MISSIONSAUFTRAGSANALYSE IN COSA ALS FUNKTIONSMODUL EINES PILOTENASSISTENZSYSTEMS

C. Groth[1,2], C. Meitinger[1], D. Donath[1], A. Schulte[1]
[1]Universität der Bundeswehr München
Professur für Flugmechanik & Flugführung
85577 Neubiberg
[2]bis Dezember 2005
info@conrad-groth.de, [claudia.meitinger|diana.donath|axel.schulte]@unibw.de

KURZFASSUNG

Dieser Beitrag beschreibt die Funktionalität „Auftragsinterpretation“ als Teil eines wissensbasierten Assistenzsystems am Beispiel einer Hubschrauberrettungsmission. Zunächst werden Assistenzsysteme, die Piloten bei Überwachung, Entscheidungsfindung und Problemlösung zur Seite stehen sollen, als Ausprägung kognitiver und kooperativer Automation unter Zuhilfenahme des Arbeitssystems charakterisiert. Anschließend wird der Kognitive Prozess – ein Modell menschlicher Informationsverarbeitung – als grundlegender Theorieansatz und die Kognitive Systemarchitektur COSA als Basis für die Implementierung vorgestellt. Schließlich werden das Konzept, die Implementierung und einige Ergebnisse der Auftragsinterpretation erläutert.

1. EINLEITUNG

Vor dem Hintergrund immer komplexer werdender Flugmissionen ändern sich die Anforderungen an Piloten sowohl im Flugzeug- als auch im Hubschraubercockpit. Um die Missionserfüllung sicherstellen und darüber hinaus die Flugsicherheit erhöhen zu können, wird typischerweise immer mehr Automation im Cockpit eingeführt. Dies führt zu einem veränderten Aufgabenspektrum der Piloten: Statt der tatsächlichen Ausführung einzelner Aufgaben überwachen sie Automation, die die Aufgaben ausführt und entscheiden, welche Automationsfunktionen sie einsetzen um ein gegebenes Missionsziel zu erreichen. Während die Belastung durch die Aufgabenausführung mit Hilfe der zusätzlichen Automation verringert wird, steigt die Belastung durch Überwachungs- und Entscheidungsaufgaben, für die keine Unterstützung zur Verfügung steht. Dadurch entsteht eine neue Klasse automationsbedingter Unfälle [1].

Assistenzsysteme als Ausprägung kognitiver und kooperativer Automation sollen Piloten bei Überwachung, Entscheidungsfindung und Problemlösung zur Seite stehen und im Gegensatz zu sog. konventioneller Automation nicht nur spezifische Teilaufgaben ausführen, sondern wie der Pilot im Hinblick auf das übergeordnete Missionsziel agieren (vgl. Abschnitt 2).

Als Theorieansatz für die Entwicklung derartiger Assistenzsysteme kann der Kognitive Prozess verwendet werden, ein Modell menschlicher Informationsverarbeitung, das zielgerichtetes Verhalten nachbildet. Für die Implementierung kognitiver Systeme auf Basis des Kognitiven Prozesses steht COSA (Cognitive System Architecture) zur Verfügung (vgl. Abschnitt 3).

Damit ein Assistenzsystem einen Piloten adäquat unterstützen kann braucht es ein Verständnis bezüglich der vorherrschenden Situation. Ein Element der Situation ist dabei der Auftrag, der erfüllt werden soll. Mit entsprechendem Wissen kann es den Piloten z.B. bei der Entscheidung unterstützen, ob ein Auftrag angenommen werden kann. Eine derartige Funktionalität wird am Beispiel einer vereinfachten Hubschrauberrettungsmission exemplarisch in COSA umgesetzt (vgl. Abschnitte 4-6). Sie stellt dem Piloten die Auftragsauffassung des Systems, d.h. das Verständnis des Systems vom Auftrag, eine Aufgabenliste mit den für die Durchführung der Mission erforderlichen Aufgaben sowie eine Abschätzung der für den Auftrag benötigten Ressourcen an Zeit, Treibstoff, Personal und Infrastruktur zur Verfügung, die dieser als Grundlage für eine Entscheidung verwenden kann. Zukünftig soll zudem eine Entscheidungsempfehlung ausgesprochen werden.

2. ASSISTENZSYSTEME ALS TEILKOMPONENTE IM ARBEITSSYSTEM

2.1. Das Arbeitssystem

Das Arbeitssystem [2] kann als Paradigma für die Beschreibung des Zusammenwirkens von Mensch und Maschine bei der Bearbeitung einer definierten Arbeitsaufgabe verwendet werden. Im Bereich der Flugführung wurde es angewendet, um den Einsatz konventioneller und kognitiver Automation im Flugzeugcockpit zu beschreiben [3][4].

Das Arbeitssystem (vgl. Abbildung 1) ist definiert durch eine Arbeitsaufgabe, welche auch als Auftrag bezeichnet werden kann [5]. Dabei muss zwischen dem gegebenen Auftrag und dem vom Arbeitssystem verstandenen Auftrag unterschieden werden [5]. Er wird entweder von einer anderen, leitenden Organisation an das Arbeitssystem gegeben oder kann vom menschlichen Bediener selbst erzeugt oder modifiziert werden. Da ein Arbeitssystem üblicherweise in eine Umgebung eingebettet ist, stellen Umgebungsbedingungen einschließlich der zur

Verfügung stehenden Informationen und Ressourcen zu beachtende Randbedingungen für die Bearbeitung der Arbeitsaufgabe dar. An den Ausgängen des Arbeitssystems können der gegenwärtige Zustand des Arbeitsprozesses, d.h. der Fortschritt der Arbeitsaufgabe, sowie das endgültige Ergebnis beobachtet werden.

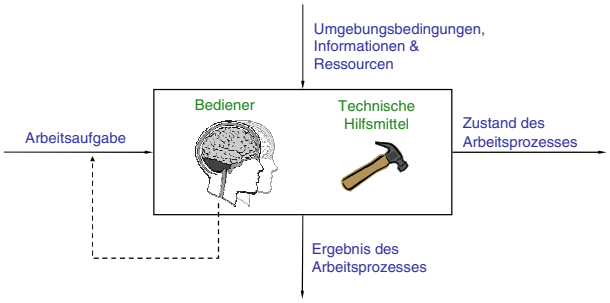


BILD 1. Arbeitssystem

Der klassische Entwurf des Arbeitssystems sieht die folgenden beiden Komponenten vor, die in ihm kombiniert werden, um ein bestimmtes Resultat auf Basis des gegebenen Arbeitsziels zu erreichen:

– Der *Bediener* ist diejenige Komponente des Arbeitssystems, die die Erreichung der gesamten Arbeitsaufgabe zum Ziel hat und zu diesem Zweck technische Hilfsmittel einsetzt, diese koordiniert und deren Aufgabenausführung überwacht. Er stellt also die Entscheidungskomponente des Systems dar und verfügt über die höchste Autorität. Üblicherweise ist der Bediener ein Mensch, kann jedoch auch ein Team von Menschen sein [3].
– Die *technischen Hilfsmittel* repräsentieren im Umfeld der Prozesssteuerung üblicherweise die dem Bediener zur Verfügung stehende Automation, z.B. einen Autopiloten. Charakteristisch für die einzelnen Arbeitsmittel ist, dass sie lediglich Teilaufgaben bearbeiten und kein Verständnis von der übergeordneten Arbeitsaufgabe des Arbeitssystems haben.

2.2. Kognitive und kooperative Automation im Arbeitssystem

Im Allgemeinen weist konventionelle Automation keine Fähigkeiten im Bereich kognitiver Funktionalitäten wie Informationsintegration und Entscheidungsfindung auf [6]. Die Bearbeitung von Aufgaben, welche das Vorhandensein solcher Fähigkeiten benötigen, bleibt daher dem Menschen überlassen. Um diesen Defiziten konventioneller Automation entgegenzuwirken, werden so genannte künstliche kognitive Einheiten (*Artificial Cognitive Units*, ACUs) als Ergänzung zum menschlichen Operateur in das Arbeitssystem integriert (vgl. Abbildung 2).

Wird eine ACU auf Seiten der technischen Hilfsmittel eingeführt (Abbildung 2, rechts), so bildet sie mit den vorhandenen technischen Systemen ein teilautonomes System, welches als Hilfsmittel in ein Arbeitssystem eingebettet ist und vom Bediener Teilaufgaben zugewiesen bekommt. Ein teilautonomes System unterscheidet sich von einem Arbeitssystem hauptsächlich dadurch, dass die zu erfüllende (Teil-)Aufgabe nicht modifiziert werden kann (vgl. [4][7]). Es stellt *kognitive Funktionalitäten* im Bereich des Aufbaus von Situationsverständnis, zielebasierter Aufgabenausführung, rationaler Entscheidungsfindung, sowie Problemlösung und Planung zur Verfügung. Damit ist es der ACU möglich, rational und für den Bediener nachvollziehbar zu agieren.

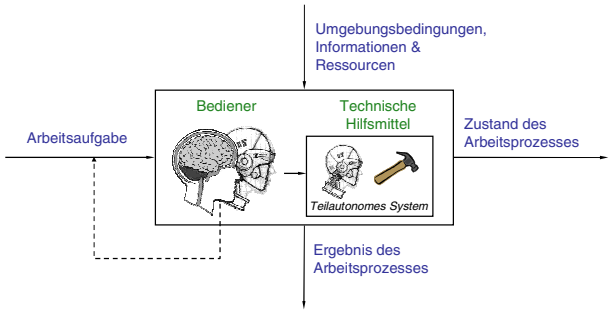


BILD 2. Einführung von ACUs in ein Arbeitssystem

Wird die ACU auf Seiten der menschlichen Komponente in Arbeitssystem eingeführt (Abbildung 2, links), so wird der Bediener durch ein Team bestehend aus Mensch und ACU ersetzt, welches gemeinsam das Ziel der Auftragserfüllung verfolgt. Dazu muss die ACU zusätzlich über *kooperative Fähigkeiten* verfügen, die aktuell verfügbaren Ressourcen des Menschen einschätzen können und fähig sein, zielorientiert zu interagieren.

Eine ACU, die über sowohl kognitive als auch kooperative Fähigkeiten im beschriebenen Sinne verfügt, wird nach [8] als (intelligentes) *Assistenzsystem* bezeichnet.

2.3. Charakteristika von Assistenzsystemen

Assistenzsysteme sind also eng mit den oben skizzierten Ansätzen kognitiver und kooperativer Automation verknüpft. Ziel solcher Assistenzsysteme ist es, das Beanspruchungsniveau eines Operateurs vor dem Hintergrund seiner zu bearbeitender Aufgaben und der aktuell vorherrschenden Situation auf einem angemessenen Niveau zu halten. Unter Berücksichtigung der von Billings aufgestellten Forderungen an *„Human Centred Automation“* [1] leitete Onken [9] in diesem Zusammenhang zwei grundlegende Forderungen an Assistenzsysteme als Spezifikation für einen entsprechenden Systementwurf ab:

– *Grundforderung (1)*
„Die Aufmerksamkeit des Operateurs ist im Rahmen der Präsentation des Gesamtbildes der Situation auf die situationsgemäß objektiv dringlichste Aufgabe bzw. Teilaufgabe zu lenken“. [9]
– *Grundforderung (2)*
„Ist die Grundforderung (1) erfüllt und tritt eine Situation mit Überforderung des Operateurs ein, dann ist diese Situation mit technischen Mitteln auf eine normale, handhabbare zurückzuführen.“ [9]

Um Grundforderung (1) erfüllen zu können, muss ein Assistenzsystem Situationsbewusstsein bezüglich der Missionsdurchführung besitzen, wozu unter anderem ein Verständnis des Auftrags (vgl. Abschnitte 4-6 dieses Beitrags), der aktuellen Flugsituation und des Missionsfortschritts gehört. Grundforderung (2) verlangt nach Anpassung der Automation an die aktuellen Bedürfnisse des Menschen (vgl. [10]).

3. ENTWICKLUNG KOGNITIVER SYSTEME

3.1. Theorie des Kognitiven Prozesses

Als Basis für die Modellierung intelligenten, menschähnlichen Verhaltens im Rechner dient der Kognitive Prozess [11]. Er ist an die menschliche Informationsverarbeitung angelehnt, ohne diese exakt nachzubilden. Ziel ist die Erzeugung rationalen, vorhersehbaren und verständlichen Verhaltens.

Das nach außen sichtbare Verhalten des Kognitiven Prozesses ist durch den gestrichelten Pfeil am linken Bildrand in Abbildung 3 angedeutet. Für einen Beobachter, der sich in der Umwelt befindet, sind die inneren Abläufe, die basierend auf Zielen zu dem beobachtbaren Verhalten führen, dabei ähnlich einer „black box“ unsichtbar.

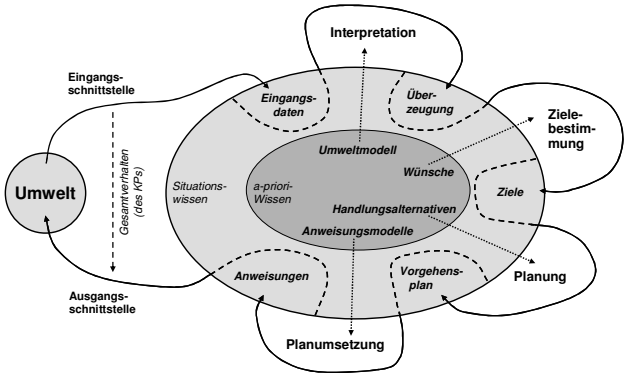


BILD 3. Der Kognitive Prozess

Der Kognitive Prozess basiert auf einem wissensbasierten Ansatz, d.h. applikationsspezifisches **Wissen** (ovale Bereiche in Abbildung 3, sog. „Rumpf“) wird von applikationsunabhängiger **Verarbeitung** (äußere Pfeile in Abbildung 3, sog. „Transformatoren“) getrennt. Das Wissen kann wiederum in *a-priori Wissen*, welches vom Entwickler einer Applikation modelliert werden muss, und *Situationswissen*, welches zur Laufzeit des Systems erzeugt wird, unterteilt werden. Transformatoren generieren auf Basis eines Teils des Situationswissens und des a-priori Wissens neues Situationswissen. Dabei ist jedem Transformator ein gewisser Teil des a-priori Wissens, angedeutet durch die gepunkteten Pfeile in Abbildung 3, sowie durch die Pfeilspitze ein Zielbereich im Situationswissen zugeordnet. Im Gegensatz zum eindeutig zugeordneten Zielbereich kann jeder Transformator im gesamten Situationswissen lesen, liest aber primär in jenem Bereich, in dem der Transformatorpfeil beginnt.

Die Informationsverarbeitung im Kognitiven Prozess erfolgt grundsätzlich parallel, d.h. alle Transformatoren verarbeiten gleichzeitig Wissen. Sie kann aber dennoch sequenziell beschrieben werden: Der Transformator *Interpretation* erzeugt aus den Eingangsdaten, die über die Eingangsschnittstelle aufgenommen werden, die Überzeugung, welche ein Verständnis bezüglich des Zustands der Umwelt darstellt. Basis dafür sind die Umweltmodelle, welche z.B. Objekte der Umwelt, Relationen zwischen diesen Objekten oder abstrakte Gebilde wie z.B. einen Auftrag beschreiben. Vor dem Hintergrund der modellierten Wünsche, d.h. potentiell zu verfolgender Ziele, werden aus der Überzeugung *Ziele bestimmt*, die gegenwärtig verfolgt werden sollen und damit die gewünschte Zielsituation beschreiben. Mit Hilfe von Handlungsalternativen werden die Schritte ausgewählt, die nötig sind, um die aktuelle Situation in die Zielsituation zu überführen (*Planung*). Dieser Plan wird in einzelne Anweisungen zerlegt, welche zur Ausführung mittels der Ausgangsschnittstelle an entsprechende Effektoren in der Umwelt gegeben werden können (*Planumsetzung*).

3.2. Implementierung von auf dem Kognitiven Prozess basierenden Systemen

Zur Implementierung von Systemen, welche auf Grundlage der Theorie des Kognitiven Prozesses entwickelt worden sind, steht die kognitive Systemarchitektur COSA (*Cognitive System Architecture*, [12]) zur Verfügung. Diese enthält bereits eine Implementierung der applikationsunabhängigen Mechanismen des Kognitiven Prozesses, d.h. der Wissensverarbeitung, so dass ein Systementwickler lediglich das anwendungsspezifische a-priori Wissen modellieren muss. Hierfür wird üblicherweise die Programmiersprache CPL (*Cognitive Programming Language*) verwendet, welche ebenfalls Teil von COSA ist, da sie die Möglichkeit bietet, komplexe Verhaltensmuster auf einem hohen Abstraktionsgrad zu beschreiben. Außerdem wird die Implementierung von Umweltmodellen, Wünschen, Handlungsalternativen und Anweisungsmodellen wie sie aus dem Kognitiven Prozess bekannt sind unterstützt.

Ein Modell im Sinne des Kognitiven Prozesses entspricht in CPL einer Klasse, welche durch Attribute und Verhalten beschrieben wird. Im Gegensatz zu Klassen wie sie aus der objektorientierten Programmierung bekannt sind, steuern die Klassen dabei nicht nur die Belegung der Attribute, sondern darüber hinaus auch die Erzeugung und Zerstörung von Instanzen.

4. KONZEPT AUFTRAGSINTERPRETATION

Dieser Abschnitt erläutert das Konzept der „Auftragsinterpretation“ als Teilfunktionalität eines Assistenzsystems. Als Anwendungsbeispiel wird dabei eine Hubschrauberrettungsmission verwendet, d.h. die Rettung von Verletzten an einer Unfallstelle mit einem Hubschrauber. Eine zeitkritische Phase während einer solchen Hubschrauberrettungsmission ist der Startvorgang, bei dem der Pilot vor dem Abflug entscheiden muss, ob eine angeforderte Mission durchgeführt werden kann oder nicht. Während der Pilot nun seinerseits eine zeitaufwändige Überschlagsrechnung des benötigten Treibstoffs durchführt, wird das Assistenzsystem in die Lage versetzt, ebenfalls den Auftrag zu analysieren und dem Piloten gegebenenfalls eine Empfehlung auszugeben, ob bzw. unter welchen Bedingungen die Mission durchführbar ist. So kann es z.B. erforderlich sein, nach der Ablieferung des Patienten im Krankenhaus zunächst nachzutanken, bevor der Rückflug zur Basis beginnen kann. Alternativ könnte auch nur ein Arzt an der Unfallstelle abgesetzt werden, der Patient aber nicht vom Hubschrauber geborgen werden. Die endgültige Entscheidung über die Durchführung der Mission liegt allerdings weiterhin beim Piloten.

4.1. Einordnung in die Gesamtarchitektur

Als grundlegende Theorie für die Konzeption der Auftragsinterpretation wird der Kognitive Prozess (vgl. Abschnitt 3.1) verwendet. Da die Auftragsinterpretation objektive Eingangsdaten (z.B. den von einer externen Stelle erteilten Auftrag, die aktuelle Flugphase) anhand von Modellen (z.B. zur Beschreibung einer Hubschrauberrettungsmission und deren Teilaufgaben) interpretiert, um dem Assistenzsystem für die weitere Arbeit eine Überzeugung hinsichtlich des eingegangenen Auftrags (z.B. Auftragsauffassung und Ressourcenbedarf für die Mission) bereitzustellen, kann diese Funktionalität dem Transformator „Interpretation“ innerhalb des Kognitiven Prozesses zugeordnet werden.

4.2. Teilfunktionalitäten der Auftragsinterpretation

In Abbildung 4 sind die Eingangs- und Ausgangsgrößen im Sinne einer funktionalen Abhängigkeit dargestellt, die für die Auftragsinterpretation benötigt bzw. durch die folgenden Teilfunktionalitäten zur Verfügung gestellt werden:

- Bestimmung der Auftragsauffassung
- Bestimmung der Aufgabenliste
- Bestimmung des Gesamtressourcenbedarfs

Bei den links in Abbildung 4 an den Pfeilen angetragenen Eingangsinformationen handelt es sich um Situationswissen, das von der Auftragsinterpretation verarbeitet wird. Die Pfeile am rechten Abbildungsrand beinhalten ebenfalls Situationswissen, welches durch die Auftragsinterpretation erzeugt wird. Am unteren Abbildungsrand ist das von der Auftragsinterpretation benötigte a-priori Wissen dargestellt.

Ordnet man den oben genannten Bereichen der Abbildung 4 Begriffe des Kognitiven Prozesses zu, so entspricht das Wissen an den linken Pfeilen sowohl Eingangsdaten (z.B. Auftrag) als auch Überzeugungen (z.B. Flug-/Missionsphase), der zentrale schraffierte Bereich dem Transformator Interpretation, das a-priori Wissen den Umweltmodellen und das Wissen an den rechten Pfeilen der Überzeugung.

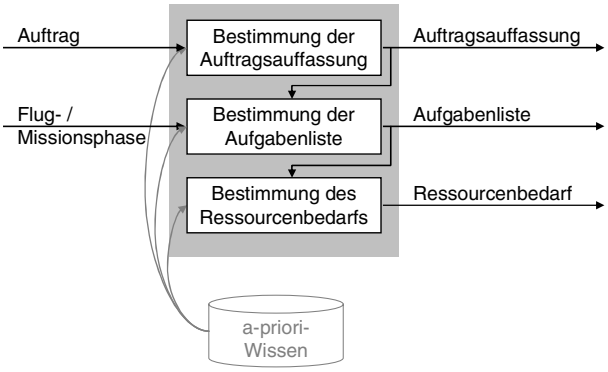


BILD 4. Teilfunktionalitäten der Auftragsinterpretation

4.2.1. Auftragsauffassung

Die Auftragsauffassung umfasst die Vervollständigung des Auftrags durch Annahmen sowie die Bestimmung einer Liste von Missionselementen, die zur Erfüllung des Auftrags erforderlich sind. Wichtig an dieser Stelle ist die Abgrenzung zwischen der extern gegebenen, objektiven Auftragsinformation und der Auftragsauffassung, die das mentale Konzept repräsentiert, welches vom Assistenzsystem (repräsentiert durch die ACU) darunter verstanden wird. Diese Unterscheidung liegt in gleicher Weise bei einem menschlichen Piloten vor, der aufgrund seines Wissens und seiner Fähigkeiten bei Eintreffen eines Auftrags sofort weiß, wie dieser zu verstehen ist, also beispielsweise, welche Aufgaben mit einer möglichen Ausführung des Auftrags verbunden sind.

Ein extern erteilter Auftrag muss mindestens den Unfallort und kann zusätzlich das Krankenhaus, zu dem der Patient geflogen werden soll, den Ort, an dem ein zusätzlicher Arzt aufgenommen werden kann, sowie den Zielort des Hubschraubers enthalten. Wird von der externen Stelle kein Krankenhaus angegeben, so wird auf Basis des implementierten Wissens von der Auftragsinterpretation das dem Unfallort nächstgelegene und für die Verletzungen des Patienten ausgestattete Krankenhaus ausgewählt. Sind die Verletzungen des Patienten ebenfalls unbekannt, so wird angenommen, dass zur Behandlung des Patienten ein gewöhnliches Krankenhaus ausreicht. Enthält der Auftrag keinen Zielort, so fliegt der Hubschrauber bei Missionsende zur Basis zurück. Die Auftragsauffassung umfasst somit den Unfallort, ggf. den Ort, an dem ein zusätzlicher Arzt aufgenommen werden soll, das Krankenhaus, zu dem der Patient geflogen werden soll, und den Zielort, zu dem der Hubschrauber nach Beendigung der Mission fliegen soll.

Nach der Vervollständigung des Auftrags werden für die während der Mission anzufliegenden Orte die folgenden Informationen zu Missionselementen erzeugt:

- Pick up *obj* at location *loc*
- Depose *obj* at location *loc*
- Transit to location *loc*

Für jeden anzufliegenden Ort wird dabei ein „Transit“- und – je nach Ort – ein „Pick up“- bzw. „Depose“-Element erzeugt; *obj* ist entweder der Patient oder ein Arzt. Die Reihenfolge der Missionselemente ergibt sich aus dem typischen Ablauf einer Hubschrauberrettungsmission, der in Abbildung 5 dargestellt ist. Üblicherweise erfolgt der Flug von der Heimatbasis zum Unfallort, jedoch kann es sein, dass vorher ein Arzt aufgenommen werden muss. Nach dem Unfallort wird ein Krankenhaus angeflogen, bevor der Hubschrauber zum Zielort gelangt, welcher identisch mit der Basis sein kann.

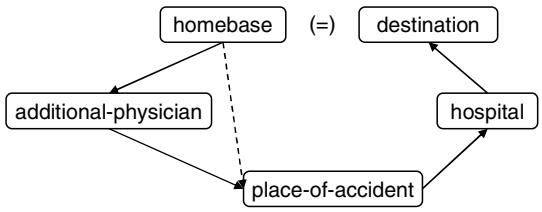


BILD 5. Typischer Ablauf einer Hubschrauber-rettungsmission

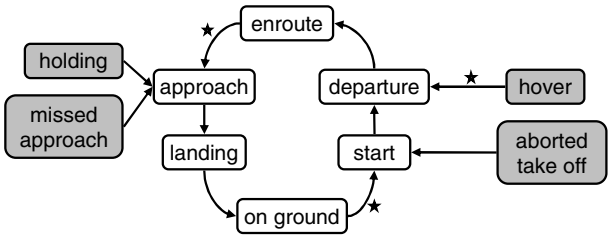
Eine vollständige Mission mit Aufnahme eines zusätzlichen Arztes sieht dann folgendermaßen aus:

- Transit to *additional-physician*
- Pick up *physician* at *additional-physician*
- Transit to *place-of-accident*

– Pick up *patient* at *place-of-accident*
– Transit to *hospital*
– Depose *patient* at *hospital*
– Transit to *destination*

4.2.2. Bestimmung der Aufgabenliste

Zur Bestimmung der Aufgabenliste wird der Flug des Hubschraubers beginnend bei der aktuellen Flug- und Missionsphase antizipiert. Dabei wird die typische Abfolge der Flugphasen, die in der Abbildung 6 dargestellt ist, so oft durchlaufen, bis die Basis erreicht und die Mission damit abgeschlossen ist.



★ = Nächste simulierte Missionsphase

BILD 6. Flugphasenübergänge

An den mit einem Stern gekennzeichneten Pfeilen wird die nächste antizipierte Missionsphase (Transit bzw. Landung) bestimmt. Eine Übersicht über die Missionsphasen und die dazugehörigen Missionselemente sowie Flugphasen gibt Tabelle 1. Es ist zu erkennen, dass zu einer Transitphase sowohl der reine Streckenflug als auch Start und Abflug gehören. Analog gehören zu einer Landephase sowohl der Landeanflug als auch die Bodenoperationen.

antizipierte Missionsphase	Missionselement-typ	Flugphasen
Transit	ACT_TRANSIT	start
		departure
		enroute
Land	ACT_PICK_UP bzw. ACT_DEPOSE	approach
		landing
		on-ground

TAB 1. Zusammenhang zwischen antizipierter Missionsphase, Missionselement und Flugphasen

Für jede antizipierte Flugphase wird der Aufgabenliste eine Aufgabe hinzugefügt. Die Zuordnungen der Aufgaben zu den Flugphasen sind in der folgenden Tabelle dargestellt.

Flugphase	Aufgabe
on-ground	ground-op
start	take-off
departure	depart
enroute	enroute
approach	approach
landing	landing

TAB 2. Zuordnung von Aufgaben zu Flugphasen

Die in Abbildung 6 grau hinterlegten Flugphasen sind irreguläre Flugphasen, d.h. Flugphasen, die bei der Antizipation eines Fluges mit normalem Ablauf ohne ungeplante Ereignisse nicht auftreten. Sollte die aktuelle Flugphase eine dieser irregulären Flugphasen sein, wird diese ignoriert und es wird die nächste Flugphase in Pfeilrichtung ausgewählt, da für die irregulären Flugphasen keine entsprechenden Aufgabentypen modelliert sind.

4.2.3. Bestimmung des Gesamtressourcen-bedarfs

Für jede Aufgabe der Aufgabenliste wird der Zeit- und Treibstoffbedarf abgeschätzt. Dabei wird für alle Aufgaben außer den Streckenflügen ein durchschnittlicher, aufgabentypischer Ressourcenbedarf angenommen. Bei den Streckenflügen wird der *Zeitbedarf* auf Basis der Entfernung zwischen den missionsrelevanten Orten grob abgeschätzt. Der *Treibstoffbedarf* wird anschließend durch die Multiplikation des Zeitbedarfs mit dem durchschnittlichen, flugartspezifischen Treibstoffverbrauch bestimmt. Mögliche Flugarten sind normaler, wirtschaftlicher oder schneller Flug. Die erforderlichen Ressourcen aller Aufgaben werden zum Gesamtressourcenbedarf der Mission aufsummiert. Zudem wird der *Infrastrukturbedarf*, d.h. die Art des Krankenhauses, sowie der *Personalbedarf*, d.h. der Bedarf an einem Notarzt, bestimmt und dem Gesamtressourcenbedarf hinzugefügt.

5. IMPLEMENTIERUNG DER AUFTRAGS-INTERPRETATION

Die Auftragsinterpretation ist in COSA als eigenständige Komponente implementiert. Dabei besteht diese lediglich aus Funktionalitäten im Bereich der Interpretation, d.h. sie enthält nur Umweltmodelle.

5.1. Testumgebung

Der Aufbau der Testumgebung ist in Abbildung 7 dargestellt. Der grau hinterlegte Bereich kennzeichnet die „COSA-Welt“, d.h. die Implementierung der eigentlichen Auftragsinterpretation. Dazu gehört zunächst der applikationsunabhängige COSA-Kernel, in den das applikationsabhängige a-prori Wissen in Form einer CPL-Datei geladen wird. Hinzu kommen die COSA-Server und -Funktionen, die die Eingangs- und Ausgangsschnittstellen abbilden.

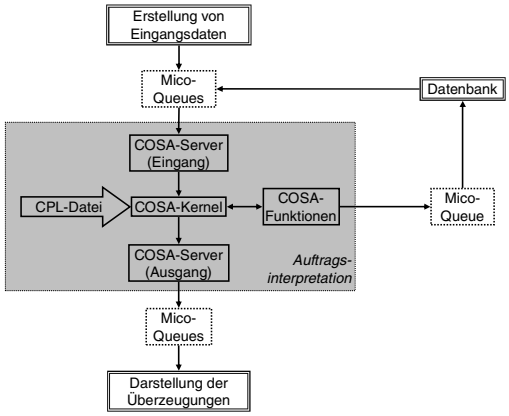


BILD 7. Aufbau der Testumgebung für die Auftrags-interpretation

Die doppelt umrahmten Begriffe kennzeichnen Hilfsprogramme, die zum Testen der Funktionalität benötigt werden. Hierzu zählen ein Programm zur Erstellung der Eingangsdaten, die im Rahmen eines Assistenzsystems von anderen Funktionalitäten zur Verfügung gestellt werden bzw. von externen Stellen wie z.B. der Rettungsleitstelle an die Hubschrauber-Crew gegeben werden, ein Programm zur Darstellung der Überzeugungen, d.h. der Ergebnisse der Auftragsinterpretation, sowie ein Programm, welches Datenbankfunktionalitäten nachbildet. Die gestrichelten Rahmen charakterisieren Server auf Basis von MICO [13], einer frei verfügbaren Implementierung des Kommunikationsstandards CORBA, welche zum Datenaustausch verwendet werden. Die Pfeile in Abbildung 7 geben stets die Kommunikationsrichtung zwischen den durch sie verbundenen Objekten an.

5.2. Klassenstruktur

Das a-priori Wissen, das das Verhalten der Auftragsinterpretation generiert, ist in Form von CPL-Klassen repräsentiert (vgl. Abschnitt 3.2, Abbildung 8). Für den Auftrag, die Auftragsauffassung, die Aufgabenliste sowie den Gesamtressourcenbedarf existiert jeweils eine Klasse, von der je eine Instanz gebildet wird.

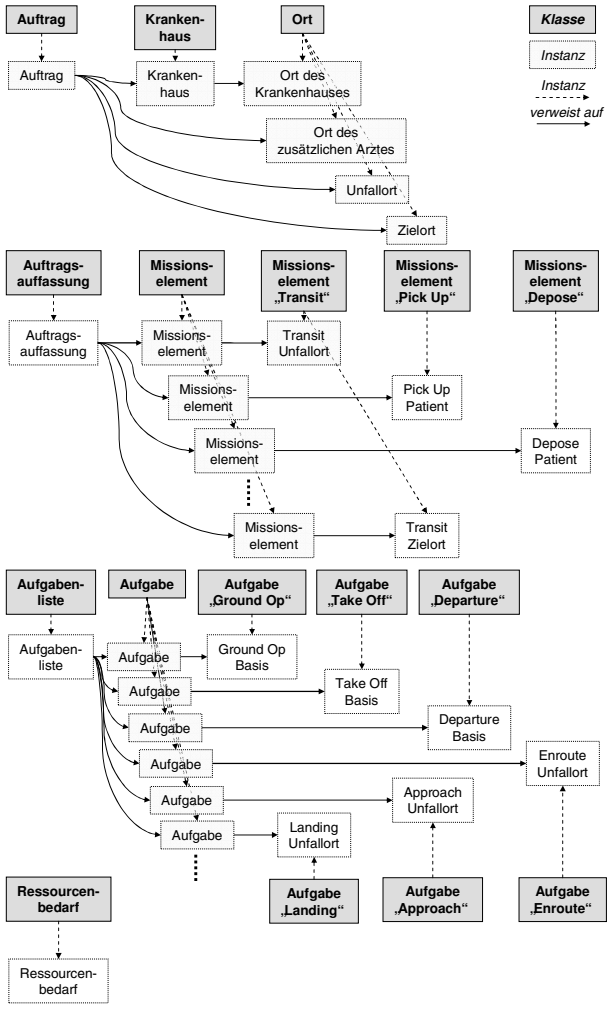


BILD 8. Übersicht über wichtige Klassen und Beispiele für Instanzen davon

Die Instanz, die den Auftrag repräsentiert, verweist auf eine Instanz der Klasse „Krankenhaus“ sowie ggf. Instanzen der Klasse „Ort“ für den Ort des zusätzlichen Arztes, den Unfallort und den Zielort. Die Instanz „Krankenhaus“ verweist ebenfalls auf eine Instanz der Klasse „Ort“ und enthält noch zusätzliche Informationen bezüglich der Infrastruktur des Krankenhauses.

Die Instanz, die die Auftragsauffassung darstellt, verweist für jedes Missionselement auf eine Instanz der Klasse „Missionselement“. Ebenso verweist die Instanz der Aufgabenliste für jede Aufgabe auf eine Instanz der Klasse „Aufgabe“. Die Instanzen von „Missionselement“ und „Aufgabe“ verweisen jeweils weiter auf eine Instanz der Klasse des entsprechenden Aufgaben- bzw. Missionselementtyps.

Die Instanz bezüglich des Gesamtressourcenbedarfs beinhaltet und bestimmt die benötigten Ressourcen, verweist aber nicht auf Instanzen weiterer Klassen.

6. ERGEBNISSE

Zur Validierung des Konzepts der Auftragsinterpretation wurde die im vorigen Abschnitt beschriebene Implementierung mit unterschiedlich umfangreichen Aufträgen sowie variierenden Randbedingungen wie Flug- und Missionsphase getestet. Exemplarisch werden hier die Ergebnisse der Auftragsinterpretation für eine Hubschrauberrettungsmission zur Rettung eines Patienten in der Altmühlgegend dargestellt, wobei vorher ein zusätzlicher Arzt am Krankenhaus Treuchtlingen aufgenommen werden soll. Der Auftrag für diesen Testfall ist in Tabelle 3 dargestellt. Es werden keine Angaben zur aktuellen Flug- und Missionsphase gemacht.

Attribut		Wert
Zusätzlicher Arzt	Name	Khs Treuchtlingen
	Typ	KOL_HOSPITAL
	Geografische Breite [°]	48.9583
	Geografische Länge [°]	10.9033
Unfallort	Name	Bäckertal
	Typ	KOL_ACCIDENT
	Geografische Breite [°]	48.8433
	Geografische Länge [°]	11.0460

TAB 3. Auftrag

In den folgenden Abbildungen sind die Auftragsauffassung, die Aufgabenliste und ein Kartenausschnitt der Flugroute als Ergebnis der Interpretation dieses Auftrags dargestellt.

Id	Action	Object	Loc-Kind	Loc-Name	Lat	Lon
1	ACT_TRANSIT		KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
2	ACT_PICK_UP	OBJ_DOCTOR	KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
3	ACT_TRANSIT		KOL_ACCIDENT	Bäckertal	48.8433	11.046
4	ACT_PICK_UP	OBJ_PATIENT	KOL_ACCIDENT	Bäckertal	48.8433	11.046
5	ACT_TRANSIT		KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
6	ACT_DEPOSE	OBJ_PATIENT	KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
7	ACT_TRANSIT		KOL_HOMEBASE	homebase	48.7067	10.77

BILD 9. Auftragsauffassung

Die Auftragsauffassung (vgl. Abbildung 9) umfasst die maximal mögliche Anzahl an Missionselementen durch die Aufnahme des Arztes. Da weder Krankenhaus noch Zielort im Auftrag gegeben sind, werden hier das Krankenhaus Eichstätt als das nächste Krankenhaus und die Basis als Standard-Zielort eingesetzt.

Id	Action	(type)	Loc-Kind	Loc-Name	Lat	Lon
1	TSK_PERFORM_GROUND_OP	at	KOL_HOMEBASE	current-location	48.7067	10.77
2	TSK_PERFORM_TAKE_OFF	at	KOL_HOMEBASE	current-location	48.7067	10.77
3	TSK_PERFORM_DEPARTURE	at	KOL_HOMEBASE	current-location	48.7067	10.77
4	TSK_PERFORM_ENROUTE	from	KOL_HOMEBASE	current-location	48.7067	10.77
		to	KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
5	TSK_PERFORM_APPROACH	at	KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
6	TSK_PERFORM_LANDING	at	KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
7	TSK_PERFORM_GROUND_OP	at	KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
8	TSK_PERFORM_TAKE_OFF	at	KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
9	TSK_PERFORM_DEPARTURE	at	KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
10	TSK_PERFORM_ENROUTE	from	KOL_HOSPITAL	Khs Treuchtlingen	48.9583	10.9033
		to	KOL_ACCIDENT	Bäckertal	48.8433	11.046
11	TSK_PERFORM_APPROACH	at	KOL_ACCIDENT	Bäckertal	48.8433	11.046
12	TSK_PERFORM_LANDING	at	KOL_ACCIDENT	Bäckertal	48.8433	11.046
13	TSK_PERFORM_GROUND_OP	at	KOL_ACCIDENT	Bäckertal	48.8433	11.046
14	TSK_PERFORM_TAKE_OFF	at	KOL_ACCIDENT	Bäckertal	48.8433	11.046
15	TSK_PERFORM_DEPARTURE	at	KOL_ACCIDENT	Bäckertal	48.8433	11.046
16	TSK_PERFORM_ENROUTE	from	KOL_ACCIDENT	Bäckertal	48.8433	11.046
		to	KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
17	TSK_PERFORM_APPROACH	at	KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
18	TSK_PERFORM_LANDING	at	KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
19	TSK_PERFORM_GROUND_OP	at	KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
20	TSK_PERFORM_TAKE_OFF	at	KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
21	TSK_PERFORM_DEPARTURE	at	KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
22	TSK_PERFORM_ENROUTE	from	KOL_HOSPITAL	Khs Eichstätt	48.9	11.1833
		to	KOL_HOMEBASE	homebase	48.7067	10.77

BILD 10. Aufgabenliste

An der Aufgabenliste (vgl. Abbildung 10) ist der stets gleiche Ablauf zwischen den einzelnen Orten durch den geschlossenen Flugphasenübergangsgraphen erkennbar.

Es ergeben sich für diese Mission geschätzt ein Zeitbedarf von 1 h 42 min und ein Treibstoffbedarf von 323 kg. Außerdem werden ein Arzt sowie ein gewöhnliches Krankenhaus benötigt.

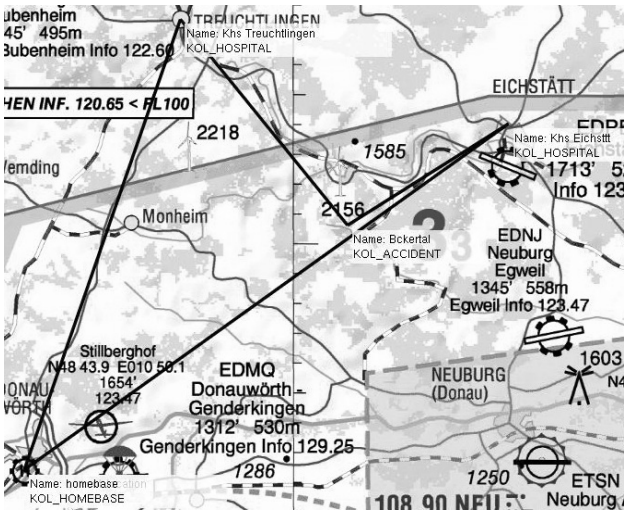


BILD 11. Karte der Flugroute

7. ZUSAMMENFASSUNG

Dieser Beitrag beschreibt das Konzept, die Implementierung und einige Ergebnisse der Funktionalität „Auftragsinterpretation“ als Teil eines wissensbasierten Assistenzsystems am Beispiel einer Hubschrauber-rettungsmission. Als grundlegender Theorieansatz wurde dabei der *Kognitive Prozess* als Modell menschlicher Informationsverarbeitung verwendet. Basis für die Implementierung war die *Kognitive Systemarchitektur COSA*, die speziell die Implementierung kognitiver Systeme, denen der Kognitive Prozess zugrunde liegt, unterstützt.

Die Auftragsinterpretation selbst kann einen Auftrag empfangen und verstehen, welches eine wesentliche Grundlage für ein Assistenzsystem ist, um einen Piloten bei Missionsdurchführung und Entscheidung über Auftragsannahme oder -ablehnung zu unterstützen. In einem ersten Schritt wird der extern gegebene Auftrag auf der Basis von a-priori Wissen über derartige Missionen vervollständigt, d.h. eine Auftragsauffassung erzeugt. Die anzufliegenden Orte werden auf Basis des Wissens zum Ablauf einer typischen Hubschrauberrettungsmission sortiert und als Missionselemente dargestellt. Beginnend bei der aktuellen Flug- und Missionsphase wird der Flug des Hubschraubers antizipiert. Der Antizipation liegen dabei die typischen Flugphasenübergänge während eines Hubschrauberfluges zugrunde. Für jede antizipierte Flugphase wird der Aufgabenliste eine Aufgabe hinzugefügt, deren Aufgabendetails zunächst lediglich aus typischen Annahmen für Standardwerte bestehen. Details, die im a-priori Wissen hinterlegt sind, werden abgefragt und anstelle der Standardwerte in der Aufgabenliste eingefügt. Für jede Aufgabe wird der Zeit- und Treibstoffbedarf bestimmt und zum Gesamtressourcenbedarf akkumuliert. Der Personal- und Infrastrukturbedarf wird anhand der Auftragsauffassung bestimmt und ebenfalls zum Gesamtressourcenbedarf hinzugefügt. Die Auftragsauffassung, die Aufgabenliste sowie der Gesamtressourcenbedarf stehen schließlich dem Assistenzsystem zur Verfügung. Dadurch wird das Assistenzsystem in die Lage versetzt, extern gegebene Aufträge (hier im Umfeld von Hubschrauber-rettungsmissionen) selbstständig im Sinne der Abwägung von Handlungsalternativen zu verstehen, gerade so, wie es auch ein geschulter Pilot tun würde. Aufbauend auf derartige Fähigkeiten wird Automationstechnologie im Cockpit künftig die Rolle eines partnerschaftlich agierenden, elektronischen Crew-Mitglieds übernehmen können.

DANKSAGUNG

Dieser Beitrag entstand aus meiner Diplomarbeit gleichen Titels angefertigt 2005 an der Professur für Flugmechanik & Flugführung, Fakultät für Luft- & Raumfahrttechnik, Universität der Bundeswehr München. Die Diplomarbeit wurde anlässlich des Deutschen Luft- & Raumfahrtkon-gresses 2006 mit dem ***Ludwig Bölkow-Stiftungspreis*** ausgezeichnet. Mein Dank gilt besonders den Betreuern der Arbeit und Koautoren dieses Beitrags.

LITERATUR

[1] Charles E. Billings. *Aviation Automation: The Search for a Human-Centred Approach.* Mahwah, NJ: Lawrence Erlbaum Associates. 1997.

[2] REFA. *Methodenlehre des Arbeitsstudiums, Teil 1-3*. 7. Auflage. Verband für Arbeitsstudien und Betriebsorganisation. München: Carl Hanser Verlag. 1984.

[3] Reiner Onken. Cognitive Cooperation for the Sake of the Human-Machine Team Effectiveness. In: *RTO-Meeting Proceedings MP-088, HFM-084: The Role of Humans in Intelligent and Automated Systems.* Warschau, Polen, 7. - 9. Oktober 2002.

[4] Axel Schulte & Claudia Meitinger. Human Factors in the Guidance of Autonomous Vehicles: Oxymoron or Tautology? The Potential of Cognitive and Co-operative Operator Assistant Systems. In: Erik Hollnagel et al. (Eds.). *Intelligent Decisions? Intelligent Support?* Erscheint voraussichtlich 2007.

[5] Lutz von Rosenstiel, Walter Molt & Bruno Rüttinger. *Organisationspsychologie*. 9. Auflage. Stuttgart: Kohlhammer. 2005.

[6] M. Leroux. The role of expert systems in future cooperative tools for air traffic controllers. In: *Proccedings of the Seventh International Symposium on Aviation Psychology*, Columbus, OH. S. 335-340. 1993.

[7] Michael Kriegel & Axel Schulte. Work System Analysis of the Integration of Autonomous Functions and Intelligent Operator Assistance in UAV Guidance. In: *RTO-Meeting Proceedings HFM-135: Human Factors of Uninhabited Military Vehicles as Force Multipliers*. Biarritz, Frankreich, 9. - 11. Oktober 2006.

[8] Reiner Onken. Basic Requirements Concerning Man-Machine Interactions in Combat Aircraft. In: *Proceedings of the Workshop on Human Factors / Future Combat Aircraft*. Ottobrunn, 4. - 6. Oktober 1994.

[9] Reiner Onken. *Optimaler Automatisierungsgrad von Mensch-Maschine-Systemen.* DGLR-Bericht 94-01. 1994.

[10] Diana Donath & Axel Schulte. Weiterentwicklungs-möglichkeiten kognitiver und kooperativer Operateur-assistenz: Ein Konzeptansatz für modellbasierte adaptive Automation. In: *DGLR Bericht „Cognitive Systems Engineering in der Fahrzeug- und Prozessführung“*. Karlsruhe, 24. - 25. Oktober 2006.

[11] Henrik Putzer & Reiner Onken. COSA – A generic cognitive system architecture based on a cognitive model of human behaviour. In: *International Journal on Cognition, Technology and Work*. Vol. 5, S. 140-151. 2003.

[12] Henrik Putzer. *Ein uniformer Architekturansatz für Kognitive Systeme und seine Umsetzung in ein operatives Framework.* Berlin: Köster, 2004. – Dissertation Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik

[13] MICO Homepage, http://www.mico.org, letzter Aufruf: 25.08.2006