser Arbeit liegt in der Stichprobengröße im Rahmen der experimentellen Erprobung des entwickelten Funktionsprototypen. So wurden die verschiedenen Teiluntersuchungen zur Validierung des Funktionsprototypen allesamt mit Kampfflugzeugpiloten als Probandengruppe durchgeführt. Dieses Vorgehen führte zwar zu einer höheren externen Validität der Untersuchungen, da Kampfflugzeugpiloten über eine entsprechende militärische Ausbildung verfügen, um Funktionen gezielt einsetzen und bewerten zu können. Im Gegenzug führte die Auswahl dieses spezialisierten Teilnehmerkreises jedoch zu einer geringeren internen Validität, da verschiedene Untersuchungsergebnisse aufgrund der geringen Stichprobengröße nicht statistisch signifikant waren.

Hier hätte die Möglichkeit bestanden, die Detailuntersuchungen zu den Delegationsmethoden sowie zur Auftragskritik mit einer weiteren, weniger qualifizierten Probandengruppe durchzuführen, um bezüglich der quantitativen Aussagen wie Interaktionszeiten mehr statistisch signifikante Ergebnisse zu erreichen. Da bei einem solchen Versuchsaufbau auch die Unterschiede zwischen Novizen und ausgebildeten Kampfflugzeugpiloten ermittelt werden können, lassen sich aus diesen Ergebnissen ggf. weitere Aussagen über die Intuitivität der Interaktion, den Schulungsbedarf und die Anpassbarkeit des Systems an unterschiedliche Fähigkeitsniveaus ableiten.

Simulationsumgebung

Eine weitere Limitation dieser Arbeit liegt in der unvollständigen Abbildung der Arbeitsumgebung in einem zukünftigen Kampfflugzeug. Die Untersuchungen bildeten zwar die wesentlichen Elemente im Hinblick auf die Infrastruktur, die Missionsszenarien und die Cockpitaufgaben ab, allerdings werden verschiedene Aspekte der heutigen Arbeitsumgebung

von Kampfflugzeugpiloten nicht abgedeckt.⁷⁶ Dies könnte dazu führen, dass sich die im Rahmen der Arbeit entwickelten Interaktionskonzepte & Methoden in der Praxis als weniger geeignet herausstellen als in den vorgestellten Untersuchungen.

Ein wichtiger Aspekt, der in der Simulationsumgebung nicht abgebildet wurde, ist die Bewegung des bemannten Kampfflugzeugs. Die damit verbundenen Kräfte und Vibrationen haben das Potenzial, die Nutzerinteraktion mit dem System erheblich zu stören.

Die Funkkommunikation zwischen den verschiedenen Missionsteilnehmern wurde in den Untersuchungen ebenfalls nicht abgebildet. In heutigen Kampfflugzeugmissionen ist die Funkkommunikation ein sehr wichtiges Werkzeug zur Koordination zwischen den Missionsteilnehmern. Auch wenn dieser Koordinationsaufwand durch übersichtliche Kartendarstellungen sowie durch die Führung unbemannter Systeme reduziert werden dürfte, ist noch unklar, inwieweit die Funkkommunikation erhalten bleibt und damit einen Störfaktor insbesondere für die auditiv-verbale Modalitätenpaarung darstellt.

Im Sinne der Übertragbarkeit der Ergebnisse sollten im Idealfall weitere Versuche durchgeführt werden, in denen die nicht berücksichtigten Aspekte untersucht werden, oder es sollte zumindest abgeschätzt werden, welche Rolle diese Faktoren zukünftig spielen werden.

6.3 Weiterer Forschungsbedarf

Aus den Ergebnissen und Limitationen dieser Arbeit ergeben sich verschiedene Hinweise auf Themen, die im Rahmen zukünftiger Forschungsarbeiten weiter untersucht werden sollten. Im Folgenden werden diese Themen vorgestellt.

Führungskonzepte

Das Thema der Rollen und Aufgaben von Mensch und Automation ist eine Kernfrage bei der Auslegung von Mensch-Maschine-Systemen und sollte daher ein Schwerpunkt künftiger Forschungsarbeiten sein. Dieser Arbeit hat durch die Untersuchung des Führungskonzeptes der assistierten auftragsbasierten Führung einen Beitrag zur Beantwortung dieser Frage geleistet, allerdings gibt es auch andere vielversprechende Formen der Zusammenarbeit, die im Rahmen dieser Arbeit nicht untersucht wurden.

Eine dieser Formen ist das Führungskonzept der *Effect-based Guidance*. In diesem Führungskonzept erteilt der Nutzer keine Aufträge, sondern definiert stattdessen die Effekte, die das bemannt-unbemannte Team auf dem Schlachtfeld erzielen soll. Gegenüber der auftragsbasierten Führung könnte dies den Vorteil haben, dass eine größere Flexibilität darüber besteht, welche Aufgaben zur Erreichung des gewünschten Effektes eingesetzt werden und welche Kräfte zur Bearbeitung dieser Aufgaben herangezogen werden. Die Herausforderung wird bei diesem Ansatz darin bestehen, dem Nutzer zu vermitteln, auf welche Art und Weise das System die Effekte erzielen möchte.

⁷⁶ Von denen die meisten Aspekte auch in zukünftigen Arbeitsumgebungen relevant sein dürften.

Ein vielversprechender alternativer Führungsansatz ist die *auftragsbasierte Führung von Teams*. Hierzu müsste das Führungskonzept der auftragsbasierten Führung so angepasst werden, dass Aufträge auch an Teams von unbemannten Plattformen vergeben werden können, deren Mitglieder sich im Rahmen der Auftragserfüllung selbstständig untereinander koordinieren. In diesem Zusammenhang sollte auch die Möglichkeit der Ad-hoc-Bildung von Teams geprüft werden, d. h., dass die Nutzer je nach Bedarf Teams mit variabler Größe bilden und auflösen können.

Interaktionsmethoden zur UAV-Führung

Die Ergebnisse dieser Arbeit zeigen auch den Forschungsbedarf hinsichtlich der Interaktionsmethoden im Rahmen der UAV-Führung an. In diesem Zusammenhang sollte genauer untersucht werden, inwiefern die dargestellten Limitationen dieser Arbeit die Anwendbarkeit der Interaktionsmethoden in einem künftigen Kampfflugzeug beeinträchtigen könnten. Außerdem sollte die Ausgestaltung von einzelnen Interaktionsmethoden überarbeitet werden, um diese robuster gegenüber Störfaktoren zu machen. Ein wichtiger Störfaktor, der in diesem Zuge adressiert werden sollte, ist eine hohe Objektdichte auf dem Display (Display Clutter).

Hinsichtlich der Interaktionsmethoden bleibt ein wichtiger Untersuchungspunkt, welche Effekte die Bereitstellung mehrerer Interaktionsmethoden bringt. Diesbezüglich sollte untersucht werden, ob die Bereitstellung mehrerer Methoden auch dann vorteilhaft ist, wenn die Methoden robuster gegenüber einzelnen Störgrößen sind oder die Nutzer mit anderen Nebenaufgaben beschäftigt sind. In diesem Zuge könnte auch der hier vorgeschlagene Lösungsansatz, zwei diametral unterschiedliche Methoden zur Verfügung zu stellen, untersucht werden.

Ein weiterer interessanter Untersuchungspunkt ist in diesem Zusammenhang auch das fehlende Bewusstsein über die Verfügbarkeit von Methoden. Hier sollte Ursachenforschung betrieben werden, warum Nutzer in gewissen Situationen eine geeignete Interaktionsmethode nicht bedenken und durch welche Maßnahmen man diese Methode ins Bewusstsein der Nutzer bringen könnte.

Automatisierte Entscheidungsunterstützung

Im Themengebiet der automatisierten Entscheidungshilfen besteht ebenfalls noch Forschungsbedarf. Primär sollte sich die Forschung in diesem Kontext darauf konzentrieren, klare Richtlinien zu entwickeln, in welchen Anwendungsfällen eine kritikbasierte Entscheidungshilfe zu wählen ist und in welchen Anwendungsfällen eine vorschlagsbasierte Entscheidungshilfe die bessere Wahl darstellt.

Zu diesem Zweck sollten die spezifischen Vor- und Nachteile von vorschlags- bzw. kritikbasierten Entscheidungshilfen für unterschiedliche Anwendungen ermittelt werden. In diesem Zusammenhang sollte auch die vorgeschlagene Lösung, auf den oberen Ebenen der Entscheidungsfindung mit einem kritikbasierten System und auf den unteren Ebenen mit einem vorschlagsbasierten System zu unterstützen, untersucht werden.

Ein weiterer wichtiger Untersuchungspunkt ist der Einfluss von Feedback auf automatisierte Entscheidungshilfen. In der durchgeführten Untersuchung hatte das System die Auswirkungen von Entscheidungen durch ein visuelles Feedback sehr deutlich gemacht. Solche verlässlichen Vorhersagen sind jedoch bei anderen Entscheidungen oft nicht möglich, sodass sich die Frage stellt, inwieweit die Ergebnisse der Untersuchung durch das Feedback beeinflusst wurden. Um diese Frage zu beantworten, sollten weitere Studien durchgeführt werden, in denen das Feedback variiert und die Auswirkungen auf die Entscheidungsunterstützung untersucht werden.

Methoden zur UAV-Verhaltensgenerierung

Weiteren Forschungsbedarf gibt es auch hinsichtlich der Methoden zur Verhaltensgenerierung eines Plattformsteuerungsagenten. Im Rahmen dieser Arbeit bildeten BehaviorTrees das Kernelement der Verhaltensgenerierung, wofür insbesondere die Flexibilität und die Transparenz dieses Ansatzes ausschlaggebend waren. Diese Methode eignete sich zwar auch zur Erzeugung von komplexeren Verhaltensweisen und konnte somit alle Auftragsstypen und Parametrisierungen abbilden, allerdings kritisierten die Probanden das Verhalten der UAVs als stellenweise nicht taktisch sinnvoll. Dies betraf insbesondere Situationen, in denen das UAV eine Vielzahl von Faktoren berücksichtigen müsste, um ein optimales Verhalten zu erzeugen.⁷⁷

Andere Methoden könnten hier einen entscheidenden Vorteil gegenüber BehaviorTrees bieten, da ein BehaviorTree eine fallabhängige Beschreibung des Verhaltens erfordert, die für solche Situationen nur schwer zu definieren ist. Dies verdeutlicht den Bedarf an weiterer Forschung, um zu klären, wie eine Vielzahl verschiedener Faktoren in die Verhaltensgenerierung integriert werden kann, ohne dabei die Transparenz des Verhaltens für den Nutzer zu beeinträchtigen.

Ein vielversprechender Forschungsansatz, der in diesem Zuge untersucht werden sollte, ist die Kombination von modellbasierten Methoden und Deep Reinforcement Learning Methoden in einer gemeinsamen Architektur. Bleibt man beim BehaviorTree als modellbasierte Methode, so könnte dieser beispielsweise die wesentlichen Entscheidungen darüber treffen, welche Verhaltensweisen die Plattform durchführen soll und ein Reinforcement Learning Modell ist anschließend für die hochpräzise Ausführung dieser Verhaltensweisen verantwortlich. Alternativ könnte das Reinforcement Learning Modell auch in die Ablaufsteuerung des BehaviorTrees integriert werden und beispielsweise mit darüber entscheiden, welche Abschnitte der Baumstruktur ausgeführt werden sollen.

⁷⁷ Beispielsweise wenn ein UAV während eines Auftrags von Flugabwehrstellungen und Kampfflugzeugen bedroht wird.

A. Literaturverzeichnis

Abdat, F., Maaoui, C., & Pruski, A. (2011). Human-Computer Interaction Using Emotion Recognition from Facial Expression. *2011 UKSim 5th European Symposium on Computer Modeling and Simulation*, 196–201. <https://doi.org/10.1109/EMS.2011.20>

Akamatsu, M., Mackenzie, I. S., & Hasbroucq, T. (1995). A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics*, 38(4), 816–827. <https://doi.org/10.1080/00140139508925152>

Alden, D. G., Daniels, R. W., & Kanarick, A. F. (1972). Keyboard Design and Operation: A Review of the Major Issues. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 14(4), 275–293. <https://doi.org/10.1177/001872087201400401>

Arora, S. J., & Singh, R. P. (2012). Automatic speech recognition: A review. *International Journal of Computer Applications*, 60(9). <https://doi.org/10.5120/9722-4190>

Artigas, J., Balachandran, R., Riecke, C., Stelzer, M., Weber, B., Ryu, J.-H., & Albuschaeffer, A. (2016). KONTUR-2: Force-feedback teleoperation from the international space station. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1166–1173. <https://doi.org/10.1109/ICRA.2016.7487246>

Basil, M. D. (2012). Multiple Resource Theory. In N. M. Seel (Ed.), *Encyclopedia of the Sciences of Learning* (pp. 2384–2385). Springer US. https://doi.org/10.1007/978-1-4419-1428-6_25

BDLI. (2021). *Das Future Combat Air System: Übersicht*. Bundesverband der Deutschen Luft- und Raumfahrtindustrie e. V. <https://www.bdli.de/sites/default/files/2021-06/%C3%9Cbersicht%20FCAS.pdf>

Bieg, H.-J., Chuang, L. L., Fleming, R. W., Reiterer, H., & Bülthoff, H. H. (2010). Eye and pointer coordination in search and selection tasks. *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10*, 89. <https://doi.org/10.1145/1743666.1743688>

Billings, C. E. (1997). *Aviation automation: The search for a human-centered approach*. CRC Press. <https://doi.org/10.1201/9781315137995>

Boehm, F., Clauss, S., Brueggenwirth, S., & Schulte, A. (2012). Cognitive UAV resource management allowing task-based mission execution under data link limitations. *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, 8B3-1-8B3-11. <https://doi.org/10.1109/DASC.2012.6382437>

Boyne, W. J. (2002). *Air Warfare: An International Encyclopedia*. ABC-CLIO. <https://doi.org/10.5040/9798216190332>

Bragdon, A., Nelson, E., Li, Y., & Hinckley, K. (2011). Experimental analysis of touch-screen gesture designs in mobile environments. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 403–412. <https://doi.org/10.1145/1978942.1979000>

Brand, Y., & Schulte, A. (2017). Model-based prediction of workload for adaptive associate systems. *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1722–1727. <https://doi.org/10.1109/SMC.2017.8122864>

Brand, Y., & Schulte, A. (2021). Workload-adaptive and task-specific support for cockpit crews: Design and evaluation of an adaptive associate system. *Human-Intelligent Systems Integration*, 3(2), 187–199. <https://doi.org/10.1007/s42454-020-00018-8>

Bratman, M. E., Israel, D. J., & Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(3), 349–355. <https://doi.org/10.1111/j.1467-8640.1988.tb00284.x>

Breuninger, J. (2020). *Suitability of Touch Gestures and Virtual Physics in Touchscreen User Interfaces for Critical Tasks*. <https://doi.org/10.13140/RG.2.2.16704.15363>

Brooke, J. (1996). SUS-A quick and dirty usability scale. In *Usability evaluation in industry* (Vol. 189, pp. 4–7). CRC Press. <https://doi.org/10.1201/9781498710411-35>

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1), 14–23. <https://doi.org/10.1109/JRA.1986.1087032>

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1–3), 139–159. [https://doi.org/10.1016/0004-3702\(91\)90053-M](https://doi.org/10.1016/0004-3702(91)90053-M)

Brown, A. (2021, December 13). *Text To Speech—Foundational Knowledge (Part 2)*. Medium. <https://towardsdatascience.com/text-to-speech-foundational-knowledge-part-2-4db2a3657335>

Brumitt, B. L., & Stentz, A. (1998). GRAMMPS: A generalized mission planner for multiple mobile robots in unstructured environments. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 2, 1564–1571. <https://doi.org/10.1109/ROBOT.1998.677360>

Buscher, G., Van Elst, L., & Dengel, A. (2009). Segment-level display time as implicit feedback: A comparison to eye tracking. *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 67–74. <https://doi.org/10.1145/1571941.1571955>

Calhoun, G., & Draper, M. (2006). 11. Multi-Sensory Interfaces for Remotely Operated Vehicles. In N. J. Cooke, H. L. Pringle, H. K. Pedersen, & O. Connor (Eds.), *Human Factors of Remotely Operated Vehicles* (Vol. 7, pp. 149–163). Emerald Group Publishing Limited. [https://doi.org/10.1016/S1479-3601\(05\)07011-6](https://doi.org/10.1016/S1479-3601(05)07011-6)

Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1988). An empirical comparison of pie vs. Linear menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '88*, 95–100. <https://doi.org/10.1145/57167.57182>

Casali, S. P., Williges, B. H., & Dryden, R. D. (1990). Effects of Recognition Accuracy and Vocabulary Size of a Speech Recognition System on Task Performance and User Acceptance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 32(2), 183–196. <https://doi.org/10.1177/001872089003200206>

Chen, J. Y. C., Barnes, M. J., Selkowitz, A. R., & Stowers, K. (2016). Effects of Agent Transparency on human-autonomy teaming effectiveness. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 001838–001843). IEEE. <https://doi.org/10.1109/SMC.2016.7844505>

Clauss, S., & Schulte, A. (2014). Task delegation in an agent supervisory control relationship capability awareness in a cognitive agent. In *Proceedings: 2014 IEEE International Conference on Systems, Man and Cybernetics* (pp. 825–830). <https://doi.org/10.1109/SMC.2014.6974013>

Cohen, P. R., Dalrymple, M., Moran, D. B., Pereira, F. C., & Sullivan, J. W. (1989). Synergistic use of direct manipulation and natural language. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Wings for the Mind - CHI '89*, 227–233. <https://doi.org/10.1145/67449.67494>

Cohen, P. R., & Oviatt, S. L. (1995). The role of voice input for human-machine communication. *Proceedings of the National Academy of Sciences*, 92(22), 9921–9927. <https://doi.org/10.1073/pnas.92.22.9921>

Colledanchise, M., & Ögren, P. (2017). *Behavior Trees in Robotics and AI: An Introduction*. <https://doi.org/10.48550/ARXIV.1709.00084>

Copp, T., & Weisgerber, M. (2021, May 12). *The Air Force Is Planning For a Future Without the F-22*. Defense One. <https://www.defenseone.com/technology/2021/05/air-force-planning-future-without-f-22/174001/>

Dekker, S. W. A., & Woods, D. D. (2002). MABA-MABA or Abracadabra? Progress on Human-Automation Co-ordination. *Cognition, Technology & Work*, 4(4), 240–244. <https://doi.org/10.1007/s101110200022>

Dennis, L. A., & Fisher, M. (Eds.). (2023). *Autonomous Systems Architectures*. In *Verifiable Autonomous Systems: Using Rational Agents to Provide Assurance about Decisions Made by Machines* (pp. 13–23). Cambridge University Press; Cambridge Core. <https://doi.org/10.1017/9781108755023.002>

Dix, A. (2004). *Human-computer interaction* (3rd ed.). Pearson Education.

Doll, T. J., & Folds, D. J. (1986). Auditory signals in military aircraft: Ergonomics principles versus practice. *Applied Ergonomics*, 17(4), 257–264. [https://doi.org/10.1016/0003-6870\(86\)90127-4](https://doi.org/10.1016/0003-6870(86)90127-4)

Duchowski, A. T. (2007). *Eye Tracking Techniques*. In *Eye tracking methodology: Theory and practice* (2nd ed). Springer. http://doi.org/10.1007/978-1-84628-609-4_5

Dudek, M., & Schulte, A. (2022a). Effects of Tasking Modalities in Manned-Unmanned Teaming Missions. *AIAA SCITECH 2022 Forum*. <https://doi.org/10.2514/6.2022-2478>

Dudek, M., & Schulte, A. (2022b). Meaningful guidance of unmanned aerial vehicles in dynamic environments. *Proceedings of the 1st International Conference on Cognitive Aircraft Systems*. <https://doi.org/10.5220/0011946500003622>

Dudek, M., & Schulte, A. (2023). Comparison of Delegation Methods for Task-Based UAV Guidance. *Proceedings of the 22nd International Symposium of Aviation Psychology*. https://corescholar.libraries.wright.edu/isap_2023/46

Dudek, M., & Schulte, A. (2024a). Avoiding automation bias through critique? Experimental evaluation of a decision-support critiquing system for UAV task delegation. In D. De Waard, D. Manzey, K. Brookhuis, F. Siebert, K. Karrer-Gauß, S. Winkler, F. Di Nocera, A. Toffetti, & T. Franke (Eds.), *Proceedings of the Human Factors and Ergonomics Society Europe Chapter 2024 Annual Conference*. <https://www.hfes-europe.org/wp-content/uploads/2024/05/Dudek2024.pdf>

Dudek, M., & Schulte, A. (2024b). Experimental Evaluation of UAV Task Delegation Methods. *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, 1–8. <https://doi.org/10.1109/ICHMS59971.2024.10555792>

Endsley, M. R. (2017). From Here to Autonomy. *Human Factors*, *59*(1), 5–27. <https://doi.org/10.1177/0018720816681350>

Endsley, M. R., & Kaber, D. B. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, *42*(3), 462–492. <https://doi.org/10.1080/001401399185595>

ETAP GSS Phase 3. (2015). *Report R2.1—Operational Assumptions and System Data Base* (Global System Study). European Technology Acquisition Programme.

Fessenden, T. (2017, April 23). *Modal & Nonmodal Dialogs: When (& When Not) to Use Them*. Nielsen Norman Group. <https://www.nngroup.com/articles/modal-nonmodal-dialog/>

Fitts, P. M. (1951). *Human engineering for an effective air-navigation and traffic-control system*.

Friedrich, M., Richards, D., & Vollrath, M. (2022). Icon Design for Representing Safety-Critical Aircraft Functions to Support Supervisory Control of Remotely Piloted Aircraft Systems. *Aerospace*, *9*(4), 181. <https://doi.org/10.3390/aerospace9040181>

Fu Li, Wang Qi, Xu Jin, Zhou Yuandong, & Zhu Kun. (2012). Target assignment and sorting for multi-target attack in Multi-aircraft coordinated based on RBF. *2012 24th Chinese Control and Decision Conference (CCDC)*, 1935–1938. <https://doi.org/10.1109/CCDC.2012.6244311>

Gable, N. (2024). *Commercial Active Stick: An Active Role*. BAE Systems. <https://www.baesystems.com/en/feature/an-active-role>

Gallersdörfer, F. (2021). *Experimente im Flugsimulator zu Unterschieden bzgl. Missionsperformance bei verschiedenen Eingabemodalitäten (Touch/Voice)* [Unveröffentlichte Bachelorarbeit]. Universität der Bundeswehr.

Gertler, J. (2020). *Air Force Next-Generation Air Dominance Program: An Introduction* [Congressional Research Service Report]. LIBRARY OF CONGRESS WASHINGTON DC. <https://apps.dtic.mil/sti/pdfs/AD1169997.pdf>

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., & Ng, A. Y. (2014). *Deep Speech: Scaling up end-to-end speech recognition* (Version 2). arXiv. <https://doi.org/10.48550/ARXIV.1412.5567>

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology* (pp. 139–183). Elsevier. [https://doi.org/10.1016/s0166-4115\(08\)62386-9](https://doi.org/10.1016/s0166-4115(08)62386-9)

Hartley, K. (1999). The eurofighter development programme. *Air & Space Europe*, 1(3), 46–50. [https://doi.org/10.1016/S1290-0958\(00\)88427-0](https://doi.org/10.1016/S1290-0958(00)88427-0)

Hartmann, E. (2023). Assessment of a hybrid cursor control device for highly dynamic cockpit environments. In D. de Waard, V. Hagemann, L. Onnasch, A. Toffetti, D. Coelho, A. Botzer, M. de Angelis, K. Brookhuis, & S. Fairclough (Eds.), *Proceedings of the Human Factors and Ergonomics Society Europe Chapter 2023 Annual Conference*.

Heilemann, F. (2022). *Skalierbare Delegation und adaptive Planungsassistenz zur Führung bemannt-unbemannter Teams: Konzept, Implementierung und Evaluation*. Verlag Dr. Hut.

Heilemann, F., Lindner, S., & Schulte, A. (2021). Experimental evaluation of tasking and teaming design patterns for human delegation of unmanned vehicles. *Human-Intelligent Systems Integration*, 3(3), 223–240. <https://doi.org/10.1007/s42454-021-00036-0>

Heilemann, F., & Schulte, A. (2019). Interaction Concept for Mixed-Initiative Mission Planning on Multiple Delegation Levels in Multi-UCAV Fighter Missions. In W. Karwowski & T. Ahram (Eds.), *Intelligent Human Systems Integration 2019* (pp. 699–705). Springer Nature Switzerland.

Heilemann, F., & Schulte, A. (2020a). Experimental evaluation of an adaptive planning assistance system in manned unmanned teaming missions. *Augmented Cognition. Human Cognition and Behavior*, 371–382. https://doi.org/10.1007/978-3-030-50439-7_25

Heilemann, F., & Schulte, A. (2020b). Time Line Based Tasking Concept for MUM-T Mission Planning with Multiple Delegation Levels. In T. Z. Ahram, W. Karwowski, A. Vergnano, F. Leali, & R. Taïar (Eds.), *Intelligent Human Systems Integration 2020* (1st ed. 2020, pp. 1014–1020). Springer International Publishing. https://doi.org/10.1007/978-3-030-39512-4_154

Hofmann, J. (2022). *Gestaltung von Feedbackarten bei der Führung unbemannter Luftfahrzeuge unter Nutzung verschiedener Modalitäten* [Unveröffentlichte Masterarbeit]. Universität der Bundeswehr.

Holzinger, A. (2002). Finger instead of mouse: Touch screens as a means of enhancing universal access. *Universal Access Theoretical Perspectives, Practice, and Experience*, 387–397. https://doi.org/10.1007/3-540-36572-9_30

Horst, J. V. D., & Noble, J. (2010). Distributed and Centralized Task Allocation: When and Where to Use Them. *2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop*, 1–8. <https://doi.org/10.1109/SASOW.2010.8>

Hossain, M. S., Muhammad, G., Song, B., Hassan, M. M., Alelaiwi, A., & Alamri, A. (2015). Audio–Visual Emotion-Aware Cloud Gaming Framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(12), 2105–2118. <https://doi.org/10.1109/TCSVT.2015.2444731>

Itti, L. (2004). Automatic Foveation for Video Compression Using a Neurobiological Model of Visual Attention. *IEEE Transactions on Image Processing*, 13(10), 1304–1318. <https://doi.org/10.1109/TIP.2004.834657>

Jacob, R. J. (1991). The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems (TOIS)*, 9(2), 152–169. <https://doi.org/10.1145/123078.128728>

Jarrett, D. N. (2017). *Cockpit Engineering* (1st ed.). Routledge. <https://doi.org/10.4324/9781315260020>

Joe, J. C., O’Hara, J., Hugo, J. V., & Oxstrand, J. H. (2015). Function Allocation for Humans and Automation in the Context of Team Dynamics. *Procedia Manufacturing*, 3, 1225–1232. <https://doi.org/10.1016/j.promfg.2015.07.204>

Johnson, J. (2010). *Designing with the mind in mind: Simple guide to understanding user interface design rules*. Morgan Kaufmann Publishers/Elsevier. <https://doi.org/10.1016/c2009-0-20318-7>

Jurafsky, D., & Martin, J. H. (2024a). Automatic Speech Recognition and Text-to-Speech. In *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (3. ed. [Draft.]). Prentice Hall. <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>

Jurafsky, D., & Martin, J. H. (2024b). N-gram Language Models. In *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (3. ed. [Draft.]). Prentice Hall. <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>

Justh, E. W., & Krishnaprasad, P. S. (2002). *A simple control law for UAV formation flying*. Technical Report 2002-38, Institute for Systems Research.

Këpuska, V. (2011). Wake-up-word speech recognition. *Speech Technologies*, 237–262. <https://doi.org/10.5772/16242>

Khamis, A., Hussein, A., & Elmogy, A. (2015). Multi-robot Task Allocation: A Review of the State-of-the-Art. In A. Koubâa & J. R. Martínez-de Dios (Eds.), *Cooperative Robots and Sensor*

Networks 2015 (pp. 31–51). Springer International Publishing. https://doi.org/10.1007/978-3-319-18299-5_2

Kingston, D., Rasmussen, S., & Humphrey, L. (2016). Automated UAV tasks for search and surveillance. *2016 IEEE Conference on Control Applications (CCA)*, 1–8. <https://doi.org/10.1109/CCA.2016.7587813>

Körber, M. (2019). Theoretical Considerations and Development of a Questionnaire to Measure Trust in Automation. In S. Bagnara, R. Tartaglia, S. Albolino, T. Alexander, & Y. Fujita (Eds.), *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)* (Vol. 823, pp. 13–30). Springer International Publishing. https://doi.org/10.1007/978-3-319-96074-6_2

Kühn, A. (2022). *Entwicklung einer multimodalen Schnittstelle zur Zuweisung von Aufträgen an unbemannte Luftfahrzeuge* [Unveröffentlichte Masterarbeit]. Universität der Bundeswehr.

Kumar, E. (2011). *Natural language processing*. I.K. International Publishing House.

Kumar, M., Paepcke, A., & Winograd, T. (2007). EyePoint: Practical pointing and selection using gaze and keyboard. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 421–430. <https://doi.org/10.1145/1240624.1240692>

Kun, A. L., Paek, T., & Medenica, Z. (2007). The effect of speech interface accuracy on driving performance. *Interspeech*, 1326–1329. <https://doi.org/10.21437/interspeech.2007-406>

Larson, J., & Hawkins, G. (2023). Speed-accuracy tradeoffs in decision making: Perception shifts and goal activation bias decision thresholds. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 49(1). <https://doi.org/10.31234/osf.io/6ycaf>

Law, B., Atkins, M. S., Kirkpatrick, A. E., & Lomax, A. J. (2004). Eye gaze patterns differentiate novice and experts in a virtual laparoscopic surgery training environment. *Proceedings of the Eye Tracking Research & Applications Symposium on Eye Tracking Research & Applications - ETRA'2004*, 41–48. <https://doi.org/10.1145/968363.968370>

Legras, F., & Coppin, G. (2007). Autonomy spectrum for a multiple UAVs system. *COGIS'07-Cognitive Systems with Interactive Sensors*.

Lindner, S., Schwerd, S., & Schulte, A. (2019). Defining Generic Tasks to Guide UAVs in a MUM-T Aerial Combat Environment. In W. Karwowski & T. Ahram (Eds.), *Intelligent Human Systems Integration 2019* (pp. 777–782). Springer Nature Switzerland. https://doi.org/10.1007/978-3-030-11051-2_118

Lockheed Martin. (2017, April 10). *U.S. Air Force, Lockheed Martin Demonstrate Manned/Unmanned Teaming*. Media - Lockheed Martin. <https://news.lockheedmartin.com/2017-04-10-U-S-Air-Force-Lockheed-Martin-Demonstrate-Manned-Unmanned-Teaming>

Lyons, M. J. (2004). Facial gesture interfaces for expression and communication. *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, 1, 598–603. <https://doi.org/10.1109/ICSMC.2004.1398365>

Maes, P. (1991). The agent network architecture (ANA). *ACM SIGART Bulletin*, 2(4), 115–120. <https://doi.org/10.1145/122344.122367>

Maier, S., Kiam, J. J., & Schulte, A. (2024). Adaptive Mission Planning: Evaluation of a Hybrid Cognitive Mixed-Initiative Planning Assistant in Manned-Unmanned Teaming Operations. *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3492–3499. <https://doi.org/10.1109/SMC54092.2024.10831141>

Maier, S., & Schulte, A. (2021). Concept for Cross-platform Delegation of Heterogeneous UAVs in a MUM-T Environment. In M. Zallio, C. Raymundo Ibañez, & J. H. Hernandez (Eds.), *Advances in Human Factors in Robots, Unmanned Systems and Cybersecurity* (Vol. 268, pp. 3–9). Springer International Publishing. https://doi.org/10.1007/978-3-030-79997-7_1

Mardenbegi, D., & Qvarfordt, P. (2015). Creating gaze annotations in head mounted displays. *Proceedings of the 2015 ACM International Symposium on Wearable Computers - ISWC '15*, 161–162. <https://doi.org/10.1145/2802083.2808404>

Markovich, T., Honig, S., & Oron-Gilad, T. (2020). Closing the feedback loop: The relationship between input and output modalities in human-robot interactions. *International Workshop on Human-Friendly Robotics*, 29–42. https://doi.org/10.1007/978-3-030-42026-0_3

McCowan, I. A., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P., & Boulard, H. (2004). *On the use of information retrieval measures for speech recognition evaluation* (IDIAP Research Report Nos. 04–73). <https://infoscience.epfl.ch/handle/20.500.14299/228600>

Metzger, U., & Parasuraman, R. (2005). Automation in future air traffic management: Effects of decision aid reliability on controller performance and mental workload. *Human Factors*, 47(1), 35–49. <https://doi.org/10.1518/0018720053653802>

Miller, C. (2016). Sub-patterns for Human-Autonomy Teaming: Variations on a Delegation Theme. In D. Harris (Ed.), *Engineering Psychology and Cognitive Ergonomics* (pp. 198–210). Springer International Publishing. https://doi.org/10.1007/978-3-319-40030-3_21

Miller, C., Funk, H., Goldman, R., Meisner, J., & Wu, P. (2005). Implications of adaptive vs. Adaptable UIs on decision making: Why “automated adaptiveness” is not always the right answer. *Proceedings of the 1st International Conference on Augmented Cognition*, 22–27.

Miller, C., Funk, H., Wu, P., Goldman, R., Meisner, J., & Chapman, M. (2005). The Playbook™ Approach to Adaptive Automation. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 49(1), 15–19. <https://doi.org/10.1177/154193120504900105>

Miller, C., & Parasuraman, R. (2007). Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human Factors*, 49(1), 57–75. <https://doi.org/10.1518/001872007779598037>

Mund, D., & Schulte, A. (2018). *Model- and Observation- based Workload Assessment and Activity Determination in Manned-Unmanned Teaming Missions*. 33rd EAAP Conference, Dubrovnik, Croatia.

Munteanu, C., Baecker, R., Penn, G., Toms, E., & James, D. (2006). The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 493–502. <https://doi.org/10.1145/1124772.1124848>

Murthy, L. R. D., Mukhopadhyay, A., Yellheti, V., Arjun, S., Thomas, P., Dilli Babu, M., Singh Saluja, K. P., JeevithaShree, D. V., & Biswas, P. (2020). Evaluating Accuracy of Eye Gaze Controlled Interface in Military Aviation Environment. *2020 IEEE Aerospace Conference*, 1–12. <https://doi.org/10.1109/AERO47225.2020.9172480>

Neves, P., & Câmara, A. (2020). Multisensory HCI Design with Smell and Taste for Environmental Health Communication. In C. Stephanidis, A. Marcus, E. Rosenzweig, P.-L. P. Rau, A. Moallem, & M. Rauterberg (Eds.), *HCI International 2020—Late Breaking Papers: User Experience Design and Case Studies* (pp. 441–463). Springer International Publishing. https://doi.org/10.1007/978-3-030-60114-0_31

Ng, H., Tao, D., & Or, C. K. (2013). Age differences in computer input device use: A comparison of touchscreen, trackball, and mouse. *Advances in Information Systems and Technologies*, 1015–1024. https://doi.org/10.1007/978-3-642-36981-0_96

Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain Computer Interfaces, a Review. *Sensors*, 12(2), 1211–1279. <https://doi.org/10.3390/s120201211>

Norman, D. A. (2013). *The design of everyday things* (Revised and expanded edition). Basic Books.

Obrist, M., Velasco, C., Vi, C., Ranasinghe, N., Israr, A., Cheok, A., Spence, C., & Gopalakrishnakone, P. (2016). Sensing the future of HCI: Touch, taste, and smell user interfaces. *Interactions*, 23(5), 40–49. <https://doi.org/10.1145/2973568>

O’Connell, S. D., Castor, M., Pousette, J., & Krantz, M. (2012). Eye Tracking-Based Target Designation in Simulated Close Range Air Combat. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 56(1), 46–50. <https://doi.org/10.1177/1071181312561030>

Olson, W. A., & Wuennenberg, M. G. (2001). Autonomy based human-vehicle interface standards for remotely operated aircraft. *20th DASC. 20th Digital Avionics Systems Conference (Cat. No.01CH37219)*, 2, 7D3/1-7D3/9. <https://doi.org/10.1109/DASC.2001.964199>

Öncan, T. (2007). A Survey of the Generalized Assignment Problem and Its Applications. *INFOR: Information Systems and Operational Research*, 45(3), 123–141. <https://doi.org/10.3138/infor.45.3.123>

Onken, R., & Schulte, A. (2010). *System-Ergonomic Design of Cognitive Automation* (Vol. 235). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-03135-9>

Oviatt, S. (2022). Multimodal Interaction, Interfaces, and Analytics. In J. Vanderdonckt, P. Palanque, & M. Winckler (Eds.), *Handbook of Human Computer Interaction* (pp. 1–29). Springer International Publishing. https://doi.org/10.1007/978-3-319-27648-9_22-1

Oviatt, S., & Cohen, P. (2000). Perceptual user interfaces: Multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3), 45–53. <https://doi.org/10.1145/330534.330538>

Oviatt, S., Coulston, R., & Lunsford, R. (2004). When do we interact multimodally?: Cognitive load and multimodal communication patterns. *Proceedings of the 6th International Conference on Multimodal Interfaces*, 129–136. <https://doi.org/10.1145/1027933.1027957>

Oviatt, S., Schuller, B., Cohen, P. R., Sonntag, D., Potamianos, G., & Krüger, A. (Eds.). (2017). *The handbook of multimodal-multisensor interfaces. Volume 1: Foundations, user modeling, and common modality combinations* (First edition). Association for Computing Machinery.

Parasuraman, R., Galster, S., Squire, P., Furukawa, H., & Miller, C. (2005). A Flexible Delegation-Type Interface Enhances System Performance in Human Supervision of Multiple Robots: Empirical Studies With RoboFlag. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(4), 481–493. <https://doi.org/10.1109/TSMCA.2005.850598>

Parasuraman, R., & Riley, V. (1997). Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 39(2), 230–253. <https://doi.org/10.1518/001872097778543886>

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans : A Publication of the IEEE Systems, Man, and Cybernetics Society*, 30(3), 286–297. <https://doi.org/10.1109/3468.844354>

Patel, R., Rudnick-Cohen, E., Azarm, S., Otte, M., Xu, H., & Herrmann, J. W. (2020). Decentralized Task Allocation in Multi-Agent Systems Using a Decentralized Genetic Algorithm. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 3770–3776. <https://doi.org/10.1109/ICRA40945.2020.9197314>

Pearson, G., & Weiser, M. (1986). Of moles and men: The design of foot controls for workstations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 333–339. <https://doi.org/10.1145/22627.22392>

Peixoto, N., Nik, H. G., & Charkhkar, H. (2013). Voice controlled wheelchairs: Fine control by humming. *Computer Methods and Programs in Biomedicine*, 112(1), 156–165. <https://doi.org/10.1016/j.cmpb.2013.06.009>

Poupyrev, I., & Maruyama, S. (2003). Tactile interfaces for small touch screens. *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, 217–220. <https://doi.org/10.1145/964696.964721>

Pulliam, R., Price, H., Bongarra, J., Sawyer, C., & Kisner, R. (1983). *Methodology for allocating nuclear power plant control functions to human or automatic control*. BioTechnology, Inc.; Oak Ridge National Laboratory. <https://doi.org/10.2172/5836278>

Qiao, F., Sherwani, J., & Rosenfeld, R. (2010). Small-vocabulary speech recognition for resource-scarce languages. *Proceedings of the First ACM Symposium on Computing for Development*, 1–8. <https://doi.org/10.1145/1926180.1926184>

Rabhi, Y., Mrabet, M., & Fnaiech, F. (2018). A facial expression controlled wheelchair for people with disabilities. *Computer Methods and Programs in Biomedicine*, 165, 89–105. <https://doi.org/10.1016/j.cmpb.2018.08.013>

Reichenbach, J., Onnasch, L., & Manzey, D. (2011). Human performance consequences of automated decision aids in states of sleep loss. *Human Factors*, 53(6), 717–728. <https://doi.org/10.1177/0018720811418222>

Renaud, K., & Cooper, R. (2000). Feedback in human-computer interaction—Characteristics and recommendations. *South African Computer Journal*, 2000(26), 105–114. <https://journals.co.za/doi/abs/10.10520/EJC27895>

Rieth, M., & Hagemann, V. (2022). Automation as an equal team player for humans? – A view into the field and implications for research and practice. *Applied Ergonomics*, 98, 103552. <https://doi.org/10.1016/j.apergo.2021.103552>

Rosenhoover, F. A. (1987). AFTI/F-16 Voice Interactive Avionics Evaluation. *SAE Technical Paper*. Aerospace Technology Conference and Exposition. <https://doi.org/10.4271/871751>

Roth, E. M., Sushereba, C., Militello, L. G., DiIulio, J., & Ernst, K. (2019). Function Allocation Considerations in the Era of Human Autonomy Teaming. *Journal of Cognitive Engineering and Decision Making*, 13(4), 199–220. <https://doi.org/10.1177/1555343419878038>

Rovira, E., McGarry, K., & Parasuraman, R. (2007). Effects of imperfect automation on decision making in a simulated command and control task. *Human Factors*, 49(1), 76–87. <https://doi.org/10.1518/001872007779598082>

Russell, S., & Norvig, P. (with Davis, E., & Edwards, D.). (2016). *Artificial intelligence: A modern approach* (Third edition, Global edition). Pearson.

Schelle, A., & Stütz, P. (2018). Gestural Transmission of Tasking Information to an Airborne UAV. In S. Yamamoto (Ed.), *Human Interface and the Management of Information. Interaction, Visualization, and Analytics* (pp. 318–335). Springer International Publishing. https://doi.org/10.1007/978-3-319-92043-6_27

Schmidt-Skipiol, F. J., & Hecker, P. (2015). Tactile Feedback and Situation Awareness—Improving Adherence to an Envelope in Sidestick-Controlled Fly-by-Wire Aircrafts. *15th AIAA Aviation Technology, Integration, and Operations Conference*. 15th AIAA Aviation Technology, Integration, and Operations Conference, Dallas, TX. <https://doi.org/10.2514/6.2015-2905>

Schmitt, F., & Schulte, A. (2016). Mixed-Initiative Missionsplanung für Multi-UAV Szenarien. *Kognitive Systeme*, 2016(1). <https://doi.org/10.17185/dupublico/42383>

Schmitt, F., & Schulte, A. (2018). Experimental Evaluation of a Scalable Mixed-Initiative Planning Associate for Future Military Helicopter Missions. In D. Harris (Ed.), *Engineering Psychology and Cognitive Ergonomics* (pp. 649–663). Springer International Publishing. https://doi.org/10.1007/978-3-319-91122-9_52

Schulte, A., & Donath, D. (2018). A Design and Description Method for Human-Autonomy Teaming Systems. In W. Karwowski & T. Ahram (Eds.), *Intelligent Human Systems Integration* (pp. 3–9). Springer International Publishing. https://doi.org/10.1007/978-3-319-73888-8_1

Schulte, A., & Donath, D. (2020). Systems level design patterns approach for HAT. *Human-Autonomy Teaming: Supporting Dynamically Adjustable Collaboration*. NATO STO Task Group Report HFM-247. Neuilly-Sur-Seine, France.

Schulte, A., Donath, D., & Lange, D. S. (2016). Design Patterns for Human-Cognitive Agent Teaming. In D. Harris (Ed.), *Engineering Psychology and Cognitive Ergonomics* (pp. 231–243). Springer International Publishing. https://doi.org/10.1007/978-3-319-40030-3_24

Schweigert, R., Schwind, V., & Mayer, S. (2019). EyePointing: A Gaze-Based Selection Technique. *Proceedings of Mensch Und Computer 2019*, 719–723. <https://doi.org/10.1145/3340764.3344897>

Schwerd, S., & Schulte, A. (2020). Experimental Validation of an Eye-Tracking-Based Computational Method for Continuous Situation Awareness Assessment in an Aircraft Cockpit. In D. Harris & W.-C. Li (Eds.), *Engineering Psychology and Cognitive Ergonomics. Cognition and Design* (pp. 412–425). Springer International Publishing. https://doi.org/10.1007/978-3-030-49183-3_32

Schwerd, S., & Schulte, A. (2021). *Situation Awareness Adaptive Alerting in an Aircraft Cockpit: A Simulator Study*. <https://doi.org/10.31224/osf.io/9gmnv>

Schwerd, S., & Schulte, A. (2024). Triggering cockpit alerts using an eye-tracking-based measure of monitoring performance. *Aviation Psychology and Applied Human Factors*.

Sells, C., & Griffiths, I. (2007). *Programming WPF: Building Windows UI with Windows Presentation Foundation*. (2nd ed). O'Reilly.

Sharma, R., Pavlovic, V. I., & Huang, T. S. (1998). Toward multimodal human-computer interface. *Proceedings of the IEEE*, 86(5), 853–869. <https://doi.org/10.1109/5.664275>

Shaw, R. L. (1985). *Fighter combat: Tactics and maneuvering*. Naval Institute Press. https://www.google.de/books/edition/Fighter_Combat/hKHLwAEACAAJ

Shen, W. (2019). *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*. CRC Press. <https://doi.org/10.1201/9780429182112>

Sheridan, T., & Verplank, W. (1978). *Human and Computer Control of Undersea Teleoperators*. Defense Technical Information Center. <https://doi.org/10.21236/ada057655>

Shree Dv, J., Murthy, L. R. D., Saluja, K. S., & Biswas, P. (2018). Operating Different Displays in Military Fast Jets Using Eye Gaze Tracker. *Journal of Aviation Technology and Engineering*, 8(1), 31. <https://doi.org/10.7771/2159-6670.1184>

Sibert, L. E., & Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 281–288. <https://doi.org/10.1145/332040.332445>

Sprenger, S. (2020). Ist es noch Taktilität oder schon Haptik? In *Haptik am User Interface* (pp. 31–32). transcript Verlag. <https://doi.org/10.1515/9783839451342-006>

Sucan, I. A., Moll, M., & Kavraki, L. E. (2012). The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4), 72–82. <https://doi.org/10.1109/MRA.2012.2205651>

Tao, D., Yuan, J., Liu, S., & Qu, X. (2018). Effects of button design characteristics on performance and perceptions of touchscreen use. *International Journal of Industrial Ergonomics*, 64, 59–68. <https://doi.org/10.1016/j.ergon.2017.12.001>

Theißing, N., & Schulte, A. (2016). Designing a support system to mitigate pilot error while minimizing out-of-the-loop-effects. *Engineering Psychology and Cognitive Ergonomics*, 439–451. https://doi.org/10.1007/978-3-319-40030-3_43

Thomas, P. (2018). Performance, characteristics, and error rates of cursor control devices for aircraft cockpit interaction. *International Journal of Human-Computer Studies*, 41–53. <https://doi.org/10.1016/j.ijhcs.2017.08.003>

Thomas, P., Biswas, P., & Langdon, P. (2015). State-of-the-Art and Future Concepts for Interaction in Aircraft Cockpits. In M. Antona & C. Stephanidis (Eds.), *Universal Access in Human-Computer Interaction.: Access to Interaction* (pp. 538–549). Springer International Publishing. https://doi.org/10.1007/978-3-319-20681-3_51

U. S. Naval Institute. (2024, November 7). Report to Congress on Air Force Next-Generation Air Dominance Fighter. *USNI News*. <https://news.usni.org/2024/11/07/report-to-congress-on-air-force-next-generation-air-dominance-fighter>

Uhrmann, J., & Schulte, A. (2011). Task-based Guidance of Multiple UAV Using Cognitive Automation. In T. Bossomaier (Ed.), *Cognitive 2011: The Third International Conference on Advanced Cognitive Technologies and Applications*. Curran.

Uhrmann, J., & Schulte, A. (2012). Concept, Design and Evaluation of Cognitive Task-based UAV Guidance. In *J. Adv. Intell. Syst: Vol 5 no1&2* (pp. 145–158).

U.S. Army. (2023). *Brevity: Multi-Service Tactics, Techniques, and Procedures for Multi-Service Brevity Codes*. Air Land Sea Space Application (ALSSA) Center.

https://www.alssa.mil/Portals/9/Documents/mttps/brevity_2023.pdf?ver=FmQkpQ3BYGk3XkAUd_vQWA%3d%3d

U.S. Department of Defense. (2012). *MIL-STD-1472: Human Engineering Design Criteria for Military Systems, Equipment, and Facilities* (No. MIL-STD-1472G; Version H). U.S. Government Printing Office. <https://www.dtic.mil/dtic/tr/fulltext/u2/a606015.pdf>

Van Baelen, D., Van Paassen, M. M. (René), Ellerbroek, J., Abbink, D. A., & Mulder, M. (2021). Flying by Feeling: Communicating Flight Envelope Protection through Haptic Feedback. *International Journal of Human-Computer Interaction*, 37(7), 655–665. <https://doi.org/10.1080/10447318.2021.1890489>

Van Erp, J., & Self, B. (2008). Introduction to tactile displays in military environments. *Tactile Displays for Orientation, Navigation and Communication in Air, Sea, and Land Environments*, 1–1.

Vidulich, M. A., & Wickens, C. D. (1985). Stimulus-central processing-response compatibility: Guidelines for the optimal use of speech technology. *Behavior Research Methods, Instruments, & Computers*, 17(2), 243–249. <https://doi.org/10.3758/BF03214390>

Weinstein, C. J. (1990). Opportunities for advanced speech processing in military computer-based systems. *Proceedings of the Workshop on Speech and Natural Language - HLT '90*, 433–452. <https://doi.org/10.3115/116580.116728>

Wickens, C. D. (1984). Processing resources and attention, varieties of attention. R. Parasuraman and D. Davis, Eds. *Academic Press*.

Wickens, C. D. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2), 159–177. <https://doi.org/10.1080/14639220210123806>

Wickens, C. D. (2024). The Multiple Resource Theory and Model. Some Misconceptions in Data Interpretations. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 10711813241260740. <https://doi.org/10.1177/10711813241260740>

Wickens, C. D., Hollands, J. G., Banbury, S., & Parasuraman, R. (2016). *Engineering psychology and human performance* (Fourth edition). Routledge, Taylor & Francis Group. <https://doi.org/10.4324/9781315665177>

Wickens, C. D., Vidulich, M., & Sandry, D. (1981). Factors Influencing the Performance Advantage of Speech Technology. *Proceedings of the Human Factors Society Annual Meeting*, 25(1), 705–709. <https://doi.org/10.1177/1071181381025001185>

Wills, C. (2016). *Unmanned Combat Air Systems in Future Warfare: Gaining Control of the Air*. Palgrave Macmillan UK. <https://doi.org/10.1057/9781137498496>

Wintersberger, P., Janotta, F., Peintner, J., Löcken, A., & Riener, A. (2021). Evaluating feedback requirements for trust calibration in automated vehicles. *It - Information Technology*, 63(2), 111–122. <https://doi.org/10.1515/itit-2020-0024>

Wolkov, N., Gasparyan, D., Hird, K., Mappes, G., & Barros, G. (2024). *Russian Offensive Campaign Assessment, August 26, 2024*. ISW (Institute for the Study of War). <https://www.understandingwar.org/report/russian-offensive-campaign-assessment-august-26-2024>

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115–152. <https://doi.org/10.1017/S0269888900008122>

Xie, J., Gao, L., Peng, K., Li, X., & Li, H. (2019). Review on flexible job shop scheduling. *IET Collaborative Intelligent Manufacturing*, 1(3), 67–77. <https://doi.org/10.1049/iet-cim.2018.0009>

Yamato, M., Inoue, K., Monden, A., Torii, K., & Matsumoto, K. (2000). Button selection for general GUIs using eye and hand together. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 270–273. <https://doi.org/10.1145/345513.345339>

Yannakakis, G. N., & Togelius, J. (2018). *Artificial intelligence and games*. Springer International Publishing AG. <https://doi.org/10.1007/978-3-319-63519-4>

Young, D. (2014). *Learning game AI programming with Lua: Leverage the power of Lua programming to create game AI that focuses on motion, animation, and tactics* (1st ed). Packt Publishing.

Zabcikova, M., Koudelkova, Z., Jasek, R., & Lorenzo Navarro, J. J. (2022). Recent advances and current trends in brain-computer interface research and their applications. *International Journal of Developmental Neuroscience*, 82(2), 107–123. <https://doi.org/10.1002/jdn.10166>

Zafra, M., Hunder, M., Rao, A., & Kiyada, S. (2024, March 26). How drone combat in Ukraine is changing warfare. *Reuters*. <https://www.reuters.com/graphics/UKRAINE-CRISIS/DRONES/dwpkeyjwkp/>

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems the CHI Is the Limit - CHI '99*, 246–253. <https://doi.org/10.1145/302979.303053>

Zühlke, D. (2012). *Nutzergerechte Entwicklung von Mensch-Maschine-Systemen: Ueware-Engineering für technische Systeme* (2., neu bearb. Auflage). Springer. <https://doi.org/10.1007/978-3-642-22074-6>

Zwerger, P. (2021). „Kein Gefühl“: F-35-Pilot hadert mit dem Touchscreen-Cockpit. <https://www.flugrevue.de/militaer/erfahrungsbericht-f-35-pilot-hadert-mit-dem-touchscreen-cockpit/>

B. Abkürzungsverzeichnis

Abkürzung	Bedeutung
AI	Air Interdiction
AWACS	Airborne Early Warning and Control System
BDA	Battle Damage Assessment
BOGEY	Potenziell feindliches Luftfahrzeug
C2	Command & Control
CAP	Combat Air Patrol
CCD	Cursor Control Device
COMAO	Composite Air Operation
DEAD	Destruction of Enemy Air Defences
DVI	Direct Voice Input
FCAS	Future Combat Air System
FJSP	Flexibles Job-Shop-Scheduling-Problem
FL	Flight Level
FLOT	Forward Line of Own Troops
FPV	First-Person-View
FSM	Finite-State-Machine
GPS	Global Positioning System
GUI	Graphical User Interface
HOTAS	Hands On Throttle And Stick
HUD	Head-up Display
MFD	Multifunction Display
MHDD	Multifunction Head Down Display
MRT	Multiple Resource Theory
MUM-T	Manned-Unmanned Teaming
MV	Missionsvignette
NGAD	Next Generation Air Dominance
OCA	Offensive Counter Air
RNN	Recursive Neural Network
ROE	Rules of Engagement
SAM	Surface to Air Missile
SCR	Stimulus – Central Processing – Response
SEAD	Suppression of Enemy Air Defences
SER	Sentence-Error-Rate
SHDD	Side Head Down Display
TOT	Time-over-Target
TTS	Text-to-Speech
UAV	Unmanned Aerial Vehicle
VR	Virtual Reality
WER	Word-Error-Rate

C. Abbildungsverzeichnis

ABBILDUNG 1. MISSIONSVIGNETTE A; A) MISSIONSÜBERSICHT B) KRÄFTEANSATZ & ROLLEN	4
ABBILDUNG 2. MISSIONSVIGNETTE B; A) MISSIONSÜBERSICHT B) KRÄFTEANSATZ & ROLLEN	5
ABBILDUNG 3. MISSIONSVIGNETTE C; A) MISSIONSÜBERSICHT B) KRÄFTEANSATZ & ROLLEN	6
ABBILDUNG 4. BESCHREIBUNG DER FÄHIGKEITEN VON MENSCH UND AUTOMATION NACH FITTS (1951).	8
ABBILDUNG 5. ENTSCHEIDUNGSMATRIX ZUR FUNKTIONSALLOKATION (VGL. PULLIAM ET AL. (1983) & JOE ET AL. (2015)).	9
ABBILDUNG 6. ANWENDUNG DES INTERAKTIONSMODELLS IM MILITÄRISCHEN KONTEXT. (VGL. PARASURAMAN ET AL. (2000) SOWIE LEGRAS & COPPIN (2007)).	11
ABBILDUNG 7. EINGANGS- UND AUSGANGSGRÖßEN EINES ARBEITSPROZESSES NACH ONKEN & SCHULTE (2010).	12
ABBILDUNG 8. NETZWERK AUS ARBEITSPROZESSEN (VGL. SCHULTE & DONATH (2018)).	13
ABBILDUNG 9. AKTEURE IM ARBEITSSYSTEM UND MÖGLICHE ROLLEN.	14
ABBILDUNG 10. A) HETERARCHISCHE BEZIEHUNG ZWISCHEN NUTZER UND WORKER-AGENT. B) HIERARCHISCHE BEZIEHUNG ZWISCHEN NUTZER UND TOOL-AGENT.	14
ABBILDUNG 11. A) ASSISTENZMUSTER B) DELEGATIONSMUSTER.	15
ABBILDUNG 12. SYMBOLISCHE UNTERSCHIEDUNG DER BEIDEN AUSPRÄGUNGEN DES FEEDBACK-MODUS.	16
ABBILDUNG 13. INTERAKTIONSMODALITÄTEN DES MENSCHEN (VGL. SHARMA ET AL. (1998)).	17
ABBILDUNG 14. VORHERSAGE DER AUFGABENLEISTUNG EINER EINZELAUFGABE IN ABHÄNGIGKEIT VON EIN- UND AUSGABEMODALITÄT NACH DEM SCR-KOMPATIBILITÄTSMODELL (VIDULICH & WICKENS, 1985).	20
ABBILDUNG 15. 4-DIMENSIONALES RESSOURCENMODELL NACH WICKENS (2024).	21
ABBILDUNG 16. VERSCHIEDENE EINGABEELEMENTE FÜR KAMPFFLUGZEUGCOCKPITS. A) BEDIENELEMENTE AM EUROFIGHTER- SCHUBHEBEL (ROT: MECHANISCHE TASTEN UND SCHALTER; GELB: FINGERSTICK ALS CCD) B) MECHANISCHE TASTEN ZUR OPTIONS-AUSWAHL AM DISPLAY-RAND C) BERÜHRUNGSEMPFINDLICHES DISPLAY	23
ABBILDUNG 17. MÖGLICHER HYBRIDER AGENTENAUFBAU (VGL. DENNIS & FISCHER (2023))	27
ABBILDUNG 18. MÖGLICHES NETZWERK AUS ARBEITSPROZESSEN ZUR DURCHFÜHRUNG EINER MISSION.	29
ABBILDUNG 19. WECHSELWIRKUNG ZWISCHEN ARBEITSPROZESS „AIR SUPERIORITY PACKAGE OPERATION“ UND ARBEITSOBJEKTEN IN DER MISSIONSPHASE INGRESS.	30
ABBILDUNG 20. INITIALES ARBEITSSYSTEMDESIGN OHNE KOGNITIVE AGENTEN.	31
ABBILDUNG 21. ARBEITSSYSTEMENTWURF UNTER NUTZUNG DER AUFTRAGSBASIERTEN FÜHRUNG.	32
ABBILDUNG 22. ARBEITSSYSTEM MIT SCHEDULER ZUR AUFGABENVERTEILUNG.	32
ABBILDUNG 23. ARBEITSSYSTEMENTWURF BEI NUTZUNG EINER ZIELBASIERTEN DELEGATIONSFORM.	33
ABBILDUNG 24. ARBEITSSYSTEMENTWURF MIT PROAKTIVER UNTERSTÜTZUNG DES NUTZERS. AUS GRÜNDEN DER ÜBERSICHTLICHKEIT SIND DIE BEZIEHUNGEN ZWISCHEN NUTZER UND PLATTFORMSTEUERUNGSAGENTEN SOWIE ZWISCHEN MISSIONSMANAGEMENTAGENT UND PLATTFORMSTEUERUNGSAGENTEN JEWEILS NUR EINMALIG DARGESTELLT.	34
ABBILDUNG 25. ENTWURFSMUSTER „ASSISTIERTE AUFTRAGSBASIERTE FÜHRUNG“. AUS GRÜNDEN DER ÜBERSICHTLICHKEIT SIND DIE BEZIEHUNGEN ZWISCHEN NUTZER UND PLATTFORMSTEUERUNGSAGENTEN SOWIE ZWISCHEN DELEGATIONSAGENT UND PLATTFORMSTEUERUNGSAGENTEN JEWEILS NUR EINMALIG DARGESTELLT.	35
ABBILDUNG 26. ACTOR-RELATIONSHIP-ACTOR TUPEL DER ASSISTIERTEN AUFTRAGSBASIERTEN FÜHRUNG.	36
ABBILDUNG 27. FUNKTIONALE ARCHITEKTUR DER ASSISTIERTEN AUFTRAGSBASIERTEN FÜHRUNG.	38
ABBILDUNG 28. PROZESSSCHRITTE DER AUFTRAGSDELEGATION.	41
ABBILDUNG 29. MÖGLICHE AUFTRAGSSPEZIFIKATION ZUR AUFKLÄRUNG EINES ZIELOBJEKTES.	41
ABBILDUNG 30. BEISPIEL EINES MISSIONSPLANES FÜR EINE BEMANNT-UNBEMANNTE FLIGHT.	42
ABBILDUNG 31. DELEGATION EINES AUFTRAGS OHNE REVIEW-SCHRITT (ROTER PFEIL) UND MIT REVIEW-SCHRITT (BLAUE PFEILE). .	43
ABBILDUNG 32. VERVOLLSTÄNDIGUNG DER AUFTRAGSSPEZIFIKATION DURCH DELEGATIONSAGENT.	44
ABBILDUNG 33. VERTEILUNG EINES AUFTRAGS DURCH DELEGATIONSAGENT.	45
ABBILDUNG 34. VERSCHIEDENE FORMEN DER ENTSCHEIDUNGSUNTERSTÜTZUNG A) KRITIKBASIERT B) VORSCHLAGSBASIERT.	46
ABBILDUNG 35. MODULE UND SCHNITTSTELLEN DES DELEGATIONSAGENTEN.	47
ABBILDUNG 36. AUFGABE DES PLATTFORMSTEUERUNGSAGENTEN.	49
ABBILDUNG 37. MODULE UND SCHNITTSTELLEN DES PLATTFORMSTEUERUNGSAGENTEN.	50
ABBILDUNG 38. KAMPFFLUGZEUGSIMULATOR DER PROFESSUR FÜR FLUGMECHANIK UND FLUGFÜHRUNG.	52
ABBILDUNG 39. ELEMENTE DES COCKPITS UND DER AUßENSICHT.	53
ABBILDUNG 40. ABLAUF VON DELEGATION MIT DER DELEGATIONSMETHODE TIMELINE.	56

ABBILDUNG 41. RADIAL-MENU NACH AUSWAHL EINES OBJEKTES (LINKS) UND ZWEITE EBENE NACH AUSWAHL DER OPTION "TASK" (RECHTS).....	57
ABBILDUNG 42. FENSTER ZUR MANUELLEN SPEZIFIKATION DES AUFLÄRUNGSAUFTRAGS.	57
ABBILDUNG 43. SCHALTFLÄCHEN ZUR AUSWAHL DES SCHEDULING-MODUS (ORANGE) UND ZUR BESTÄTIGUNG / ABLEHNUNG DES MISSIONSPLANES (GRÜN) IN MISSIONSPLANUNGSANSICHT.....	58
ABBILDUNG 44. ABLAUF DER VERSCHIEDENEN MODI DER AUFTRAGSDELEGATION MIT DER METHODE TOUCH-GESTURE	59
ABBILDUNG 45. AUFTRAGSDELEGATION MIT DELEGATIONSMETHODE VOICE. A) SPRACHBEFEHL B) PUSH-TO-TALK BUTTON.	61
ABBILDUNG 46. MODALITÄTEN UND DEREN VERWENDUNG IN DELEGATIONSMETHODE GAZE-HOTAS. A) BLICKBEWEGUNGSMESSUNG ZUR OBJEKTAUSWAHL B) KNÖPFE AUF DEN STEUERELEMENTEN ZUR STEUERUNG DES AUSWAHL- UND DES DELEGATIONSPROZESSES.	62
ABBILDUNG 47. ABLAUF VON DELEGATIONEN MIT METHODE GAZE-HOTAS.....	63
ABBILDUNG 48. A) SELEKTIONSKNOPF B) AUSWAHLMARKER C) ABRUCH-KNOPF D) WEITER-KNOPF.	64
ABBILDUNG 49. POLICY PAGE ZUR KONFIGURATION VON VERHALTENSRICHTLINIEN.....	66
ABBILDUNG 50. AUFTRAGSKRITIK NACH ABSCHLUSS DER NUTZERINTERAKTION ZUR AUFTRAGSDELEGATION.....	68
ABBILDUNG 51. AUFTRAGSKRITIK NACH DELEGATION EINES AUFLÄRUNGSAUFTRAGS MIT UNGEEIGNETER HÖHENVORGABE. DAS SYSTEM SCHLÄGT IN DIESEM FALL EINE ANPASSUNG DER AUFLÄRUNGSHÖHE VOR.	69
ABBILDUNG 52. HANDLUNGSFEEDBACK-ICONS IN VERSCHIEDENEN SITUATIONEN.....	70
ABBILDUNG 53. MELDUNG ÜBER AUTOMATISCHE KLASSIFIZIERUNG BEI FEINDESELIGER HANDLUNG.	72
ABBILDUNG 54. BEISPIELE FÜR EREIGNISMELDUNGEN. A) SPRACHMELDUNG BEI BESCHUSS B) POPUP-MELDUNG BEI ENTDECKUNG EINES BOGEYS.	72
ABBILDUNG 55. NUTZERDIALOG IN FOLGE VON PROBLEMEN BEI DER AUFLÄRUNG EINES GEBÄUDES.	73
ABBILDUNG 56. VERARBEITUNGSKETTE VON SPRACHBEFEHLEN.	74
ABBILDUNG 57. AKUSTIKMODELL-VOCODER VERFAHREN ZUR SPRACHSYNTHESE (VGL. BROWN, 2021).....	77
ABBILDUNG 58. ERZEUGUNG EINER BEOBACHTUNGSLISTE DURCH SAMPLING UM BLICKPOSITION.	79
ABBILDUNG 59. PROZESS ZUR VERVOLLSTÄNDIGUNG EINES AUFTRAGS.	82
ABBILDUNG 60. MODULARE ARCHITEKTUR DER VERHALTENSGENERIERUNG.	85
ABBILDUNG 61. FINITE-STATE-MACHINE ZUR AUFLÄRUNG EINES ZIELGEBIETS. DER ABLAUF DER FINITE-STATE-MACHINE WIRD DURCH DIE ÜBERGÄNGE ZWISCHEN DEN ZUSTÄNDEN DEFINIERT.	87
ABBILDUNG 62. DECISIONTREE ZUR AUFLÄRUNG EINES ZIELGEBIETS. DER ABLAUF DES DECISIONTREE WIRD DURCH DIE BEDINGUNGEN DEFINIERT.	87
ABBILDUNG 63. BEHAVIORTREE ZUR AUFLÄRUNG EINES ZIELGEBIETS. DER ABLAUF DES BEHAVIORTREE WIRD DURCH DEN ERFOLG/MISSERFOLG DER BLATTKNOTEN SOWIE DIE KONTROLLFLUSSKNOTEN BESTIMMT.	88
ABBILDUNG 64. PRINZIP DER TRENnung VON LOGISCHER UND PROGRAMMATISCHER EBENE.....	89
ABBILDUNG 65. ABSTRAKTE BESCHREIBUNG DES GEWÜNSCHTEN AGENTENVERHALTENS WÄHREND EINES „COMBAT AIR PATROL“-AUFTRAGS.	92
ABBILDUNG 66. ANWENDUNGSBEISPIELE VON SEQUENZEN UND FALLBACKS (VERHALTEN VEREINFACHT). A) SEQUENZ ZUR MODELLIERUNG DES VERHALTENS IM RAHMEN DER BEKÄMPFUNG EINER RADARSTELLUNG. B) FALLBACK ZUR MODELLIERUNG DER PRIORITÄTEN VON VERHALTENSWEISEN WÄHREND EINES „COMBAT AIR PATROL“-AUFTRAGS.....	93
ABBILDUNG 67. EINSATZ VON SUB-TREES IM BEHAVIORTREE. LINKS: MODELLIERUNG DES VERHALTENS ZUR AUFLÄRUNG MIT ZWEIFACHER VERWENDUNG EINES SUB-TREE ZUR OBJEKTAUFLÄRUNG. RECHTS: BAUMSTRUKTUR DES SUB-TREE ZUR OBJEKTAUFLÄRUNG.	94
ABBILDUNG 68. GRUNDSTRUKTUR DES IMPLEMENTIERTEN BEHAVIORTREE. DIE VERHALTENSWEISEN IM RAHMEN DER VERSCHIEDENEN AUFTRÄGE SIND IN SUB-TREES UNTERGEBRACHT – HIER VERDEUTLICHT AM BEISPIEL DES „DEAD“-AUFTRAGS.	94
ABBILDUNG 69. IMPLEMENTIERUNG VON BENUTZERDEFINIERTEN KNOTEN IM C++-CODE (VEREINFACHT). OBEN: KLASSENDEFINITION EINES KNOTEN ZUM ANFLIEGEN EINER ZIELPOSITION. UNTEN: VERKNÜPFUNG VON C++-KLASSENDEFINITION UND KNOTENTYP IM XML-DOKUMENT (REGISTRIERUNG).	95
ABBILDUNG 70. ERZEUGUNG UND AUSFÜHRUNG DER BEHAVIORTREE-INSTANZEN FÜR ALLE UNBEMANNTE TEAMMITGLIEDER (VEREINFACHT).	96
ABBILDUNG 71. SIMULATOR-ARCHITEKTUR MIT INTEGRIERTEN FUNKTIONEN.	97
ABBILDUNG 72. MISSIONSÜBERBLICK.....	105
ABBILDUNG 73. MISSIONSÜBERBLICK.....	116
ABBILDUNG 74. RÜCKMELDUNGEN ZU FRAGEN ZU UAV-FÜHRUNGSFUNKTIONEN.	118

ABBILDUNG 75. RÜCKMELDUNG ZUR BEREITSTELLUNG EINER DELEGATIONSMETHODE ZUR INSTANTANEN AUFTRAGSDELEGATION.	119
ABBILDUNG 76. RÜCKMELDUNGEN ZU ANWENDUNGSFÄLLEN VON INSTANTANER AUFTRAGSERTEILUNG.	120
ABBILDUNG 77. RÜCKMELDUNGEN ZUM HANDLUNGSFEEDBACK.....	121
ABBILDUNG 78. A) RÜCKMELDUNGEN ZUR EBENE DES HANDLUNGSFEEDBACKS B) RÜCKMELDUNGEN ZUR DARSTELLUNGSDAUER.	121
ABBILDUNG 79. RÜCKMELDUNGEN ZU VISUELLEN BENACHRICHTIGUNGEN IN DER TAKTISCHEN KARTE.	122
ABBILDUNG 80. RÜCKMELDUNGEN ZU AUDIOMELDUNGEN.	123
ABBILDUNG 81. RÜCKMELDUNGEN ZU NUTZERDIALOGEN.	124
ABBILDUNG 82. RÜCKMELDUNGEN ZUR EIGNUNG DER KRITIKFÄHIGKEIT ABHÄNGIG VON DER GEFÄHRDUNG DES MISSIONSZIELS. .	125
ABBILDUNG 83. RÜCKMELDUNGEN ZU FRAGEN ZUR MODELLIERUNGSGÜTE.....	126
ABBILDUNG 84. RÜCKMELDUNGEN ZUM VERHALTEN VON UAVS.	127
ABBILDUNG 85. RÜCKMELDUNG ZUR ZUVERLÄSSIGKEIT DER UAVS.	127
ABBILDUNG 86. RÜCKMELDUNGEN ZUR VERSTÄNDLICHKEIT DES VERHALTENS.....	128
ABBILDUNG 87. RÜCKMELDUNGEN ZUR VERTRAUTHEIT MIT UAVS.	128
ABBILDUNG 88. RÜCKMELDUNG ZUR INTENTION DER ENTWICKLER.	128
ABBILDUNG 89. RÜCKMELDUNGEN ZUR NEIGUNG ZU VERTRAUEN.	129
ABBILDUNG 90. RÜCKMELDUNGEN ZUM VERTRAUEN IN DIE UAVS.	129
ABBILDUNG 91. MISSIONSÜBERSICHT.....	134
ABBILDUNG 92. REIHENFOLGE DER VERSUCHSBEDINGUNGEN.	135
ABBILDUNG 93. SIMULATOR IN EXPERIMENTALKONFIGURATION.	138
ABBILDUNG 94. ZIELBEKÄMPFUNGSRATE NACH METHODE. DIE STICHPROBENGRÖÖE VON JEWEILS 24 PROBEN ERGIBT SICH AUS DREI MISSIONEN PRO METHODE UND PROBAND (KEINE DIFFERENZIERUNG NACH AUFGABENBELASTUNG).	139
ABBILDUNG 95. ANZAHL RICHTIG-POSITIVER REAKTIONEN NACH METHODE UND NEBENAUFGABE. DIE STICHPROBENGRÖÖE BETRÄGT FÜR ALLE BEDINGUNGEN ACHT PROBEN: JE EINE MISSION FÜR JEDE KOMBINATION AUS METHODE UND AUFGABENBELASTUNG PRO PROBAND.....	140
ABBILDUNG 96. REAKTIONSZEIT IN DER AUDITIV-VERBALEN NEBENAUFGABE NACH METHODE. FÜR JEDE METHODE LAG DIE THEORETISCH ERREICHBARE STICHPROBENGRÖÖE BEI 96 (12 KORREKTE STIMULI PRO MISSION, EINE MISSION JE PROBAND).	141
ABBILDUNG 97. VERWENDUNG VON DELEGATIONSMETHODEN IN MISSIONEN MIT FREIER WAHL OHNE NEBENAUFGABE (P4 UND P7 AUSGESCHLOSSEN, DA SIE NUR GAZE-HOTAS BZW. VOICE VERWENDETEN).....	142
ABBILDUNG 98. RÜCKMELDUNGEN ZU EIGENSCHAFTEN DER METHODEN.	143
ABBILDUNG 99. RÜCKMELDUNG ZU STÖRFAKTOREN (INVERTIERTE FARBBEGEBUNG BEACHTEN).....	143
ABBILDUNG 100. BERÜCKSICHTIGTE FAKTOREN BEI DER AUSWAHL EINER METHODE.	144
ABBILDUNG 101. AUSWIRKUNGEN DER BEREITSTELLUNG MEHRERER METHODEN.....	144
ABBILDUNG 102. A) BEISPIEL-CLUSTER B) GEEIGNETE ANGRIFFSKONFIGURATION MIT VISUALISIERUNG DES BEKÄMPFUNGSEFFEKTS; RECHTS: LEGENDE.	150
ABBILDUNG 103. A) SUBOPTIMALE LÖSUNG B) KRITIK-BENACHRICHTIGUNG C) VORKONFIGURIERTE LÖSUNG DES VORSCHLAGSSYSTEMS.....	151
ABBILDUNG 104. SIMULATOR-KONFIGURATION FÜR DIE STUDIE „KRITIKFÄHIGKEIT IN AUFTRAGSDELEGATION“.	154
ABBILDUNG 105. PRIMÄRZIEL-BEKÄMPFUNGSRATE NACH SYSTEMTYP.	155
ABBILDUNG 106. INTERAKTIONSDAUER NACH SYSTEMTYP UND FUNKTIONSFÄHIGKEIT.	156
ABBILDUNG 107. BEANSPRUCHUNG NACH SYSTEMTYP.	157
ABBILDUNG 108. RÜCKMELDUNGEN DER PROBANDEN ZUR BENUTZERFREUNDLICHKEIT DER SYSTEME.	157
ABBILDUNG 109. RÜCKMELDUNGEN ZUR ZUVERLÄSSIGKEIT DER SYSTEME.....	158
ABBILDUNG 110. RÜCKMELDUNGEN ZUR VERTRAUTHEIT MIT DEN SYSTEMEN.	159
ABBILDUNG 111. RÜCKMELDUNG ZUM VERTRAUEN IN DIE SYSTEME.	159
ABBILDUNG 112. RÜCKMELDUNGEN ZUR FEHLERERKENNUNG UND -BEHEBUNG.	160
ABBILDUNG 113. VERSCHIEDENE FORMEN UND KONTEXTE VON LOGISCHEN OBJEKTEN. DIE LOGISCHEN OBJEKTE WERDEN ABHÄNGIG VOM KONTEXT IN UNTERSCHIEDLICHEN FARBEN DARGESTELLT.....	195
ABBILDUNG 114. ELEMENTE VON BEFEHLEN DER DELEGATIONSMETHODE VOICE.	198
ABBILDUNG 115. ÜBERSICHT ÜBER DEN BEHAVIOR TREE ZUR ABLAUFSTEUERUNG.....	206
ABBILDUNG 116. MISSIONSROLLEN DER TEAMMITGLIEDER.	208
ABBILDUNG 117. MISSIONSROLLEN DER TEAMMITGLIEDER.	211
ABBILDUNG 118. PENETRATION DER LUFTVERTEIDIGUNG ENTLANG VORGESCHLAGENER TIEFFLUGROUTE.	212

ABBILDUNG 119. LAUNCHER IM SENSORBILD..... 213

D. Tabellenverzeichnis

TABELLE 1. DIE INITIALE DEFINITION DER AUTOMATIONSGRAD (FREI ÜBERSETZT NACH SHERIDAN & VERPLANK (1978)).	10
TABELLE 2. ARTEN VON AKTEURS-BEZIEHUNGEN NACH SCHULTE ET AL. (2016).	14
TABELLE 3. BESCHREIBUNG UND SYMBOLIK DER AUSPRÄGUNGEN DER DELEGATIONSFORM.	16
TABELLE 4. ARBEITZIELE, UMWELTBEDINGUNGEN UND AUSGABEGRÖßEN, DIE SICH AUS DEN MISSIONEN ERGEBEN.	30
TABELLE 5. AKTEURE IM ENTWURFSMUSTER DER ASSISTIERTEN AUFTRAGSBASIERTE FÜHRUNG.	36
TABELLE 6. AUFTRAGSTYPEN, ERFOLGSKRITERIEN UND PARAMETER AUSGEWÄHLTER AUFTRÄGE.	40
TABELLE 7. SCHEDULING-MODI ZUR AUFTRAGSVERTEILUNG (VGL. HEILEMANN & SCHULTE (2020B)).	45
TABELLE 8. ÜBERBLICK ÜBER DIE WESENTLICHEN EIGENSCHAFTEN DER VERSCHIEDENEN DELEGATIONSMETHODEN.	55
TABELLE 9. DELEGATIONSMODI MIT DER METHODE TOUCH-GESTURE.	59
TABELLE 10. DELEGATIONSMODI DER METHODE VOICE.	60
TABELLE 11. SYNTAX UND BEISPIELE VON SPRACHBEFEHLEN FÜR DIE VERSCHIEDENEN MODI.	61
TABELLE 12. DELEGATIONSMODI DER METHODE GAZE-HOTAS.	63
TABELLE 13. BESCHREIBUNG DER EINSTELLBAREN IDLE-VERHALTENSWEISEN.	65
TABELLE 14. KATEGORISIERUNG VON ANWENDUNGSFÄLLEN DER AUFTRAGSKRITIK.	67
TABELLE 15. MÖGLICHE FÄLLE EINER UNGEEIGNETEN AUFTRAGSSPEZIFIKATION.	68
TABELLE 16. BEISPIELE FÜR FORTSCHRITTMELDUNGEN UND DEREN BEDEUTUNG.	71
TABELLE 17. IMPLEMENTIERTE ANWENDUNGSFÄLLE VON NUTZERDIALOGEN.	73
TABELLE 18. ELEMENTE DES JOB-SHOP-SCHEDULING-PROBLEMS UND MODELLIERUNG DES PROBLEMS ZUR MISSIONSPLANUNG.	81
TABELLE 19. ERMITTLUNG DES AUFTRAGSTYP ANHAND DER EIGENSCHAFTEN DES ZIELOBJEKTES.	82
TABELLE 20. VERSCHIEDENE ARTEN DER ERMITTLUNG VON PARAMETERN.	83
TABELLE 21. IMPLEMENTIERTES VORGEHEN ZUR ERKENNUNG UND LÖSUNG VON PROBLEMFÄLLEN.	84
TABELLE 22. UNTERSUCHUNGEN IM RAHMEN DER EXPERIMENTELLEN VALIDIERUNG.	100
TABELLE 23. BEFRAGUNGSZEITPUNKTE UND ANALYSIERTE KONSTRUKTE (VGL. PARASURAMAN & RILEY (1997)).	106
TABELLE 24. DEMOGRAFISCHE DATEN DER PROBANDEN.	106
TABELLE 25. BESCHREIBUNG UND HÄUFIGKEITEN DER NUTZUNG VERSCHIEDENER FUNKTIONEN.	108
TABELLE 26. BESCHREIBUNG UND HÄUFIGKEITEN VON FEHLERFÄLLEN BEIM EINSATZ VON FÜHRUNGSFUNKTIONEN.	109
TABELLE 27. HÄUFIGKEIT UND BESCHREIBUNG VON FÄLLEN DER NICHTNUTZUNG.	110
TABELLE 28. DEMOGRAFISCHE DATEN DER PROBANDEN.	117
TABELLE 29. DEMOGRAFISCHE DATEN DER PROBANDEN.	137
TABELLE 30. ANZAHL BEKÄMPFTER FALSCHZIELE NACH METHODE (OHNE ERSTE MISSION).	140
TABELLE 31. VERWENDUNGSHÄUFIGKEIT DER METHODEN NACH AUFGABENBELASTUNG.	141
TABELLE 32. MEDIAN DER DELEGATIONSZEITEN NACH METHODE BEI BETRACHTUNG ALLER DATEN UND BEI BETRACHTUNG DES SCHNELLSTEN ZEHNTELS JEDES PROBANDEN.	145
TABELLE 33. BESCHREIBUNG DER CLUSTER-TYPEN.	150
TABELLE 34. VERHALTEN VON KRITIK- UND VORSCHLAGSSYSTEM ABHÄNGIG VON DER FUNKTIONSFÄHIGKEIT.	152
TABELLE 35. ANZAHL AN PROBEN (MÖGLICHE KONFIGURATIONEN) FÜR JEDES SYSTEM.	152
TABELLE 36. ANZAHL FALSCHER KONFIGURATIONEN NACH SYSTEMTYP UND FUNKTIONSFÄHIGKEIT.	155
TABELLE 37. ÜBERSICHT ÜBER ARTEN, ERFOLGSKRITERIEN UND PARAMETER VON AUFTRÄGEN.	196
TABELLE 38. BENENNUNGEN BEMANNTER UND UNBEMANNTER PLATTFORMEN.	198
TABELLE 39. BENENNUNGEN WEITERER OBJEKTE.	198
TABELLE 40. ABLEITUNG DES AUFTRAGSTYP ANHAND DES ZUGEHÖRIGEN SPRACHBEFEHLS.	199
TABELLE 41. ERMITTLUNG DES AUFTRAGSTYP BEI PHYSISCHEN OBJEKTEN.	200
TABELLE 42. ERMITTLUNG DES AUFTRAGSTYP BEI LOGISCHEN OBJEKTEN.	200
TABELLE 43. ERMITTLUNG DES ERFOLGSKRITERIUMS ANHAND DES AUFTRAGSTYP.	201
TABELLE 44. VERHALTEN WÄHREND VERSCHIEDENER AUFTRAGSTYPEN IM KONTEXT VON OFFENSIVE COUNTER AIR.	203
TABELLE 45. VERHALTEN IM RAHMEN DES SUB-TREE "SEADTask".	203
TABELLE 46. VERHALTEN IM SUB-TREE "NAVTask".	204
TABELLE 47. VERHALTEN IM SUB-TREE "INVESTIGATETask".	204
TABELLE 48. VERHALTEN IM SUB-TREE "EATask".	205
TABELLE 49. ÜBERSICHT ÜBER DIE MISSIONSZIELE UND DIE RANDBEDINGUNGEN DER MISSION.	207

Tabellenverzeichnis

TABELLE 50. ABLAUF DER STRIKE/CLOSE-AIR-SUPPORT MISSION.	209
TABELLE 51. ÜBERSICHT ÜBER DIE MISSIONSZIELE UND DIE RANDBEDINGUNGEN DER MISSION.	211
TABELLE 52. ABLAUF DER MISSION.	212

E. Anhang

E.1 Logische Objekte

Das Interaktionskonzept des Kampfflugzeugsimulators ermöglicht es dem Nutzer logische Objekte anzulegen, die vielseitig eingesetzt werden können. So können beispielsweise relevante Informationen wie Missionsgebiete visualisiert werden. Die logischen Objekte können außerdem als Zielobjekte für Aufträge genutzt werden. Im Kontext der Auftragsdelegation kann dies beispielsweise genutzt werden, um UAVs zu einer beliebigen Zielposition zu befehligen oder um Zielgebiete zu erstellen, innerhalb derer Feinde aufgeklärt oder bekämpft werden sollen.

Logische Objekte können unterschiedliche geometrische Formen wie Punkte, Linien, Kreise oder komplexe Flächen mit n-Ecken annehmen. Ergänzend wird jedes Objekt durch einen sogenannten Kontext definiert, der die Funktion des Objekts in der Mission beschreibt. Dabei werden die Kontexte Navigation, Air Interdiction, Offensive Counter Air und Suppression of Enemy Air Defences unterschieden.⁷⁸ Eine Übersicht über verschiedene Formen und Kontexte von logischen Objekten ist in Abbildung 113 dargestellt.

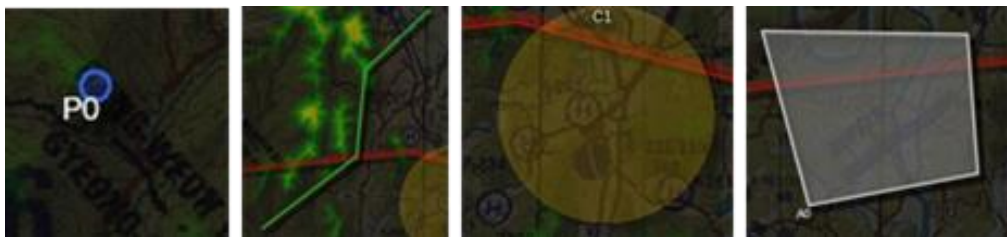


Abbildung 113. Verschiedene Formen und Kontexte von logischen Objekten.
Die logischen Objekte werden abhängig vom Kontext in unterschiedlichen Farben dargestellt.

⁷⁸ Der Kontext entspricht folglich den Kategorien von Aufträgen (wie in [3.3.2](#) beschrieben).

E.2 Liste von UAV-Aufträgen

Die folgende Tabelle 37 bietet eine vollständige Übersicht über alle Auftragsarten sowie deren mögliche Erfolgskriterien und Parameter.

Tabelle 37. Übersicht über Arten, Erfolgskriterien und Parameter von Aufträgen.

	Auftragstyp	Beschreibung	Erfolgskriterien	Parameter
Navigation	Fly-Over	Überfliegen eines Wegpunktes oder einer Serie von Wegpunkten.	<ul style="list-style-type: none"> (Letzter) Wegpunkt erreicht 	<ul style="list-style-type: none"> Flughöhe
	Loiter	Warteschleifen in einem Gebiet oder in der Nähe eines Punktes.	<ul style="list-style-type: none"> Zeit verstrichen 	<ul style="list-style-type: none"> Flughöhe
Air Interdiction	Investigate	Aufklärung eines einzelnen bodengebundenen Objektes.	<ul style="list-style-type: none"> Ein Bild Vier Bilder über einen Winkel von 90 Grad Vier Bilder über 360 Grad verteilt 	<ul style="list-style-type: none"> Depressionswinkel⁷⁹ Flughöhe Sensortyp
	Battle Damage Assessment (BDA)	Nachaufklärung eines bodengebundenen Objektes.	<ul style="list-style-type: none"> s. Investigate 	<ul style="list-style-type: none"> s. Investigate
	Scan	Aufklärung eines Gebiets zur Suche nach Objekten.	<ul style="list-style-type: none"> Area abgeflogen Erstes Objekt gefunden 	<ul style="list-style-type: none"> Anzahl und Richtung der Bilder Depressionswinkel Flughöhe Sensortyp
	Engage	Bekämpfung eines Bodenziels.	<ul style="list-style-type: none"> Payload ausgebracht Verifizierte Zerstörung 	<ul style="list-style-type: none"> Abwurfdistanz Angriffsrichtung Anzahl Effektoren Zünder
Offensive Counter Air	Combat Air Patrol	Identifizierung von Luftfahrzeugen und Bekämpfung von Feinden in der Nähe des CAPs.	<ul style="list-style-type: none"> Zeit verstrichen 	<ul style="list-style-type: none"> Taktik (Defensiv/Offensiv)
	Commit	Bekämpfung eines feindlichen Luftfahrzeugs.	<ul style="list-style-type: none"> Verifizierte Zerstörung 	<ul style="list-style-type: none"> Taktik (Defensiv / Offensiv)
	Vis-Ident	Identifizierung eines Luftfahrzeugs.	<ul style="list-style-type: none"> 1 Bild 4 Bilder 	<ul style="list-style-type: none"> -

⁷⁹ Dieser Parameter beschreibt den Winkel zwischen der Horizontalen und der Sensorachse, unter dem die Bildaufnahme erfolgt. Ein Winkel von 90° entspricht einer Aufnahme senkrecht von oben.

Suppression of Enemy Air Defences	Dead	Zerstörung einer Radarstellung.	<ul style="list-style-type: none"> • Payload ausgebracht • Verifizierte Zerstörung 	<ul style="list-style-type: none"> • Angriffsrichtung • Anzahl Effektoren • Flughöhe
	Area Suppression	Unterdrückung von Radarstellungen im Zielgebiet.	<ul style="list-style-type: none"> • Zeit verstrichen 	<ul style="list-style-type: none"> • Flughöhe • Taktik (Präventiv oder Reaktiv) <p>Falls präventiv:</p> <ul style="list-style-type: none"> • Auslöse-Intervall von Anti-Radiation-Missiles <p>Falls reaktiv:</p> <ul style="list-style-type: none"> • Anzahl Effektoren pro Ziel
	Escort Jamming	Eskortieren eines verbündeten Luftfahrzeugs.	<ul style="list-style-type: none"> • Zeit verstrichen 	<ul style="list-style-type: none"> • Jamming-Typ • Positionierung um Teammitglied • Verhalten bei Einflug in Radarstellung
	Target Jamming	Elektronischer Angriff auf Radarstellung.	<ul style="list-style-type: none"> • Zeit verstrichen 	<ul style="list-style-type: none"> • Jamming-Typ • Flughöhe
Mehrzweck	Swarm Release	Ausbringen eines Schwarms.	<ul style="list-style-type: none"> • Payload ausgebracht 	<ul style="list-style-type: none"> • Abwurfdistanz • Anzahl Plattformen • Flughöhe

E.3 Vokabular der Delegationsmethode Voice

Im Folgenden werden die verschiedenen Wörter dargestellt, die im Kontext von Sprachbefehlen verwendet werden können. Als Beispiel hierfür wird ein Sprachbefehl zur Delegation eines Auftrages mit Reihenfolgenvorgabe genutzt, da dieser Befehlstyp alle Elemente enthält, die in einem Sprachbefehl enthalten sein können (Abbildung 114).

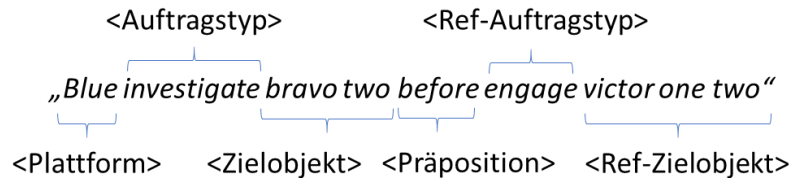


Abbildung 114. Elemente von Befehlen der Delegationsmethode Voice.

Beim Befehlselement „Plattform“ wird das Vokabular durch die möglichen Namen der bemannten und unbemannten Plattformen definiert. Unbemannte Plattformen werden zur einfacheren Zuordnung nach Farben benannt, bemannte Plattformen erhalten einen Rufnamen (Tabelle 38).

Tabelle 38. Benennungen bemannter und unbemannter Plattformen.

Plattformtyp	Namen
Bemannt	<ul style="list-style-type: none"> • Raven • Eagle
Unbemannt	<ul style="list-style-type: none"> • Blue • Orange • White • Brown • Green • Olympic

Beim Befehlselement „Zielobjekt“ wird das Vokabular ebenfalls durch die Benennung der jeweiligen Zielobjekte bestimmt. Dieser Name folgt bei allen Objekten⁸⁰ einem festen Schema und besteht immer aus einem Buchstaben und bis zu zwei Ziffern. Der Buchstabe wird im Sprachbefehl durch das NATO-Alphabet repräsentiert. Beispiele für Benennungen von Objekten und die zugehörigen Befehle sind in Tabelle 39 dargestellt.

Tabelle 39. Benennungen weiterer Objekte.

Objekttyp	Beispiel-Benennung	Beispiel-Befehlselement
Gebäude (Building)	B1	„Bravo one“
Radar	R21	„Romeo two one“
Flugabwehr (SAM)	S6	„Sierra six“

Das Befehlselement „Präposition“ besteht aus zwei Wörtern („Before“ & „After“)

⁸⁰ Mit Ausnahme der Plattformen des eigenen Teams (wie zuvor beschrieben).

Das Befehlselement „Auftragstyp“ umfasst mehrere Begriffe. Dabei wird nicht jeder Auftragstyp durch einen eigenen Begriff abgebildet, da Begriffe in der Sprache je nach Kontext unterschiedliche Bedeutungen haben können. Tabelle 40 zeigt die kontextabhängige Zuordnung von Auftragstypen zu Sprachbefehlen.

Tabelle 40. Ableitung des Auftragstyps anhand des zugehörigen Sprachbefehls.

Sprachbefehl	Zielobjekt	Auftragstyp
„Engage“	Gebäude & Fahrzeuge	Engage
	Radare und Flugabwehrstellungen	Dead
	Kampfflugzeuge	Commit
	Area (Sead Kontext)	Area Suppression
	Linien und Areas (Offensive Counter Air Kontext)	Combat Air Patrol
„Investigate“	Gebäude, Fahrzeuge & Punkte (Air Interdiction Kontext)	Investigate
	Area (Air Interdiction Kontext)	Scan
„Assess“	Gebäude & Fahrzeuge	Battle Damage Assessment
„Fly“	Punkt (Navigation Kontext)	Fly Over
„Marshall“	Circle (Navigation Kontext)	Loiter

Die Elemente des Sprachbefehls, die den Referenz-Auftragstyp und das Referenz-Zielobjekt angeben, verwenden das selbe Vokabular wie der Auftragstyp resp. das Zielobjekt.

E.4 Auftragsvervollständigung

E.4.1 Ermittlung des Auftragstyps

Im Folgenden wird erläutert, wie der Delegationsagent den Auftragstyp während der Auftragsvervollständigung anhand der Eigenschaften des Zielobjekts auswählt.

Für physische Objekte wird diese Auswahl anhand des Objekttyps, der Zugehörigkeit und des Status getroffen (Tabelle 41).

Tabelle 41. Ermittlung des Auftragstyp bei physischen Objekten.

Einflussfaktoren			Auftragstyp
Objekttyp	Zugehörigkeit	Status	
Gebäude & Fahrzeuge	Unbekannt	Unbekannt	Investigate
	Verbündet & Neutral	Alle	Investigate
	Feindlich	Funktional	Engage
		Bekämpft	BDA
Flugzeuge	Unbekannt	Unbekannt	Vis-Ident
	Feindlich	Funktional	Commit
	Verbündet	Funktional	Escort-Jamming
Radare & Flugabwehrstellungen	Feindlich	Funktional	Dead
		Bekämpft	BDA
	Verbündet oder Neutral	Alle	Investigate

Bei logischen Objekten⁸¹ sind der Objekttyp und der Kontext des logischen Objektes für die Auswahl des Auftragstypen entscheidend (Tabelle 42).

Tabelle 42. Ermittlung des Auftragstyps bei logischen Objekten.

Einflussfaktoren		Auftragstyp
Objekttyp	Kontext	
Punkt	Navigation	Fly Over
	Air Interdiction	Investigate
	Suppression of Enemy Air Defences	Dead
Linie	Navigation	Fly Over
	Offensive Counter Air	Combat Air Patrol
Circle	Navigation	Loiter
Area	Air Interdiction	Scan
	Suppression of Enemy Air Defences	Area Suppression
	Offensive Counter Air	Combat Air Patrol

⁸¹ Eine Erläuterung von logischen Objekten findet sich in Anhang E.1.

E.4.2 Ermittlung des Erfolgskriteriums

Im Folgenden wird beschrieben, wie im Rahmen der Auftragsvervollständigung das Erfolgskriterium ausgewählt wird. Der entscheidende Faktor hierfür ist der Auftragstyp (Tabelle 43).

Tabelle 43. Ermittlung des Erfolgskriteriums anhand des Auftragstyps.

Auftragstyp	Erfolgskriterium
Fly Over	Letzter Wegpunkt abgeflogen
Loiter	Zeit verstrichen
Investigate	4 Bilder über einen Winkel von 90°
Battle Damage Assessment	4 Bilder über einen Winkel von 90°
Scan	Area abgeflogen
Engage	Payload ausgebracht
Combat Air Patrol	Zeit verstrichen
Commit	Verifizierte Zerstörung
Vis-Ident	1 Bild
Dead	Payload ausgebracht
Area Suppression	Zeit verstrichen
Escort Jamming	Zeit verstrichen
Target Jamming	Zeit verstrichen
Swarm Release	Payload ausgebracht

E.5 Modellierung des BehaviorTree und resultierendes Agentenverhalten

Dieser Abschnitt befasst sich ausführlich mit der Modellierung des BehaviorTrees und dem daraus resultierenden Agentenverhalten. Zur Veranschaulichung wird das Verhalten des Agenten im Folgenden anhand der grafischen Repräsentation des BehaviorTrees in Abbildung 115 erläutert.

Auf oberster Ebene des BehaviorTree steht ein ReactiveFallback.⁸² Die Anordnung der Kindknoten bewirkt, dass der BehaviorTree den Selbstschutz priorisiert, gefolgt von der Auftragsdurchführung. Falls kein defensives Verhalten notwendig ist und auch kein Auftrag durchgeführt werden soll, führt der BehaviorTree die Idle-Verhaltensweisen aus, die mithilfe der Verhaltensrichtlinien (4.2.2) spezifiziert wurden. Im Folgenden werden diese drei Segmente der Baumstruktur näher erläutert.

Selbstschutz

Der Selbstschutz wird vom Sub-Tree „DefensivelfNecessary“ wahrgenommen. Dieser Sub-Tree überprüft, ob der Raketenwarnempfänger eine ankommende Rakete wahrnimmt. Falls dies der Fall ist, wird der Nutzer informiert und es werden Schutzmaßnahmen eingeleitet, die das Ausstoßen von Chaffs & Flares (Deutsch: Düppel und Täuschkörper) sowie kinematische Ausweichbewegungen beinhalten. Falls der Nutzer mithilfe von Verhaltensrichtlinien spezifiziert hat, dass die Auftrags Erfüllung Priorität gegenüber dem Selbstschutz hat, entfallen die Ausweichbewegungen.

Auftragsdurchführung

Die Auftragsdurchführung wird von der ReactiveSequence „ExecuteTaskIfExists“ übernommen. Im Rahmen dieser Sequenz wird zunächst überprüft, ob die Plattform einen Auftrag zugewiesen hat (Bedingungsknoten „HasTask“). Falls dies der Fall ist, wird im Knoten „CheckTaskConditions“ überprüft, ob die Erfolgskriterien des Auftrags erfüllt sind. Der Knoten generiert außerdem Meldungen, falls der Auftrag nicht erfüllt werden kann oder sich die Bearbeitung des Auftrags verzögert (z. B. aufgrund geänderter Randbedingungen). Im Anschluss wird mithilfe des Switch-Knotens der entsprechende Sub-Tree für den jeweiligen Auftragstyp ausgeführt.

⁸² Eine Beschreibung der Knotentypen ist in der Dokumentation des BehaviorTree-Frameworks zu finden (<https://www.behaviortree.dev/>). Diese Arbeit basiert auf Version 3.3.0.

Der Sub-Tree „OCATask“ regelt den Ablauf aller Aufträge im Kontext von Offensive Counter Air. Das Verhalten des Agenten ist dabei abhängig vom Auftragstyp (Tabelle 44).

Tabelle 44. Verhalten während verschiedener Auftragstypen im Kontext von Offensive Counter Air.

Auftragstyp	Verhalten
Combat Air Patrol	Der Agent führt eine der folgenden Verhaltensweisen aus (Zuerst genannte haben Priorität): <ol style="list-style-type: none"> 1. Bekämpfung feindlicher Luftfahrzeuge in der Nähe des CAP 2. Aufklärung von nicht klassifizierten Luftfahrzeugen in der Nähe 3. Anfliegen des CAPs 4. „Racetrack“-Pattern am CAP abfliegen
Commit	Der Agent führt folgende Schritte aus: <ul style="list-style-type: none"> • Annäherung an feindliches Luftfahrzeug • Bekämpfung des Luftfahrzeugs
Vis-Ident	Der Agent führt folgende Schritte aus: <ul style="list-style-type: none"> • Annäherung an nicht klassifiziertes Luftfahrzeug • Generierung von Bildern des Luftfahrzeugs

Im Sub-Tree „EngageTask“ ist das Verhalten des Agenten während der Durchführung des gleichnamigen Auftragstyps definiert. Der Ablauf der Auftragsdurchführung ist wie folgt: Der Agent steuert einen Anflugpunkt⁸³ an, von dem aus das Zielobjekt in Waffenreichweite ist, und greift es dann an.

Der Sub-Tree „SeadTask“ definiert das Verhalten im Rahmen von Aufträgen vom Typ „Dead“ und „Area Suppression“. Das Verhalten des Agenten unterscheidet sich nach Auftragstyp (Tabelle 45).

Tabelle 45. Verhalten im Rahmen des Sub-Tree "SeadTask".

Auftragstyp	Verhalten
Dead	Der Agent führt folgende Schritte aus: <ul style="list-style-type: none"> • Anfliegen der Radarstellung • Bekämpfung der Radarstellung mit radarsuchender Rakete
Area Suppression	Der Ablauf hängt von der vorgegebenen Taktik ab. <p>Bei Taktik „Präventiv“ werden folgende Schritte ausgeführt:</p> <ul style="list-style-type: none"> • Anfliegen des Zielgebietes • Regelmäßiges Auslösen von radarsuchenden Raketen entsprechend des vorgegebenen Zeitintervalls <p>Bei Taktik „Reaktiv“ ist der Ablauf wie folgt:</p> <ul style="list-style-type: none"> • Anfliegen des Zielgebietes • Überfliegen des Zielgebietes und Sensieren von Radarstellungen • Bekämpfung aller sensierten Radarstellungen

⁸³ Die Position dieses Anflugpunktes hängt davon ab, welche Angriffsrichtung der Nutzer vorgegeben hat. Falls der Nutzer keine Vorgabe gemacht hat, wählt der Agent die nächstgelegene Position in Reichweite.

Der Sub-Tree „NavTask“ definiert das Verhalten in Aufträgen vom Typ „Fly Over“ und „Loiter“. In Tabelle 46 ist dieses Verhalten beschrieben.

Tabelle 46. Verhalten im Sub-Tree "NavTask".

Auftragstyp	Verhalten
Fly Over	Das Agentenverhalten hängt von der Art des logischen Objektes ab: <ul style="list-style-type: none"> • Punkt: Überfliegen des Punktes • Linie: Vollständiges Abfliegen der Linie
Loiter	Das Verhalten hängt von der Art des logischen Objektes ab: <ul style="list-style-type: none"> • Punkt: Umkreisen des Punktes • Kreis/sonstige Fläche: Warteschleifen innerhalb der Fläche

Das Verhalten des Agenten im Rahmen von Aufklärungsaufträgen („Investigate“, „Scan“ und „Battle Damage Assessment“) ist im Sub-Tree „InvestigateTask“ beschrieben. Tabelle 47 beschreibt diese Verhaltensweisen.

Tabelle 47. Verhalten im Sub-Tree "InvestigateTask".

Auftragstyp	Verhalten
Investigate & Battle Damage Assessment	Der Agent führt folgende Schritte aus: <ul style="list-style-type: none"> • Anfliegen des Zielobjektes • Umkreisen⁸⁴ des Zielobjektes und Aufnahme von Aufklärungsbildern
Scan	Der Agent führt folgende Schritte aus: <ul style="list-style-type: none"> • Anfliegen des Zielgebietes • Systematisches Abfliegen des Zielgebietes • Falls ein Objekt gefunden wird: Umkreisen des Zielobjektes und Aufnahme von Aufklärungsbildern

⁸⁴ Die Umkreisung wird in der vorgegebenen Höhe durchgeführt. Der Radius des Kreises hängt von der Höhe und dem vorgegebenen Depressionswinkel ab.

Im Sub-Tree „EATask“ wird das Verhalten in den Aufträgen der elektronischen Kampfführung („Target Jamming“ und „Escort Jamming“) beschrieben. Das Verhalten des Agenten kann anhand von Tabelle 48 nachvollzogen werden.

Tabelle 48. Verhalten im Sub-Tree "EATask".

Auftragstyp	Verhalten
Target Jamming	Der Agent führt folgende Schritte aus: <ul style="list-style-type: none"> • Anfliegen der Flugabwehr- bzw. Radarstellung bis zum Beginn des Wirkungsbereiches • Bewegung entlang der Grenzen des Wirkungsbereiches und Jamming des Zielobjektes
Escort Jamming	Der Agent führt folgende Schritte aus: <ul style="list-style-type: none"> • Anfliegen des zu unterstützenden Teammitglieds • Einnehmen der vorgegebenen Formationsposition. Falls die Position bereits besetzt ist, wird eine andere Position eingenommen • Jamming zum Schutz des Teammitglieds (Escort Jamming) • Falls das unterstützte Teammitglied in den Wirkungsbereich einer Flugabwehr- bzw. Radarstellung fliegt, hängt das Verhalten vom Parameter „Verhalten bei Einflug in Radarstellung“ ab. Der Agent zeigt dabei eine der folgenden Verhaltensweisen: <ul style="list-style-type: none"> ○ Beibehalten der Formation & Escort Jamming ○ Target Jamming der Radarstellung wie oben beschrieben

Das Verhalten des Agenten im Rahmen der Durchführung eines „Swarm Release“ wird im gleichnamigen Sub-Tree definiert. Dabei steuert der Agent einen Absetzpunkt⁸⁵ an und stößt am Absetzpunkt den Schwarm aus.

Idle

Der Sub-Tree „Idle“ regelt das Verhalten des Agenten, wenn kein Auftrag vorliegt. Dieses Verhalten orientiert sich an den vom Nutzer vorgegebenen Idle-Verhaltensrichtlinien, die ausführlich in Abschnitt [4.2.2](#) beschrieben sind.

⁸⁵ Die Position des Absetzpunktes hängt von der Art des Schwarms ab. Bei Schwärmen zur Unterdrückung von Radarstellungen wird der Schwarm außerhalb des Wirkungsbereiches der Radarstellung ausgebracht, während Aufklärungs-Schwärme im Zielgebiet ausgebracht werden.

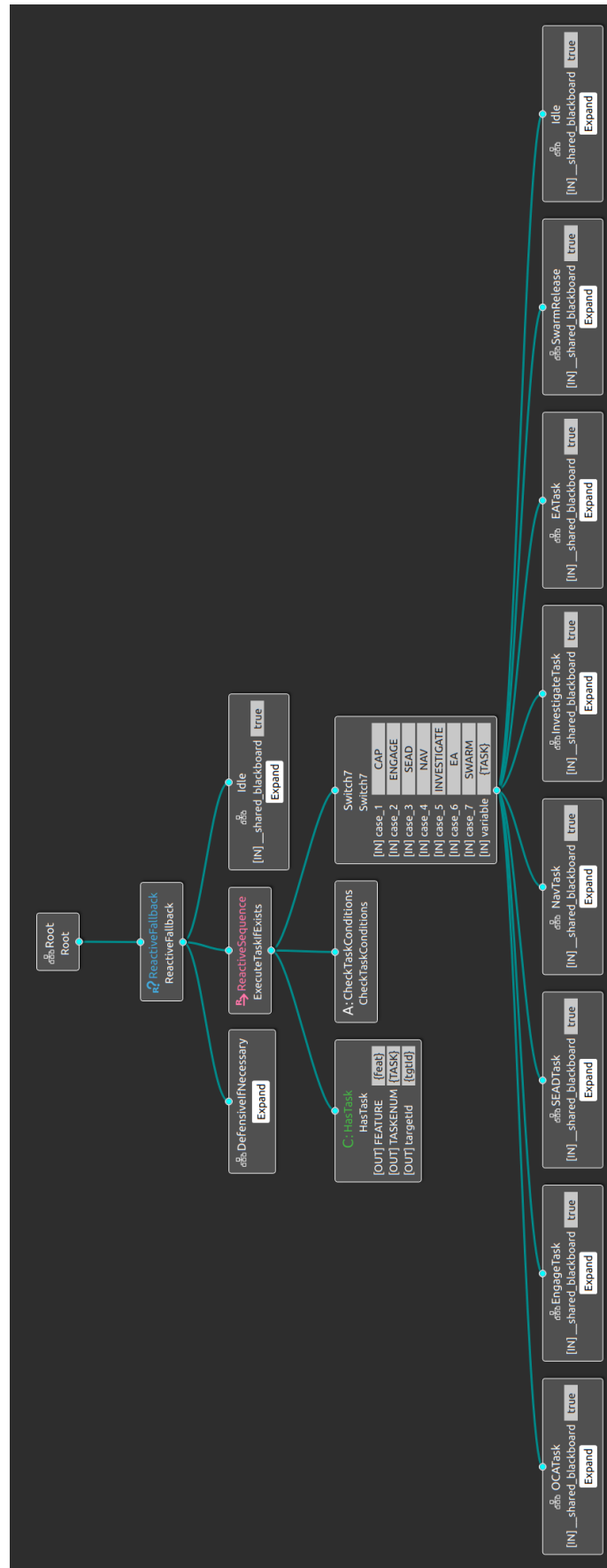


Abbildung 115. Übersicht über den BehaviorTree zur Ablaufsteuerung.

E.6 Anhang zu Validierungsstudien

E.6.1 Detaillierte Missionsbeschreibung zur Studie „Nutzung von Führungsfunktionen“

Dieser Abschnitt bietet eine vollständige Missionsbeschreibung der in Abschnitt 5.2 beschriebenen Studie zur Nutzung von Führungsfunktionen.

Missionsziel

Ziel der Mission war zunächst, eine verdeckte Bekämpfung von zwei ballistischen Raketenabschussvorrichtungen durchzuführen. Nachdem dieses Missionsziel erreicht wurde, erhielten die Probanden ein weiteres Missionsziel: Die Unterstützung von Bodentruppen bei der Durchführung einer Operation (Luftnahunterstützung; Tabelle 49).

Tabelle 49. Übersicht über die Missionsziele und die Randbedingungen der Mission.

Missions-Element	Beschreibung
Primärziele	<ol style="list-style-type: none"> 1. Bekämpfung von zwei ballistischen Abschussvorrichtungen (Launcher): <ul style="list-style-type: none"> • Bekämpfung eines voraufgeklärten Launchers auf feindlichem Militärflughafen (nachfolgend <i>Air Base Launcher</i> genannt). • Aufspüren einer zweiten Abschussvorrichtung in urbanem Gebiet (nachfolgend <i>Urban Launcher</i>). • Klassifizieren des Urban Launchers durch den Piloten. • Bekämpfung des Urban Launchers. • Verifizierung der Zerstörung beider Launcher durch den Piloten. 2. Luftnahunterstützung von vorrückenden Bodentruppen: <ul style="list-style-type: none"> • Bekämpfung von verschiedenen Bodenzielen.
Randbedingungen & Einsatzregeln	<ul style="list-style-type: none"> • Verdecktes Eindringen in feindlichen Luftraum • Keine neutralen oder verbündeten Kollateralschäden durch Luft-Boden-Angriffe • Beeinträchtigung der Aufklärung durch Wolken: <ul style="list-style-type: none"> ○ Leichte Bewölkung (FEW) bei 1500 Fuß über Grund ○ Bedeckt bei 3000 Fuß über Grund.

Kräfteansatz

Die Mission wurde mit einem Team aus einem bemannten Kampfflugzeug und drei unbemannten Luftfahrzeugen durchgeführt. Der Bewaffnungsschwerpunkt des Teams lag aufgrund der Missionsziele auf präzisionsgelenkter Luft-Boden Munition, wobei jedes UAV für mehrere Missionsrollen ausgerüstet war (Abbildung 116).





Teammitglied	Missionsrolle
Foxtrot 	Kommando-Plattform & Luft-Boden Bekämpfung
Purple 	Mehrzweck mit Fähigkeit zum Ausbringen eines Schwarms
Orange 	Mehrzweck mit Schwerpunkt Luft-Boden
Blue 	Mehrzweck mit Schwerpunkt Luft-Boden

Abbildung 116. Missionsrollen der Teammitglieder.

Missionsablauf

Militärische Missionen werden in verschiedenen Missionsphasen durchlaufen, die aufeinander aufbauen und in denen jeweils verschiedene Aspekte im Vordergrund stehen.

Wie im Missionsziel beschrieben bestand die Mission aus zwei voneinander unabhängigen Aspekten: Einem Strike Anteil zur Bekämpfung von Abschussvorrichtungen und einem Close-Air-Support Anteil zur Unterstützung von Bodentruppen. Der daraus resultierende Ablauf der Mission ist in Tabelle 50 beschrieben.

Tabelle 50. Ablauf der Strike/Close-Air-Support Mission.

Missions-aspekt	Missions-phase	Beschreibung
Strike	Missions-start	Der Pilot startet sein eigenes Luftfahrzeug an einem Flughafen in verbündetem Gebiet, die UAVs stehen in einem naheliegenden Holding für die Mission bereit.
	Überqueren der Frontlinie	Der Pilot führt die UAVs und die eigene Plattform über die Frontlinie. Die Route und Höhe kann vom Piloten frei gewählt werden, es wird allerdings eine Flughöhe unter 3000 ft. (Meereshöhe) empfohlen, um eine Entdeckung durch Radarstellungen zu vermeiden.
	Wirken im Zielgebiet	Der Pilot bringt mit UAV Purple einen Schwarm aus, um die Flugabwehrstellung S5 zu unterdrücken, die den Luftraum über dem Zielgebiet schützt. Parallel dazu muss der Pilot ein UAV günstig positionieren, um feindliche Kampfflugzeuge in Schach zu halten, welche von zwei naheliegenden Militärflughäfen aufsteigen. Diese beiden Aktionen ermöglichen dem Piloten das Wirken im Zielgebiet. Die weiteren UAVs kann der Pilot nun einsetzen, um die beiden Launcher zu finden bzw. zu bekämpfen. Beide Launcher werden jeweils durch unaufgeklärte Flugabwehrstellungen geschützt, welche beim ersten Kontakt mit einem Schwarm oder UAV aktiv werden und bekämpft werden müssen. Nach der Unterdrückung der Flugabwehrstellungen kann der Pilot die beiden Launcher bekämpfen und nachaufklären. Dieser Prozess gestaltet sich beim Urban Launcher aufwändiger, da dieser zunächst gefunden werden muss. Während der Aufklärung des urbanen Gebiets erkennt das UAV vier Fahrzeuge, jedoch ist nur eines davon der gesuchte Launcher. Der Urban Launcher steht außerdem in unmittelbarer Nähe zu einem Krankenwagen, der Pilot muss die Parameter bei der Vergabe der Bekämpfungsaufträge also mit Bedacht wählen.
	Rückkehr Richtung Frontlinie	Nach der Bestätigung der Zerstörung beider Launcher erhält der Pilot über Funk die Anweisung nach Süden Richtung Frontlinie zu fliegen und sich für weitere Missionsziele bereit zu halten.

Close-Air-Support	Unterstützung bei Flussüberquerung	Nahe der Frontlinie erhält der Pilot über Funk die Anweisung, verbündete Bodentruppen durch die Bekämpfung von feindlichen Fahrzeugen bei der Überquerung eines Flusses zu unterstützen. Die verbündeten Truppen bestehen aus einer <i>West Group</i> aus Panzern, die versuchen eine Brücke zu überqueren und einer <i>East Group</i> aus gepanzerten Amphibienfahrzeugen, die eine direkte Überquerung des Flusses weiter östlich anstreben. Um die verbündeten Gruppen zu unterstützen, muss der Pilot Bekämpfungsaufträge an seine UAVs verteilen. Bei der <i>West Group</i> muss er darauf achten, keine verbündeten Truppen zu gefährden.
	Vorrücken der Bodentruppen	Nach der Bekämpfung der jeweiligen Feinde von <i>West</i> und <i>East Group</i> setzen sich die Gruppen in Bewegung. Der Pilot wird über Funk angewiesen, sich für weitere Ziele bereit zu halten.
	Beschuss der <i>West Group</i> durch Artillerie	Nach dem Vorrücken der <i>West Group</i> erhält der Pilot die Meldung, dass die <i>West Group</i> durch feindliche Artillerie bekämpft wird. Der Pilot erhält in diesem Zuge den Auftrag, die Artillerie in einem Gebiet aufzuspüren und zu zerstören.
	Vorrücken der Gruppen zu einem naheliegenden Staudamm	<i>West</i> und <i>East Group</i> rücken Richtung eines Staudamms vor. Sobald eine Gruppe den Staudamm erreicht hat, erhält der Pilot den Befehl, zum Startflughafen zurückzukehren.

E.6.2 Detaillierte Missionsbeschreibung zur Studie „Bewertung des Systems“

Dieser Abschnitt bietet eine vollständige Missionsbeschreibung der in Abschnitt 5.3 beschriebenen Studie zur Bewertung des Systems.

Missionsziel

Ziel der Mission war es, eine verdeckte Bekämpfung von zwei ballistischen Raketenabschussvorrichtungen (*Launcher*) durchzuführen. Die Launcher befanden sich auf dem Gebiet eines verfeindeten Staates und die Position der Launcher konnte nur auf ein 100 km² Zielgebiet eingegrenzt werden (Tabelle 51).

Tabelle 51. Übersicht über die Missionsziele und die Randbedingungen der Mission.

Missions-Element	Beschreibung
Primärziel	Bekämpfung von zwei ballistischen Launchern: <ul style="list-style-type: none"> • Lokalisieren der Launcher im Suchgebiet. • Bekämpfung der Launcher. • Verifizierung der Zerstörung beider Launcher.
Randbedingungen & Einsatzregeln	<ul style="list-style-type: none"> • Verdecktes Vorgehen bis zur Bekämpfung der Zielsysteme. • Klassifizierung der Systeme und Bestätigung der Bekämpfung durch menschliches Teammitglied. • Keine Kollateralschäden an Zivilisten oder Infrastruktur durch Luft-Boden-Angriffe. • Beeinträchtigung der Aufklärung durch Wolken: <ul style="list-style-type: none"> ○ Leichte Bewölkung (FEW) bei 1500 Fuß über Grund. ○ Bedeckt bei 3000 Fuß über Grund.

Kräfteansatz

Die Mission wurde mit einem Team aus einer bemannten und drei unbemannten Plattformen durchgeführt. Jedes Teammitglied hatte eine spezifische Bewaffnung, wodurch die Teammitglieder für verschiedene Missionsrollen vorgesehen waren (Abbildung 117).





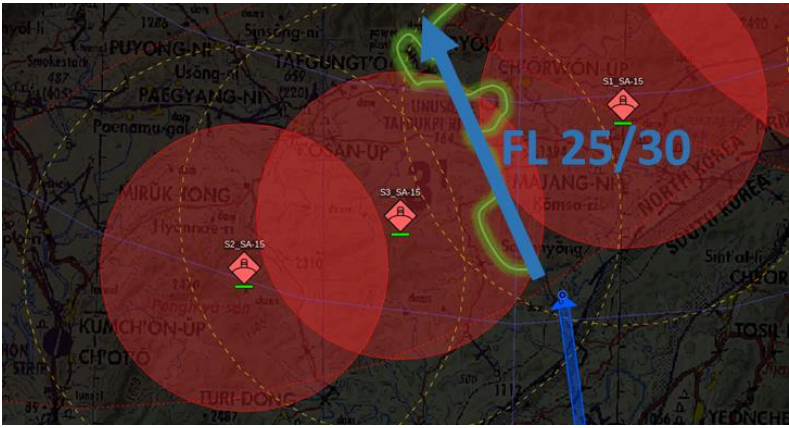
Teammitglied	Missionsrolle
Foxtrot 	Kommando-Plattform (optional Luft-Boden Bekämpfung)
Blue 	Unterdrückung von feindlichen Radarstellungen
Orange 	Luft-Boden Aufklärung & Bekämpfung
White 	Luftkampf

Abbildung 117. Missionsrollen der Teammitglieder.

Missionsablauf

Analog zur Mission der Studie „Nutzung von Führungsfunktionen“ besteht auch diese Mission aus verschiedenen Missionsphasen, der Ablauf der Mission ist in Tabelle 52 dargestellt.

Tabelle 52. Ablauf der Mission.

Missionsphase	Beschreibung
Missionsstart	Der Pilot startet sein eigenes Luftfahrzeug an einem Flughafen in verbündetem Gebiet, die UAVs stehen in einem naheliegenden Holding für die Mission bereit.
Penetration der feindlichen Luftverteidigung	<p>Das Team durchdringt die Linie von feindlichen Flugabwehrstellungen. Hierfür sollte das Team im Tiefflug entlang eines Flusses vorstoßen, um eine Entdeckung durch bereits aufgeklärte Flugabwehrstellungen zu verhindern (Abbildung 118).</p>  <p>Abbildung 118. Penetration der Luftverteidigung entlang vorgeschlagener Tiefflugroute.</p> <p>Am Ende der Luftverteidigungslinie wird der Pilot durch eine zuvor nicht aufgeklärte Flugabwehrstellung überrascht, welche das vorderste Teammitglied anstrahlt und somit sofort bekämpft werden muss.</p>
Vorrücken Richtung Zielgebiet	Nach der Bekämpfung der Flugabwehrstellung kann das Team weiter Richtung Zielgebiet vorrücken. In dieser Phase muss sich das Team gegen feindliche Kampfflugzeuge verteidigen, die in Richtung Zielgebiet unterwegs sind. Die feindlichen Kampfflugzeuge sind zwar technisch unterlegen, aber schwer von zivilem Flugbetrieb zu unterscheiden, sodass der Pilot die Flugbewegungen genau überwachen muss. Zur Unterscheidung zwischen neutralen und feindlichen Luftfahrzeugen gibt es vordefinierte Kriterien (Geschwindigkeit, Flugrichtung, Ursprung).

Wirken im Zielgebiet	<p>Um im Zielgebiet operieren zu können, muss eine angrenzende Flugabwehrstellung durch elektronischen Kampf (Jamming) unterdrückt werden. Während dieser Unterdrückung kann das Zielgebiet abgesucht werden. Im Zielgebiet befinden sich mehrere Fahrzeuge, darunter auch die beiden gesuchten Launcher (Abbildung 119).</p>  <p><i>Abbildung 119. Launcher im Sensorbild.</i></p> <p>Nach der Klassifizierung der beiden Launcher können diese bekämpft werden. Anschließend muss der Pilot die erfolgreiche Bekämpfung durch eine Nachaufklärung bestätigen.</p>
Rückkehr zum Startflughafen	<p>Im Anschluss an die Nachaufklärung führt der Pilot sein Team zurück zum Startflughafen, wobei er weiterhin mit feindlichen Flugabwehrstellungen und Kampfflugzeugen rechnen muss.</p>