

**UNIVERSITÄT DER BUNDESWEHR MÜNCHEN**

**FAKULTÄT FÜR  
LUFT- UND RAUMFAHRTTECHNIK**

**Thema der Dissertation:** Zentrale, objektorientierte Situationsrepräsentation  
angewandt auf die Handlungsziele eines  
Cockpitassistenzsystems

**Verfasser:** Anton Walsdorf

**Promotionsausschuss:**

**Vorsitzender:** .....

**1. Berichterstatter:** .....

**2. Berichterstatter:** .....

**3. Berichterstatter:** .....

**4. Berichterstatter:** .....

**Tag der Prüfung:** .....

**Mit der Promotion erlangter akademischer Grad:** .....

**(Dr. ....)**

**Neubiberg, den .....**



*Für meine Eltern*



## **Vorwort**

Die vorliegende Arbeit entstand im Rahmen meiner Arbeiten als Wissenschaftlicher Mitarbeiter am Institut für Systemdynamik und Flugmechanik an der Universität der Bundeswehr München. Viele in dieser Arbeit verwirklichten Ideen und Konzepte sind während meiner Tätigkeiten im Rahmen des Projektes „Crew Assistant Military Aircraft“ des Bundesamtes für Wehrtechnik und Beschaffung entstanden.

Motiviert und gefördert wurde diese Arbeit durch Prof. Reiner Onken. Sein unermüdliches Engagement in Form von Kommentaren und zahlreichen Diskussionen haben in den letzten Monaten entscheidend zur zügigen Fertigstellung dieser Arbeit beigetragen. Hierfür bin ich ihm sehr dankbar.

All denjenigen Menschen möchte ich danken, mit denen ich während der langen Jahre am Institut zusammenarbeiten durfte. Viel gemeinsamer Spaß und Freude an der Forschungsarbeit im Team haben dazu beigetragen, die zahlreichen beruflichen wie privaten Hürden im Leben eines Doktoranden zu bewältigen.

Gemeinsam mit und dank meines Freundes und Kollegen Henrik Putzer durfte ich sehr schöne, fröhliche Stunden während des Laboralltages erleben. Nicht zuletzt haben auch die vielen Fachdiskussionen mit Ihm zu zahlreichen guten Anregungen geführt. Vielen Dank hierfür.

Ganz besonderer Dank gilt meiner Freundin Julia, die im letzten Jahr viel Verständnis und Geduld für die vielen Wochenend- und Feierabendarbeiten während der Fertigstellung dieser Arbeit gezeigt hat.

Unterhaching, im Mai 2002



# Inhaltsverzeichnis

<b>1</b>	<b><u>EINLEITUNG</u></b>	<b>1</b>
1.1	FORDERUNGEN AN MENSCH-MASCHINE-SYSTEME	2
1.2	STAND DER TECHNIK BEI COCKPITASSISTENZSYSTEMEN	4
1.3	ZIEL UND AUFBAU DER ARBEIT	15
<b>2</b>	<b><u>ANALYSE DER ANFORDERUNGEN</u></b>	<b>18</b>
2.1	INNERE KONSISTENZ	18
2.2	KOOPERATION	19
2.2.1	EFFEKTIVE KOOPERATION	22
2.3	ERKLÄRBARKEIT DES VERHALTENS	23
<b>3</b>	<b><u>SYSTEMANALYSE</u></b>	<b>25</b>
3.1	INFORMATIONSVERRARBEITUNG DES MENSCHEN	25
3.2	DER KOGNITIVE PROZESS	29
3.2.1	SITUATIONSINTERPRETATION	32
3.2.2	SITUATIONS DIAGNOSE	33
3.2.3	PLANUNG UND ENTSCHEIDUNGSFINDUNG	34
3.2.4	AUFGABENDISPOSITION UND AUSFÜHRUNG	36
<b>4</b>	<b><u>ENTWURF</u></b>	<b>25</b>
4.1	SITUATIONSWISSEN	38
4.2	OBJEKTORIENTIERTE SITUATIONSREPRÄSENTATION	42
4.2.1	RELATIONALE STRUKTUREN DES SITUATIONSWISSENS	43
4.2.2	IMPLIZITE REPRÄSENTATION VON SITUATIONSWISSEN	49
4.3	BEITRÄGE DER TEILPROZESSE ZUM SITUATIONSWISSEN	50
4.3.1	SITUATIONSINTERPRETATION	52
4.3.2	SITUATIONS DIAGNOSE	54
4.3.3	PLANUNG UND ENTSCHEIDUNGSFINDUNG	69
4.3.4	AUFGABENDISPOSITION	71
4.4	WISSENSERWERB	73
4.4.1	PROBLEMSTELLUNG	74
4.4.2	METHODIKEN	75
4.5	RECHNERREPRÄSENTATION DES SITUATIONSWISSENS	76
4.5.1	WISSENSBASIS DER SITUATIONSELEMENTE	79
4.6	GESAMTSPEZIFIKATION	86

<b>5</b>	<b><u>IMPLEMENTIERUNG</u></b>	<b>87</b>
<b>5.1</b>	<b>SOFTWARE-ARCHITEKTUR</b>	<b>87</b>
5.1.1	MULTI-AGENTEN-SYSTEM	87
5.1.2	COMMON OBJECT REQUEST BROKER ARCHITECTURE	90
5.1.3	IDENTIFIKATION VON KLASSEN IN CORBA	91
5.1.4	UMSETZUNG DER SPEZIFIKATION IN C++	98
<b>5.2</b>	<b>WERKZEUGE</b>	<b>101</b>
5.2.1	MICO – OBJECT REQUEST BROKER	103
5.2.2	LEDA - BIBLIOTHEK	104
5.2.3	GRAPH-MONITOR	104
5.2.4	DIRECT-RATIO-BEDIENERSCHNITTSTELLE	105
5.2.5	FLIGHT-PROGRESS-MONITOR	105
<b>6</b>	<b><u>MODELLBILDUNG</u></b>	<b>107</b>
<b>6.1</b>	<b>WISSENSAKQUISITION DER HANDLUNGSZIELE</b>	<b>108</b>
6.1.1	FLUGLEISTUNG UND –FÜHRUNG	109
6.1.2	LUFTVERKEHRSVORSCHRIFTEN	112
6.1.3	MILITÄRISCHE TRANSPORTFLUGVERFAHREN	121
6.1.4	REPRÄSENTATION DES WISSENS	135
<b>6.2</b>	<b>MAKROSKOPISCHE STRUKTUR DES MODELLS DER HANDLUNGSZIELE</b>	<b>135</b>
6.2.1	GEFÄHRDUNGSMODELL	135
6.2.2	MODELL DES MISSIONSAUFTRAGES	139
6.2.3	MODELL DER BESATZUNGSABSICHTEN	140
<b>6.3</b>	<b>ERMITTLUNG DER ZIELGEWICHTE</b>	<b>141</b>
6.3.1	INSTRUMENTALZIELE	141
6.3.2	FUNDAMENTALZIELE	142
<b>6.4</b>	<b>SENSITIVITÄTSANALYSE DER HANDLUNGSZIELE</b>	<b>144</b>
<b>6.5</b>	<b>STRUKTUR DER RELEVANTEN OBJEKTE</b>	<b>146</b>
<b>7</b>	<b><u>ERPROBUNG</u></b>	<b>148</b>
<b>7.1</b>	<b>VERSUCHSSZENARIEN AUS FLUGVERSUCHSDATEN</b>	<b>149</b>
7.1.1	FLUGVERSUCHSABLÄUFE	151
<b>7.2</b>	<b>VERSUCHSAUFBAU IM LABOR</b>	<b>153</b>
<b>7.3</b>	<b>UNTERSUCHUNGSZIELE UND BEANSPRUCHUNG DES MODELLS</b>	<b>154</b>
7.3.1	LEISTUNG DER RELATIONALEN REPRÄSENTATIONSFORM	154
7.3.2	ERPROBUNG DES MODELLS DER HANDLUNGSZIELE	164
<b>8</b>	<b><u>ZUSAMMENFASSUNG</u></b>	<b>172</b>

<b>9</b>	<b>LITERATURVERZEICHNIS</b>	<b>174</b>
<b>10</b>	<b>ANHANG</b>	<b>178</b>
<b>10.1</b>	<b>UML KOMPAKT</b>	<b>178</b>
<b>10.2</b>	<b>EINZELERGEBNISSE DER GROUP ELICITATION METHOD</b>	<b>190</b>
<b>10.3</b>	<b>SPRACHE ZUR WISSENSREPRÄSENTATION</b>	<b>198</b>
<b>10.4</b>	<b>BEISPIELE ZUR MIKROSKOPISCHEN STRUKTUR VON ZIELMODELLEN</b>	<b>206</b>



# 1 EINLEITUNG

*Am 26. April 1994 befand sich der Flug 140 der China Airlines, aus Taipei kommend, im Anflug auf den japanischen Flughafen Nagoya. Der erste Offizier führte den Airbus A300-600R manuell entlang des ILS Anflugpfades zum Aufsetzpunkt der Piste 34. In einer Höhe von 1040 Fuß aktivierte der Co-Pilot versehentlich den TOGA (Take Off Go Around) Modus. Infolgedessen leitete der Autopilot einen Steigflug mit maximaler Triebwerksleistung ein. Der Kapitän machte den Co-Piloten sofort darauf aufmerksam und wies ihn an, den TOGA Modus zu deaktivieren. Die Situation schien wieder unter Kontrolle. Der Airbus befand sich zu diesem Zeitpunkt 5,5 km von seinem geplanten Aufsetzpunkt entfernt. Kurze Zeit später aktivierte der Co-Pilot ein zweites Mal den Autopiloten. Der TOGA Modus wurde hierdurch abermals aktiv und der Airbus begann aufs Neue ein Steigmanöver. Der Co-Pilot versuchte, den TOGA Modus wie üblich durch manuelles Übersteuern zu deaktivieren und drückte den Sidestick nach vorne, um den Sinkflug fortzusetzen. Zu diesem Zeitpunkt war ihm nicht bewusst, dass eine Deaktivierung des TOGA Modus in dieser Höhe nicht möglich war, da ein einmal aktivierter TOGA Modus durch Übersteuern erst oberhalb einer bestimmten Sicherheitsmindesthöhe wieder verlassen werden kann. Während der manuellen Steuereingaben der Besatzung blieb somit dieser Modus des Autopiloten weiterhin aktiv und versuchte, gegen die Besatzung arbeitend, durch Verändern der aerodynamischen Trimmung sein Ziel „maximales Steigen“ zu erreichen. Der Besatzung war es durch Reduzieren des Schubes und kontinuierliches Drücken mittlerweile gelungen, auf den geplanten Gleitpfad zurückzukehren. Das automatische Trimmen des Flugzeuges durch den Autopiloten führte den Airbus allerdings bis zu einem Anstellwinkel von 40°. Zu diesem Zeitpunkt fuhr die Besatzung die Auftriebsklappen aus, um dem drohenden Strömungsabriss entgegenzuwirken. Die Geschwindigkeit sank weiter. Bei einem Anstellwinkel von 52° und einer Geschwindigkeit von 78 Knoten ertönte die Überziehwarnung, und das Flugzeug ging in einen nicht mehr zu kontrollierenden Flugzustand über. Um 20.15 Uhr schlug der Airbus aus 1800 Fuß Höhe nur 110 Meter neben dem Aufsetzpunkt der Piste 34 auf. An diesem Tag starben 264 Personen, 7 Personen überlebten den Absturz schwer verletzt.*

Beispielhaft an dieser Katastrophe ist, dass der Grund für den Absturz weder durch technisches Versagen, noch durch mangelnde Flugerfahrung der beiden Besatzungsmitglieder (Kapitän 8340 h, Co-Pilot 1624 h) zu erklären ist. Unfallursache ist vielmehr das immer komplexer werdende Arbeitsumfeld an einem hochtechnisierten und mit komplexen

Automatismen ausgestatteten Arbeitsplatz, an dem Mensch und Maschine immer öfter im Konflikt bezüglich ihrer verfolgten Handlungsziele stehen. Hinzu kommt, dass der Mensch typischerweise nur diejenigen Teilprobleme bearbeitet, die er aktuell erkennt. Dabei läuft er Gefahr, bei nicht erkannten Problemen in daraus entstehende zusätzliche neue Problemsituationen zu geraten, die er, wenn überhaupt, erst im Zuge der Reaktion auf das erste Problem bzw. nach dessen Lösung erkennt. Das kann dann möglicherweise zu spät sein. Dies schlägt sich auch in den Ergebnissen des Abschlussberichts der untersuchenden Behörde AAIC (Aircraft Accident Investigation Commission, Japan) nieder (siehe AAIC, 1994).

## 1.1 FORDERUNGEN AN MENSCH-MASCHINE-SYSTEME

Insbesondere in hochtechnisierten Umgebungen tritt der Effekt der Überforderung des Menschen immer häufiger zu Tage. Das ungenügende Verständnis für das technische System und die oftmals fehlende Kompetenz des Menschen (möglicherweise auch durch mangelhafte Ausbildung oder fehlendes Training) in der Bedienung solcher Systeme führen immer öfter in kritische Situationen. Immer mehr Forscher und Entwickler sind sich der Risiken der gegenwärtigen Mensch-Maschine-Systeme bewusst. Die Identifikation der Probleme und der Risiken, welche in den bisherigen Automationsprinzipien stecken, ist weit fortgeschritten. Studien über bisher angewandte Automationsprinzipien stoßen besonders auf Schwächen in der Kooperationsfähigkeit zwischen Mensch und Maschine. In der hochautomatisierten Domäne der Luftfahrt tritt dieses Problem besonders deutlich zutage. Jedoch sind ganzheitliche Lösungen und Konzepte immer noch selten zu finden. Die von (Onken,1993) gestellten Grundforderungen legen Anforderungen an ein technisches Unterstützungssystem fest. Dabei steht das Ziel der „Aufrechterhaltung des Situationsbewusstseins“ und die „Reduzierung der Beanspruchung“ des Bedieners im Vordergrund. Diese Grundforderungen lauten (Onken, 1993):

- 1) Unter Berücksichtigung des Gesamtbildes der Flugsituation muss gewährleistet sein, dass die Aufmerksamkeit der Cockpit Besatzung auf die objektiv dringlichste Aufgabe gelenkt wird.
- 2) Ist Grundforderung (1) erfüllt und ergibt sich dennoch eine Situation, welche zu einer Überforderung der Cockpit Besatzung führt, soll diese Situation durch Anwendung von technischen Hilfsmitteln in eine für die Besatzung handhabbare Situation überführt werden.

Dieser qualitative Ansatz steht im Einklang mit den Richtlinien an den Entwurf eines Mensch-Maschine-Systems, welche von (Billings, 1991) im „Human-Centred Approach“ dargestellt wurden. Billings stellt hierbei Kriterien für eine „Human-Centred Automation“ auf, die die Interaktion des Bedieners mit einem automatischen System erleichtern und das Potential einer kritischen Situation vermindern sollen. Sein Anforderungskatalog umfasst unter der Prämisse der Bedienerautorität und dessen Prozessverantwortung sowohl funktionale als auch dem Verhalten eines Mensch-Maschine-Systems zuzuordnende Eigenschaften (Billings, 1991):

- 1) Bediener sind aktiv an der Prozessführung beteiligt
- 2) Bediener erhalten ausreichende Information über die Prozesssituation
- 3) Bediener überwachen den Prozess
- 4) Automation überwacht den Bediener
- 5) Automation verhält sich vorhersehbar
- 6) Situationsbewusstsein auf Bediener- und Automationsseite über die verfolgten Absichten

Diese Forderungen erweitern den Anforderungskatalog für Automation bezüglich der Kooperationsfähigkeit automatischer Funktionen. Damit resultieren die aufeinander aufbauenden technischen Anforderungen an ein kognitives Assistenzsystem in

- Basisfähigkeiten der Kognition zur Prozessüberwachung durch Situationsanalyse, Planung und Entscheidungsfindung sowie Planausführung,
- der Kompetenz, die anstehenden Aufgaben in der entsprechenden Arbeitsdomäne aufgrund vorhandener Expertise durchführen zu können und
- der Kooperationsfähigkeit, die ein partnerschaftliches, gemeinschaftliches Handeln im Team ermöglicht.

Die bisher entwickelten Prototypen von kognitiven Assistenzsystemen beginnen mit der schrittweisen Umsetzung dieses Anforderungsprofils.

## 1.2 STAND DER TECHNIK BEI COCKPITASSISTENZSYSTEMEN

Bereits in den 60er Jahren hat die Entwicklung von intelligenten Systemen mit den Expertensystemen begonnen. Diese verfügen über eine explizite Wissensbasis, in der statisches Problemlösungswissen gespeichert ist. Im Diskurs zwischen Nutzer und Rechner werden systematisch Merkmale des Problemraums gesammelt. Durch diese interaktive System- und Situationsanalyse kann das Repertoire an gespeicherten Problemlösungen immer weiter eingegrenzt werden. Jedoch sind diese Systeme in sich abgeschlossen und nur über eine spezifische Schnittstelle mit weiterem Wissen zu versehen.

Erst in den 70er Jahren kamen kognitive Ansätze in die Schlagzeilen, und es wurden erste Erfolge in der maschinellen Wahrnehmung verzeichnet. Die Umsetzung dieser Technologien zur Führung von Fahrzeugen kam in den 80er Jahren zum Tragen, ermöglicht nicht zuletzt durch das gestiegene Leistungsvermögen der Rechner. Mit steigendem Intelligenzgrad der Maschine begann schrittweise die Ära der kognitiven Assistenzsysteme. Kognitive Assistenzsysteme unterscheiden sich von herkömmlichen Systemen durch die Funktion, ihre Umwelt umfassend wahrzunehmen und die hieraus gewonnenen Situationsinformationen in geeigneter Weise zu analysieren und, darauf aufbauend, dem Operateur Unterstützungsleistung im Kontext der Gesamtsituation zur Verfügung zu stellen. In zahlreichen Forschungs- und Entwicklungsvorhaben sind für die unterschiedlichsten Anwendungen solche Systeme konzipiert worden. Assistenzsysteme können den folgenden Leistungsstufen zugeordnet werden (abgeleitet aus McBryan, 1999):

Level 1: “Systems try to make you better at what you do”

Level 2: “Systems help you with your tasks without needing instructions”

Level 3: “Systems do what you ask them to do”

Level 4: “Systems do what they know to do”

Systeme der Stufen 1 und 2 sind kooperativ und stellen damit den anzustrebenden Meilenstein in der Entwicklung automatischer Funktionen dar. Sie erarbeiten selbständig auf Basis der bewerteten Gesamtsituation eine eigene Vorstellung bezüglich des Handelns. Mit Hilfe dieser werden dem Operateur situationsangepasst Unterstützungsleistungen geboten. Die Luft- und Raumfahrt als innovative High-Tech-Branche hat bei der Entwicklung kooperativer Systeme

eine Vorreiterrolle bewiesen und viel Wissen für die Konzeption solcher Systeme geschaffen. Nachfolgend sollen die am weitesten gediehenen Projekte kurz vorgestellt werden.

### 1.2.1.1 SmartInteraction

Das Flight-Management-System moderner Flugzeugmuster stellt eine zentrale und komplexe Komponente der Flugführung dar. Die Bedienung dieses Systems erfolgt über eine CDU (Control and Display Unit) im Cockpit. Die Unterstützung des Operateurs bei der Bedienung dieser Kernkomponente ist Ziel des Systems *SmartInteraction* (siehe Romahn, 1997). *SmartInteraction* nutzt hierzu das bekannte, regelbasierte Wissen aus Handbüchern und von Experten über die Interaktionsprozeduren in einer wissensbasierten Systemarchitektur (siehe Abbildung 1).

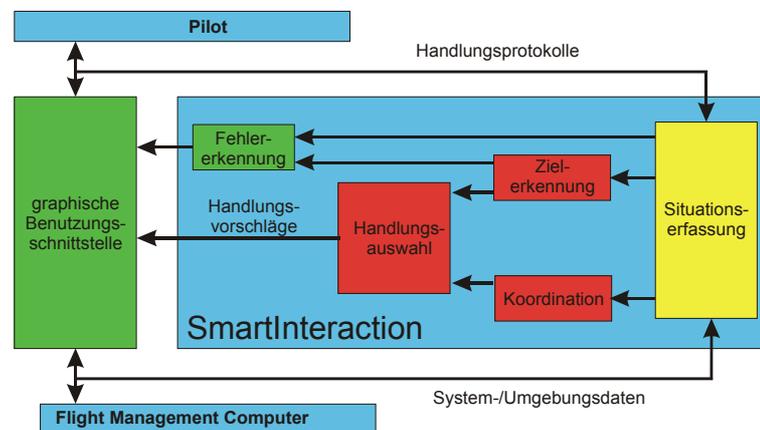


Abbildung 1: Systemarchitektur von SmartInteraction (Romahn, 1997)

Durch die Analyse der momentanen Flugsituation versucht *SmartInteraction*, die Interaktionsabsicht des Piloten zu erkennen. Auf Basis der ermittelten Handlungsalternativen für die Bedienung der CDU werden bei der Programmierung des Flight-Management-Computers Hilfestellungen gegeben. Hierfür besitzt das System eine graphische Bedienoberfläche, die sich bezüglich Funktionalität und Darstellung von herkömmlichen CDUs unterscheidet. Neben der reinen Unterstützungsfunktion in der Bedienung protokolliert *SmartInteraction* die vom Nutzer vollzogenen Eingaben zur Laufzeit, um in Bezug auf die normativen Prozeduren mögliche Bedienfehler zu erkennen.

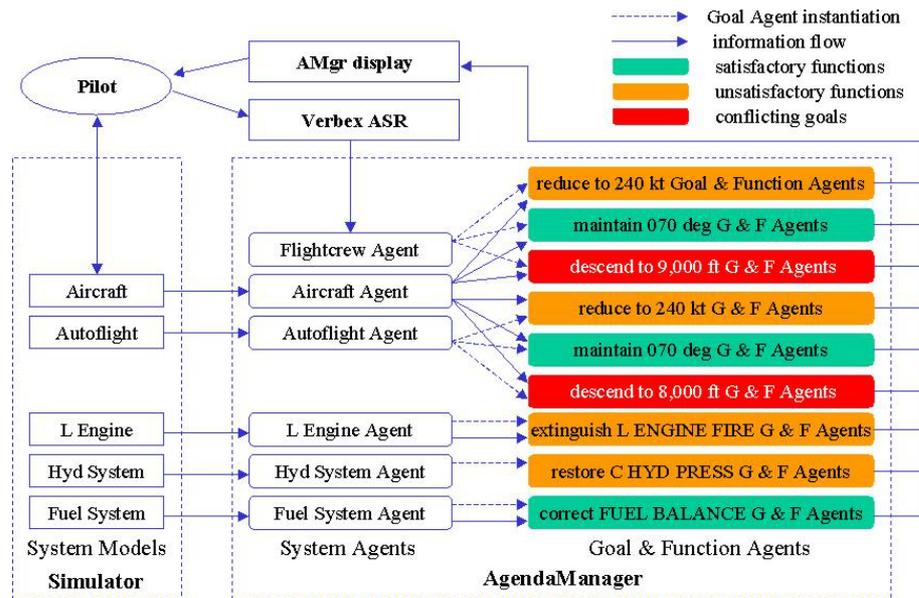
### 1.2.1.2 AgendaManager

Das High-Tech-Cockpit der jüngsten Flugzeugentwicklungen birgt eine Fülle an Aufgaben, die der Piloten erfüllen muss. Das so genannte Cockpit Task Management (CTM) ist der Prozess, bei dem der Pilot diese Aufgaben zum Zwecke der Missionserfüllung verrichtet. Dabei stellt die Auswahl der aktuell zu verrichtenden Aufgabe, gewichtet nach Dringlichkeit und Wichtigkeit, eine komplexe Situation für den Piloten dar. Nach (Funk, 1991) umfasst das Cockpit Task Management zahlreiche Ziele:

- Initiierung neuer Aufgaben
- Überwachen bereits laufender Aufgaben bezüglich ihres Zustandes, Priorisierung der Aufgaben auf Basis von Wichtigkeit, Systemzustand, Dringlichkeit
- Zuweisung menschlicher oder maschineller Ressourcen an übergeordnete Aufgaben
- Unterbrechung und nachfolgende Wiederaufnahme untergeordneter Aufgaben
- Identifikation bereits vollendeter Aufgaben

Das System *AgendaManager* stellt eine technische Unterstützung für das *Cockpit Task Management* dar und ist den Projekten *IAAS (Intelligent Air Attack System) Task Support System* (Funk, 1991) und dem *Cockpit Task Management System* (Funk, 1991) entsprungen. Hier wird der kognitive Ansatz zur Koordination dieser Cockpit Managementaufgaben verwandt. Der *AgendaManager* unterstützt den Piloten beim Erkennen und Lösen von Handlungskonflikten und der Priorisierung und Ausführung von Aufgaben an seinem Arbeitsplatz (siehe Abbildung 2).

Der *AgendaManager* ist ein Multi-Agenten-System. Die Hauptkomponenten sind die Module der *System Agents*, *Actor Agents*, *Goal Agents*, *Function Agents*, ein *Agenda Agent* und eine *AgendaManager*-Benutzeranzeige. Jedem Agenten ist hierbei ein Teilsystem aus dem Cockpit zugeordnet. Zweck des Agenten ist es, ablaufbezogene Informationen des Teilsystems bereit zu stellen und damit das Agenda Management zu erleichtern.



**Abbildung 2: Systemarchitektur des AgendaManager (Funk, 1991)**

Den Systemen *SmartInteraction* und *AgendaManager* ist gemeinsam, dass sie den Operateur nur in Teilbereichen unterstützen. Das gesamte Aufgabenspektrum der Flugführungsaufgaben

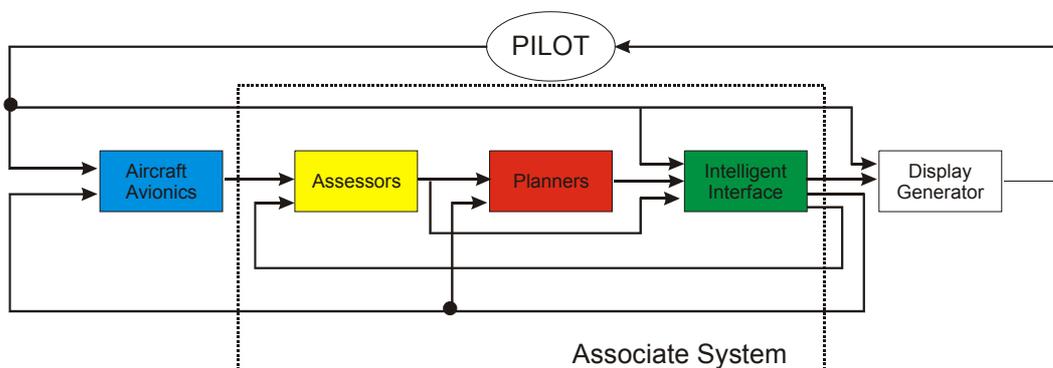
- Flugzeugsteuerung,
- Navigation,
- Kommunikation und
- Management der Systeme

bedarf umfassenderer Konzepte.

Im Bereich der Militärluftfahrt hat man schon frühzeitig versucht, den Piloten bei der Missionsdurchführung durch einen Rechner zu unterstützen. Hierbei standen die Situationsanalyse sowie Planung und Entscheidungsfindung bezüglich der strategischen Inhalte eines militärtaktischen Fluges im Vordergrund. Vertreter intelligenterer Assistenzsysteme für die Militärluftfahrt sind der *Pilot's Associate* und der *Copilote Electronique* für Kampfflugzeuge, und der *Rotorcraft Associate* für Kampfhubschrauber.

### 1.2.1.3 Pilot's Associate

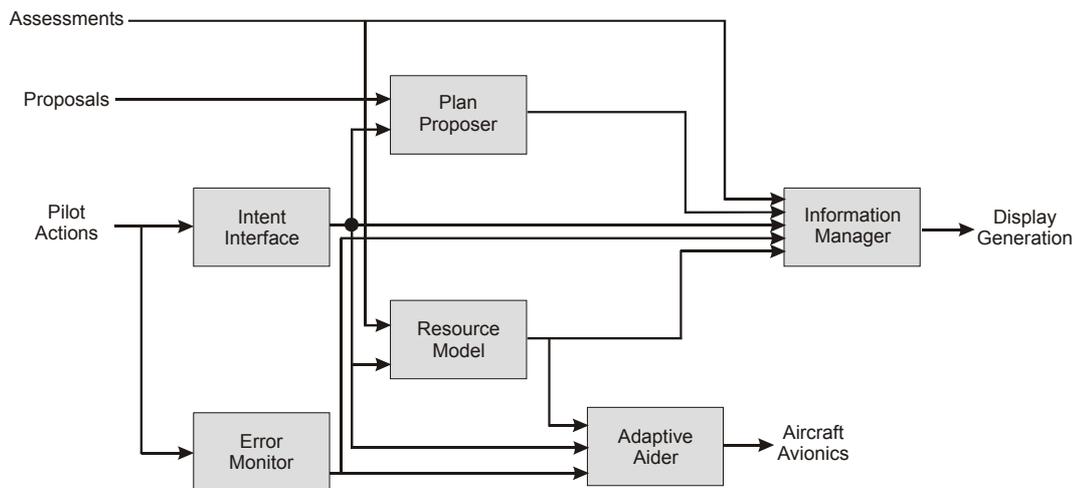
Mitte der 80er Jahre unternahm man erste Versuche, intelligente Systeme zur Unterstützung des Piloten zu entwickeln die das gesamte Spektrum der vorherrschenden Situation berücksichtigten sollten. Erstes System dieser Art ist der „Pilot's Associate“ (PA) (siehe Banks, 1991; Lizza, 1990/1991; Small, 1988/1992/1995). Dieses Forschungsvorhaben, initiiert und gefördert von der Defense Advanced Research Projects Agency (DARPA) und der US Air Force, hatte zum Ziel, den Piloten eines fortschrittlichen, einsitzigen, taktischen Kampfflugzeuges während seiner Mission beim Prozess der Entscheidungsfindung zu unterstützen. Forschungsschwerpunkt war die Mensch-Maschine-Schnittstelle, welche man durch ein intelligentes, wissensbasiertes System effizienter und effektiver in der Bedienung zu machen versuchte. Kernanforderung an den PA stellte die dem Situationskontext angepasste Aufbereitung und Darstellung von Informationen, die aus dem Situationswissen generierten Informationen zur Unterstützung beim Prozess der Entscheidungsfindung und die kooperative Umsetzung einer vom Piloten gefällten Entscheidung dar. Die Systemarchitektur des PA ist in Abbildung 3 dargestellt.



**Abbildung 3: Systemarchitektur des Pilot's Associate (aus Small, 1995)**

Die aus den Avioniksystemen gewonnenen Informationen werden vom Modul „Assessors“ aufbereitet und für die Weiterverarbeitung im Modul „Planners“ nutzbar gemacht, welches hieraus Vorschläge für ein mögliches sinnvolles Vorgehen zu erarbeiten hat. Diese werden vom Modul „Intelligent Interface“ (siehe Abbildung 4) weiterverarbeitet, um sie anschließend dem Piloten über eine graphische Anzeige darzubieten. Neben der Präsentation eines „assessment“, eines „plan“ oder anderer Informationen ist es Aufgabe des „Intelligent Interface“ Aufgaben auszuführen, den Piloten bezüglich möglicher Handlungsfehler zu überwachen und diesen im Falle des Auftretens solcher darüber zu informieren oder den Fehler durch selbstständigen Eingriff in das Flugzeugsystem zu beheben. Das Erkennen der

Absichten des Piloten dient einer zielgerichteten Analyse der Situation durch das Modul „Assessors“ und einer angepassten Planung im Modul „Planners“.



**Abbildung 4: Architektur des Moduls „Intelligent Interface“ (aus Small, 1995)**

#### 1.2.1.4 Co-Pilot Electronique

Ebenfalls die Unterstützung des Piloten eines Kampfflugzeuges ist das Ziel des *Copilote Electronique* (CPE), welcher Anfang der 90er Jahre unter Federführung der Firma Dassault und der Kooperation zahlreicher Industriefirmen entwickelt wurde (siehe Champigneux, 1989/1997). Aufgabe dieses Systems sollte es sein, während eines Kampfeinsatzes in einem durch feindliche Kräfte besetzten Gebiet die Besatzung bei ihrer Aufgabe in Planung und Entscheidungsfindung durch ein wissensbasiertes System zu unterstützen. Der Aufbau des CPE basierte auf einer Multi-Agenten-System-Architektur, welche in ihrem Aufbau dem menschlichen Problemlösungsprozess für den Bereich der Flugzeugsteuerung nachempfunden wurde (siehe Abbildung 5). Die Unterstützungsfunktionalität des CPE war für die Aufgabenbereiche des Flugzeugsystem-, des Taktik- und Missionsmanagements sowie der Mensch-Maschine-Koordination konzipiert.

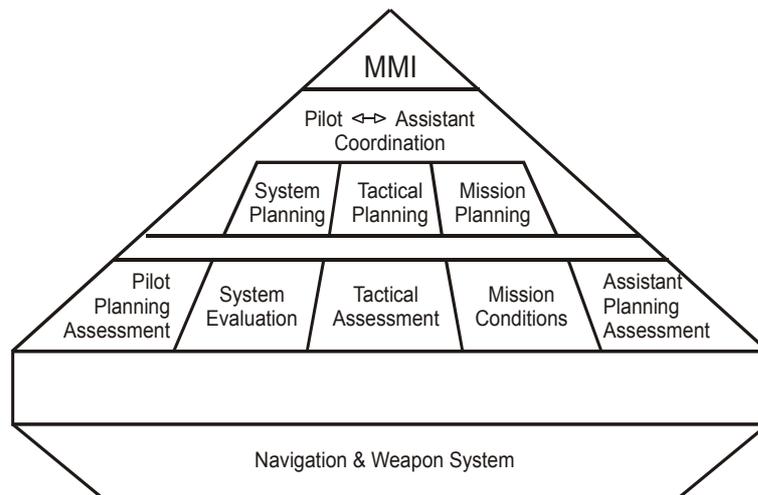


Abbildung 5: Systemarchitektur des Copilote Electronique (Champigneux, 1997)

### 1.2.1.5 Rotorcraft-Pilot's Associate – RPA

Der Rotorcraft-Pilot's Associate (siehe McBryan, 1999; Kupferer, 1999) ist ein kognitives Unterstützungssystem für Hubschrauberpiloten, entwickelt im Auftrag der US Army. Ziel ist es, die Besatzung eines militärischen Kampfhubschraubers bei ihren Entscheidungen in einem informationsüberfluteten Kampfgebiet angemessen zu unterstützen. Dabei leistet der RPA Unterstützung durch Aufgabenübernahme und durch Handlungsempfehlungen an die Besatzung. Der RPA umfasst 15 Softwaremodule, welche Funktionen übernehmen und Informationen bereitstellen: Sensor- und Datenfusionierung, digitale Geländedatenbank, digitale Kartenanzeigen, Kampfgebiet sowie interne und Teamzustands-Analyse, Waffensystemauswahl, Routen-, Angriffs- und Aufklärungs- sowie Kommunikationsplanung. Die funktionale Architektur des RPA ist in Abbildung 6 dargestellt.

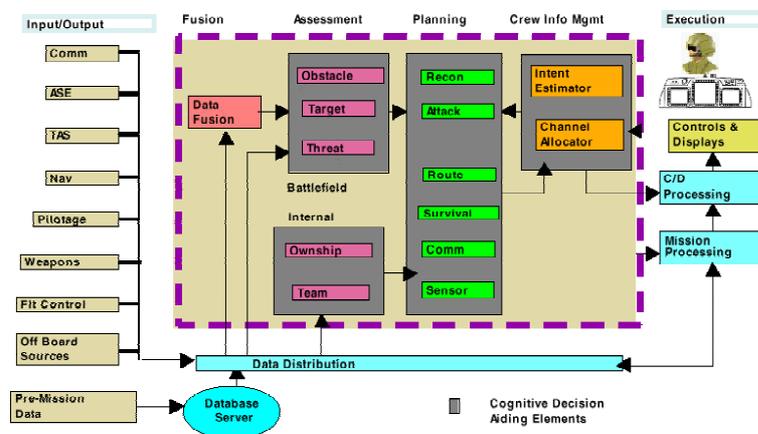


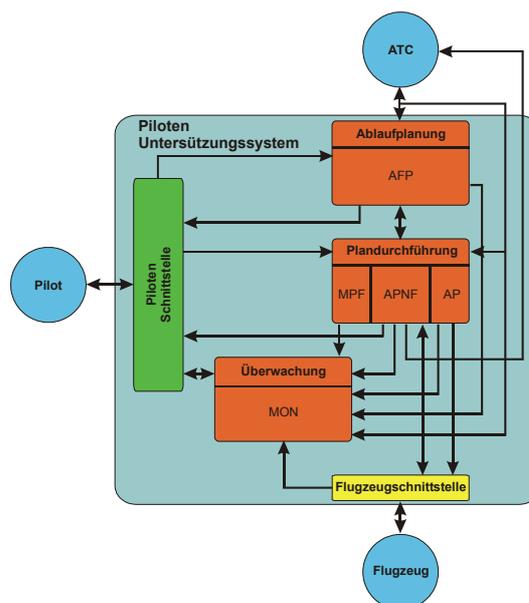
Abbildung 6: Systemarchitektur des Rotorcraft-Pilot-Associate (McBryan, 1997)

Allen bis hierher vorgestellten Systemen ist gemein, dass sie dem Operateur Unterstützungsleistung nur in Teilbereichen, sei es funktional, tätigkeits- oder missionsorientiert, bieten. Ziel der Forderungen an ein Mensch-Maschine-System ist es aber, das gesamte funktionale Spektrum durch geeignete Automationsansätze in allen Flugphasen abzudecken.

In der Luftfahrt ist dies durch die Prototypensysteme *ASPIO* (Assistant for Single Pilot IFR Operations) und *CASSY* (Cockpit Assistant System) für die Zivile Verkehrsfliegerei sowie das System *CAMA* (Crew Assistant Military Aircraft) für die militärische Transportfliegerei am weitgehendsten umgesetzt worden.

### 1.2.1.6 Assistant for Single IFR Operations – ASPIO

Das System ASPIO stellte eines der ersten Systeme dar, welches den Piloten in der Gesamtheit seiner Flugführungsaufgaben unterstützt (siehe Abbildung 7).



**Abbildung 7: Systemarchitektur des Assistant for Single Pilot IFR Operation (Dudek, 1990)**

ASPIO ist aus dem Bestreben eines Single-IFR-Fluges entstanden und basiert in seiner Arbeitsweise auf der Planung und Problemlösungsebene. Die Überwachungsfunktion behandelt Konflikte im Flugplanverlauf, welche von der Plandurchführungskomponente erkannt wurden, und bietet dem Operateur eine der Situation angemessene Unterstützung. ASPIO besteht aus 5 Modulen, welche in kleinere Teilmodule zerfallen. Die Ablaufplanung beinhaltet den AFP (Automatic Flightplanner), welcher der autonomen Flugplangenerierung

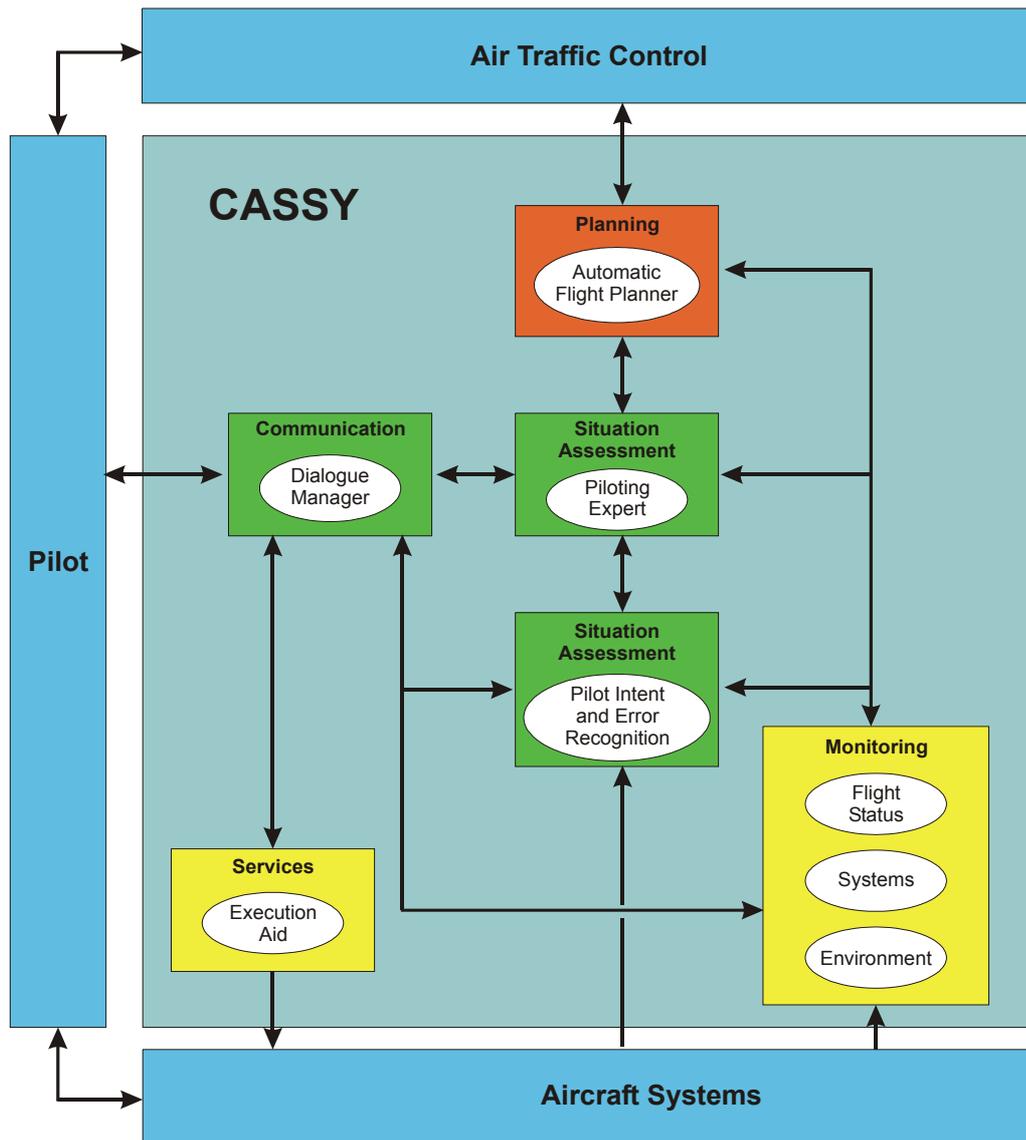
dient. Flugpläne werden der Besatzung zunächst als Vorschlag dargeboten, welche diese dann akzeptieren, modifizieren oder verwerfen kann. Die Plandurchführung bedient sich des MPF (Modell Pilot Flying), des APNF (Automatic Pilot-Non-Flying) und des AP (Autopilot). Der Vergleich des tatsächlichen mit dem erwarteten Piloten-Verhalten erfolgt durch das Modul MON (Monitor). Die Aufgaben der Module von ASPIO sind in Tabelle 1 dargestellt.

<b>Aufgabenstellung</b>	<b>Aufgabenzuweisung</b>
Überwachen der Pilotenaktionen durch Verfolgen des Flugablaufes und eintreffender Informationen	MPF / MON
Eintreffende Informationen auf Notwendigkeit der Flugplanänderung untersuchen	AFP
Callouts	MPF / APNF
Planungs- und Entscheidungsvorschläge	AFP
Navigations- und Flugleistungsberechnung auf Anweisung	APNF
Bedienaufgaben auf Anweisung	APNF
ATC-Kommunikation	MPF / APNF
Briefing	MPF / APNF
Checklistenbearbeitung	MPF / APNF

**Tabelle 1: Aufgabenzuweisung der ASPIO-Module (Dudek, 1990)**

#### 1.2.1.7 Cockpit Assistant System – CASSY

CASSY (Cockpit Assistant System) ist ein wissensbasiertes System zur Unterstützung der Cockpitbesatzung im zivilen Instrumentenflug und wurde in den Jahren 1991 bis 1994 an der Universität der Bundeswehr München, mit Unterstützung der Firma Dornier Luftfahrt GmbH Friedrichshafen entwickelt (siehe Onken, 1993/1995/1997). Ziel von CASSY ist die Erhöhung der Flugsicherheit durch eine autonome Erfassung der Flugesamtsituation. Der Gewinn an Flugsicherheit wird bei CASSY durch eine Steigerung des Situationsbewusstseins der Besatzung und im Falle einer Überlastung derselben durch technische Unterstützung bei der Aufgabenausführung gewährleistet. Dabei greift das System auf eine interne Situationsrepräsentation zurück, die den Grundstein des Situationsverständnisses ausmacht. CASSY wurde aufgrund der heterogenen Aufgabenstruktur in mehrere Module gegliedert (siehe Abbildung 8).



**Abbildung 8: Systemarchitektur des Cockpit Assistant Systems CASSY (Prévôt, 1995)**

Das Verhalten des Systems wird bei CASSY überwiegend durch den Flugplan bestimmt, der nach Vorgabe von Start und Ziel unter Berücksichtigung von Vorgaben der Besatzung, Anweisung der Flugsicherung, Wetterinformationen, Flugleistungsdaten u.v.a. einen konfliktfreien Flugplan generiert (siehe Prévôt, 1995). Ein normatives, regelbasiertes Pilotenmodell (siehe Ruckdeschel, 1997) interpretiert diesen Flugplan und leitet daraus das erwartete Pilotenverhalten ab. Die Unterstützung bezüglich des Erkennens von Fehlern und Absichten des Piloten wird durch die Interpretation des erwarteten mit dem tatsächlichen Pilotenverhaltens geleistet (siehe Wittig, 1994). Der Dialog zwischen CASSY und dem Piloten geschieht über eine eigens konzipierte Mensch-Maschine-Schnittstelle (siehe Gerlach, 1996).

### 1.2.1.8 Crew Assistant Military Aircraft – CAMA

CAMA baut auf den Kernkomponenten des erfolgreichen CASSY auf und wurde in den Jahren 1991 bis 2001 an der Universität der Bundeswehr München mit Unterstützung des Deutschen Zentrums für Luft- und Raumfahrt, der European Aeronautic and Defense System AG und der Elektroniksystem und Logistik GmbH entwickelt und in Simulationskampagnen (siehe Schulte, 1998) und Flugversuchen (siehe Frey, 2001) von erfahrenen Operateuren getestet. CAMA bietet über die Funktionalität von CASSY hinaus Unterstützungspotential für eine militärische Transportmission, welche sich neben den instrumentenflugähnlichen Reisefluganteilen durch taktische Flugabschnitte in Bodennähe unter feindlicher Bedrohung auszeichnet. CAMA hat das Konzept des kognitiven Unterstützungssystems weiter konkretisiert und hat sich von Aufbau (siehe Abbildung 9) und Architektur weiter zu einem kognitiven, kooperativen Assistenzsystem entwickelt.

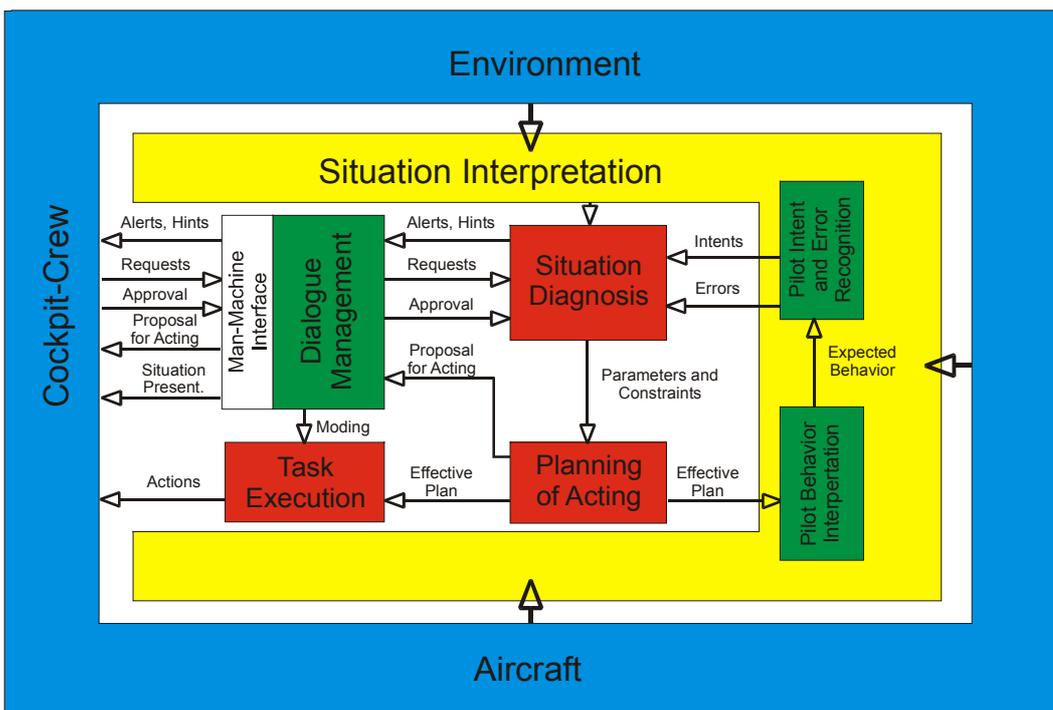


Abbildung 9: Systemarchitektur des Crew Assistant Military Aircraft (Walsdorf, 1998; Frey, 1999; Strohal, 1999; Lenz, 2000; Stütz, 2000)

### 1.3 ZIEL UND AUFBAU DER ARBEIT

Das Situationsverständnis kognitiver Systeme zum Zwecke einer Verbesserung der Kooperationsfähigkeit zwischen dem Menschen und der Maschine weiter zu steigern, ist Ziel dieser Arbeit. Aufgabe soll es hier sein, einen neuartigen Systemansatz zu entwickeln, der wesentlich dazu beiträgt, eine konsistente, zielorientierte Funktionsweise der kognitiven Teilfunktionen eines kognitiven Assistenzsystems sicherzustellen. Dies ist insbesondere durch den beachtlichen Anstieg an Komplexität innerhalb der Assistenzsysteme erforderlich geworden. Der Systemansatz soll dabei den Anforderungen genügen, die Handlungsziele zentral und in objektorientierter Form im System zu repräsentieren. Darüber hinaus soll es möglich sein, diese Handlungsziele zum Zwecke des gemeinschaftlichen Situationsbewusstseins zwischen Maschine und Mensch kommunizieren zu können. Das Situationswissen des Assistenzsystems soll hierfür in symbolischer Form als dessen „mentaler Zustand“ repräsentiert werden. Dies ist eine wichtige Voraussetzung für die Fähigkeit des Assistenzsystems, seine Handlungsziele für eine effektive Kooperation mit der Cockpitbesatzung und die daraus abgeleiteten Handlungen selbst zu erläutern, d.h. erklären zu können.

Der Entwurfs- und Entwicklungsprozess soll dabei sowohl den Anforderungen einer systematischen wissenschaftlichen Arbeit als auch den Erfordernissen von SW-Standards bei der Implementierung genügen. Für den Aufbau dieser Arbeit wurde daher eine Struktur gewählt, die dem Entwurfsprozess objektorientierter Systeme entspricht. Die Arbeit gliedert sich in die Abschnitte

- Analyse der Anforderungen,
- Systemanalyse,
- Entwurf,
- Implementierung,
- Modellbildung und
- Erprobung.

Die entwickelten Konzepte werden frühzeitig im Standard der Unified Modelling Language (UML) dargestellt (siehe Anhang 10.1). Ziel der UML ist es, die SW- Entwicklung und - Pflege bei hoher Komplexität handhabbar zu machen. Durch die Strukturierung des

Anwendungssystems in einzelne, überschaubare Teilfunktionen wird der Problemraum für den Entwickler verständlich gehalten.

Es werden zunächst im Rahmen der Analyse der Anforderungen die Leistungsmerkmale an die *zentrale, objektorientierte Situationsrepräsentation* definiert. Hierzu zählen insbesondere die innere Konsistenz, eine effektive Kooperation zwischen Mensch und Maschine sowie die Erklärungsfähigkeit bezüglich der verfolgten Handlungsziele.

Die Voraussetzung für die Umsetzung dieser Anforderungen, insbesondere für eine effektive Mensch-Maschine-Kooperation, ist die maschinelle Analyse der Situation. Im Rahmen der Systemanalyse wird die Funktionsweise der maschinellen Situationsanalyse aus den Erkenntnissen der menschlichen Informationsverarbeitung abgeleitet. Die Grundlagen der Kognition werden zur Analyse des Gesamtsystems nachfolgend erörtert.

Auf Basis des gewonnenen Systemansatzes wird ein Entwurf erstellt, welcher den gestellten Anforderungen an eine *zentrale, objektorientierte Situationsrepräsentation* genügt. Die für die Modellbildung entscheidende Aufgabe der Wissenserhebung wird abschließend erörtert, und es wird die Frage beantwortet, welche Repräsentationsform für das gewonnene Wissen herangezogen werden kann, die für Mensch wie Maschine verständlich ist.

Die Umsetzung der *zentralen, objektorientierten Situationsrepräsentation* in eine geeignete Software-Architektur erfolgt mit Hilfe standardisierter Entwicklungsverfahren unter Berücksichtigung von etablierten Ansätzen, die dem Fachgebiet der künstlichen Intelligenz entnommen werden. Die in dieser Arbeit gewählte Implementierung stellt ein verteiltes Multi-Agenten-System dar. Die verwandten Softwarewerkzeuge werden abschließend vorgestellt.

Die Modellbildung zielt hier auf den Anwendungsfall eines militärischen Transportfluges. Basierend auf den erarbeiteten Methoden zu Wissensakquisition und Wissensrepräsentation erfolgt die Erstellung entsprechender *expliziter „mentaler“ Modelle*. Die ermittelten Handlungsziele stellen hier den Inhalt der *zentralen, objektorientierten Situationsrepräsentation* eines kognitiven Cockpitassistenzsystems.

Die Erprobung der *zentrale, objektorientierte Situationsrepräsentation* erfolgt hinsichtlich des Modells der Handlungsziele unter Laborbedingungen anhand aufgezeichneter Daten, die während der Flugversuchserprobung eines kognitiven Cockpitassistenzsystems gewonnen wurden. Die Bewertung der Kompliziertheit der Situation und die auftretende

Situationskomplexität sind Teil der quantitativen Untersuchungen. Ziel der qualitativen Untersuchung ist es, die in den Systementwurf eingeflossenen Anforderungen anhand möglichst realitätsnaher Belastungen zu bestätigen.

## 2 ANALYSE DER ANFORDERUNGEN

Die Handlungsziele eines kognitiven Assistenzsystems sollen zentral und in einer objektorientierten Form repräsentiert werden. Hiermit soll

- die innere Konsistenz des kognitiven Systems verbessert,
- eine effektive Mensch-Maschine-Kooperation gefördert und
- das Systemverhalten dem Operateur erklärbar gemacht werden.

Dafür bedarf es eines interdisziplinären Ansatzes. Die Argumentationskette für den gewählten Weg soll in diesem Kapitel kurz dargestellt werden. Die innere Konsistenz eines kognitiven Systems ist durch die übergeordneten Handlungsziele bestimmt, die wiederum durch den Prozess der Entscheidungsfindung das Systemverhalten bestimmen. Entsprechend der Forderung an ein kooperatives Mensch-Maschine System werden nachfolgend verschiedene Kooperationsprinzipien diskutiert. Das Ergebnis dieser Analyse wird dem Systementwurf zu Grunde gelegt. Ferner werden die Voraussetzungen für eine effektive Kooperation anhand von Studien zur Gruppenarbeit erarbeitet. Diejenigen Mechanismen, mit denen dem Operateur das intentionale Verhalten des Systems erklärt wird, werden abschließend erörtert.

### 2.1 INNERE KONSISTENZ

Die Verbesserung der inneren Konsistenz dient in erster Linie dem Prozess der Entscheidungsfindung. Der Begriff der Entscheidungsfindung wird in unterschiedlicher Weise definiert. Zum einen wird das Entscheiden als Unterklasse des Problemlösens und damit als Bewertung von Alternativen gegenüber vorgegebenen Handlungszielen und deren Auswahl gesehen, zum anderen wird das Problemlösen als untergeordneter Schritt im Entscheidungsprozess verstanden. Diese Vorstellung über das Entscheiden trifft sowohl für den Menschen als auch für das Assistenzsystem zu. Folgende Handlungen einer Gruppe von Individuen (hier: Mensch und Assistenzsystem) sind für den Entscheidungsfindungsprozess typisch (von Rosenstiel, 1995):

- Ziele setzen und/oder vereinbaren,
- Ziele auf bestimmten Wegen zu erreichen suchen und
- Mittel für die Zielerfüllung beschaffen.

Eisenführ und Weber führen aus (Eisenführ, 1993):

- Ohne Klarheit über die Ziele ist keine rationale Entscheidung möglich. Einsicht in die Ziele fördert auch das Erkennen besserer Alternativen und lohnender Entscheidungschancen.

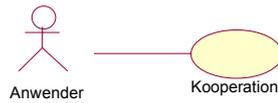
Für die Entscheidungsfindung ist damit das Wissen über die verfolgten Handlungsziele von hoher Wichtigkeit. Die Ziele bilden die Basis des Handelns und führen z.B. durch die Bewertung von Handlungsalternativen zu einer Entscheidung bezüglich des weiteren Vorgehens. In kognitiven Systemen ist der Prozess der Entscheidungsfindung oftmals verteilt, d.h. an einer Entscheidung sind mehrere Unterprozesse beteiligt. Diesen Unterprozessen müssen die übergeordneten Handlungsziele bekannt sein, damit sie zielgerichtet eine Entscheidung erarbeiten können. Der Grad an Übereinstimmung der diesen Unterprozessen bekannten Handlungsziele führt zur inneren Konsistenz. Es deutet vieles darauf hin, dass dies mit einer *zentralen, objektorientierten Situationsrepräsentation* und einem *Modell der Handlungsziele* auf die sicherste Weise gewährleistet wird.

## 2.2 KOOPERATION

Die Entscheidungsfindung in einer Gruppe kann auch in Konflikte münden, wie (von Rosenstiel, 1995) ausführt. Den verschiedenen Handlungsschritten zur Entscheidungsfindung ordnet er spezifische Konfliktarten zu:

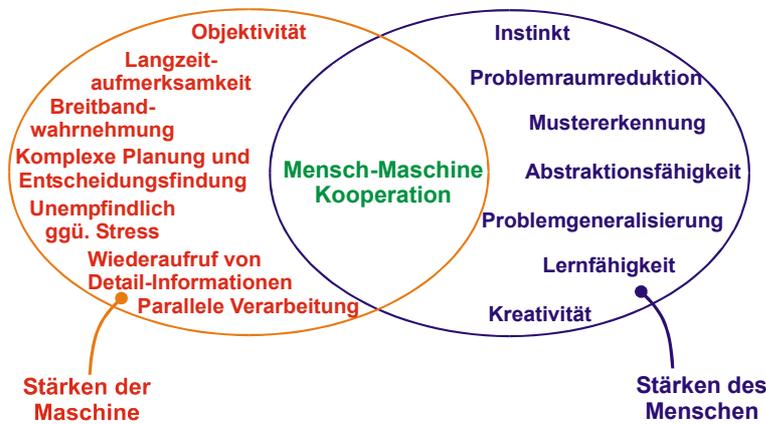
- Uneinigkeit über die Ziele,
- Uneinigkeit über die Wege und
- Uneinigkeit über die einzusetzenden Mittel.

Die Konfliktbewältigung muss im Sinne eines guten Zusammenspiels in der Gruppe Mensch-Assistenzsystem gelöst werden, indem Kooperation zum Handlungsziel gemacht wird. Kooperation ist demzufolge von zentraler Bedeutung für eine sinnvolle Entscheidungsfindung in einer Gruppe von Individuen. Hiermit ergibt sich ein erster Schritt zur Systemspezifikation mit dem Use-Case der „Kooperation“ (siehe Abbildung 10).



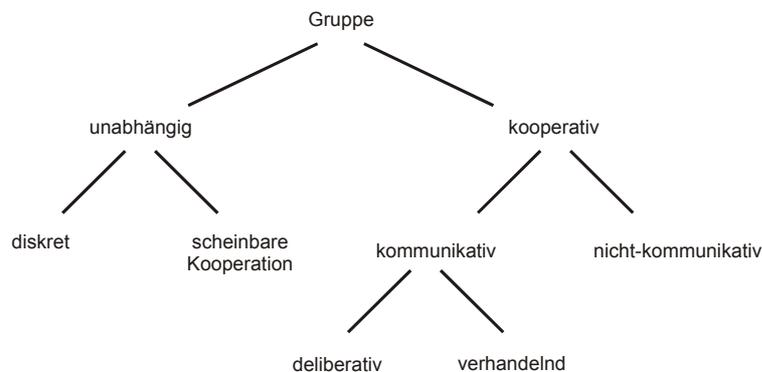
**Abbildung 10: Use-Case Diagramm „Kooperation“**

Die Fähigkeit zur Kooperation mit dem Menschen stellt an eine Maschine besondere Anforderungen. Mensch und Maschine können zunächst als eine Kooperationseinheit bzw. als ein Team betrachtet werden, welches auf der Basis einer vorherrschenden Situation gemeinsam auf ein Ziel hinarbeiten sollte. Dabei ist es das Ziel, die dem Menschen eigenen Schwächen durch die Stärken der Maschine sinnvoll zu kompensieren (siehe Abbildung 11).



**Abbildung 11: Synergieeffekte aus der Mensch-Maschine-Kooperation (Walsdorf, 1999)**

Doran und Franklin (Doran, 1997) klassifizieren Kooperation wie in Abbildung 12 dargestellt.

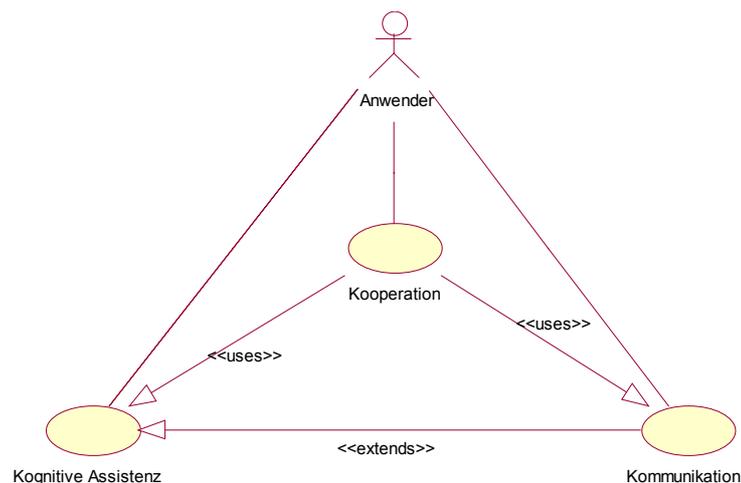


**Abbildung 12: Prinzipien der Kooperation (Doran, 1997)**

Verfolgen die Mitglieder einer Gruppe ausschließlich ihre eigenen Ziele, so handeln sie *unabhängig* voneinander. Liegt keinerlei inhaltlicher Zusammenhang zwischen diesen Zielen vor, so bezeichnet Franklin dieses System als *diskret*. Eine Kooperation liegt explizit nicht vor. Verfolgen die Mitglieder einer Gruppe inhaltlich ihre eigenen, aber gleichen Ziele, so erscheint dies einem außenstehenden Betrachter als *scheinbar kooperativ*. Echte kooperative Systeme besitzen explizite *Mechanismen zur Kooperation*. Franklin unterscheidet hierbei *kommunikative* und *nicht-kommunikative* Mechanismen. Die nicht-kommunikative Kooperation stützt sich auf die Reflexion des Handelns in der Umwelt. Von kommunikativer Kooperation ist die Rede, wenn Mitglieder einer Gemeinschaft zielgerichtet und explizit Informationen untereinander austauschen.

Die Kommunikation zum Zwecke der Kooperation wird weiter in *deliberativ* und *verhandelnd* unterschieden. Die gemeinsame Planung und Abstimmung des Vorgehens ist Eigenschaft der *deliberativen* Kooperation, bei der bereitgestellte Lösungen durch Wettbewerb selektiert werden. Bei der *verhandelnden* Kooperation werden Aufgaben verteilt und Konflikte gemeinsam gelöst.

Um den Anforderungen an ein Mensch-Maschine-System gerecht zu werden und die Kooperation in einem Team aus Mensch und Maschine zu ermöglichen, muss eine Kommunikation über die gemeinsamen Handlungsziele erfolgen können. Diese ist im Use-Case-Diagramm „Kommunikative Kooperation“ dargestellt (siehe Abbildung 13).



**Abbildung 13: Use-Case Diagramm „Kommunikative Kooperation“**

Die Handlungsziele sind zum Zweck einer expliziten Darstellung hierfür in *objektorientierter* Form zu repräsentieren.

### 2.2.1 EFFEKTIVE KOOPERATION

(Orasanu, 1995) hat Faktoren für Gruppenarbeit untersucht und dabei folgende Gruppeneigenschaften herausgearbeitet:

- hohes Situationsbewusstsein,
- hoher Grad an 'Metakognition',
- effiziente Verteilung der gemeinsamen kognitiven Ressourcen und
- verteilte mentale Modelle.

Aus diesen Faktoren ergeben sich weitere Anforderungen hinsichtlich kognitiver Fähigkeiten zur Sicherstellung eines effektiven, kooperativen Zusammenwirkens. Hier wird auch von mentalen Modellen als wesentliche, die Kooperation unterstützende Komponenten gesprochen.

Unter mentalen Modellen versteht man individuelle Denkmodelle, die das Verständnis eines Sachverhalts prägen. Mentale Modelle helfen, günstige Lösungspfade bei der Problembehandlung schnell zu finden. Mentale Modelle sind dynamisch, d.h. sie werden mit zunehmendem Verständnis eines Sachverhalts, eines Objektes oder eines Prozesses überarbeitet und angepasst. Zwischen Mensch und Maschine verteilte, konsistente mentale Modelle stellen einen Grundstein für eine effektive Kooperation dar und erhöhen die Leistung des kooperativen Problemlösens. Sowohl eine dem menschlichen mentalen Modell angepasste Betrachtungsweise der Welt als auch ein gemeinsamer Detaillierungsgrad in der Modelltiefe sind hierfür in einem technischen System zu berücksichtigende Faktoren.

Die bis zu dieser Stelle entwickelten Entwurfsempfehlungen in Form der Grundforderungen von Onken (Onken, 1993) und die von Orasanu (Orasanu, 1995) genannten Faktoren für eine effektive Gruppenarbeit ergeben für die Spezifikation einer *zentralen, objektorientierten Situationsrepräsentation* das Use-Case-Diagramm „Human-Centred-Approach“ (siehe Abbildung 14).

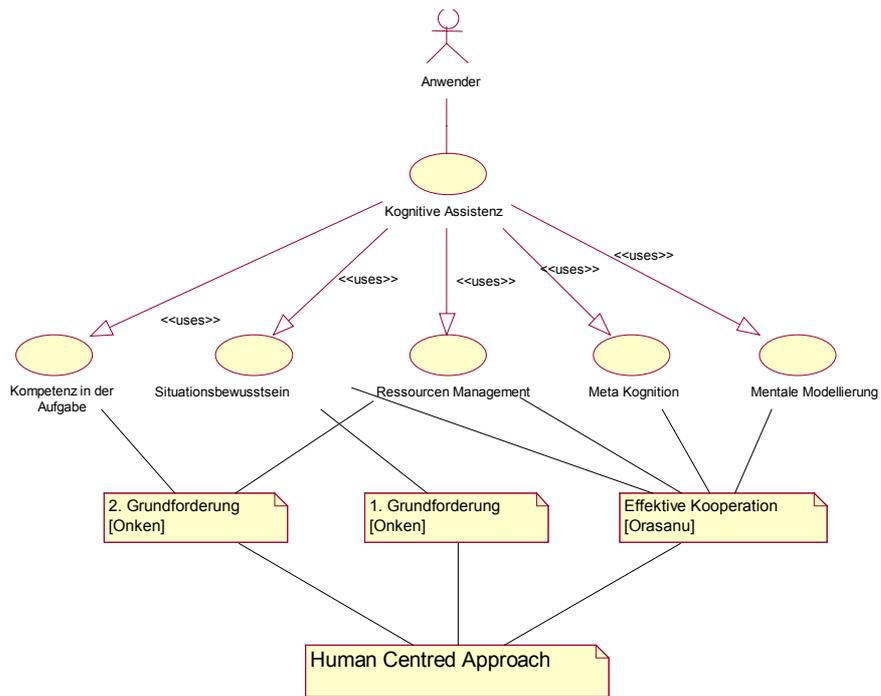
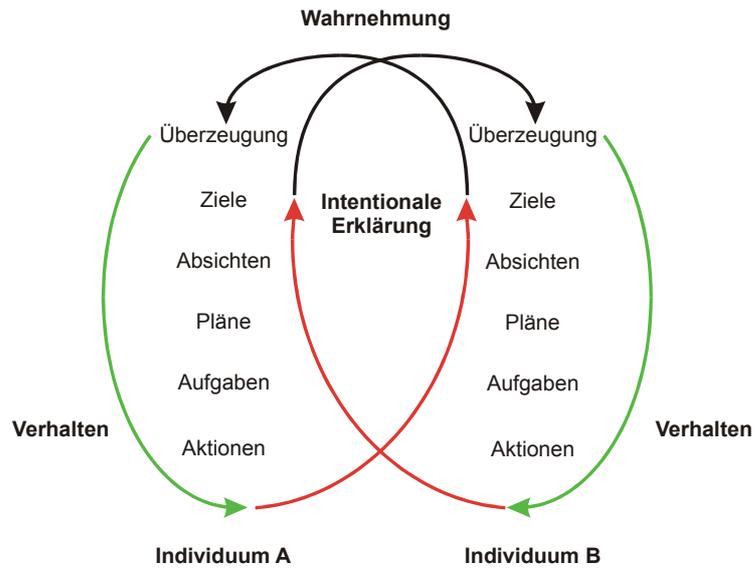


Abbildung 14: Use-Case-Diagramm „Human Centred Approach“

## 2.3 ERKLÄRBARKEIT DES VERHALTENS

Eine effektive Kooperation zwischen Mensch und Maschine mit Vermeidung oder Auflösung von Konflikten ist jedoch dann erst möglich, wenn beide Partner das Verhalten des anderen verstehen, ihre Ziele plausibel machen bzw. erklären können, möglichst ohne dass viel aktiv kommuniziert werden muss. Die Forderung nach einer Erklärungsfähigkeit des Verhaltens der Maschine gegenüber dem Menschen durch die *intentionale Erklärung* (Schank, 1986) ist nicht explizit im Human Centred Approach beschrieben, führt aber zum gemeinsamen Situationsverständnis zwischen Mensch und Maschine.

Intentionale Erklärung ist der Versuch, eine Theorie über die Schlussfolgerungen zu erstellen, welche das Handeln der Individuen bestimmen (siehe Abbildung 15).



**Abbildung 15: Situationsverständnis durch *intentionale* Erklärung**

Schank (Schank, 1986) nennt für die intentionale Erklärung die Handlungsziele und deren Priorisierung als entscheidend. Dies würde bedeuten, dass mit Hilfe von explizit vorliegenden Handlungszielen eine intentionale Erklärung des Systemverhaltens erfolgen könnte.

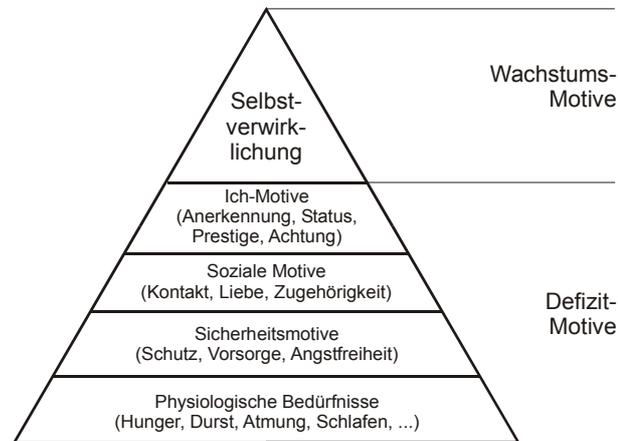
### 3 SYSTEMANALYSE

Grundlegend für kognitive Kooperation und allen hier genannten Anforderungen gemeinsam ist, dass die relevanten Informationen bezüglich der Situation zur Verfügung stehen und weiterverarbeitet werden können. Um kooperativ handeln zu können, bedarf es damit einer Repräsentation des Situationswissens und der Möglichkeit der prozessdienlichen Verarbeitung dieses Wissens. Dafür ist von Interesse, was diesbezüglich von der Informationsverarbeitung des Menschen bekannt ist. Für die Realisierung einer *zentralen, objektorientierten Situationsrepräsentation* scheint es sinnvoll, für die sequentielle Informationsverarbeitung als Medium solche mentalen Strukturen zu wählen, die für den Teilprozess der Wahrnehmung den Aufbau von *Erwartungen* an die Umwelt ermöglichen. Auf die menschliche Informationsverarbeitung aufbauend, wird hier anhand eines Modells die Funktionsweise des kognitiven Prozesses analysiert, der Entsprechendes beinhaltet. Die in der Kognitionspsychologie gewonnenen Erkenntnisse stellen hierzu das nötige Wissen bereit. Für die Repräsentation der Handlungsziele sollten die mentalen Strukturen zusätzlich ein effektives Management der Aufmerksamkeitsressourcen unterstützen. Um den damit gestellten Anforderungen zu entsprechen, sind die kognitiven Teilprozesse hinsichtlich ihrer Aufgabe in der Informationsverarbeitung und der dabei verwandten mentalen Strukturen zu analysieren.

#### 3.1 INFORMATIONSVERRARBEITUNG DES MENSCHEN

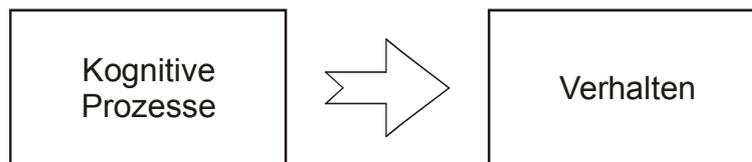
Die Kognitionspsychologie hat sich Ende der 50er Jahre als eine Teildisziplin der Allgemeinen Psychologie entwickelt und befasste sich vordringlich mit den kognitiven Funktionen und den kognitiven Prozessen des Menschen. Einzelwissenschaften – insbesondere der Psychologie, Physiologie, Informatik, Linguistik, Semiotik und Philosophie – haben die Kognitionstheorie entscheidend mitbestimmt und geformt. Sie wurde dabei wesentlich durch die Informationstheorie sowie kybernetische Modellvorstellungen beeinflusst. Das *Verhalten* des Menschen wird in der Organisationspsychologie als eine Funktion mit den Größen *Mensch* und *Situation* betrachtet, welche auf Seiten des Individuums in die Variablen *Motivation* und *Fähigkeit* zerfällt. Die Verhaltenswissenschaft benutzt den Begriff der Motivation zur Erklärung beobachteten Verhaltens, welches beim Menschen wiederum auf der Verfolgung von Zielen basiert.

Maslow (Maslow, 1943/1954) ging Mitte des 20. Jahrhunderts davon aus, dass es unabhängig vom gesammelten Erfahrungsschatz beim Menschen fünf Motivgruppen gibt (siehe Abbildung 16).



**Abbildung 16: Motivationspyramide (Maslow, 1943/1954)**

Die von Maslow favorisierte Erkenntnis hat sich über viele Jahre der psychologischen Forschung hin weiter bekräftigt. Jedoch ist die *motivgetriebene* Beeinflussung des Verhaltens in bezug auf Intensität, Richtung und Form mehr und mehr in den Hintergrund getreten, und der kognitive Anteil hat einen immer höheren Stellenwert erhalten. In den 70er Jahren führt diese Theorie das Motivationsmodell von (Bolles, 1974) ad extremum, welcher das Verhalten des Menschen allein durch die kognitiven Prozesse bestimmt sieht (siehe Abbildung 17).



**Abbildung 17: Kognitives Motivationsmodell (Bolles, 1974)**

Kognition bezeichnet u.a. alle diejenigen Vorgänge und Inhalte, die mit dem Wahrwerden und Erkennen der Umwelt zusammenhängen, und ist dabei Ausdruck für jeden Prozess, durch den ein Individuum Kenntnis von einem Umweltobjekt erhält oder sich seiner Umwelt in Form von Objektmengen und den Zusammenhängen der darin enthaltenen Einzelobjekte bewusst wird. Kognition basiert hierbei fundamental auf der intelligenten Verarbeitung dieser Umweltinformationen. Kognition ist zum anderen die geistige Tätigkeit zur Akquisition, Verarbeitung, Organisation und Nutzung von Wissen. Dies umfasst eine selektive

Aufmerksamkeit, Wahrnehmung, Unterscheidung, Interpretation, Klassifikation, Erinnerung und Wiedererkennung, Bewertung, Folgern und Ableiten. Das sich aus diesem Spektrum ergebende Leistungsvermögen des Menschen wird durch die Belastung (Verfahren, Aufgabe, Umwelt) und die daraus resultierende Beanspruchung bestimmt. Die Fragen, die es hier für die Entwicklung einer *Repräsentation des Situationswissens in objektorientierter Form* zu beantworten gilt, stellen sich durch das Ziel, diese Eigenschaften einer Maschine zu vermitteln. Welche Mechanismen sind für die Verarbeitung heranzuziehen, welche Repräsentationsformen, Speicher und Zugriffsmethoden auf die unterschiedlichen Wissenseinheiten sind zu wählen? Antwort hierauf geben die Erkenntnisse, die über die Informationsverarbeitung des Menschen bekannt sind.

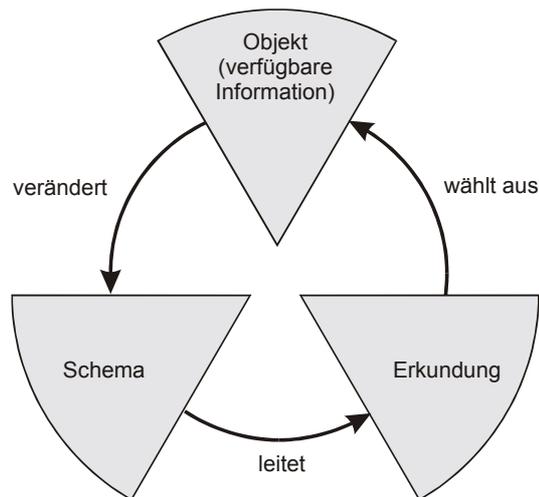
Verhalten basiert, wie von Bolles dargestellt, auf kognitiven Prozessen. Diese lassen sich zunächst vereinfacht als Teil des *Recognition Act Cycle* darstellen. Dieses generell anwendbare Schema für die Durchführung von Aufgaben enthält funktionale Elemente der *Informationsaufnahme* und der *Informationsverarbeitung*. Es umfasst die Bereiche

- Wahrnehmung,
- zentrale Verarbeitung und Entscheidungsfindung,
- Übertragung und
- Speicherung der Information.

Der *Recognition Act Cycle* wird zudem in die Bereiche

- Kommunikation (explizit) und
- Kognition (implizit)

unterteilt. Wegen der unzureichenden Beobachtbarkeit dieser Funktionen, soll als Ausgangspunkt für die Konzeption einer *Repräsentation des Situationswissens in objektorientierter Form* der von (Neisser, 1976) erarbeitete Wahrnehmungszyklus dienen (siehe Abbildung 18).



**Abbildung 18: Wahrnehmungszyklus (Neisser, 1976)**

Neisser versteht Wahrnehmung als kontinuierliche Tätigkeit. Jedoch sieht er sie nicht nur als *Informationsaufnahme*, sondern auch als *Informationsverarbeitung*. Dies bedeutet, dass zur Wahrnehmung nach seinem Verständnis wesentlich mehr kognitive Tätigkeit gehört als von den Befürwortern des *ökologischen Ansatzes* zugestanden wird. Neisser greift den aus dem Informationsverarbeitungsansatz stammenden Gedanken der Interaktion zwischen daten- und konzeptgesteuerter Wahrnehmung auf und postuliert, dass der erwachsene Mensch sehr erfahren im Wahrnehmen seines Alltages ist und folglich *Erwartungen* über das aufgebaut hat, was er wahrnehmen wird. Diese *Erwartungen* lenken die aktive Erkundung der Welt. Dabei unterscheidet er beim Wahrnehmungsprozess zwei Phasen (Neisser, 1976):

- **Präattentive Phase**

Sie umfasst Prozesse, die vor einer bewussten Verarbeitung von Wahrnehmungsinformationen ablaufen.

- **Attentive Phase**

Die Wahrnehmung wird hierbei durch die Reizgegebenheiten einerseits, durch die Erwartungen andererseits gesteuert.

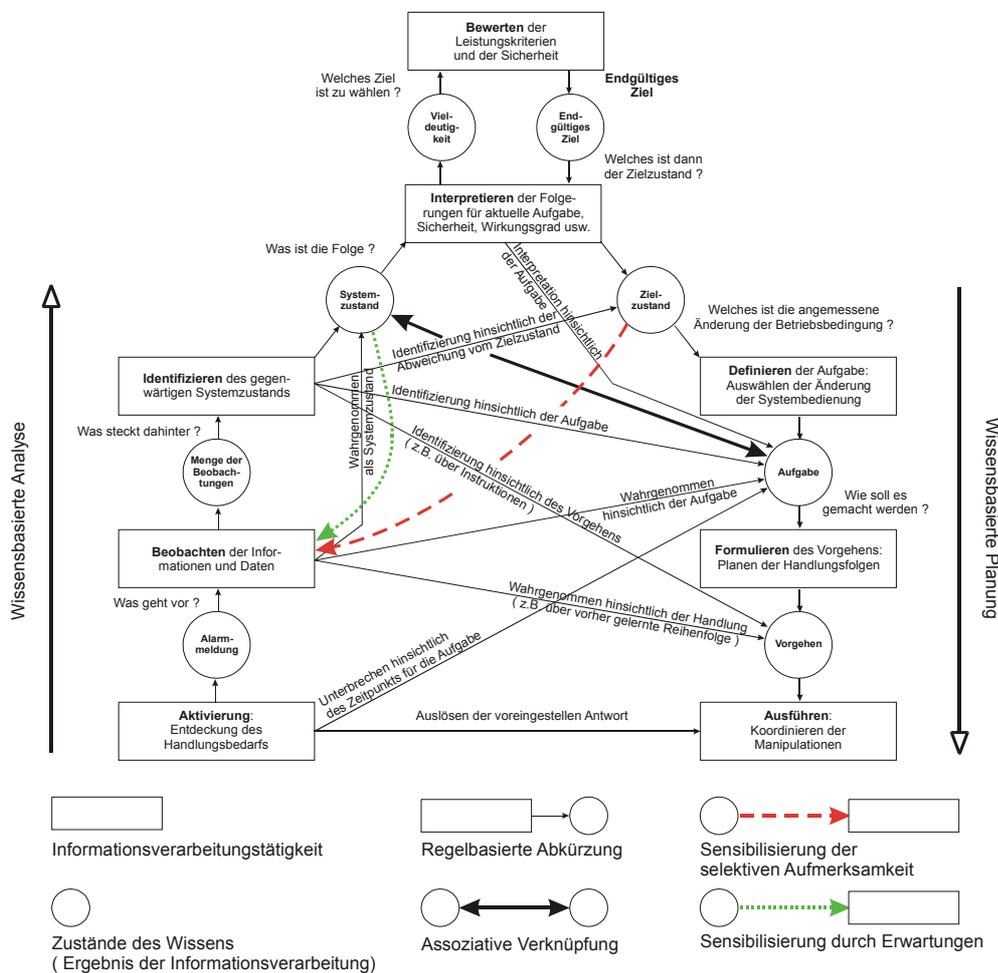
Neissers Betonung des konzeptgesteuerten Aufsuchens von Informationen lässt Wahrnehmung als kognitiv-konstruktive Handlung erscheinen.

Die Diskussion des Wahrnehmungszyklus vernachlässigt jedoch die für eine Umsetzung einer *Repräsentation des Situationswissens in objektorientierter Form* wichtigen inneren Zusammenhänge der menschlichen Informationsverarbeitung, welche damit nur implizit vorhanden scheinen und deren Funktionsweisen somit unbeantwortet bleiben. Welche

mentalen Strukturen liegen vor, auf die die Informationsverarbeitung des Menschen zurückgreift? Um diese Frage zu beantworten, bedarf es einer detaillierten Betrachtung des kognitiven Prozesses.

### 3.2 DER KOGNITIVE PROZESS

Die kognitiven Teilprozesse findet man schematisch dargestellt im Leiterdiagramm (siehe Abbildung 19) von Rasmussen (Rasmussen, 1983/1991) wieder. Das Leiterdiagramm bietet für die Konzeption einer *zentralen, objektorientierten Situationsrepräsentation* durch definierte und strukturierte Teilprozesse eine gute Grundlage. Es soll in dieser Arbeit weiter konkretisiert und hier zunächst einer genaueren Diskussion unterzogen werden.



**Abbildung 19: Leiterdiagramm (nach Rasmussen, 1983) und Zusammenhang von selektiver Aufmerksamkeit und dem Einfluss von Erwartungen**

Das Leiterdiagramm stellt über der

- fertigkeitbasierten,
- regelbasierten und
- wissensbasierten

Verhaltensebene den sequentiellen Informationsverarbeitungsprozess des Menschen dar. Die informationsverarbeitenden Teilprozesse ermitteln innerhalb der *wissensbasierten Analyse* aus dem zur Verfügung stehenden Kontextwissen das Situationswissen auf einer höheren Abstraktionsebene (Signale  $\Rightarrow$  Zeichen  $\Rightarrow$  Symbole). Die *wissensbasierte Analyse* stellt in anderen Worten ausgedrückt den Erkenntnisprozess bezüglich der momentanen Situation dar. Dieses Situationswissen wird während der *wissensbasierten Planung* unter Hinzuziehung der im Hintergrundwissen gespeicherten „*mentalen*“ Modelle in ein geplantes Handeln, d.h. ein Vorgehen überführt. „*Mentale*“ Modelle spielen somit eine Schlüsselrolle im Prozess der Planung und Entscheidungsfindung.

Über die *sequentielle Informationsverarbeitung* hinaus existieren jedoch kognitive Fähigkeiten, die durch die unidirektionale Betrachtung von Rasmussen im menschlichen Wahrnehmungsprozess nicht erklärt werden können.

So besitzt der Mensch elementare *Erwartungen* an das wahrzunehmende Element. Erwartungen stellen Annahmen oder Wahrscheinlichkeiten bezüglich des zukünftigen Zustandes/Verhaltens der beobachteten Situationsbereiche oder einzelner Situationselemente dar. Erwartungen werden immer ausgehend vom momentanen Systemzustand gehegt:

**Reize  $\Rightarrow$  Erwartete Wahrnehmung  $\Leftarrow$  Systemzustand**

Dies deutet darauf hin, dass der Mensch die Informationen, die er über seine Sinnesorgane aufnimmt, mit bereits vorhandenen mentalen Strukturen verbindet. Im Zusammenhang mit mentalen Strukturen spricht man von Elementen wie Schemata, Bildern, Symbolen, Konzepten oder Kategorien und Aussagen. Ein Schema ist hierbei eine abstrakte Repräsentation von Charakteristiken einer Situation und beinhaltet ein Gerüst von Merkmalen/Besonderheiten, welche die betrachtete Situation von anderen unterscheidet. Mit steigendem Abstraktionsgrad erfolgt die Annäherung an die Symbolebene. Symbole sind

willkürliche Bezeichnungen von Dingen. Im Gegensatz zu Schemata, die ein spezifisches Wissen aus *a-priori* und *a-posteriori* Wissen darstellen, sind Symbole willkürliche Repräsentationen von solchen Gegebenheiten. Symbole werden zur Entwicklung höherer kognitiver Elemente, so genannter Konzepte genutzt. Konzepte stellen wiederum eine Menge von Symbolen dar und werden zur Situationsklassifikation herangezogen.

Während der kognitiven Verarbeitung passt der Mensch seinen Informationsbedarf den Handlungszielen und dem entsprechenden Vorgehensplan an. Er besitzt ein effektives Management seiner Wahrnehmungsressourcen. Dabei werden durch den Informationsbedarf der verfolgten Ziele defizitäre Informationen lokalisiert und die Aufmerksamkeitsressourcen situationsgerecht hierauf verteilt:

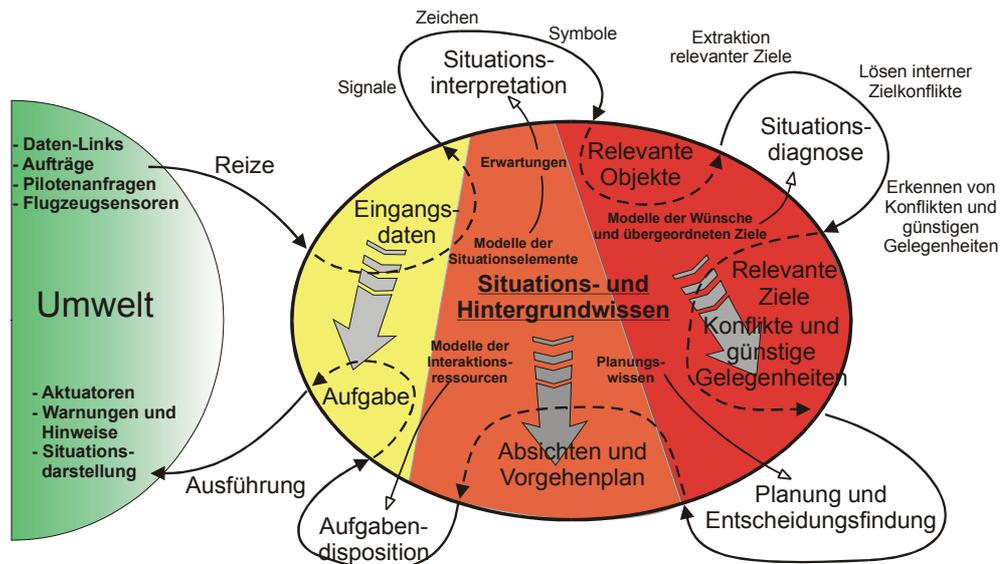
**Reize  $\Rightarrow$  *Selektive Aufmerksamkeit*  $\Leftarrow$  Zielzustand**

Die *sequentielle Informationsverarbeitung*, ihre Sensibilisierung durch die *erwartete Wahrnehmung* und das effektive Ressourcenmanagement durch die *selektive Aufmerksamkeit* sind im Leiterdiagramm (siehe Abbildung 19) dargestellt.

Aus den gewonnenen Erkenntnissen der menschlichen Informationsverarbeitung und dem Modell des menschlichen Verhaltens lässt sich ein Modell des kognitiven Prozesses mit den fünf Teilprozessen

- Wahrnehmung und Situationsinterpretation,
- Situationsdiagnose,
- Planung und Entscheidungsfindung,
- Aufgaben-Disposition und
- Handeln (Ausführung der Aktionen)

erstellen (siehe Abbildung 20).



**Abbildung 20: Modell des Kognitiven Prozesses (Walsdorf, 2000)**

### 3.2.1 SITUATIONSINTERPRETATION

Die Wahrnehmung der die Situation ausmachenden Inhalte bedeutet, dass das Individuum bereits bekannte, im Hintergrundwissen abgelegte „mentale“ Modelle mit den aus der Umwelt aufgenommenen Reizen in Beziehung bringt. Für den Teilprozess der Situationsinterpretation führt auch Dörner aus:

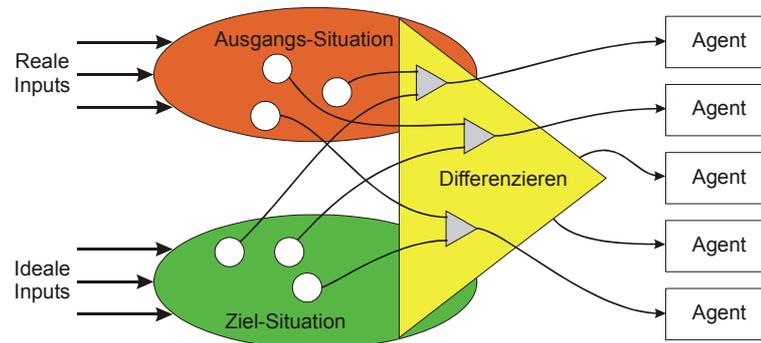
*„Mit der reinen Sammlung von Informationen ist es [...] nicht getan. Man muss [...] die Informationen möglichst so integrieren, dass sich eine Art Gesamtbild, ein Modell der Realität, mit der man umgeht, ergibt. Eine ungegliederte Anhäufung von Informationen über diese oder jene Merkmale der Situation vermehrt allenfalls noch die Unübersichtlichkeit und ist keine Entscheidungshilfe. Es muss alles irgendwie zusammenpassen; man braucht keine Informationshaufen, sondern ein Bild von der Sache, damit man Wichtiges von Unwichtigem trennen kann und weiß, was zusammengehört und was nicht. Man braucht Strukturwissen.“ (Dörner, 1979)*

Dabei durchlaufen die aufgenommenen Umweltreize einen Abstraktionsprozess. „Mentale“ Modelle sind im Hintergrundwissen abgelegt und werden vom Individuum selbständig erlernt. Maschinen können diese auch durch den Prozess des Wissenserwerbs von externer Stelle vermittelt werden. Die Situationsinterpretation hat die Repräsentation bekannter „mentaler“ Modelle in einer der Situation entsprechenden Instanziierung in einem Speichermedium zum Ergebnis. Beim Menschen ist das Speichermedium sein Gedächtnis. Dieser Inhalt bildet das

Wissen bezüglich der Situation und kann sowohl kurz- als auch langfristig im Gedächtnis bereitgehalten werden. Das Situations- oder Kontextwissen eines Individuums existiert parallel zum Hintergrundwissen, welches exklusiv im Langzeitgedächtnis abgelegt ist.

### 3.2.2 SITUATIONS DIAGNOSE

Die Diagnose der Situation hat zur Aufgabe, die gewünschte *Ziel-Situation* festzustellen und die Differenz zur gegenwärtigen *Ausgangs-Situation* zu ermitteln. Dieser Zusammenhang ist von Minsky durch die *Differenz-Maschine* dargestellt worden (siehe Abbildung 21).



**Abbildung 21: Differenz-Maschine (Minsky, 1985)**

Die *Situationsdiagnose* ermittelt auf der Basis der erfassten Situation und der ihr bekannten potentiellen Ziele diejenige Menge von Handlungszielen, welche in diesem Zustand relevant ist. Eisenführ und Weber (Eisenführ, 1993) weisen hierbei auf den hohen mentalen Aufwand hin, der für das Erkennen der eigenen Ziele vonnöten ist. Die relevanten Ziele spielen für den informationsverarbeitenden Prozess und damit für die gesamte Kognitionsleistung eine Schlüsselrolle. Durch die Ziele wird das perzeptive Verhalten in Form der Aufmerksamkeitsselektion gesteuert. Die Repräsentation eines Ziels ist durch den *Zustand einer idealen Situation* gegeben. Hierdurch werden diejenigen Objekte bestimmt, die für das Situationsverständnis von Relevanz sind. Einfach ausgedrückt könnte man auch sagen, dass sich die Wahrnehmung nur auf diejenigen Situationselemente zu konzentrieren bräuchte, die für die Differenzbestimmung zur Zielsituation erforderlich sind. Thematisch werden die *Situationsinterpretation* und die *Situationsdiagnose* der *Situationsanalyse* zugeschrieben.

Da es aufgrund der – typischerweise – eingeschränkten Ressourcen nicht möglich ist, alle möglichen Handlungsalternativen zwischen zwei Situationszuständen als potentielle Kandidaten für eine Problemlösung heranzuziehen, ist eine Problemraumreduktion nötig. Die Reduktion des Problemraumes und damit die Reduktion der Komplexität des Lösungsraumes

erfolgt durch die Aufspaltung der Ziele in Teilziele. Eisenführ und Weber führen hierzu die Unterscheidung in Fundamental- und Instrumentalziele an, welchen sie folgende Eigenschaften zuschreiben (Eisenführ, 1993):

- Fundamentalziele sind von Instrumentalzielen zu unterscheiden. Für die Bewertung von Alternativen sollten möglichst nur solche Ziele herangezogen werden, die innerhalb des jeweiligen Entscheidungskontextes fundamental sind.
- Das System von (Fundamental-)Zielen, die in einer bestimmten Entscheidungssituation von Bedeutung sind, sollte bestimmte Anforderungen erfüllen: Vollständigkeit, Redundanzfreiheit, Messbarkeit, Unabhängigkeit und Einfachheit.
- Für jedes (Fundamental-)Ziel, das zur Bewertung der Alternativen herangezogen werden soll, ist eine Zielvariable zu bestimmen, die die Zielerreichung möglichst treffend und eindeutig wiedergibt.

Mit der Aktivierung von im Gedächtnis gespeicherten Zielvorstellungen ist die Ermittlung von möglichen Handlungsalternativen verbunden. Die *Situationsdiagnose* ist derjenige Prozess, der dem Problemlösen vorangestellt ist. Die Bereitstellung von Handlungsalternativen erfolgt durch Mengen von bekannten Operatoren. Operatoren stellen Lösungen von in sich geschlossenen Teilproblemen dar. Diese werden der Problemlösungseinheit zur Auswahl und anschließenden Ausführung zugeführt. Probleme bzw. Konflikte mit den relevanten Zielen werden zunächst lokalisiert, interpretiert und in geeigneter Form repräsentiert. Diese Hürden zu überwinden ist dann Aufgabe der Problemlösungseinheit, d.h. des Prozesses der *Planung und Entscheidungsfindung*. Die Wahl einer geeigneten Instanz, welche auf die Lösung eines Konfliktfalls spezialisiert ist (z.B. Experte, Agent, gespeicherte Problemlösung), liegt ebenfalls im Zuständigkeitsbereich der *Situationsdiagnose*. Diese hat dann die Differenz zwischen den beiden Zuständen, der realen und der idealen Situation, zu verringern.

### 3.2.3 PLANUNG UND ENTSCHEIDUNGSFINDUNG

Der Teilprozess der *Planung und Entscheidungsfindung* dient der Problemlösung bezüglich der erfassten Situation und den darin erkannten Konflikten, basierend auf einer relevanten Zielmenge. Die Problemlösungstätigkeit stellt sich im allgemeinen durch das Ermitteln fehlender Informationen dar. Nach (Klix, 1971) können Problemstellungen in drei Klassen

vorliegen. Der erste Problemtyp beinhaltet eine bekannte Ausgangs-Situation und bekannte Operatoren. Jedoch ist hier die Ziel-Situation unbekannt. Im zweiten Fall sind die Operatoren und die Ziel-Situation bekannt, jedoch fehlen Informationen über die Ausgangs-Situation. Die Planungstätigkeit des Menschen ist ausschließlich Gegenstand der dritten Problemklasse, bei der die Ausgangs- und die Ziel-Situation bekannt sind, jedoch die vorzunehmenden Operationen zum Erreichen der Ziel-Situation fehlen. Probleme sind durch eine objektive Situationskomponente und eine subjektive Persönlichkeitskomponente charakterisiert. Die Persönlichkeitskomponente umfasst das Wissen und die Erfahrung und ist durch Lernen veränderbar. *Planung und Entscheidungsfindung* stellen Anforderungen an den Menschen in Bezug auf sein kognitives Leistungsvermögen. Problemstellungen sind durch ihre Komplexität und Vernetztheit der Situationskomponenten charakterisiert. *Planung* ist die Grundlage für den *Entscheidungsfindungsprozess* und stellt die mentale Vorbereitung zukünftigen Handelns dar. Das Handeln wird dann durch die Entscheidung eingeleitet. Entscheiden heißt, aus einer Menge von geeigneten Handlungsalternativen so zu wählen, dass man die gesetzten Ziele am günstigsten erreicht.

Um eine verfolgte Ziel-Situation zu erreichen, wird ein *Vorgehen* geplant, welches zielgerichtet durch Anwenden von geeigneten Operatoren diese Situation erzeugt. Der Planungsprozess besteht somit aus der Suche nach Operatoren in einem Alternativen- oder Lösungsraum. Die Distanz zwischen der Ausgangs-Situation und der Ziel-Situation und die Komplexität des Lösungsraumes entsprechen der Schwierigkeit des zu lösenden Problems. Der Planungsvorgang kann hierbei mit Situationen konfrontiert werden, welche eine vollständige Lösung, d.h. das Erfüllen aller Ziele, ausschließt. Zur Lösung solcher Probleme bedarf es einer *Strategie*. (Wilensky, 1983) spricht auch von *Meta-Themen*, welche einen Planer bei der Erzeugung von Lösungen leiten. Er führt die folgenden Strategien an, die bei der Planung einer Problemlösung Anwendung finden (Wilensky, 1983):

- „Vergeude keine Ressourcen“,
- „Befriedige so viele Ziele wie möglich“,
- „Maximiere den Erfülltheitsgrad der Ziele“ und/oder
- „Vermeide unmögliche Ziele“.

Der Problemlösungsprozess bedient sich eines ähnlich strukturierten Zustandsraumes wie der Prozess, der die Situation zu verstehen versucht. Beide arbeiten mit semantisch gleichen

Objekten. Der gesamte Entscheidungsprozess basiert dabei auf einer Reihe von Informationen über (Mag, 1977)

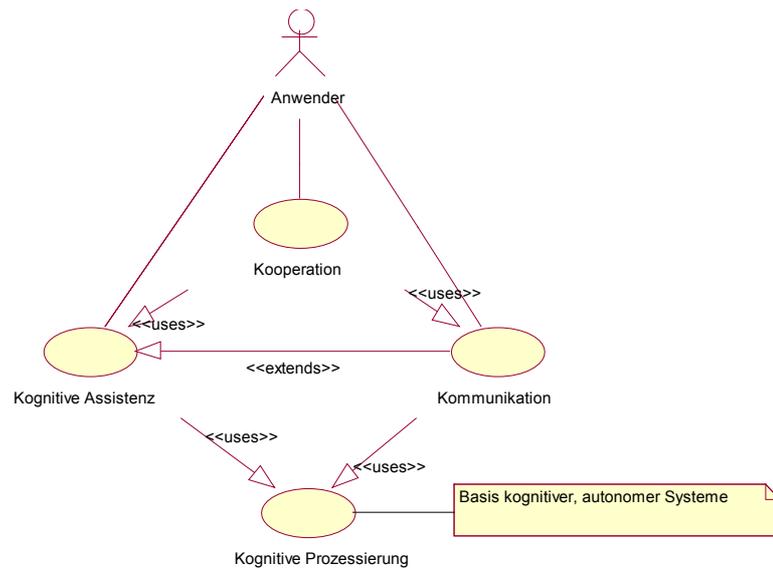
- Umweltlagen,
- Eintrittswahrscheinlichkeiten der Umweltlagen,
- Ziele,
- Handlungsmöglichkeiten,
- Ergebnisse bzw. Zielbeiträge und
- Entscheidungskriterien.

Die Umsetzung der Problemlösung erfolgt in den Bereichen der *Aufgabendisposition und Ausführung*.

#### 3.2.4 AUFGABENDISPOSITION UND AUSFÜHRUNG

Der Teilprozess der *Aufgabendisposition* und die sich hieran anschließende Umsetzung sind eng mit den vorausgehenden Prozessschritten verbunden, da diese hierfür bereits die Basis gelegt haben. Aufgaben sind durch eine Menge von makroskopischen Handlungen charakterisiert. Jede Handlung umfasst eine Sequenz von Handlungsschritten, denen jeweils ein Potential einer Ausführungsressource anhängig ist. Ziel der *Aufgabendisposition* ist es, diese Ressourcen so zu koordinieren, dass eine optimale *Ausführung* der Einzel-Aktionen erfolgen kann. Der Teilprozess der *Ausführung* stellt den letzten Schritt im kognitiven Prozess dar und dient der Umsetzung der anstehenden Handlungsschritte mittels der Interaktionsinstrumentarien, den Operatoren.

Mit den gewonnenen Kenntnissen des kognitiven Prozesses kann der Entwurf der *zentralen, objektorientierten Situationsrepräsentation* weitergeführt werden. Für das Use-Case-Diagramm ergibt sich der Zusammenhang in Form der kognitiven Mensch-Maschine-Kooperation (siehe Abbildung 22).



**Abbildung 22: Use-Case-Diagramm „Kognitive Mensch-Maschine-Kooperation“**

Der kognitive Prozess stellt hierbei die Basis des intelligenten, autonomen Systems dar und ist in seinen grundlegenden Funktionen spezifiziert.

Durch das Studium der kognitiven Teilprozesse sind deren Aufgaben und Funktionsweisen dargestellt worden. Nun gilt es, die gewonnenen Erkenntnisse in eine Repräsentationsform umzusetzen, die den in Kapitel 2 gestellten Anforderungen genügt.

## 4 ENTWURF

Entsprechend den Anforderungen aus Kapitel 2 fließen die in Kapitel 3 gesammelten Erkenntnisse nun in den Entwurf einer *zentralen, objektorientierten Situationsrepräsentation*. Den Ausgangspunkt hierfür stellt die im kognitiven Prozess zu verankernde Repräsentation und Verarbeitung des Situations- und Hintergrundwissens dar, auf der die Funktionen der kognitiven Teilprozesse, beginnend mit der *Situationsanalyse*, beruhen. Diejenigen Eigenschaften, die das Wissen über eine Situation charakterisieren, werden dargestellt und hinsichtlich einer Umsetzung in einer *zentralen, objektorientierten Situationsrepräsentation* erörtert. Aus den erarbeiteten Zusammenhängen gilt es eine Repräsentationsform zu entwickeln, die den „mentalen“ Zustand des kognitiven Systems bezüglich der verfolgten Handlungsziele darstellt, was im folgenden diskutiert wird.

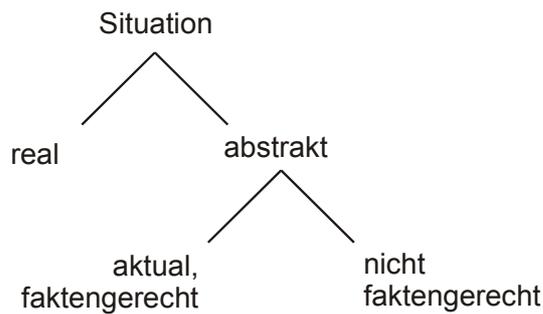
### 4.1 SITUATIONSWISSEN

Das intelligente Verhalten und die damit verbundene kognitive Verarbeitung von Informationen ist neben dem im Hintergrund abgelegten Langzeitwissen in Form von „mentalen“ *Modellen* insbesondere durch das so genannte *Situationswissen* motiviert, in dem u.a. die „mentalen“ *Modelle* entsprechend der wahrgenommenen Situation instanziiert werden. Situation ist ein in der Umgangssprache sehr vielseitig verwendeter Begriff. Onken hat den Situationsbegriff aus Sicht der Verhaltensforschung wie folgt beschrieben:

*„Die Situation besteht aus einer Momentaufnahme von allen in ihr erkennbaren Besonderheiten, die (wie z.B. die Aufgabenschwierigkeit) ihre eigentliche Natur ausmachen und die Fähigkeiten und Anstrengungen des Handelns erfordern, um die bestehende Situation in ein erwünschtes Ergebnis zu überführen. Dazu gehört auch die bis dahin bestehende Vorstellung über den weiteren Fortgang, und es zählen dazu die Situationselemente aus der Vergangenheit, die für die weitere Situationsentwicklung von Bedeutung sind.“ (Onken, 1999)*

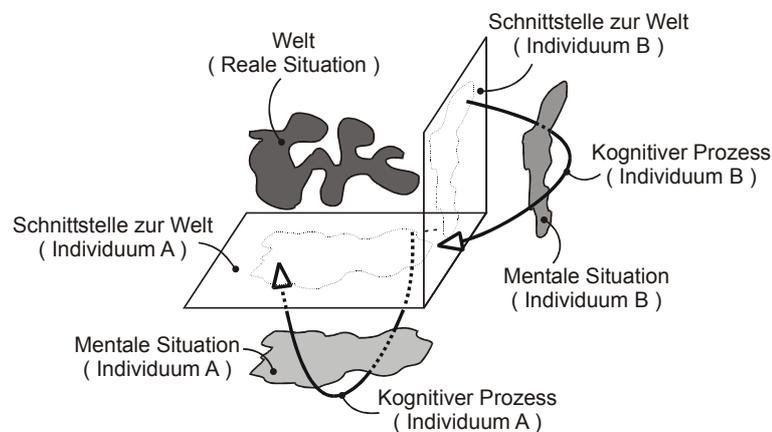
Der Aufbau und die Eigenschaften des Wissens über die Situation ist für die Entwicklung einer *zentralen, objektorientierten Situationsrepräsentation* von hoher Wichtigkeit. Barwise und Perry (Barwise, 1987) haben in ihren Arbeiten zur Situationssemantik Aufbau und Eigenschaften der Situation sehr ausführlich analysiert und unter metaphysischen wie auch

erkenntnistheoretischen Gesichtspunkten betrachtet. Dabei haben sie zunächst Klassen von Situationen erarbeitet. Sie unterscheiden *statische Situation* (Zustand) von *dynamischer Situation* (Ereignis) und *reale Situation* (Ausschnitte der Welt) von *abstrakter Situation* (abstrakte Modellvorstellungen, die benutzt werden, um reale Situationen zu repräsentieren). Abstrakte Situationen werden von Barwise und Perry weiter in *aktuale, faktengerechte* (sie entsprechen genau den realen Situationen) und *nicht-faktengerechte* Situationen unterschieden (siehe Abbildung 23).



**Abbildung 23: Klassifikation des Situationsbegriffs (Barwise, 1987)**

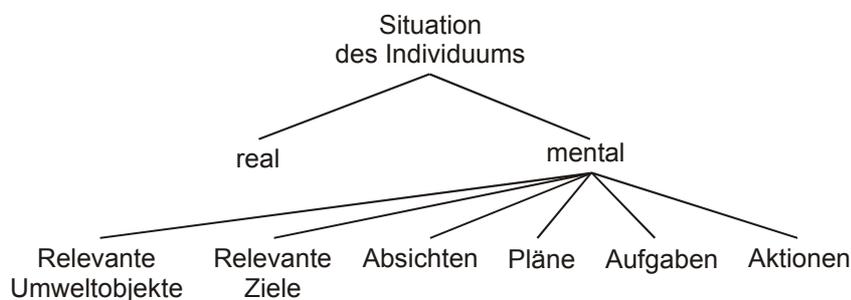
Auf den von Barwise und Perry erarbeiteten Klassen basierend, soll das Bild, welches sich das Individuum von der *realen Situation* macht, als *mentale Situation* bezeichnet werden. Die *mentale Situation* sei dabei ausschließlich Ergebnis des kognitiven Prozesses (siehe Abbildung 24) und wird dadurch vom Begriff der *abstrakten Situation* unterschieden, wobei letztere in *aktual-faktengerechte* und *nicht-faktengerechte* Situation zu unterscheiden ist. Die *mentale Situation* wird dem Individuum aufgrund seiner Subjektivität immer als *aktual, faktengerecht* erscheinen.



**Abbildung 24: Mentale Repräsentation der Welt**

Die Güte der Repräsentation der *mentalen Situation* in Bezug zur *realen Situation* ist unter anderem von den *mentalen Strukturen* und den in ihnen organisierten *mentalen Modellen* eines Individuums abhängig. *Mentale Modelle* sind in der Regel individuell vereinfachte und auf wesentliche Komponenten reduzierte Abbilder der Realität, die das Situationsverständnis prägen. *Mentale Modelle* können für bestimmte Aspekte eines Gegenstandsbereichs hinreichend sein, für andere jedoch nicht.

Für die Entwicklung einer *zentralen, objektorientierten Situationsrepräsentation* sind Komponenten zu identifizieren, die innerhalb des kognitiven Prozesses zum Aufbau der mentalen Situation benötigt werden. Diese *mentalen Situationskomponenten* werden aus den Modellvorstellungen des menschlichen Verhaltens abgeleitet (siehe Abbildung 25).



**Abbildung 25: Klassifikation des Situationsbegriffs im Kontext der Kognition**

Mentale Situationskomponenten werden im kognitiven Prozess von den kognitiven Teilprozessen aktiviert und repräsentieren darin die *mentale Situation* des Individuums. Die Eigenschaften der mentalen Situationskomponenten werden ausgehend von der *realen Situation* erarbeitet.

Die *reale Situation* entspricht dem objektiven Wissen bezüglich der Welt mitsamt der in ihr enthaltenen Individuen, deren mentalen Einstellungen sowie deren Zustände in Vergangenheit, Gegenwart und Zukunft. Diese Betrachtung grenzt subjektives von objektivem Situationswissen ab und stellt das Vollständigkeitsaxiom bezüglich der Situation dar. Aufgrund der theoretischen Betrachtung ist dies nicht im Konflikt mit dem relativen, da subjektiven, Bewusstseinsbild eines Individuums.

Die *Lage* ist objektiv und eine Teilmenge der realen Situation. Sie beinhaltet nur diejenigen Situationskomponenten, welche nicht dem kognitiven Gesamtprozess entwachsen sind, sondern diesem alleinig als wahrzunehmendes Umweltobjekt gegenüberstehen.

Die Interpretation der *realen Situation* führt zu den *Überzeugungen* bezüglich der Umwelt, wobei eine selbstreflektierende Betrachtung des Individuums nicht auszuschließen ist. *Überzeugungen* werden in die Kategorien *informationstragend* und *abstrakt* unterteilt, wobei erstere Wissen konstituieren und letztere nicht. Die Gesamtheit der *Überzeugungen* stellt das Situationsbild dar. Die Vorstellung über die Situation, welches aus den *Überzeugungen* entsteht, ist nicht alleine von ebenjenem Zustand abhängig, sondern ebenso vom Kontext der Situationen, in die sie eingebettet sind. Man spricht hier vom Referenzfeld der *Überzeugung*. Das Situationsbild, welches sich aus den *Überzeugungen* heraus ableitet, beinhaltet die *Erwartungen* und damit eine Beurteilung der zukünftigen Situation.

Aus den *Überzeugungen* und der *Erwartung* ergeben sich *Wünsche* bezüglich dem Eintreffen oder dem Vermeiden derselben. *Wünsche* implizieren an sich noch keinerlei Aussage über deren mögliche Umsetzbarkeit. *Wünsche*, deren Umsetzung prinzipiell möglich ist, bilden den potentiellen Handlungsspielraum und werden *relevante Ziele* genannt. Aus den relevanten Zielen leitet sich der Informationsbedarf ab. Sie sind dadurch für das Management der Aufmerksamkeitsressourcen entscheidend.

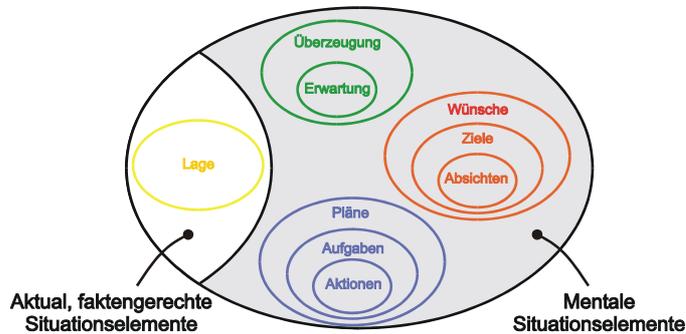
Diejenige Untermenge der *Ziele*, an deren Umsetzung momentan gearbeitet wird, nennt man *Absichten*. *Ziele* können oftmals nicht gleichzeitig verfolgt werden, so dass eine Priorisierung nach subjektiver Wichtigkeit durchgeführt werden muss. *Absichten* sind somit in ihren Eigenschaften mit den *Zielen* identisch.

Das *Vorgehen* stellt eine Menge von Handlungsfolgen und deren Abhängigkeiten untereinander dar. Es ist durch einen *Plan* oder mehrere, den Einzelabsichten zugeordneten Teilpläne repräsentiert, welche auf Grundlage der *Überzeugungen* und der *Absichten* ermittelt wurden und zu deren Erfüllung führen. Pläne sind direkt mit der Situation der *Absichten* verbunden und müssen ihnen daher permanent angepasst bzw. nachgeführt werden.

*Pläne* repräsentieren eine Aufgabenschar. *Aufgaben* formulieren Operatoren auf der momentanen *Lage*, um diese in einen erwünschten, andersartigen Zustand einer vor bestimmten Zielsituation zu überführen. Die Auswahl der geeigneten Operatoren ist Teil des Planungsprozesses.

*Aufgaben* können aus einer Vielzahl von Einzelaktionen zusammengesetzt sein, welche sequentiell oder parallel ausgeführt werden können. Die Entscheidung, wann welche *Aktion* ausgeführt werden soll, richtet sich nach den für die Ausführung benötigten *Ressourcen*.

Den diskutierten Situationselementen lassen sich Mengen von aktuellen, faktengerechten und mentalen Situationselementen zuordnen (siehe Abbildung 26).

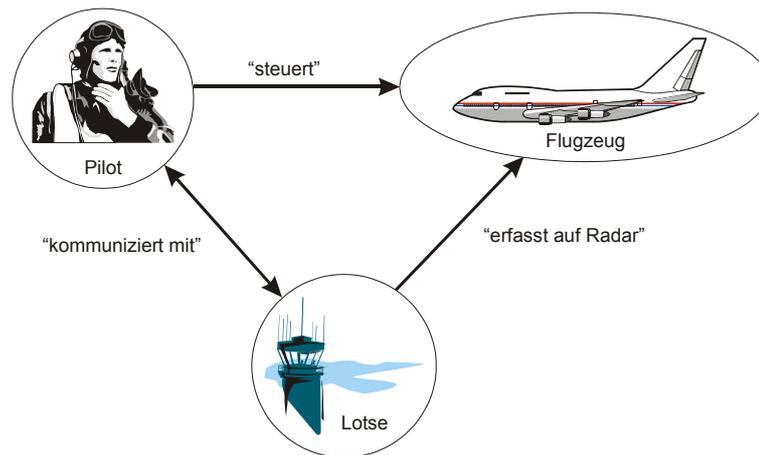


**Abbildung 26: Situationselemente des kognitiven Prozesses**

## 4.2 OBJEKTORIENTIERTE SITUATIONSREPRÄSENTATION

Für den Entwurf einer *zentralen, objektorientierten Situationsrepräsentation* ist eine theoretische Betrachtung der mit dem Situationsbegriff in Zusammenhang stehenden Grundlagen nötig. Die formale Beschreibung der Situation soll zunächst in Anlehnung an die Diskussion des Situationsbegriffes nach Barwise und Perry (Barwise, 1987) erfolgen. Wie man nachfolgend sehen wird, bedarf es jedoch für die vollständige Modellierung des kognitiven Prozesses weiterführender Grundelemente und Konzepte.

Für die Darstellung des *Situationswissens in objektorientierter Form* in einer technischen Anwendung ist eine Situationsrepräsentation zu entwickeln, welche sowohl das relationale als auch das implizite Situationswissen mit den Vorzügen der objektorientierten Umsetzung verbindet. Hierfür wurde in dieser Arbeit eine symbolische Repräsentationsform konzipiert, die in den Knoten  $N$  eines Graphen  $G$  die instanziierten Modelle der Situationselemente aufnimmt. Sie bietet den Vorzug, sowohl den Anforderungen der maschinellen Interpretation und Weiterverarbeitbarkeit gerecht zu werden, als auch die für die Kommunikation mit dem Menschen benötigte Aussagefähigkeit zu besitzen. Im Gegensatz zu herkömmlichen datenorientierten Repräsentationsformen besitzt eine objektorientierte Repräsentation zahlreiche Vorteile. Ein Beispiel für diese Situationsrepräsentationsform ist in Abbildung 27 dargestellt.



**Abbildung 27: Symbolische Situationsrepräsentation eines strukturierten Situationsobjektes**

Im gezeigten Beispiel beinhaltet die Situation die Situationselemente „Pilot“, „Flugzeug“ und „Lotse“. Jedes dieser Situationselemente stellt für sich eine Instanziierung eines „*mentalen*“ Modells innerhalb der Situationsrepräsentation dar. Die Menge der zu einem bestimmten Zeitpunkt repräsentierten Situationselemente sind in Beziehung zueinander gesetzt, was durch die Relationen zwischen den Situationselementen erfolgt. In obigem Beispiel sind das die Relationen „steuert“, „erfasst auf Radar“ und „kommuniziert mit“.

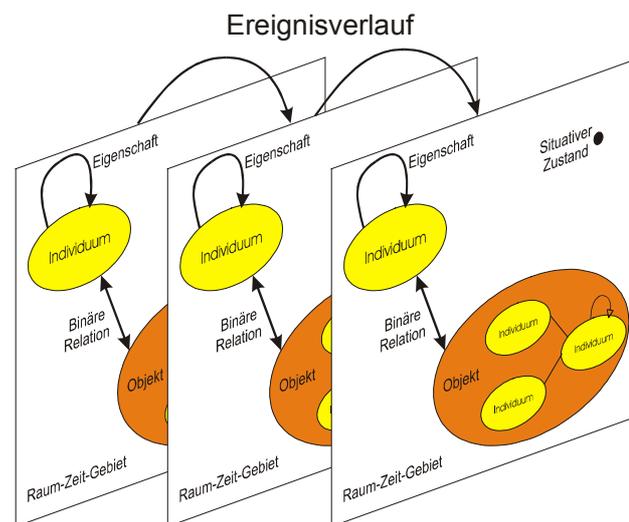
Identische Situationsobjekte können per Definition in verschiedenen Kontexten (d.h. Graphen) verschiedene Rollen spielen. Dabei sind unterschiedliche Elementarmerkmale relevant. Die Umkehrung ist ebenfalls erlaubt, so dass verschiedene Situationselemente in einem Graphen dieselbe Rolle spielen können.

#### 4.2.1 RELATIONALE STRUKTUREN DES SITUATIONSWISSENS

Barwise und Perry haben den Begriff der Situation ausführlich diskutiert und anhand einer Menge von bestimmten Grundelementen beschrieben. Damit können sowohl unterschiedliche Situationsarten (z.B. real oder abstrakt), als auch Zustände (statische Situationen) oder Ereignisverläufe (dynamische Situationen) behandelt werden. Sie stützen sich dabei auf die Definition folgender Grundelemente:

- Individuen,
- Relationen und
- Raum-Zeit-Gebiete.

Individuen sind in ihrer Theorie *einfach*. Relationen, also die Beziehunghaftigkeit zwischen Individuen, können *mehr-stellig* sein. *Null-stellige* Relationen heißen *Zustände* und können z.B. ein Raum-Zeit-Gebiet deklarieren. *Ein-stellige* Relationen werden zur Definition von *Eigenschaften* eines Individuums benutzt. *Zwei-stellige* bzw. *binäre* Relationen stellen die Beziehungen zwischen zwei Individuen dar und bilden damit eine Klasse einer Situation oder eine in ihr enthaltene Teilmenge. Raum-Zeit-Gebiete können eine endliche Menge von Individuen und Relationen zwischen den Individuen enthalten und sind selbst durch Relationen miteinander verbunden. Raum-Zeit-Gebiete können sich räumlich oder zeitlich überlappen, oder ein Gebiet kann einem anderen vorangestellt sein. Barwise und Perry verwenden zudem den Begriff des Objekts als Beschreibung eines einfachen oder komplexen Konstrukts aus Individuen. Die Beschreibung der Situation durch die genannten Grundelemente ist in Abbildung 28 dargestellt.



**Abbildung 28: Grundelemente der Situation nach Barwise und Perry (Barwise, 1987)**

Zunächst soll davon ausgegangen werden, dass es sich beim vorliegenden Problem um ein geschlossenes handelt, so dass damit die gesamte *Situationsrepräsentation in objektorientierter Form* durch ein Raum-Zeit-Gebiet repräsentiert und durch eine Menge  $S$  von Teilsituationen  $s$  vollständig dargestellt werden kann. Damit existiert ein größtes Raum-Zeit-Gebiet, welches alle anderen Raum-Zeit-Gebiete enthält. Dieses größte Raum-Zeit-Gebiet soll Gesamtsituation genannt werden.

Inhalt von  $s$  sei eine Menge von Individuen, hier Situationselemente  $E$  genannt, welche Modelle von realen oder abstrakten Elementen der Welt darstellen, und eine Menge von Relationen  $R$ . Hieraus ergibt sich:

$$S = \{E, R\}$$

Bereits aus diesem sehr einfachen Situationsmodell lässt sich ein Ähnlichkeitsmaß und damit ein Verständnis der Situation ableiten. Situationsverständnis soll hier das Wissen über den Inhalt, den Aufbau und die Semantik einer relational repräsentierten Situation sein. Eine Klassifikation von unterschiedlichen Situationen muss also Inhalt, Aufbau und Semantik der kognitiven Grundelemente berücksichtigen. Für das Maß der Ähnlichkeit ist damit eine Metrik festzulegen, die in dieser Arbeit auf dem Zusammenhang

	<b>Inhalt</b>	<b>(Situationselemente)</b>
+	<b>Aufbau</b>	<b>(Situationselemente + Relationen)</b>
+	<b>Semantik</b>	<b>(Situationselemente + (Relationen + Relationsattribute))</b>
=	<b>Situationsverständnis</b>	

basiert. Für dieses qualitativ definierte Ähnlichkeitsmaß ist eine quantitative Größe zu bestimmen, mit der der Grad der Ähnlichkeit zweier Situationen festgestellt werden kann.

Eine Ähnlichkeit bezüglich des Inhalts einer Situation sei gegeben, wenn die Situationselemente einer Situation A sich mit den Situationselementen einer Situation B decken. Instanzen der Situationselemente existieren damit jeweils in der Menge A und auch in der Menge B. Dies soll für die Situationselemente durch den Ausdruck

$$\textit{Ähnlichkeit}_{\textit{Inhalt}} = E_A * E_B$$

und – der Vollständigkeit wegen – für die Relationen durch den Ausdruck

$$\textit{Ähnlichkeit}_{\textit{Inhalt}} = R_A * R_B$$

formuliert sein. Die Ähnlichkeit bezüglich des Aufbaus einer Situation sei durch die Relationen untereinander gegeben. Der strukturelle Aufbau einer Situation sei durch den Ausdruck

$$\textit{Ähnlichkeit}_{\textit{Aufbau}} = \{E_A, R_A\} \oplus \{E_B, R_B\}$$

formuliert, wobei der Aufbau einer Situation zunächst ohne Einbeziehung der Relationsattribute charakterisiert ist. Die vollkommene Ähnlichkeit zwischen zwei Situationen liegt dann erst vor, wenn identische Mengen von Situationselementen und Relationen vorliegen. Zusätzlich müssen diese in gleicher Weise miteinander in Beziehung stehen und die Relationen in ihren Attributen übereinstimmen. Die Semantik einer Situation sei dann durch den Ausdruck

$$\textit{Ähnlichkeit}_{\textit{Semantik}} = \{E_A, R_A\} \otimes \{E_B, R_B\}$$

formuliert. Aus diesen Zusammenhängen lässt sich ein Ähnlichkeitsvektor erstellen, der die drei Stufen des Situationsverständnisses abbildet:

$$\begin{pmatrix} \textit{Inhalt} \\ \textit{Aufbau} \\ \textit{Semantik} \end{pmatrix} = \begin{pmatrix} (E_A * E_B) \\ \{E_A, R_A\} \oplus \{E_B, R_B\} \\ \{E_A, R_A\} \otimes \{E_B, R_B\} \end{pmatrix} = \textit{Ähnlichkeit}(S_A, S_B)$$

Die Verknüpfung von mehreren Ähnlichkeitsvektoren erfolgt über einen Minimumoperator, so dass das Ergebnis der Verknüpfung das Minimum der Ähnlichkeitsvektoren liefert.

#### 4.2.1.1 Objektorientierung

Die Darstellung der Situation basiert auf den Situationselementen und deren Relationen untereinander. Hierfür lassen sich die Ansätze der objektorientierten Wissensdarstellung sowie die algebraische Repräsentation zu deren Strukturierung nutzen. Objektorientierung ist die Strukturierung von Wissen in Klassen und Instanzen. Der entscheidende Vorteil ist das hohe Maß an Modularität und Sicherheit in Bezug auf Seiteneffekte. Im Gegensatz zu herkömmlichen Repräsentationsformen, in denen der prozedurale bzw. funktionale Aspekt im Mittelpunkt ist, steht hier das Objekt im Mittelpunkt. Ein Objekt stellt ein Konzept, eine Abstraktion oder ein reales Ding mit definierten Grenzen und Bedeutungen dar. Mit dem Objekt sind Aktionen assoziiert, deren Anwendung auf das Objekt sinnvoll erscheint. Ihre innere Struktur wird damit nach außen hin bedeutungslos. Einzige Relevanz hierbei ist ihr Eingabe- bzw. Ausgabe-Verhalten. Dem Prozeduraufruf in nicht objektorientierten Repräsentationsformen entspricht die Versendung einer Nachricht an ein Objekt. Die

Reaktion auf die Aktivierung der Methode kann die Veränderung des internen Zustandes des Objektes, eine Antwortinformation oder das Versenden einer Folgenachricht an ein weiteres Objekt bewirken. Der interne Zustand eines Objektes ist meist durch die Belegung lokaler Zustandsvariablen definiert. Für die Objektrepräsentation sind verschiedene Konzepte vorgesehen:

- **Methoden:** Sind der reaktive Teil der Nachrichten und bestehen aus einem die Nachricht erfassenden und einem die Reaktion beschreibenden Teil.
- **Slots:** Dienen der Objektstrukturierung und sind mit Werten belegt.
- **Vererbungshierarchie:** Die Übernahme von Objekteigenschaften erfolgt durch die Einbettung von Objekten in eine Vererbungshierarchie. Klassen sind durch die Spezialisierung eines Verhaltens einer bereits bestehenden Klasse spezifiziert und können Eigenschaften und Methoden von dieser erben.
- **Regeln:** Sie beschreiben das Verhalten von Objekten unabhängig von eingehenden Nachrichten durch einen Bedingungs- und einen Aktionsteil.
- **Constraints:** Abhängigkeiten bzw. Bedingungen zwischen den Slots, die es aufrecht zu erhalten gilt, sind durch die Constraints definiert und werden durch die Regeln beschrieben.

Die objektorientierte Repräsentation bietet als Vorzug eine hohe inhaltliche Strukturiertheit des Wissens. Dem steht entgegen, dass eine Repräsentation von Relationen und damit die Beschreibung strukturierter Elemente nur eingeschränkt möglich ist. Der Einbau von neuen Wissensseinheiten in die Vererbungshierarchie ohne Verletzung der Konsistenz erweist sich ebenfalls als schwierig. Mit der Objektorientierung werden die folgenden Zwecke verfolgt (Richter, 1989):

- Erhöhung der Übersichtlichkeit und Sicherheit durch Blockbildung,
- mehr Klarheit durch Verbergen unwichtiger Details und
- Anwendungsunabhängigkeit durch die Möglichkeit der Datenabstraktion.

Weitere, mehr technische Ziele, sind die Erhöhung der Kompaktheit der Repräsentation und die Möglichkeit, Gemeinsamkeiten zwischen Objekten durch Vererbung untereinander weitergeben zu können. Eine mehrfache Implementierung ähnlicher Sachverhalte ist dadurch

nicht mehr nötig. Objekte, zwischen denen eine Vererbungsrelation besteht, bilden einen Graphen und sind in einem Kommunikationsmodell organisiert.

#### 4.2.1.2 Relationale Strukturen

Elemente, die in einer relationalen Struktur repräsentiert werden, können durch eine relationale Algebra, d.h. einen markierten Graphen, beschrieben werden. Ein markierter Graph ist definiert durch (Wysotzki, 1998):

$$\begin{aligned}
 \text{Graph} &= (N, E, I_N, I_E, f, g) \\
 N &: \text{Menge der Knoten} \\
 E &: \text{Menge der Kanten} \\
 I_N &: \text{Knotenmarkierung} \\
 I_E &: \text{Kantenmarkierung} \\
 f, g &: \text{Knoten - bzw. Kantenmarkierung}
 \end{aligned}$$

Bei dieser Darstellung werden die Situationselemente auf die Knoten, die Relationen auf die Kanten des Graphen abgebildet. Wysotzki hat diese Art von Situationsrepräsentation ausführlich diskutiert. Die Darstellung erfolgt durch die sogenannte Adjazenzmatrix, in der auf die Diagonalelemente die Knoten (das sind hier die Situationselemente) und auf die Nicht-Diagonalelemente die Kanten (das sind hier die Relationen) eingetragen werden.

$$\left( \begin{array}{ccc}
 \textit{Pilot} & \textit{steuert} & \textit{kommuniziert mit} \\
 \textit{steuert}^T & \textit{Flugzeug} & \textit{erfasst auf Radar}^T \\
 \textit{kommuniziert mit} & \textit{erfasst auf Radar} & \textit{Lotse}
 \end{array} \right)$$

$\textit{steuert}^T$  : die zu  $\textit{steuert}$  konverse Relation *wird gesteuert von*

Jedoch ist die Zuordnung der Situationselemente zu den Diagonalelementen der Adjazenzmatrix (Nummerierung, Anordnung) nicht eindeutig. Daraus ergeben sich  $n!$  mögliche Anordnungen bzw. sogenannte isomorphe Varianten. Isomorphie zwischen zwei strukturierten Objekten liegt vor, wenn

$$PA(\text{Graph})P^{-1} = A(\text{Graph}')$$

$P$ : Permutationsmatrix

$A$ : Adjazenzmatrix

$\text{Graph}$ : Abstrakter (markierter) Graph

Isomorphie ist also durch eine (relationserhaltende) Simultanvertauschung von Zeilen und Spalten charakterisiert. Dem Erkennen strukturierter Objekte muss deshalb immer ein Isomorphietest vorangestellt werden, da ja die Anordnung der Diagonalelemente willkürlich ist.

Diese rein algebraische Repräsentation von Situationsobjekten durch einen Graphen birgt jedoch einige Problematiken. Nach Wysotzki sind dies die Ausrichtung auf einen fest zugeschnittenen Diskursbereich und die damit verbundene begrenzte Verwendbarkeit bei multiplen Anforderungen, der mangelnde Bezug auf die Merkmalsbeschreibung der im Graphen abgelegten Begriffe, die statische Beschreibung stereotyper Situationen und Ereignisverläufe und die nur unzureichende Repräsentation von Regelwissen. Die mangelnde Beschreibung des Situationsobjektes mit der Repräsentation seiner charakteristischen Eigenschaften ist einer der schwerwiegendsten Nachteile, die es für die Repräsentation von Situationswissen zu behandeln gilt.

#### 4.2.2 IMPLIZITE REPRÄSENTATION VON SITUATIONSWISSEN

Neben der symbolischen Repräsentation durch Situationselemente und Relationen existieren *implizite* Wissensformen. Hierzu zählt z.B. Wissen, welches in Karten vorhanden ist und keinerlei explizite Information über die abgebildeten Objekte und deren Relationen untereinander mit sich führt. Für die Verarbeitung werden die Merkmale und metrischen Beziehungen zwischen Objekten direkt in analoger Form dargestellt. Es bedarf also weiterer Grundelemente der Situation, die über diejenigen von Barwise und Perry hinausgehende. Die Repräsentation der bildhaften Teile der Situation soll durch die Menge  $\mathbf{K}$  von Kontinuen  $\mathbf{k}$  beschrieben werden. Kontinuen sollen hier als bildtragende Objekte verstanden sein. Damit unterscheiden sie sich fundamental von den objektorientierten Situationselementen und werden gesondert repräsentiert. Hieraus ergibt sich eine Erweiterung des einfachen, relationalen Ansatzes:

$$S = \{E, K, R\}$$

Kontinuen sind in vielfältiger Form präsent. Die bildhafte Repräsentation (z.B. aus dem Sehvermögen) projiziert Kontinuen unmittelbar. Der gewichtigste Vorteil einer impliziten Repräsentationsform besteht darin, dass nicht die Gesamtheit aller möglichen Relationen zwischen den Situationselementen gespeichert werden muss, sondern dass sich diese Inhalte implizit in einem Kontinuum befinden und von dort nach Bedarf entzogen werden können, um in eine symbolische Repräsentation übertragen zu werden. Hierfür sind jedoch intelligente Erkennungsverfahren für die Ermittlung von Situationselementen und Relationen innerhalb dieses Kontinuums nötig. Zum heutigen Zeitpunkt ist eine direkte, analoge Repräsentation von solchem Wissen in Digitalrechnern nicht möglich.

#### 4.2.2.1 Hyper-Karten

Implizites Wissen lässt sich jedoch durch Vektorenmengen für typische Situationen approximieren und damit konkret darstellen. Hauptmerkmalsvektoren  $\vec{p}$  repräsentieren hierbei einen n-dimensionalen Punkt eines Kontinuums der Parametermenge  $P$ :

$$\vec{p} = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \quad , \text{ mit } p_i \in P$$

Die Menge  $S_K$  aller möglichen kontinuierlich repräsentierten Situationen, also die Menge aller möglichen Hauptmerkmalsvektoren spannt einen Hyperraum der Dimension  $N$  auf, welcher durch  $P_H$  repräsentiert wird.

$$S_K = P_H \quad , \text{ mit } P_H \in P^N .$$

### 4.3 BEITRÄGE DER TEILPROZESSE ZUM SITUATIONSWISSEN

Für eine vollständige Beschreibung der Situation ergibt sich eine große Zahl an Instanzen von Grundelementen. Der kognitive Prozess verarbeitet typischerweise jedoch nur eine reduzierte Menge all dieser Instanzen, nämlich diejenigen, welche sich aus dem Situationskontext und den Ergebnissen der kognitiven Teilprozesse ergeben. Die Selektion dieser Elemente erfolgt durch die Fähigkeit zur *Selektiven Aufmerksamkeit*.

Zunächst soll das dem kognitiven Prozess bekannte Situationsbild, welches dem inneren Zustand der *Situationsrepräsentation* entspricht, *aktuelle Situation*  $S_a$  genannt werden.  $S_a$

umfasst die Menge aller als relevant erfassten bzw. ermittelten momentanen Situationselemente  $E_a$ , Kontinuen  $K_a$  und deren Relationen  $R_a$  zueinander.  $S_a$  stellt damit den Teil von  $S$  dar, welcher dem kognitiven Prozess bewusst ist. Es sei:

$$S_a = E_a \cup K_a \cup R_a$$

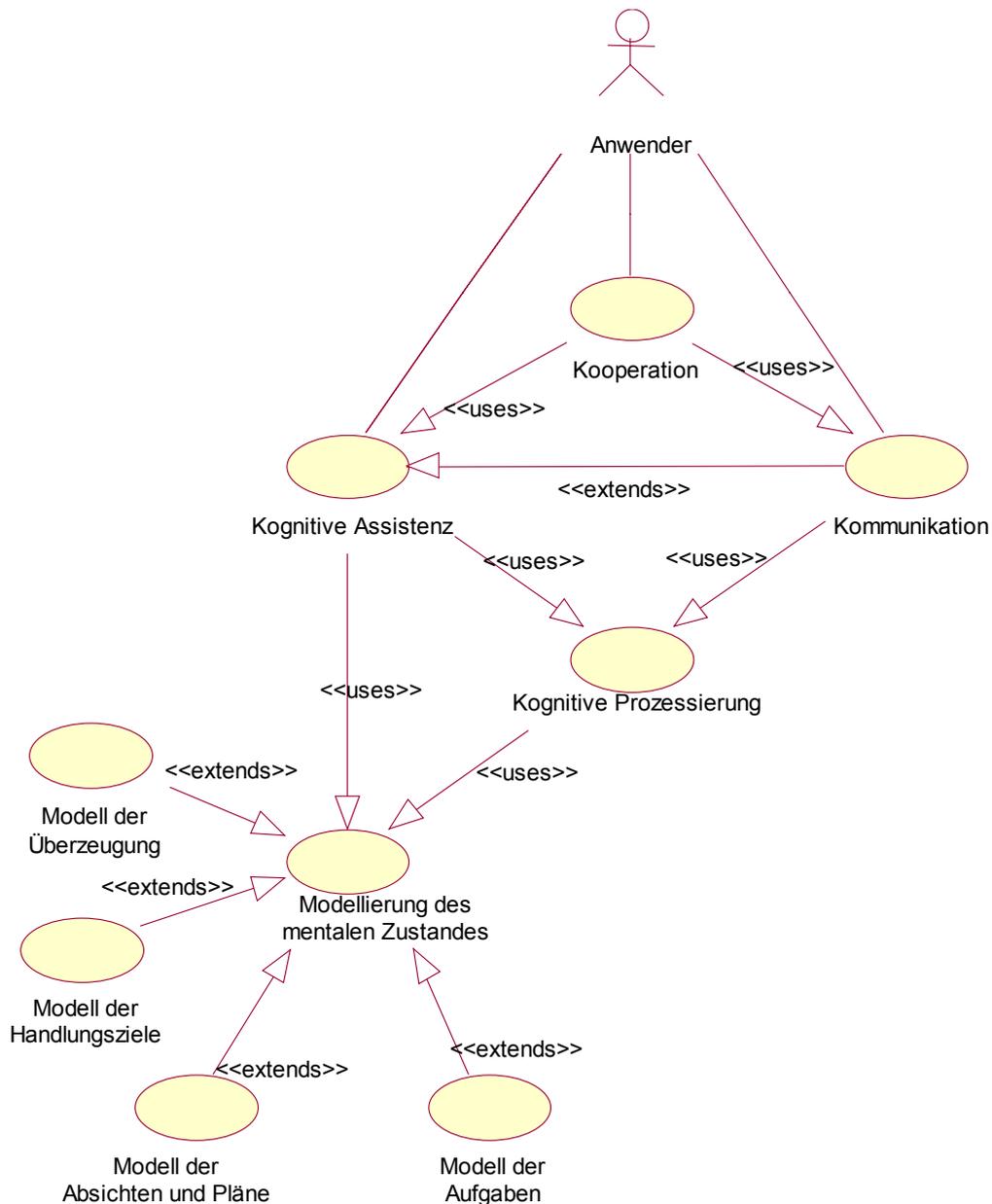
Die Menge der aktuellen Situationselemente  $E_a$ , der Kontinuen  $K_a$  und der Relationen  $R_a$  lässt sich bezüglich der Ergebnisse der kognitiven Teilprozesse unterteilen, wobei sich der folgende Zusammenhang ergibt:

$$S_a = E_{InptData} \cup E_{ReIvObj} \cup E_{ReIvGoals} \cup E_{IntentPlan} \cup E_{Tasks} \cup K_a \cup R_a$$

Entsprechend der Annahme, dass Situationen durch die Konstrukte aus Situationselementen, Kontinuen und Relationen beschrieben werden können, ergeben sich aus obiger Gleichung durch Zusammenfassen weitere typische Bereiche der Situation. Diese sind auf die sie generierenden kognitiven Teilprozesse bezogen, welche weiter unten noch im Detail diskutiert werden. Es ergibt sich hieraus die vereinfachende Annahme:

$$S_a = S_{InptData} \cup \underbrace{S_{ReIvObj} \cup S_{ReIvGoals} \cup S_{IntentPlan} \cup S_{Tasks}}_{\text{Abgeleitete Situationen des Kognitiven Prozesses}},$$

Ausgehend von dieser globalen Situationsbeschreibung soll nun auf Inhalt und Struktur der spezifischen Situationselemente näher eingegangen werden. Den Situationselementen wird in wissensbasierten Systemen eine *Rolle* zugewiesen, welche sie in einem spezifischen Kontext spielen. Daher muss nun für die kognitiven Grundelemente eine ihrer *Rolle* entsprechende Spezialisierung vorgenommen werden. Aus den bis hierher dargestellten Sachverhalten resultiert der Use-Case der „mentalen Modellierung“ (siehe Abbildung 29).

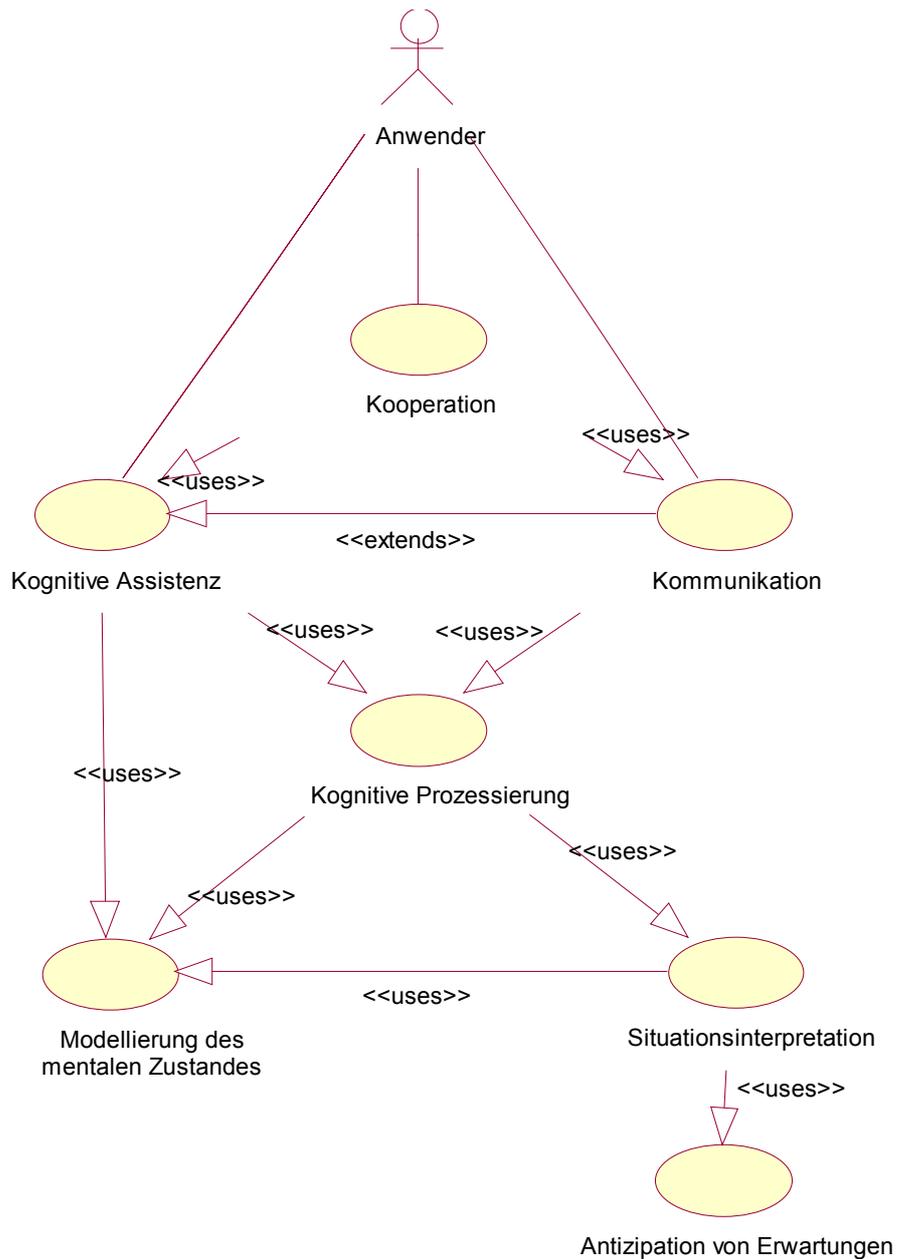


**Abbildung 29: Use-Case-Diagramm „Mentale Modellierung“**

Hierauf aufbauend werden nun formale Funktionsblöcke erstellt, mit deren Hilfe eine systematische Beschreibung des Gesamtprozesses möglich ist.

#### 4.3.1 SITUATIONSINTERPRETATION

Die Funktion der *Situation Interpretation* definiert den gleichnamigen kognitiven Teilprozess (siehe Abbildung 30).



**Abbildung 30: Use-Case-Diagramm „Situationsinterpretation“**

Ergebnis sind die für das Verhalten des Systems relevanten Lageobjekte, die aus den Eingangsdaten und den im Hintergrundwissen enthaltenen bekannten Weltmodellen ermittelt werden:

$$S_{\text{ReIvObj}} = \text{SituationInterpretation}(S_{\text{InptData}}, \text{Models of World})$$

$S_{\text{ReIvObj}}$  stellt die Überzeugung des kognitiven Systems darüber dar, in welchem Zustand es sich selbst sowie seine Umwelt sich befinden. Das Resultat des Teilprozesses der

*Situationsinterpretation* ist  $S_{RelvObj}$ .  $S_{RelvObj}$  beinhaltet die von  $e$  abgeleiteten Spezialisierungen  $O_{Relv}$ . Die Menge  $O_{Relv}$  verkörpert diejenigen abstrakten oder konkreten Situationselemente, welche für das kognitive Verhalten zum gegenwärtigen Zeitpunkt von Relevanz sind. Kern von  $O_{Relv}$  ist eine Menge  $P$  von Hauptmerkmalen  $p$  und eine Menge  $F$  von Eigenschaften  $f$ , welche ein Modell für das entsprechende Weltobjekt bilden. Hieraus ergibt sich

$$O_{RelvObj} = \{F, P\}.$$

$S_{RelvObj}$  stellt entsprechend obiger Gleichung eine Untermenge von  $S_a$  dar:

$$S_{RelvObj} = \{O_{RelvObj}, R_{RelvObj}\} \supseteq S_a$$

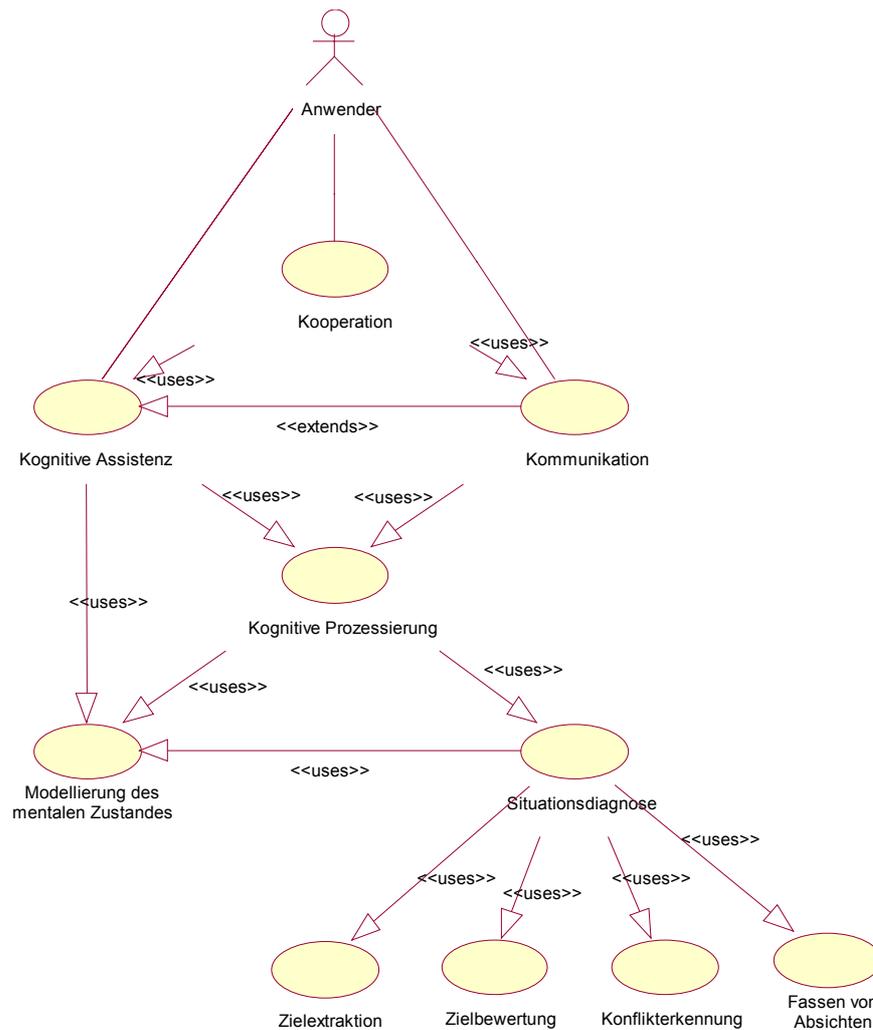
### 4.3.2 SITUATIONS DIAGNOSE

Die Funktion der *SituationDiagnosis* ist durch 4 Funktionsblöcke definiert, wobei zwei davon von besonderer Relevanz sind (siehe Abbildung 31).

Aufgabe ist es, aus den Wünschen, die dem System durch die zu Grunde liegende Spezifikation bekannt sind, diejenigen Ziele zu ermitteln, welche auf Basis der gewonnenen Überzeugung  $S_{RelvObj}$  relevant und damit aktuell anstrebenswert sind. Die Ermittlung der Zielsituation  $S_{RelvGoals}$ , als erster Schritt im Teilprozess der *Situationsdiagnose*, ist für das gesamte Verhalten des kognitiven Systems entscheidend.  $S_{RelvGoals}$  stellt den mentalen Zustand hinsichtlich der Ziele des Systems dar. Die Menge aller möglichen *Wünsche*  $g$  sei  $G$ . *Wünsche* stellen diejenigen Ziele dar, welche im Rahmen des betrachteten Prozesses unabhängig von deren Relevanz grundsätzlich möglich sind. Ziele sind im Hintergrundwissen abgelegt und werden erst durch die vorliegende Situation aktiviert und damit durch ihre Relevanz bewusst gemacht.

Die sich hieraus ergebende Teilsituation ist:

$$S_{RelvGoals} = \text{SituationDiagnosis}_1(S_{RelvObj}, \text{Desires})$$



**Abbildung 31: Use-Case-Diagramm „Situationsdiagnose“**

Im Teilprozess der *Situationsdiagnose* wird jeder im Hintergrundwissen abgelegte Wunsch  $g$  gegenüber der Überzeugung zur aktuellen Situation  $S_{RelvObj}$  bewertet. Ergebnis dieser Analyse ist die Relevanz des entsprechenden Ziels in der momentanen Situation.

Jedem Ziel ist aus diesem Grund ein Relevanzkriterium  $S_{CI}$  zugeordnet, welches dieser Bewertung zugrunde gelegt wird.  $S_{CI}$  beschreibt alle diejenigen Situationen, in denen das Ziel anzustreben ist und damit Relevanz erlangt.

$$\text{IF } S_{RelvObj} \in S_{CI} \text{ THEN } g \rightarrow g_r \text{ ENDIF}$$

Ein einfaches Beispiel soll diesen Vorgang verdeutlichen. Es existiere das Ziel „Vermeide Flug durch Gewittergebiete“. Dieses Ziel sei in all denjenigen Situationen relevant, in denen sich

- das Flugzeug in der Luft befindet,
- ein Gewittergebiet im Raum-Zeit-Gebiet bekannt ist und
- die Annäherungsgeschwindigkeit des Flugzeugs auf das Gewittergebiet positiv ist.

Das Relevanzkriterium  $\mathcal{S}_{CI}$  resultiert damit aus den drei Teilsituationen

$$\mathcal{S}_{CI} = s_{CI_1} \cup s_{CI_2} \cup s_{CI_3},$$

nachfolgend in Form der Adjazenzmatrizen entsprechend dargestellt:

$$s_{CI_1} = (\text{Flugzeug} \quad \text{fliegt})$$

$$s_{CI_2} = (\text{Gewittergebiet} \quad \text{ist bekannt})$$

$$s_{CI_3} = \begin{pmatrix} \text{Flugzeug} & \text{nähert sich} \\ \text{nähert sich}^T & \text{Gewittergebiet} \end{pmatrix}$$

Aufgabe der *Situationsdiagnose* ist es nun festzustellen, inwieweit die bekannte Situation  $\mathcal{S}_{RelvObj}$  mit dem Relevanzkriterium übereinstimmt. Für das vorliegende Beispiel sei die im Teilprozess der *Situationsinterpretation* gewonnene Situation

$$\mathcal{S}_{RelvObj} = s_{RelvObj_1} \cup s_{RelvObj_2} \cup s_{RelvObj_3}$$

mit den Teilsituationen

$$s_{RelvObj_1} = (\text{Flugzeug} \quad \text{fliegt})$$

$$s_{RelvObj_2} = (\text{Gewittergebiet} \quad \text{ist bekannt})$$

$$s_{RelvObj_3} = \begin{pmatrix} \text{Flugzeug} & \text{entfernt sich} \\ \text{entfernt sich}^T & \text{Gewittergebiet} \end{pmatrix}$$

Das Ähnlichkeitsmaß zwischen der im Teilprozess der *Situationsinterpretation* ermittelten Situation  $\mathcal{S}_{RelvObj}$  und dem Relevanzkriterium  $\mathcal{S}_{CI}$  ist zu bestimmen. Aus dem Vergleich der beiden Situationen ergeben sich zunächst die Ähnlichkeitsvektoren:

$$\begin{pmatrix} 100\% \\ 100\% \\ 100\% \end{pmatrix} = \text{Ähnlichkeit}(s_{CI_1}, s_{RevObj_1})$$

$$\begin{pmatrix} 100\% \\ 100\% \\ 100\% \end{pmatrix} = \text{Ähnlichkeit}(s_{CI_2}, s_{RevObj_2})$$

$$\begin{pmatrix} 100\% \\ 0\% \\ 0\% \end{pmatrix} = \text{Ähnlichkeit}(s_{CI_3}, s_{RevObj_3})$$

und damit eine Ähnlichkeit für das Relevanzkriterium  $S_{CI}$  in Bezug zur Situation  $S_{ReivObj}$  durch die Verknüpfung durch den Minimum-Operator von

$$\begin{pmatrix} 100\% \\ 0\% \\ 0\% \end{pmatrix} = \text{Ähnlichkeit}(S_{CI}, S_{ReivObj}).$$

Für die Bewertung der Ziel-Relevanz ist nun zu entscheiden, welchen Grad an Situationsverständnis man der Bewertung zu Grunde legen soll, wobei in diesem Beispiel die Relevanz schon nach der inhaltlichen Ähnlichkeit der Situationen als nicht zutreffend zu beurteilen wäre. Die Bewertung der Relevanz ist damit abgeschlossen.

Die Menge  $G_r$  aller relevanten Ziele  $g_r$  und die diesen zugeordneten Relationen  $R_{ReivGoal}$  bilden den Anteil  $S_{ReivGoals}$  der Beschreibung der Gesamtrepräsentation:

$$S_{ReivGoals} = \{G_r, R_{ReivGoal}\} \supseteq S_a$$

Relevante Ziele werden unmittelbar in die Situationsrepräsentation eingebracht. Sie sind dann Teil der „*mentalen*“ Situation.

Der sich hieran anschließende zweite Schritt der *Situationsdiagnose* führt auf der Basis der gegebenen Ausgangs- und der bestimmten Ziel-Situation eine Situationsbewertung hinsichtlich der Erfüllbarkeit und darüber hinaus einer möglichen günstigeren Erfüllung der aktuell vorgegebenen Ziele durch.

$$\begin{pmatrix} S_{Conflict} \\ S_{Opportunity} \end{pmatrix} = \text{SituationDiagnose}_2(S_a \setminus S_{RelvGoals}, S_{RelvGoals})$$

Hierbei wird die Untermenge  $S_a \setminus S_{RelvGoals}$  gegenüber  $S_{RelvGoals}$  diagnostiziert. Das Ergebnis dieses Prozesses sind Bewertungen, welche von *Zielkonflikt* bis zu *günstiger Gelegenheit* reichen. Erfasste Situationen der realen Welt, deren Inhalt sich als Teil der für die Erfüllung eines Zieles hinreichenden Kriteriums  $S_{C2}$  darstellt, führen ein relevantes Ziel  $g_r$  in ein erfülltes Ziel  $g_r^*$  über.

$$\text{IF } S_a \in S_{C2} \text{ THEN } g_r(S_a \setminus S_{RelvGoals}) \rightarrow g_r^* \text{ ENDIF}$$

Konflikte  $\bar{g}_r$  ergeben sich hingegen durch divergente Situationsbilder:

$$\text{IF } S_a \notin S_{C2} \text{ THEN } g_r(S_a \setminus S_{RelvGoals}) \rightarrow \bar{g}_r \text{ ENDIF}$$

#### 4.3.2.1 Eigenschaften der Ziele

Das *Modell der Handlungsziele* ist derjenige Bereich, der eine besondere Brisanz besitzt. Der Anspruch an Güte und Vollständigkeit des Modells ist hier besonders hoch, da sich die Modelleigenschaften und Modellinhalte auf die wissensbasierte und deren untergeordnete regelbasierte und fertigkeitbasierte Verhaltensebene direkt auswirken. Eine bezüglich einer interpretierten Situation unzureichend oder gar falsch ermittelte Zielmenge führt konsequenterweise zu einem fehlerhaften Verhalten. Das hohe Abstraktionsniveau in Bezug auf die Situation erschwert jedoch die Modellierung. Die Aktivierung der Ziele erfolgt über ein sehr breites und komplexes Situationsspektrum, was diesen Teil des Modells umso aufwendiger macht.

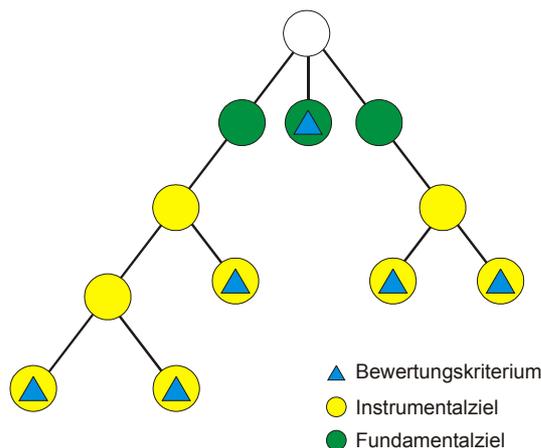
Die Zielsituation ist Teil der Gesamtsituation und ebenfalls ein Konstrukt aus Situationselementen, mit Relationen und Relationsattributen. Die Differenz zwischen der Ausgangs- und der Ziel-Situation stellt den Problemraum dar, den es mit Hilfe von bekannten Operatoren zur Lösung von Einzelproblemen zu überwinden gilt.

Entscheidungen, und damit das Verhalten in einer Situation, sind umso schwieriger zu generieren, je komplexer sich der Problemraum dem Betrachter darstellt. Daher ist die Dekomposition, die Zerlegung des Problems in einfachere Komponenten, anzustreben.

Gleiches lässt sich für die Bestimmung der Zielmenge sagen. Die Verwendung von hierarchischen Strukturen erleichtert deren Handhabung beim Problemlösen. Zielhierarchien sind durch Klassen von Unterzielen (konkretisieren eines abstrakteren Ziels) und Oberzielen (abstrahieren einer Menge von Zielen) definiert und dienen dem Präzisieren einer Zielsituation, indem die Unterziele die Bedeutung des übergeordneten Ziels erklären, bestimmen und es messbar machen.

Hierfür sind Operatoren *op* zu definieren, die aus den Erfülltheitsgraden der Einzelziele den Erfülltheitsgrad des Oberziels ermitteln. Operatoren können den Erfülltheitsgrad auf verschiedenartige Weise bestimmen. Jedoch ist auch bei der Auswahl der Operatoren Sorgfalt geboten. Erfülltheitsgrade von konkurrierenden Instrumentalzielen können z.B. durch einen Maximum-Operator in den Erfülltheitsgrad des Fundamentalziels eingehen. Bei diesem Operator ist das Erreichen des Fundamentalziels schon bei der Erfülltheit eines Instrumentalziels gewährleistet. Ein anderes Beispiel stellt der Minimum-Operator dar. Er kommt zum Einsatz, wenn das Fundamentalziel erst dann erfüllt sein darf, wenn es alle ihm untergeordneten Instrumentalziele ebenfalls sind. Während Minimum- und Maximum-Operator als Ergebnis einen diskreten Erfülltheitsgrad liefern, können Operatoren wie der Mittelwert oder der Produkt-Operator auch kontinuierliche Werte zum Ergebnis haben.

Eisenführ und Weber (Eisenführ, 1993) haben bereits die Unterscheidung in Fundamental- und Instrumentalziele diskutiert (siehe Abbildung 32).



**Abbildung 32: Fundamental- und Instrumentalziele (Eisenführ, 1993)**

Fundamentalziele stellen demgemäss Ziele dar, welche um ihrer selbst Willen verfolgt werden. Sie sind jedoch immer nur in einem bestimmten Kontext fundamental.

Instrumentalziele werden verfolgt, um eine positive Wirkung auf die Erreichung eines übergeordneten, fundamentaleren Ziels zu erzielen. Bei der Entwicklung einer Zielhierarchie kann sowohl *Top-down* als auch *Bottom-up* verfahren werden.

Die Eigenschaften eines Ziels sollen hier für die Überlegungen einer anschließenden Modellierung nochmals geschlossen dargestellt werden, sie lauten:

- Vernetztheit (Teil-/Unterziele)
- Relevanz (Situationskontext)
- Polarität (positiv/negativ)
- Grad der Erfülltheit
- Gewichtung (Priorität)

Die Abbildung dieser Eigenschaften in einem Modell ist grundlegend mit dessen Repräsentationsform verbunden. Auf diese Problematik soll an späterer Stelle eingegangen werden.

#### 4.3.2.1.1 Vernetztheit

Die *Vielzieligkeit* oder *Polytelie* ist ein Merkmal von Zielen. Es bedeutet, dass eine Situation mehreren Zielen entsprechen sollte. Die Entsprechung bezüglich mehrerer Ziele bedeutet eine Vernetztheit unter den Situationsmerkmalen und damit auch eine Vernetztheit unter den Zielen. Dabei sind eine positive und eine negative Vernetztheit zu unterscheiden. Positiv miteinander vernetzte Ziele sind einfacher zu handhaben, als es negative sind. Die negative Verknüpfung der Ziele untereinander führt zu Zielkonflikten.

#### 4.3.2.1.2 Relevanz

Ziele sind nicht einfach da, sondern müssen durch „intensives Nachdenken“ erarbeitet und formuliert werden. Jedoch lassen sich relevante Ziele anhand folgender Ausgangspunkte identifizieren (Eisenführ, 1993):

- Mängel des Ausgangszustands,
- Vergleich von Alternativen,
- strategische Ziele,
- externe Vorgaben und

- die Ziele anderer betroffener Personen.

Ziele aus Mängeln und Ziele aus dem Vergleich von erkannten Alternativen bedürfen einer Lernfähigkeit auf der wissensbasierten Verhaltensebene. Für die hier vorliegende Arbeit hat man sich auf *strategische Ziele* und Ziele aus *externen Vorgaben* beschränkt, da diese für das hier betrachtete Anwendungsgebiet besonders typisch und wichtig sind. Strategische Ziele sind nicht an eine bestimmte Entscheidungssituation gebunden, sondern haben ein übergreifendes Naturell. Klar definierte und bewusste strategische Ziele sind in konkreten Situationen zu berücksichtigen. Die Aktivierung dieser Ziele ist durch ein Kriterium zur Relevanz des Ziels zu beschreiben, welches bereits unter 4.3.2 erläutert wurde.

#### 4.3.2.1.3 Polarität

Ziele können in die verschiedensten Kategorien unterteilt werden. Zunächst ist es jedoch von Bedeutung, ob es sich um ein *positives* oder *negatives* Ziel handelt. Positive Ziele stellen Situationen dar, die angestrebt werden, sind damit also spezifisch und haben bekannte Kriterien für die Erfüllung des Ziels. Negative Ziele hingegen sind recht global definiert, da sie eine aktuelle Situation beschreiben, die verändert werden soll, ohne die Veränderung (also den positiven Zielzustand) selbst genau zu kennen. Sie sind durch recht wenige Kriterien festgelegt und damit einfach. Positive Ziele sind spezifisch und besitzen gewöhnlich einen höheren Detaillierungsgrad bzw. eine genauere Beschreibung, als sie die negativen Ziele haben; mit der Konsequenz, dass damit auch die Erreichung des Ziels viel spezifischer zu sein hat. Dörner (Dörner, 1979) unterscheidet globale Ziele noch hinsichtlich deren *Klarheit*. *Unklare* Ziele sind Ziele, welche keinerlei Kriterien besitzen, die für die Bestimmung der Erfülltheit des Zieles heranzuziehen sind. Bei unklaren Zielen ist der Zustand, der angestrebt werden soll, nicht im Detail bekannt, muss also auf anderem Wege ermittelt werden.

#### 4.3.2.1.4 Grad der Erfülltheit

Der Grad der Erfülltheit des Zieles repräsentiert ein Maß dafür, inwieweit die bewertete Situation der Zielsituation entspricht. Der Grad der Erfülltheit wird durch einen Zugehörigkeitsgrad  $t$  repräsentiert:

$$t \in \mathfrak{R}[0...1]$$

#### 4.3.2.1.5 Gewichtung

Zielkonflikte stellen einen natürlichen Zustand dar, der durch entsprechende Instrumentarien behandelt werden soll. Zunächst ist dies hier mit einer Gewichtung  $w$  der Ziele möglich, indem wichtige von weniger wichtigen Zielen abgegrenzt werden:

$$w \in \mathfrak{R}[0...1]$$

#### 4.3.2.2 Ermittlung der Zielgewichte

Die hierarchische Struktur der Ziele und der Einfluss von Instrumental- auf Fundamentalziele sind bei der Gewichtung und Verknüpfung durch die Operatoren zu berücksichtigen. Zunächst soll ein Theorem für die Erfülltheit zu Grunde gelegt werden, welches die Erfülltheit der Fundamentalziele aus der Menge der Instrumentalziele sicherstellt. Operatoren haben unabhängig von ihrer Charakteristik bei vollständiger Erfülltheit aller Instrumentalziele dem entsprechenden Fundamentalziel den Erfülltheitsgrad 1 zuzuordnen.

$$op(w_i \cdot 1) = 1$$

Die Dynamik des Zielmodells und die sich damit zur Laufzeit ändernde Zahl von Fundamental- bzw. Instrumentalzielen, bedarf einer Berücksichtigung dieser Forderung bei den Verknüpfungsoperatoren. Hierzu dient der Faktor  $C_w$ , welcher sicherstellt, dass für jede Menge an Instrumentalzielen und jeden Wert von  $t$  die vollständige Erfülltheit des Fundamentalziels gewährleistet ist.

$$C_w = \frac{1}{op(w_i \cdot 1)}$$
$$C_w \cdot op(w_i \cdot t_i) = t_{\text{Oberziel}}$$

Jedoch stellt die Ermittlung der Gewichte selbst einen nicht trivialen Prozess dar, dem besondere Aufmerksamkeit geschenkt werden muss. Für die Ermittlung der Zielgewichte existieren verschiedene Verfahren, die in der Literatur (siehe Eisenführ, 1993) unter den Kapiteln *Entscheidung unter Sicherheit bei mehreren Zielen* behandelt werden. Sie sollen an dieser Stelle kurz Erläuterung finden. Die im Rahmen dieser Arbeit zu betrachtende Theorie basiert auf der *Multiattributiven Wertfunktion*, d.h. es existiert Inferenzwissen, abhängig von

einer Menge von Situationsattributen, welches einer Ausgangs-Situation einen Erfülltheitsgrad in Abhängigkeit von der Distanz zur Zielsituation zuweist.

#### 4.3.2.2.1 Trade-Off-Verfahren

Die Bestimmung der Zielgewichte nach dem *Trade-Off Verfahren* (Eisenführ, 1993) behandelt Zielkonflikte, bei denen der Entscheider indifferent ist. Grundlage des Verfahrens ist eine Wertfunktion  $f(\dots)$ , die eine zu bewertende Alternative in das Werteintervall  $[0...1]$  überführt und damit den Grad der Erfülltheit ermittelt. Aus einer Indifferenzaussage zwischen den Alternativen

$$a = (a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{j-1}, a_j, a_{j+1}, \dots, a_m)$$

$$b = (a_1, \dots, a_{i-1}, b_i, a_{i+1}, \dots, a_{j-1}, b_j, a_{j+1}, \dots, a_m)$$

kann mittels

$$w_i f_i(a_i) + w_j f_j(a_j) = w_i f_i(b_i) + w_j f_j(b_j)$$

auf die Gewichtung der Attribute  $X_i$  und  $X_j$  geschlossen werden. Mit  $m-1$  solcher Gleichungen hat man ein eindeutig lösbares Gleichungssystem, deren Unbekannte die Gewichtungen  $w$  darstellen.

#### 4.3.2.2.2 Swing-Verfahren

Das *Swing-Verfahren* (Eisenführ, 1993) basiert auf der Quantifizierung des Entscheiders über die gesamte Menge der Alternativen auf einer Werteskala von 0 bis 100. Bei einer n-fachen Alternativenmenge und den gesammelten Bewertungen  $x$  ergeben sich die Gewichte aus

$$w_i = \frac{x_i}{\sum_{j=1-n} x_j}$$

#### 4.3.2.2.3 Direct-Ratio-Verfahren

Das in der Praxis meist verwandte Verfahren ist das *Direct-Ratio-Verfahren* (Eisenführ, 1993). Aufgabe der Entscheidungsträger ist es, zunächst aus einer Menge von Alternativen diese ihrer *subjektiven Wichtigkeit* nach zu ordnen und im Anschluss eine quantitative Aussage  $r$  über das Verhältnis der Wichtigkeiten der Attribute und Alternativen zu geben. Ein

Ergebnis könnte somit sein, dass eine Alternative *A* zweifach wichtiger ist als eine Alternative *B*. Die Einzelgewichte ergeben sich dann aus

$$w_i = \frac{r_i}{\sum_{j=1-n} r_j}$$

Eine kritische Betrachtung der drei Verfahren lässt folgende Schlüsse zu. *Trade-Off* basiert auf dem Problem des *indifferenten* Entscheiders, und Ziel ist es hierbei, sensitive Merkmale für eine Entscheidung zu identifizieren und aus dem Ergebnis eine Priorisierung abzuleiten. Das *Swing* Verfahren bedarf einer Bewertung aller möglichen Alternativen, stellt also für den in dieser Arbeit behandelten Anwendungsfall ein Problem dar, da das Verfahren für die hohe Komplexität der Anwendung nicht praktikabel erscheint. *Direct-Ratio* ist ein einfaches, robustes Verfahren und scheint daher für eine Anwendung geeignet. Jedoch sind mit dem *Direct-Ratio*-Verfahren durch die verwandte Systematik Fehlerquellen verbunden, die jedoch durch eine nachgeschaltete Sensitivitätsanalyse der gewonnenen Gewichte größtenteils eliminiert werden können.

#### 4.3.2.3 Sensitivitätsanalyse

Der Ermittlung der Ziel-Gewichte folgt eine Sensitivitätsanalyse. Insbesondere bei komplexen Zielmengen muss geprüft werden, wie signifikant der Einfluss von einzelnen Parameteränderungen auf das Gesamtergebnis ist. Für dieses nicht-triviale Problem müssen zunächst folgende vereinfachende Annahmen getroffen werden (Eisenführ, 1993):

$$w \in [0,1]$$

$$\frac{w_i}{w_j} = \text{bekannt}$$

Hieraus lassen sich je nach Operator verschiedene Sensitivitätsfunktionen ableiten. Beispiele sind hier für den Summen- und den Produktoperator gegeben:

$$op(w) : C_w \cdot \sum_i w_i = 1$$

Summen-Operator:

$$w_j = (1 - C_w \cdot w) \cdot \frac{w_j}{C_w \cdot \sum_i w_i - w}$$

$$\text{Produkt-Operator: } \begin{aligned} op(w) : C_w \cdot \prod_i w_i &= 1 \\ w_j &= \frac{1}{w} \cdot \frac{w \cdot w_j}{C_w \prod_i w_i} \end{aligned}$$

Durch die Verläufe der Gewichte und die Bewertungen von Lösungsalternativen lassen sich kritische Kombinationen von Gewichtungen ermitteln. Probleme der Überbewertung von Einzelzielen durch zu hohe Gewichtungen oder Kombinationen von Gewichtungen können dadurch erkannt werden.

#### 4.3.2.4 Zielrepräsentation durch Hyperkarten

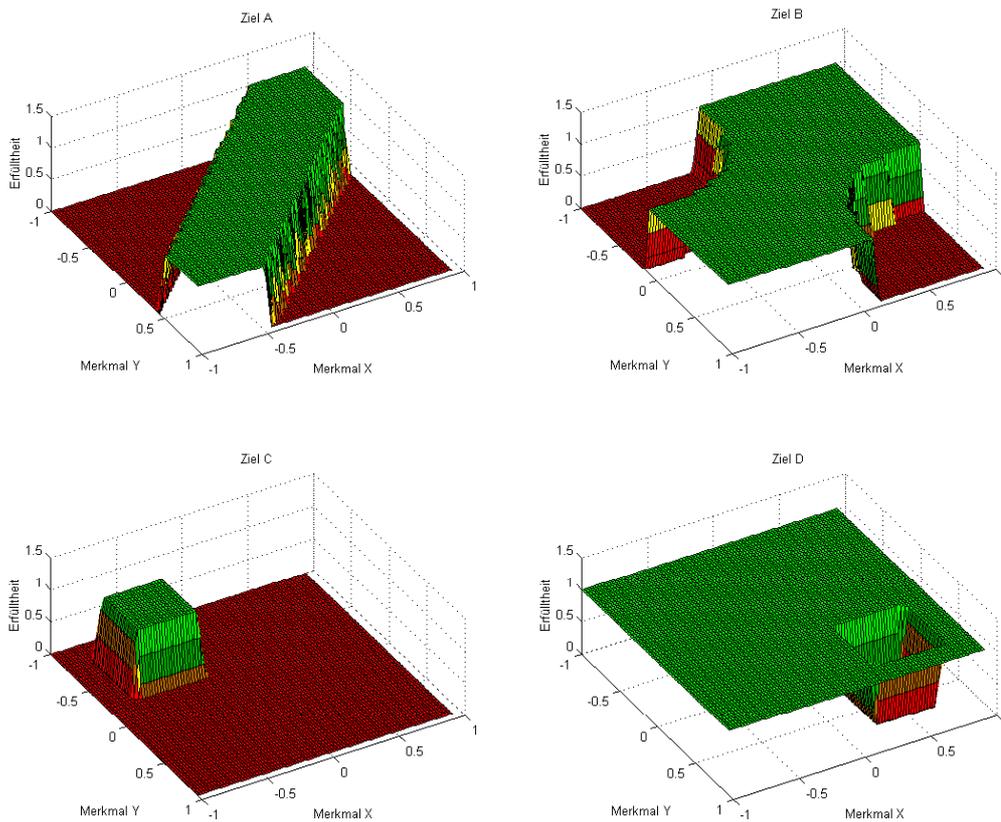
Zielsituationen sind für eine maschinelle Verarbeitung in implizites Situationswissen zu überführen. Dabei ist jeder Merkmalsvektor und damit die durch ihn repräsentierte Situation gegenüber einem Ziel zu bewerten. Hieraus ergibt sich eine Dimensionssteigerung des Situationsraumes, aufgespannt durch den Hauptmerkmalsvektor, durch den Erfülltheitsgrad  $t$ :

$$\vec{p}_t = \begin{pmatrix} t \\ p_1 \\ \vdots \\ p_n \end{pmatrix}, \quad \begin{aligned} t &\in \mathfrak{R}[0...1] \\ p_i &\in P \end{aligned}$$

Die Bewertung einer Situation bezüglich eines Ziels ist dann Teil der Repräsentation eines Hauptmerkmalsvektors. Die sich hieraus ergebende Situationsbeschreibung lautet:

$$g(S_K) = P_{H_t}, \quad P_{H_t} \in P_t^M$$

Ein kurzes Beispiel soll das Konzept erläutern. Für den Hyperraum der Merkmale  $x$  und  $y$ , aufgetragen über der Erfülltheit der Hauptmerkmalsvektoren, sind in Abbildung 33 Hyperkarten der Fundamentalziele  $g_A$ ,  $g_B$ ,  $g_C$  und  $g_D$  dargestellt, welche implizite Ziel-Situationen veranschaulichen.



**Abbildung 33:** Hyperkarten der Ziele  $g_A$  (links oben),  $g_B$  (rechts oben),  $g_C$  (links unten) und  $g_D$  (rechts unten)

In Abbildung 33 oben sind Hyper-Karten der Ziele  $g_A$  (links) und  $g_B$  (rechts) mit ausgeprägten Bereichen dargestellt, in denen ein Gebiet von Hauptmerkmalsvektoren die Zielkriterien erfüllen. In Abbildung 33 unten sind ein *positives* Ziel  $g_C$  (links) und ein *negatives* Ziel  $g_D$  (rechts) dargestellt, welche in ihrer konkreten und lokalen Ausprägung Situationsräume beschreiben, die für die Erfüllung des Ziels notwendig bzw. ausschließend sind.

Bei der Repräsentation von einzelnen Zielsituationen durch Hyperkarten stellt sich ebenfalls das Problem der Polytelie. Hier gilt es nun, aus der Menge von wichtigen und weniger wichtigen Zielsituationen diejenige Gesamtzielsituation zu ermitteln, welche der gesamten Zielmenge am ehesten genügt. Hierzu muss also eine Verknüpfung der unterschiedlichen Hyper-Karten erfolgen. Dazu soll zunächst noch einmal der Hauptmerkmalsvektor betrachtet werden. Gegeben seien die ähnlichen Hauptmerkmalsvektoren  $\vec{p}_{t_A}$  und  $\vec{p}_{t_B}$  für die Situationen  $s_A$  und  $s_B$  :

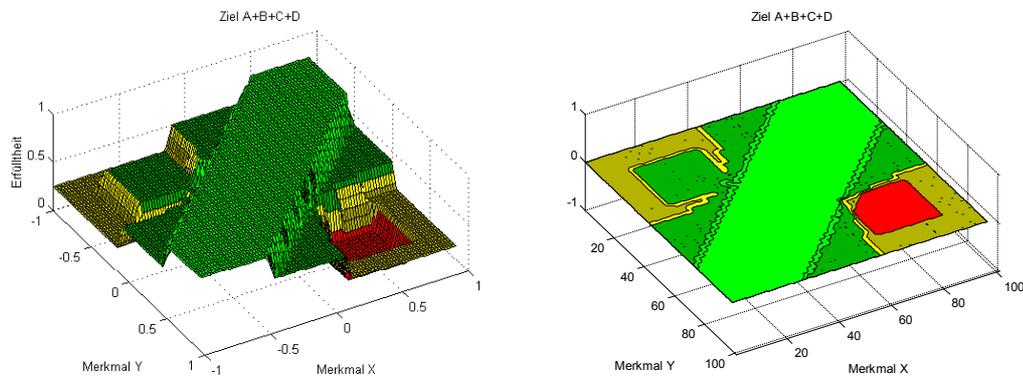
$$\bar{p}_{t_A} = \begin{pmatrix} t_A \\ p_{A_1} \\ \vdots \\ p_{A_n} \end{pmatrix} \quad \bar{p}_{t_B} = \begin{pmatrix} t_B \\ p_{B_1} \\ \vdots \\ p_{B_n} \end{pmatrix}, \text{ mit } p_{A_i} = p_{B_i}, i = 1 \dots n$$

Die Identifikation der Erfülltheit dieser beiden Hauptmerkmalsvektoren bezüglich der Ziele  $A$  und  $B$  erfolgt über den Einzel-Erfülltheitsgrad  $t$ . Ist demnach eine gemeinsame Zielsituation zu bestimmen, so müssen die Erfülltheit  $t_A$  und  $t_B$  miteinander in Relation gesetzt werden. Dies geschieht wieder über einen Verknüpfungsoperator  $op(t_A, t_B)$ , mit den unter 4.3.2.1.5 erarbeiteten Randbedingungen.

$$\bar{p}_{t_A} \circ \bar{p}_{t_B} \rightarrow t_A \circ t_B$$

$$\circ \hat{=} op(T) \quad , \text{ mit } t \in T_{p_{AB}}$$

Die Ermittlung einer übergeordneten Zielsituation für ein Instrumentalziel aus den verschiedenen Hyper-Karten der Abbildung 33 ist in Abbildung 34 gezeigt.



**Abbildung 34: Hyperkarten (Profil-/Konturansicht) gleichgewichteter Zielsituationen**

Hier sind die Zielsituationen der vier einzelnen Ziele überlagert, woraus sich eine neue Zielsituation ergibt, welche alle Situationen der Einzelziele beinhaltet. Als Operator für die Verknüpfung der Fundamentalziele  $g_A, g_B, g_C$  und  $g_D$  wurde hier der Mittelwertoperator

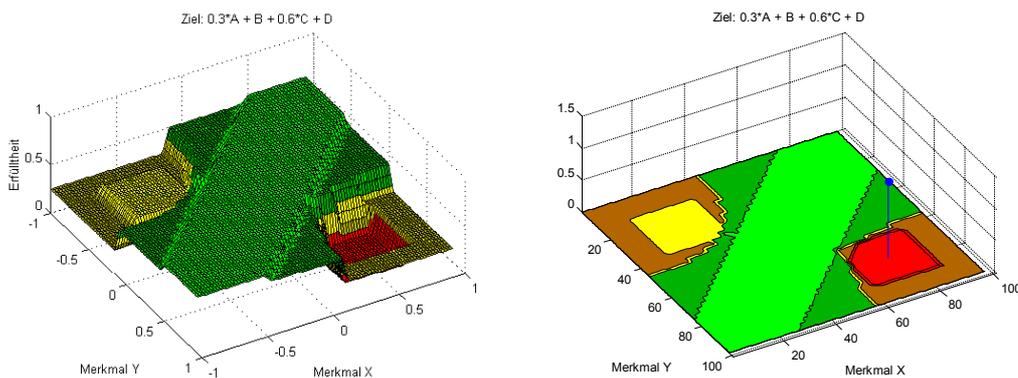
$$\circ \hat{=} op(T) = \frac{\sum_{i=A}^{i=k} t_i}{i} \quad , t \in T$$

verwandt. Um für den Fall der Polytelie wichtige von weniger wichtigen Zielen zu unterscheiden, bedarf es einer Priorisierung der Einzelziele. Hierzu werden den Einzelzielen Prioritäten durch Gewichte  $w$  zugeordnet, wodurch sich diese bezüglich ihrer Wichtigkeit unterscheiden lassen. Hieraus ergibt sich für die Verknüpfung der gewichteten Einzelziele  $A$  und  $B$  eine Unterscheidung in  $w=0.0$  (unwichtig) bis  $w=1.0$  (wichtig):

$$\vec{p}_{t_A} \circ \vec{p}_{t_B} \rightarrow (t_A \cdot w_A) \circ (t_B \cdot w_B)$$

$$\circ \hat{=} op(T, w) \quad , \text{ mit } \begin{matrix} t \in T_{p_{AB}} \\ w \in R[0...1] \end{matrix}$$

Nachfolgende Abbildung 35 zeigt den Einfluss einer Priorisierung ( $w_A=0.3$ ,  $w_B=1.0$ ,  $w_C=0.3$ ,  $w_D=1.0$ ) auf die Hyper-Karten der Zielsituation.



**Abbildung 35: Hyperkarten (Profil-/Konturansicht) gewichteter Zielsituationen**

Hyper-Karten der Zielsituation beschreiben somit vollständig den gesuchten Zielzustand, indem sie jeden bekannten und bezüglich der Zielmenge bewerteten Situationshauptmerkmalsvektor bereitstellen. Hierdurch werden insbesondere die Funktionen der Situationsdiagnose unterstützt, bei denen es (neben der Bestimmung der Zielmenge) um die Identifikation von *Zielkonflikten* und *günstigen Gelegenheiten* geht. Dabei ist mit einer zu diagnostizierenden, bekannten Ausgangssituation  $p$  aus der Hyper-Karte  $S_K$  der zugehörige Wert  $t_p$  zu extrahieren.

Entscheidender Vorteil einer Zielrepräsentation durch implizites Situationswissen in Form von Hyperkarten ist die Kompensierung von potentiellen internen Zielkonflikten. Durch die

Überlagerung der impliziten, gewichteten Zielsituation können kritische Situationen, welche zwar einzelne, aber weniger wichtige Ziele erfüllen, jedoch wichtige Ziele verletzen, identifiziert und dargestellt werden. Wie nachfolgend gezeigt wird, unterstützt die Repräsentation von Zielsituationen in Form von Hyperkarten universelle, von der Domäne unabhängige Algorithmen zur Planung und Entscheidungsfindung.

### 4.3.3 PLANUNG UND ENTSCHEIDUNGSFINDUNG

Der kognitive Prozess verfügt mit den Teilprozessen der *Situationsanalyse* über ein Verständnis bezüglich der Ausgangs-Situation und der Distanz, die es zu verringern gilt, um die bekannte Zielsituation zu erreichen. Damit stellt sich die Aufgabe der *Planung und Entscheidungsfindung* (siehe Abbildung 36) zum Zwecke der Problemlösung. Neben dem dauerhaft abgelegten Planungswissen sind die Resultate aus *Situationsinterpretation* und *Situationsdiagnose* zu berücksichtigen.

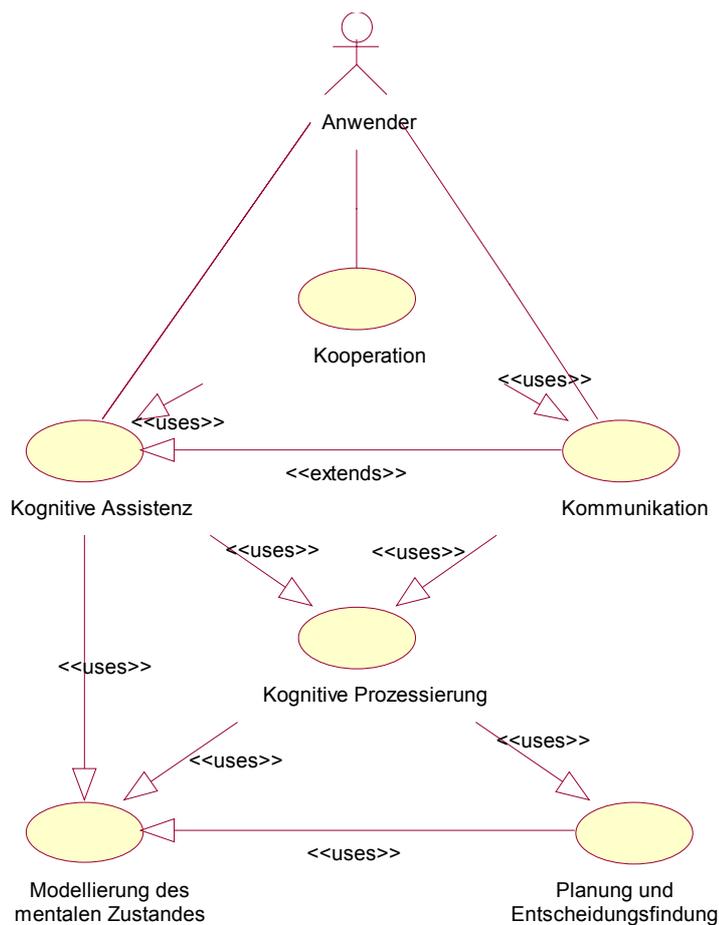


Abbildung 36: Use-Case-Diagramm „Planung und Entscheidungsfindung“

Das Ergebnis dieses Teilprozesses besteht in dem Vorgehensplan  $S_{IntentPlan}$  der diejenigen Handlungsabsichten und zugeordnete Aktionen enthält, welche zum Erreichen der Zielsituation umgesetzt werden müssen.

$$S_{IntentPlan} = Planning(S_{ReivObj}, S_{Cfct}, S_{Oppt.}, \text{Planning - Knowledge})$$

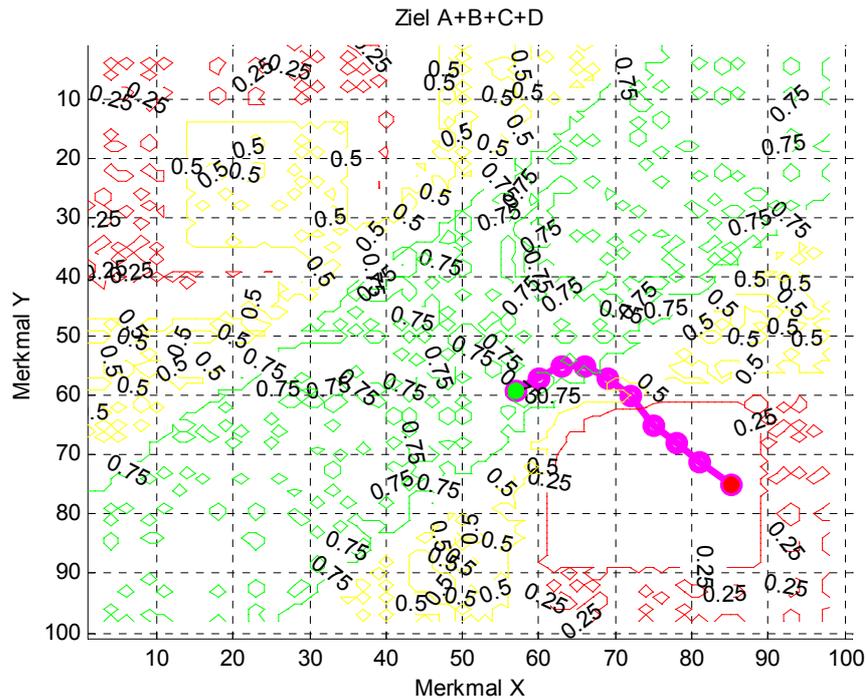
$S_{IntentPlan}$  beinhaltet das beabsichtigte Vorgehen in Form von Absichten  $I$ , Aufgaben  $T$  und deren Relationen  $R_{Task}$  untereinander:

$$S_{IntentPlan} = \{I, T, R_{Task}\} \supseteq S_a$$

Eine Absicht  $i$  kennzeichnet das bevorstehende Vorgehen, ein Ziel erreichen zu wollen. Damit sind Absichten direkt mit den Zielen verwandt und identisch mit deren Struktur. Absichten  $I$  werden im Unterprozess der Planung und Entscheidungsfindung gefasst und leiten das Handeln. Planung bedeutet die Bewertung und Auswahl geeigneter Transformationen oder Operatoren. Ziel ist es, aus einer ermittelten Ausgangs-Situation in eine Ziel-Situation zu gelangen. Hierzu müssen diese Operatoren bereits bekannt sein. Das Erreichen einer Zielsituation ohne vorheriges Wissen über geeignete Operatoren impliziert den Prozess des Lernens, auf welchen jedoch an dieser Stelle nicht näher eingegangen werden soll.

Aufgabe der *Planung und Entscheidungsfindung* im kognitiven Prozess ist die Auswahl von Operatoren, welche Lösungen von in sich geschlossenen Teilproblemen darstellen. Die Abarbeitung einer Menge  $T$  von Aufgaben  $t$ , welche sich aus einer Menge von Operatoren zusammensetzen, ermöglicht aufgrund der vorliegenden Situation und der gefassten Absichten ein Erreichen der relevanten Ziele.

Da der Planungsprozess nicht Gegenstand dieser Arbeit ist, soll hierauf nur kurz mit einem exemplarischen Planungsverfahren eingegangen werden, das den Ansatz der Hyperkarten-Repräsentation nutzt. Auf Basis der in der *Situationsdiagnose* bewerteten Merkmalsvektoren ist in Abbildung 37 eine Lösungssuche aus einem als *Zielkonflikt* bewerteten Gebiet der Hyperkarte in ein Gebiet mit *Zielerfüllung* dargestellt.

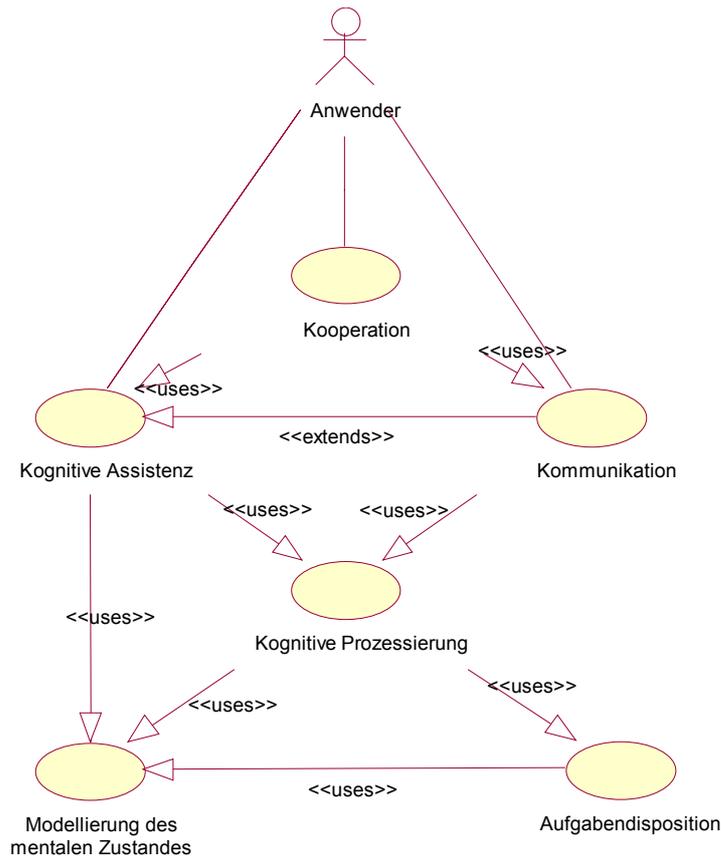


**Abbildung 37: Lösungssuche im Problemraum einer gewichteten Hyperkarte (Konturansicht mit Linien gleicher Zielerfülltheit)**

Im Teilprozess der Planung und Entscheidungsfindung gilt es, auf der Hyper-Karte einen möglichst optimalen Weg von der gegenwärtigen Ausgangs-Situation zu einer erwünschten Ziel-Situation zu ermitteln. Ergebnis der Planung stellt der Pfad in Abbildung 37 vom Ausgangszustand (Koo. 85,75) mit einer Zielerfülltheit von 0.25 zum gewünschten Zielzustand (Koo. 57,60) mit einer Erfüllung von 0.75 dar.

#### 4.3.4 AUFGABENDISPOSITION

Die Abarbeitung des Vorgehenplans  $S_{IntentPlan}$  geschieht im Einklang mit der gewonnenen *Überzeugung* durch das **Scheduling** (siehe Abbildung 38).



**Abbildung 38: Use-Case-Diagramm „Ausführung“**

Dies bedeutet, dass eine zustands- und ereignisgebundene Verarbeitung der im Vorgehenplan abgelegten Aktionen unter Berücksichtigung der hierfür notwendigen Ressourcen zu erfolgen hat. Das Ergebnis ist die momentane Aufgabensituation  $S_{Task}$ , die diejenigen Aufgaben umfasst, an deren Abarbeitung gearbeitet wird.

$$S_{Task} = Scheduling(S_{RelvObj}, S_{IntentPlan}, \text{Models of Ressources})$$

Eine Aufgabe  $t$  beinhaltet eine Menge  $A$  an Aktionen  $a$ . Diese Handlungsfolgen können durch die Nutzung bereitgestellter Ressourcen (z.B. Energie, Zeit, Material) abgearbeitet werden:

$$S_{Tasks} = \{T, A, R_{Task}\} \supseteq S_a$$

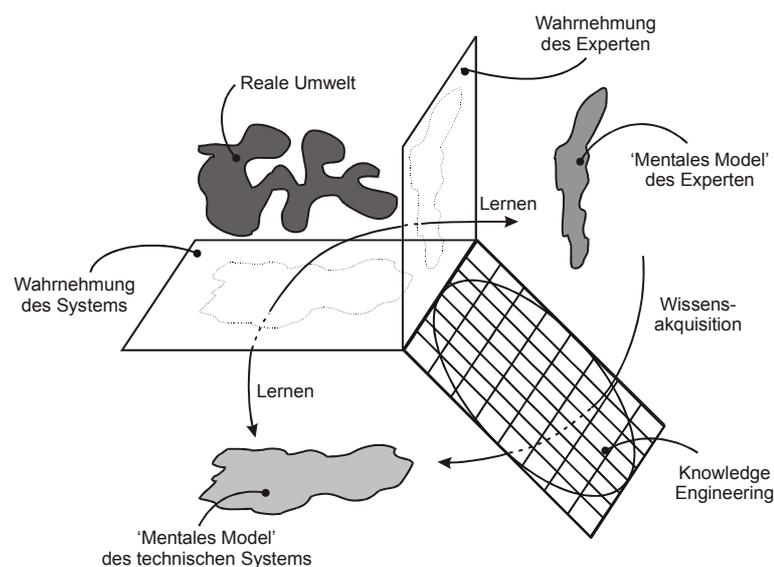
Die formale Darstellung des kognitiven Prozesses ist hiermit abgeschlossen. Es stellt sich nun die Frage, mit welchem Wissen die konzipierten Strukturen zu füllen sind.

## 4.4 WISSENERWERB

Was bedeutet es, Wissen über eine Situation zu besitzen? Welchen Quellen ist dieses Wissen zu entziehen? Wie ist die Struktur einer *zentralen, objektorientierten Situationsrepräsentation* mit diesem Wissen über die Situation der Handlungsziele zu füllen? Im Alltag sagt man, dass ein Individuum „etwas weiß“, wenn es auf Fragen bezüglich eines bestimmten Sachverhaltes hin eine Antwort geben kann, die den Fragenden zur Genüge befriedigt. Hieraus ergibt sich jedoch, dass der Wissensgrad nur subjektiv und relativ sein kann, da der Zustand der Befriedigung z.B. bei einem anderen Individuum erst nach sehr viel umfangreicheren Aussagen erreicht werden könnte. Wissen ist damit vorläufig und nicht absolut und wird oft in Tatsachen, Meinungen und Glaubensfragen unterschieden. Minsky hat Wissen in einer Form dargestellt, welche sich auf eine mentale Repräsentation der Welt stützt. Er definiert Wissen wie folgt:

*„Ein Individuum A weiß dann etwas über einen Sachverhalt S, wenn es ein mentales Modell M von S besitzt, welches hinreichend gut ist, um über S Fragen zu beantworten.“ (Minsky, 1985)*

Die Güte des Wissens scheint somit eng mit der Qualität der im Hintergrundwissen gespeicherten „*mentalen*“ Modelle zusammenzuhängen. Die Wissenserhebung hierfür geschieht für ein technisches System im Prozess des Knowledge-Engineerings (siehe Abbildung 39).



**Abbildung 39: Wissenserwerb durch Lernen und Akquisition**

Die Expertise für die aufgabenbezogenen Fähigkeiten wird hier gewonnen, welche es dann – in einer Wissensbasis vorgehalten – dem System ermöglicht, den Operateur bei seiner Aufgabe kompetent und angepasst zu unterstützen. Neben einer ausreichenden Bandbreite des Wissens in den problemspezifischen Bereichen des Systems, trägt die Qualität des Wissens entscheidend zur Unterstützungsleistung und damit zur Akzeptanz eines solchen Systems bei.

Zur Wissensakquisition zählt auch die Übertragung von Expertise in eine formale, gut strukturierte Repräsentationsform. In der Regel besteht die Aufgabe des Knowledge-Engineerings darin, menschliche Expertise auf Spezialgebieten zu erfassen und sie einem breiten Personenkreis zugänglich zu machen. Die Wissensakquisition konzentriert sich damit überwiegend auf das Wissen von Experten, da deren Wissen besonders „kostbar, da selten“ ist. Zahlreiche Verfahren und Methoden zur Wissensakquisition sind in den unterschiedlichsten Anwendungsgebieten entwickelt worden, um die Analyse, Strukturierung und Repräsentation des Expertenwissens in eine für den Rechner verständliche Form zu ermöglichen.

#### 4.4.1 PROBLEMSTELLUNG

Jede Methode zur Wissensakquisition ist mit grundlegenden Schwierigkeiten konfrontiert, deren Ursachen von Karbach auf quantitative und qualitative Aspekte der menschlichen Expertise zurückgeführt werden (Karbach, 1990)

##### 4.4.1.1 Quantitative Probleme

Die Arbeitsgebiete, in denen Experten eingesetzt werden, zeichnen sich nach (Karbach, 1990) gerade dadurch aus, dass sie sehr komplex sind und lange Einarbeitungszeiten verlangen. Entsprechend dem komplexen Anwendungsgebiet verfügt der Experte über eine quantitativ große Menge an Wissen über Konzepte und Zusammenhänge, das er durch Lernen und Erfahrung erworben hat. Zudem kann er auf eine große Anzahl von vergangenen Anwendungsfällen zurückgreifen. Schwieriger jedoch als die quantitativen Gründe sind die qualitativen Gründe.

##### 4.4.1.2 Qualitative Probleme

Nach (Karbach, 1990) gibt es keine Definition von Expertise und von dem, was Expertise ausmacht. Dementsprechend ist der Input der Wissensakquisition vage, d.h. der Knowledge Engineer und der Experte wissen nicht genau, was alles im technischen System erfasst werden muss, und wann der Abbildungsprozess vollständig durchgeführt wurde. Durch die

weitgehend unbekannte Form und Entität der Repräsentation des Expertenwissens ist es schwierig, durch Wissenserhebungstechniken das Wissen zugänglich zu machen. Zusätzlich bereitet die implizite Expertise des Experten Probleme, da sich sein Wissen nur in bestimmtem Handlungswissen, also oftmals nur in der Konfrontation mit konkreten Problemsituationen, manifestiert. Formale Methoden, mit deren Hilfe man implizites und kompiliertes Wissen in explizite Wissensrepräsentationsstrukturen überführt, sind noch unbekannt. Es gibt keine angemessenen Repräsentationsformalismen für nicht-monotone Schlussfolgerungen, Schließen über Raum und Zeit oder ‚Common Sense‘. Damit existiert noch keine adäquate Darstellung des Wissens durch Formalismen zur Repräsentation von menschlichem Wissen in der Künstlichen Intelligenz.

#### 4.4.2 METHODIKEN

Zur Extraktion von Wissen existieren zwei grundsätzlich verschiedene Ansätze.

##### 4.4.2.1 Rapid Prototyping

Anhand einer kleinen Menge von erhobenen Fallbeispielen wird ein Prototyp des Systems konstruiert, der aufgrund weiterer Erhebungen sukzessive erweitert und verfeinert wird, bis er die gewünschten Anforderungen erfüllt. Von Vorteil sind hierbei kurze Zyklen für die Validierung der Hypothesen bezüglich der Struktur von Fakten und der Inferenzstrategien sowie der Richtigkeit des Wissens. Die Prüfung der Vollständigkeit und des Systemverhaltens kann durch den Experten erfolgen. Von Nachteil ist die Bestimmung der Repräsentation der Expertise in der Wissensbasis durch die Implementierungsformalismen. Daher entspricht die Repräsentation nicht der Begriffswelt des Experten.

##### 4.4.2.2 Modellierung von Expertenwissen

Hierbei enthält der Systemprototyp implizite Modelle der Expertise. Von Vorteil ist dabei die klare Trennung von Analyse und Implementierung bei transparenter Interpretation des gewonnenen Wissens. Das erstellte Modell enthält keinerlei Implementierungsdetails und ist damit im Falle einer Neuimplementierung nicht davon betroffen. Die Analyse mündet in eine relativ vollständige Beschreibung der Expertise in einer expliziten Struktur. Das Ergebnis dient als Spezifikation für die Implementierung. Neben dem Verlust der Vorteile des Rapid-Prototypings fallen die mangelnde Überprüfbarkeit des dynamischen Ablaufs der Modelle und die fehlende formale Semantik in den Modellen als Nachteile ins Gewicht. Der modellbasierte Entwurf der Wissensbasis ist in Abbildung 40 dargestellt.

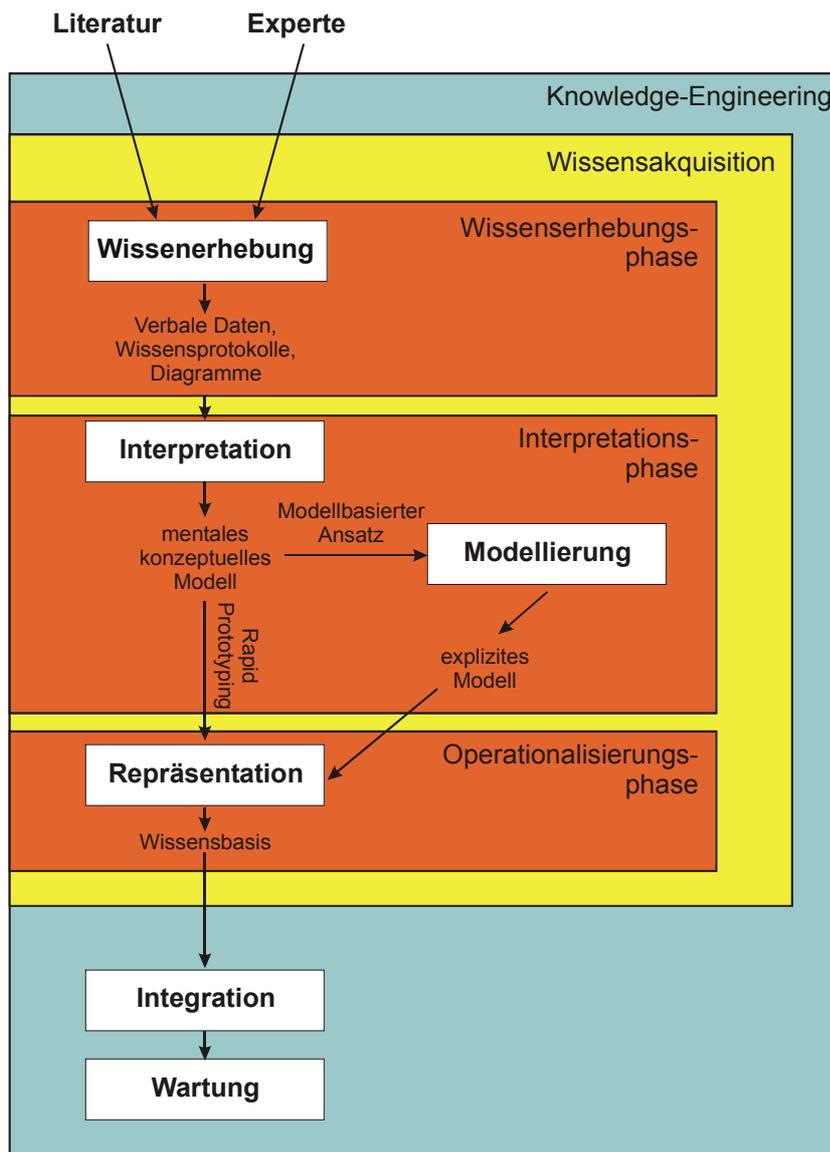


Abbildung 40: Modellbasierter Entwurf (Reimer, 1991)

Nun gilt es, für die spezifizierten kognitiven Komponenten eine geeignete Rechnerrepräsentation zu entwickeln, welche das gewonnene Wissen aufnehmen und dem kognitiven Prozess für eine Weiterverarbeitung bereitstellen kann.

#### 4.5 RECHNERREPRÄSENTATION DES SITUATIONSWISSENS

Für die Umsetzung der Repräsentation auf ein Rechnersystem sind folgende Formen denkbar (Wysotzki, 1998):

- propositional (Logik, natürlichsprachige Sätze)
- algebraisch (Graphen, Frames, semantische Netze)

- objektorientiert (Vererbungshierarchien)
- analog (bildliche oder ikonische Repräsentation)
- prozedural (Algorithmen, Programme)

Die logische Repräsentation ist die adäquate Repräsentation für Inferenzprozesse. Die algebraische Repräsentation – klassisch die Beschreibung von Objekten durch Werte-Listen (Merkmalsvektoren) – ist die Grundlage für ganzheitliche Muster- und Strukturvergleichsprozesse. Objektorientierte und algebraische Repräsentationsformen verfügen jede für sich über Eigenschaften, welche für eine Repräsentation von kognitivem Situationswissen benötigt werden. Die Beschreibung der Situationselemente durch Modelle in Form von Objekten und deren Organisation in relationalen Strukturen verspricht diejenigen Eigenschaften in einer *zentralen, objektorientierten Situationsrepräsentation* zu vereinen, die es für eine effiziente und der Realität entsprechende Art und Weise bedarf.

Brachmann und Levesque verbinden mit dem Begriff der Wissensrepräsentation:

*„... das Aufschreiben von Symbolen – Repräsentationsstrukturen genannt –, die in einer erkennbaren Weise einem Ausschnitt einer zu repräsentierenden Welt entsprechen.“ (Brachmann, 1979)*

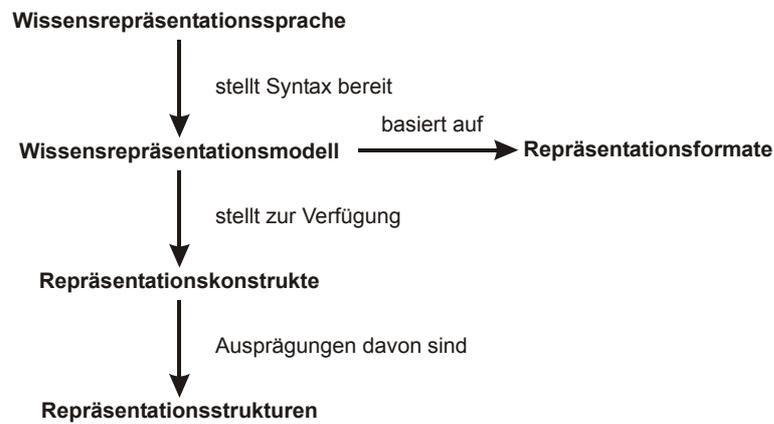
Reimer ergänzt diese Definition noch um den Aspekt der Wissensinterpretation, welche eine Vorschrift zur Formulierung und Auswertung von Anfragen (optional auch von Änderungsoptionen) auf diese Struktur umfasst. Damit steht eine Repräsentation stellvertretend für eine Menge von Sachverhalten – repräsentierte Welt genannt – und ist somit ein Modell für diese.

Hieraus definiert Reimer den Begriff der Repräsentation wie folgt:

*„Von einer Repräsentation sprechen wir, wenn zusätzlich zu einer Menge von Repräsentationsstrukturen Angaben dazu vorliegen, wie die Strukturen der Repräsentation auf die Merkmale der repräsentierten Welt abzubilden sind. Die Angaben stellen die Interpretationsvorschrift dar.“ (Reimer, 1991)*

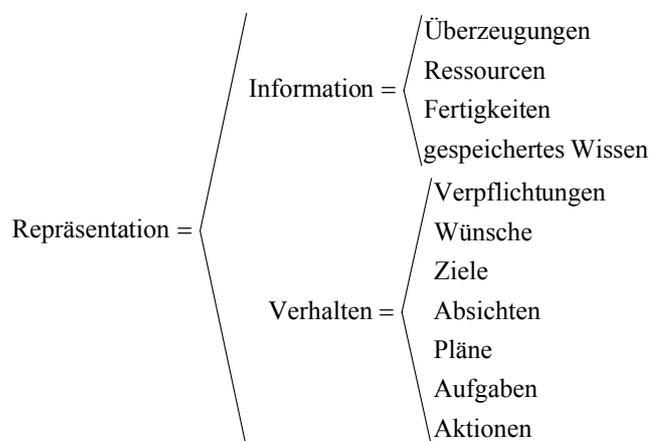
Repräsentationsstrukturen können explizit oder implizit sein, wobei bei der impliziten Repräsentation der Sachverhalt erst noch ermittelt bzw. berechnet werden muss, was man Inferenzieren bzw. Schlussfolgern nennt. Gleichartige Repräsentationsstrukturen (Klassen)

werden in Repräsentationsformaten zusammengefasst. Das Wissensrepräsentationsmodell umfasst ein oder mehrere Repräsentationsformate. Die Erfassung dieser Repräsentationskonstrukte in einem Wissensrepräsentationsmodell erfolgt in der Regel über eine Wissensrepräsentationssprache, die eine Syntax hierzu bereitstellt (siehe Abbildung 41).



**Abbildung 41: Begriffsübersicht der Wissensrepräsentation (Reimer, 1991)**

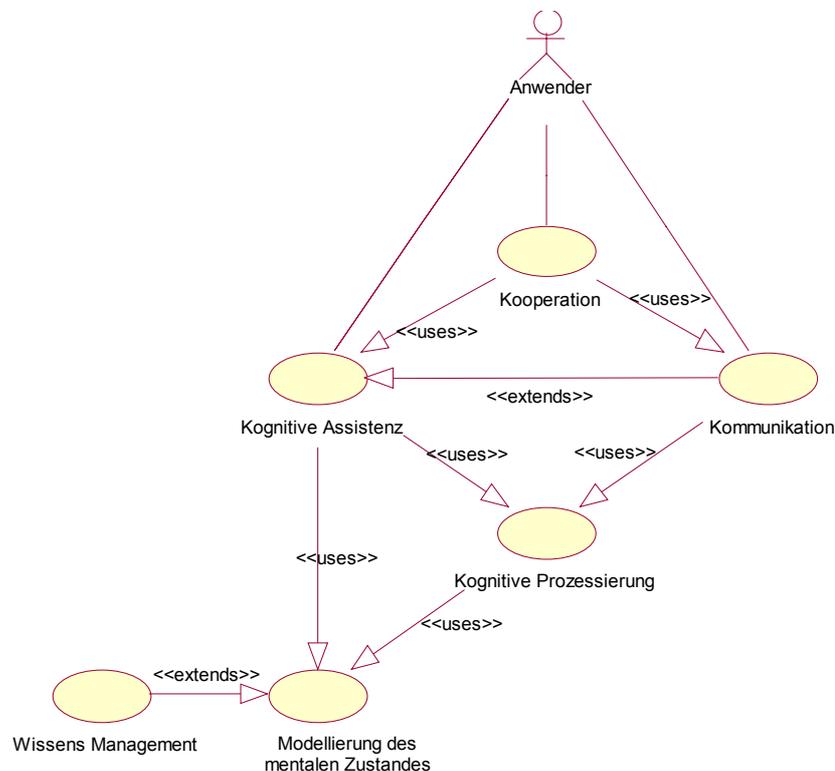
Operationelle Modelle ermöglichen neben der Modellierungsarbeit das Prototyping von wissensbasierten Systemen. Karbach (Karbach, 1990) nennt darüber hinaus als Vorteil einer Wissensrepräsentationssprache den festgelegten Rahmen für die Akquisition und Formulierung von Wissen. Als Nachteil identifiziert er die mangelnde Flexibilität bei der Abbildung von Expertenwissen und die Erweiterungs- und Anpassungsmöglichkeiten der Modelle. Die Repräsentation des Wissens für ein kognitives System kann gemäß Abbildung 42 in die Klassen der statischen Informationen und die dynamischen, das Verhalten des Systems hervorrufenden, Bereiche strukturiert werden.



**Abbildung 42: Kognitive Repräsentationsklassen**

#### 4.5.1 WISSENSBASIS DER SITUATIONSELEMENTE

Für die Repräsentation des Wissens, welches im kognitiven Prozess verarbeitet werden soll, wurden im Rahmen dieser Arbeit eine individuelle Repräsentationsform und eine Verwaltungsordnung (siehe Abbildung 43) erarbeitet.



**Abbildung 43: Use-Case Diagramm „Wissensmanagement“**

Die Wissensbasis stützt sich auf eine Repräsentationssprache, in der das erworbene Expertenwissen abgelegt ist, und dient dem kognitiven System als Quelle für die verwandten *Situationselemente*. Bei der Entwicklung dieser Sprache war es das Ziel, einen möglichst hohen Anwendungsbezug zum kognitiven Prozess zu gewährleisten, jedoch Unabhängigkeit bezüglich der Domäne zu erreichen. Hieraus ist eine einfache, aber ausreichende Sprache zur Strukturierung der kognitiven Komponenten hervorgegangen. Für die Repräsentation von Situationswissens sind zahlreiche Einzelkonzepte vonnöten, die sich jedoch aus den kognitiven Komponenten (siehe Kapitel 4.3) ableiten lassen. Diese sind in einem Wissensmodell repräsentiert und gliedern sich in die separaten Konzepte Relationen, Relevante-Objekte und Ziele. Die Darstellung der Wissensrepräsentationssprache erfolgt in der BNF (Bachus-Naur-Form). Die Wissensrepräsentation stellt sich in folgendem Aufbau dar:

Situationselemente ::= Relationen  
 Relevante-Objekte  
 Ziele

#### 4.5.1.1 “Relationen”

Relationen stellen die Beziehungen zwischen Situationselementen dar. Relationen können nach Barwise und Perry (Barwise, 1987) n-stellig sein. Von besonderer Art sind die nullstelligen, einstelligen und mehrstelligen Relationen. Relationen unterscheiden sich durch ihren Typ. Der Typ oder die Art einer Relation gibt somit eine Aussage über

- das Raum-Zeit-Gebiet (bei nullstelligen Relationen),
- die Eigenschaft eines Situationselements (einstellige Relationen) und
- die Beziehungen zwischen den Situationselementen (mehrstellige Relationen).

Das Konzept einer Relation wird im hier verwandten Entwurf durch ein Text-Attribut dargestellt. Relationen beginnen mit dem Schlüsselwort `RELATION`. Relationen können beliebig oft in der Wissensbasis verwandt werden, müssen aber eindeutig sein:

Relationen ::= [ **RELATION** Name-der-Relation ]+  
 Name-der-Relation ::= *Name*

#### Beispiele

**RELATION** istTeilVon;  
**RELATION** istGestartet;  
**RELATION** kommuniziertMit;  
**RELATION** istFundamentalzielVon;  
 u.s.w.

#### 4.5.1.2 “Relevante-Objekte”

Relevante-Objekte stellen *explizite „mentale“ Modelle* von Weltobjekten dar. Sie sind gemäß Kapitel 4.3.1 durch Hauptmerkmale ***P*** und dem Objekt zugeordnete Eigenschaften ***F*** charakterisiert. Die Hauptmerkmale stellen allgemein betrachtet Situationsmerkmale dar und bilden die atomaren Bausteine der Situation. Die Hauptmerkmale ***p*** sind durch einen eindeutigen Namen und ein Datum, welches kontinuierlich, diskret oder konstant sein kann, bestimmt. Kontinuierliche Attribute werden über einen Wertebereich [***p<sub>min</sub>***, ***p<sub>max</sub>***] und die Diskretisierungsstufen ***sets(p)*** definiert.

```

Hauptmerkmal           HAUPTMERKMAL Name Merkmalstyp
Merkmalstyp           ::=Kontinuierlich Name
                       | Diskret Name
                       | Konstant Name
                       | Name

Kontinuierlich         ::= KONTINUIERLICH
                       Beschreibung-Kontinierlicher-Wert

Beschreibung-Kont.-Wert ::= [ Real , Real ; Real ]

Diskret                ::= DISKRET Beschreibung-Diskreter-Wert

Beschreibung-Diskreter-Wert ::= [ Diskrete-Wertemenge ]

Diskrete-Wertemenge   ::= Real
                       | Diskrete-Wertemenge , Real

Beschreibung-Konstanter-Wert ::= [ Real ]

Konstant               ::= KONSTANT
                       Beschreibung-Konstanter-Wert

```

### Beispiele

```

HAUPTMERKMAL Longitude   KONTINUIERLICH [7.0,15.0; 0.1] "deg";
HAUPTMERKMAL Latitude   KONTINUIERLICH[45.0,50.0; 0.1] "deg";
HAUPTMERKMAL Flaps      DISKRET [-6,1,5,14,35] "deg";
HAUPTMERKMAL MinClearAlt KONSTANT [500] "feet";
HAUPTMERKMAL Distance   KONSTANT [0.0, 10000.0; 0.2] "NM";
HAUPTMERKMAL DistToWx   Distance;

```

u.s.w.

Relevante-Objekte sind Modelle, die aus Hauptmerkmalen zusammengesetzt sind. Das Modell eines Relevanten-Objektes ist somit durch die Hauptmerkmale repräsentiert. Die Objekteigenschaften, also die Operationen, welche auf das Objekt angewandt werden können, sind uniform und werden dem Objekt erst durch die Instanziierung während der Laufzeit im Rechner zugewiesen. Sie bedürfen daher keiner Spezifikation innerhalb der Wissensrepräsentationssprache.

```

Relevante-Objekte ::= RELVEVANTES_OBJEKT Name
                   Objekt-Hauptmerkmale
Objekt-Hauptmerkmale ::= HAUPTMERKMALE { Hauptmerkmale }
Hauptmerkmale ::= Merkmal+
Merkmal ::= -> Name
           | -> Name [ Real ]
           | <-> Name
           | <-> Name [ Real ]

```

### Beispiel

```

RELVEVANTES_OBJEKT Position
HAUPTMERKMALE { -> Altitude ;
                 -> Longitude;
                 -> Latitude
               }
RELVEVANTES_OBJEKT FlpLeg
HAUPTMERKMALE { -> Position ;
                 -> Position
               }
RELVEVANTES_OBJEKT Flightplan
HAUPTMERKMALE { -> FlpLeg [ 10 ]
               }

```

#### 4.5.1.3 „Ziele“

Die Repräsentation der Ziele stellt die Beschreibung derjenigen Modelle dar, welche im Rahmen der *expliziten „mentalen“ Modellierung* des kognitiven Prozesses die höchste Komplexität besitzt. Hier sind zahlreiche Unterkonzepte vonnöten, um ein Modell zu repräsentieren. Zunächst besteht das Konzept eines Ziels aus einer informellen Beschreibung Ziel-Identifikation, einem Ziel-Gewicht zwecks Priorisierung, einer Beschreibung des Relevanzkriteriums des Ziels und einem sogenannten Ziel-Kern. Zielmodelle können sich durch deren Parametrisierbarkeit unterscheiden. Parametrisierbare Ziele werden bei der Instanziierung zunächst mit Ziel-Parametern der momentanen Situation skaliert, bevor sie aktiviert werden.

```

Ziele ::= [ Ziel-Identifikation
           Ziel-Gewicht
           Relevanzkriterium
           Zielkern]+
       | [ Ziel-Identifikation
           Ziel-Gewicht
           Relevanzkriterium
           Ziel-Parameter
           Zielkern]+

```

Die Ziel-Identifikation umfasst einen eindeutigen Identifikator, hier ist das der Name des Ziels:

```

Ziel-Identifikation ::= ZIEL Name

```

Die Gewichtung eines Ziels erfolgt durch die Angabe der Wichtigkeit Ziel-Gewicht in Prozent:

```

Ziel-Gewicht ::= GEWICHT Real

```

Wie bereits in Kapitel 4.3.2.1 dargestellt, erfolgt die Repräsentation einer Situation über relationale Strukturen. Die Relevanz eines Ziels ist somit vom Inhalt, der Struktur und der Semantik der Situation abhängig, wodurch eine geeignete Situationsbeschreibung hierfür definiert werden muss. Diese muss eine Überprüfbarkeit auf Übereinstimmung zwischen einer Ausgangs-Situation und einer relevanten Ziel-Situation ermöglichen. Ausschlaggebend für diese Übereinstimmung sind somit Relationen und Situationselemente der relationalen Struktur, die damit Charakteristika der Situation darstellen. Dies entspricht der in Kapitel 4.1 vorgestellten Metrik des Situationsverständnisses. Die hierfür entwickelte Nomenklatur basiert auf sogenannten Tupeln aus Relevanten-Objekten und Relationen. Sie ist nachfolgend dargestellt:

```

Relevanzkriterium ::= RELEVANT_WENN Situations-Tupel+

```

```

Situations-Tupel ::= < Relations-Name >
                  | < Objekt-Name >
                  | < Relations-Name:Objekt-Name >
                  | < Relations-Name:Objekt-Name [,Objekt-Name]+ >

```

Nullstellige Relationen beschreiben Raum-Zeit-Gebiete durch ein <Relations-Name> Tupel. Mehrstellige Situations-Tupel beschreiben Teilmengen von Raum-Zeit-Gebieten und beinhalten Relationen und Situationselemente in Form von <Relations-Name> und

<Objekt-Name> Listen. Eigenschaften bzw. Zustände von Situationselementen werden durch <Relations-Name:Objekt-Name> Tupel repräsentiert. Zwei- oder mehrstellige Tupel <Relations-Name:Objekt-Name,Objekt-Name,...> repräsentieren die Beziehung zwischen zwei oder mehreren Situationselementen.

Der Ziel-Kern besteht aus dem Inferenzwissen des Ziels (Beispiel siehe Anhang 10.4). Dieses dient der Bewertung einer vorgegebenen Ausgangs-Situation bezüglich deren Erfülltheit gegenüber dem Ziel. In der Regel sind die Inferenzmechanismen mit dem Wissen aus der Domäne und der Expertise der Operateure zu versehen. Der Transfer des Inferenzwissens in eine maschinelle Struktur bedarf einer gründlichen Überlegung und der Berücksichtigung sowohl der Quelle als auch der Interpretation des Wissens auf Seiten der Maschine. Um den Anforderungen an eine hinreichend genaue Modellierung gerecht zu werden, wird für die Beschreibung des Inferenzwissens auf heterogene Wissensrepräsentationsformen zurückgegriffen. Dabei werden vier verschiedene Repräsentationsformen verwandt, welche für die Repräsentation des erhobenen Wissens geeignet erschienen: Diagramme und algebraische Funktionen, Fuzzy-Regler, Karten und freie Informationen. Die Nutzung von Neuronalen Netzen erscheint in zukünftigen Konzepten hilfreich. Die Anwendung dieser Wissensformen richtet sich zum einen nach der Wissensquelle, zum anderen nach der Klasse der Ziele, welche es zu modellieren gilt.

```
Ziel-Kern      ::= INFERENZ_FUNKTION { Funktion }
                | INFERENZ_FUZZY_LOGIK { Fuzzy }
                | INFERENZ_KARTE { Karte }
                | INFERENZ_FREIE_FORM { Freie-Form ; }
                | INFERENZ_NEURONALES_NETZ { Neuronales-Netz }
```

#### 4.5.1.3.1 Funktionen und Diagramme

Zahlreiche natürliche oder technische Prozesse sind unter der idealisierten Betrachtung von Nebenbedingungen mathematisch erfassbar und daher einfach in einem Rechnersystem zu modellieren. Jedoch werden Prozesseigenschaften auch häufig in Diagrammen dokumentiert, welche das Verhalten einer Komponente oder eines Teilsystems darstellen. Die physikalischen Zusammenhänge und das damit verbundene funktionale Wissen um den Teilprozess gehen jedoch verloren. Während mathematische Funktionen die Bestimmung des Funktionswertes durch Lösen der Gleichung ermöglichen, muss dies bei Diagrammen über den Weg der Interpolation erfolgen.

#### 4.5.1.3.2 Fuzzy-Logik

Expertenwissen liegt im Gegensatz zur präzisen, meist quantitativ orientierten mathematischen Beschreibung natürlicher oder technischer Prozesse, oftmals in einer Beschreibung durch unscharfe Merkmale vor. Die Theorie der Fuzzy-Logik hilft, solche unscharfen Beschreibungen einer mathematischen Modellierung zugänglich zu machen, indem so genannte *linguistische Variablen* bestimmt werden. *Linguistische Variable* sind durch *linguistische Terme*, definiert über eine Zugehörigkeitsfunktion, belegt. So könnte die Variable *Geschwindigkeit* durch die Terme *sehr klein*, *klein*, *mittel*, *groß* und *sehr groß* unterteilt sein. Der Vorteil einer Wissensrepräsentation durch linguistische Variable und Fuzzy-Logik liegt im schnellen Entwurf von Grobmodellen mit Methoden der Theorie der unscharfen Mengen.

#### 4.5.1.3.3 Karten

Karten stellen abstraktes Wissen über definierte Merkmale in einem endlichen Raum dar. Im Gegensatz zu den Diagrammen ist hier nicht mittels Interpolation zu inferenzieren, sondern mit der Extraktion von Merkmalsmengen und deren Interpretation zu höherem Wissen. Karten stellen analoges Wissen dar, welches implizite, höhere Wissens Elemente enthält.

#### 4.5.1.3.4 Freie Wissensform

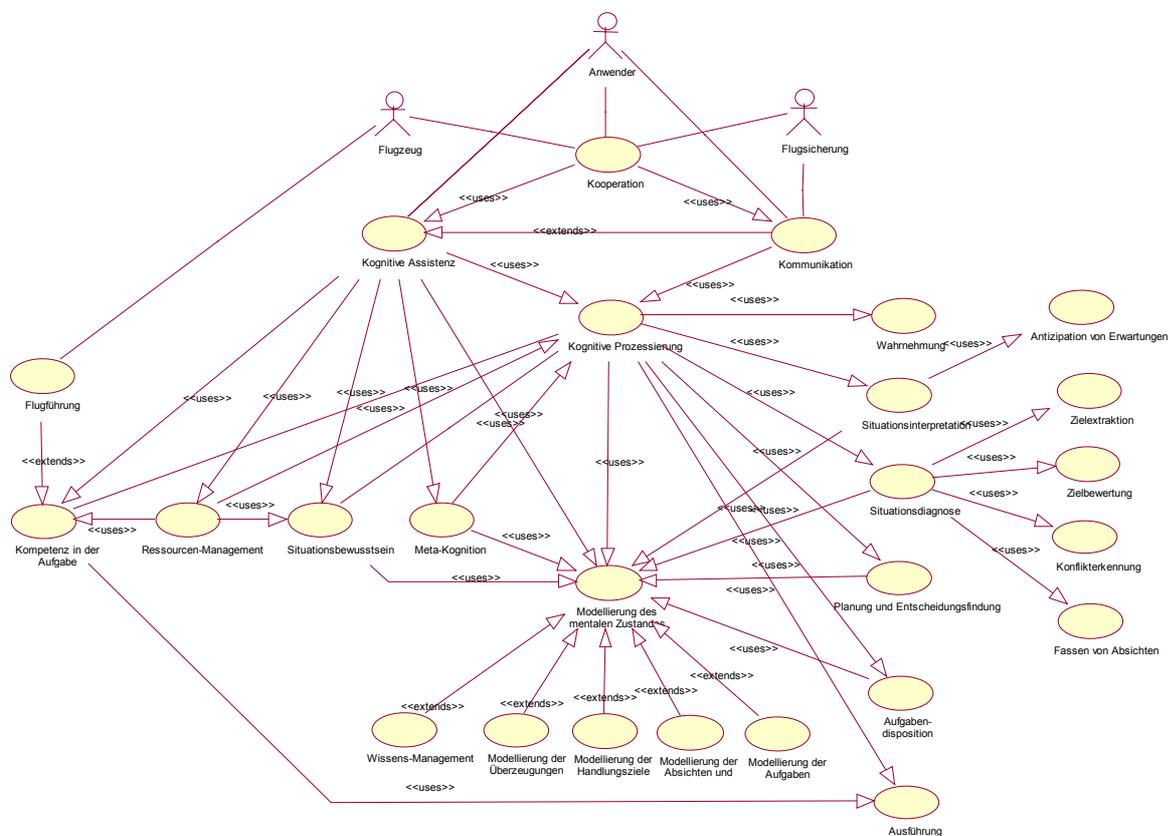
Expertenwissen, welches keiner der oben genannten Modellrepräsentationen zugeordnet werden kann, soll hier in einer freien Wissensform abgelegt werden. Insbesondere das Problem der Menge der unklaren Ziele soll durch diese Repräsentationsform berücksichtigt, und damit eine vollständige Zielmenge dem System zu Grunde gelegt werden.

#### 4.5.1.3.5 Neuronale Netze

Ein in dieser Arbeit nicht weiter verfolgter Ansatz soll an dieser Stelle kurz gestreift werden. Neuronale Netze zur Situationsklassifikation können große Teile des impliziten Wissens repräsentieren und somit auch das derzeit in der freien Wissensform abgelegte Wissen einer maschinellen Nutzung zugänglich machen. Sie dienen einer expliziten Repräsentation von gelerntem Situationswissen und könnten Teile der freien Wissensform und somit unklare Ziele darstellen und dadurch für ein technisches System nutzbar machen.

## 4.6 GESAMTSPEZIFIKATION

Aus den erarbeiteten Anforderungen und den zusammengetragenen Grundlagen lässt sich aus der domänenunabhängigen Grundspezifikation nun der Anwendungsfall eines kognitiven Cockpitassistenzsystems ableiten. Hierzu ist die Anwendung um die beteiligten Akteure „Aircraft“ und „Air Traffic Control“ (vergleiche Abbildung 10) sowie um eine Spezialisierung der „Task Competence“ durch die „Flight-Guidance“-Funktionalität zu erweitern. Die hier verwandte Gesamtspezifikation der Mensch-Maschine-Kooperation in der Flugführung für ein kognitives Cockpitassistenzsystem ist im Use-Case-Diagramm der „Mensch-Maschine-Kooperation in der Flugführung“ (siehe Abbildung 44) dargestellt.



**Abbildung 44: Use-Case-Diagramm „Mensch-Maschine-Kooperation in der Flugführung“**

## 5 IMPLEMENTIERUNG

Der Analyse der Anforderungen und der Systemspezifikation folgt die Implementierung der Softwarekomponenten. Hierbei steht die Anforderung an erster Stelle, die innere Konsistenz des kognitiven Systems zu verbessern,.

### 5.1 SOFTWARE-ARCHITEKTUR

Die Umsetzung der hier erarbeiteten theoretischen Erkenntnisse auf ein Rechnersystem ist Schwerpunkt im Fachgebiet der *Künstlichen Intelligenz* (KI). Die KI beschäftigt sich mit der Untersuchung von Ideen, die es der Maschine ermöglichen, sich intelligent zu verhalten. Dabei stützt sich die klassische KI gemäß (Brenner, 1998) auf die Schwerpunkte

- geschlossene Systeme (geringe Interaktion mit der Umwelt),
- Systeme, die in der Regel sehr komplexes, tiefgehendes Wissen innerhalb eines eng begrenzten Fachgebietes beinhalten und
- fest definierte Mechanismen zur Wissensrepräsentation mit Komponenten überwiegend statischer Natur mit bekannter Problemlösung.

Die *Verteilte Künstliche Intelligenz* (VKI) versucht, die Defizite der klassischen KI auszugleichen. Sie beschäftigt sich mit dem Entwurf verteilter Systeme und den damit auftretenden Fragestellungen. Sie umfasst die Teilgebiete der

- Parallelen KI,
- Verteiltes Problemlösen und
- Multi-Agenten Systeme.

Letztere sollen hier diskutiert und ihre Anwendung für eine *zentrale, objektorientierte Situationsrepräsentation* vorgestellt werden.

#### 5.1.1 MULTI-AGENTEN-SYSTEM

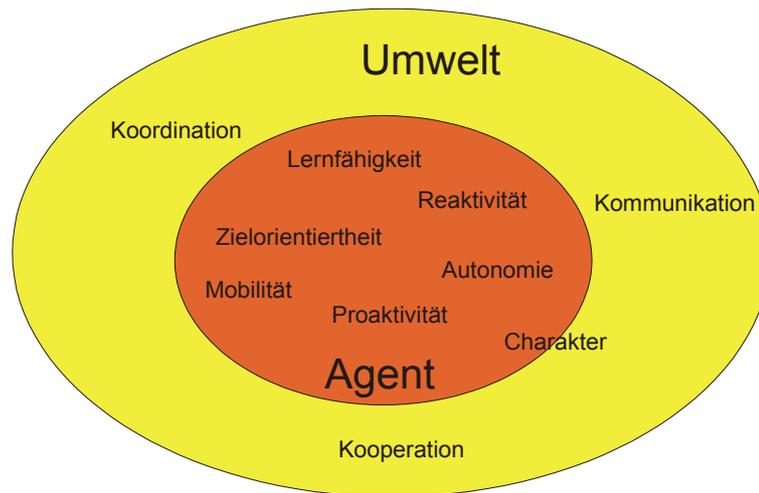
Die Implementierung einer *zentralen, objektorientierten Situationsrepräsentation* ist eng mit der Systemarchitektur eines kognitiven Assistenzsystems verbunden. Die Systemarchitektur ist diejenige Instanz eines Systems, die für die Bereitstellung und das Management der

Teilkomponenten – hier Agenten genannt – verantwortlich ist. Agenten können auf verschiedenen Architekturen aufgebaut sein. Deliberative Agenten wie die „Belief, Desire and Intention“-Agenten (BDI) (siehe Rao, 1991) unterscheiden sich z.B. von anderen Agentenarchitekturen durch ein explizites, symbolisches Modell der Welt. Überzeugungen, Wünsche, Ziele, Intentionen und Pläne stellen den Kern dieses Agententyps dar (siehe Rao, 1991) und repräsentieren deren internen mentalen Zustand.

Softwarearchitekturen besitzen gewöhnlich unterschiedliche Fähigkeiten und Eigenschaften. Die Auswahl bezüglich der Fähigkeiten einer Softwarearchitektur resultiert aus bekannten methodischen Annahmen. Die Domäne und Umgebung, in der die Architektur verwandt wird, führen ebenfalls zu charakteristischen Eigenschaften derselben. Für die Entwicklung einer *zentralen, objektorientierten Situationsrepräsentation* für ein kognitives System trifft dies ebenfalls zu. Die heterogene Struktur kognitiver Systeme und ihre vielschichtigen Komponenten führen beim Systemdesign zu Ansätzen, die sich auf die Prinzipien der *Verteilten Systeme (VS)* stützen. *Verteilte Systeme* besitzen räumlich verteilte Komponenten ohne gemeinsamen Speicherbereich. Ihre Verwaltung ist dezentral. Für die Entwicklung verteilter Systeme lassen sich Ursachen wie die Leistungsexplosion bei Prozessoren, Entwicklung schneller Netze, Fortschritte in der Softwaretechnik und Abkehr von hierarchischen Strukturen anführen. *Verteilte Systeme* bieten gegenüber Zentralrechnern Vorteile (Linnhof-Popien, 1998) wie

- Realisierbarkeit von stetigen Anpassungen der Systemgröße,
- Erweiterbarkeit,
- Expansion der Nutzungsmöglichkeit,
- Anpassbarkeit der Systemlast,
- Vereinfachung der Organisation und
- Autonome Einzelkomponenten zur Erhöhung der Robustheit.

Verteilte Systeme nutzen immer öfter die Technologie der intelligenten Softwareagenten. Intelligente Softwareagenten besitzen mannigfaltige Eigenschaften (siehe Abbildung 45).



**Abbildung 45: Charakteristiken von Intelligenzen Softwareagenten (Brenner, 1998)**

Die VKI erweitert das existierende Konzept der *Single-Agenten* um die Fähigkeiten der Kommunikation und Interaktion. Multi-Agentensysteme können dadurch Probleme lösen, welche bei der Entwicklung des Systems noch nicht vorherzusehen waren. Sie sind nicht für eine spezielle Aufgabe, sondern allgemein für das Lösen von Problemen konzipiert. Die Schwerpunkte im Bereich der Multi-Agentensysteme liegen in der Entwicklung von

- Mechanismen zur Konflikterkennung und Konfliktlösung,
- Erzeugung widerspruchsfreien Zielsystemen und
- dem strategischen Verhalten der Agenten sowie
- Organisationsstrukturen,
- Problemlösungsstrategien und
- Kooperations- und Koordinierungsmechanismen.

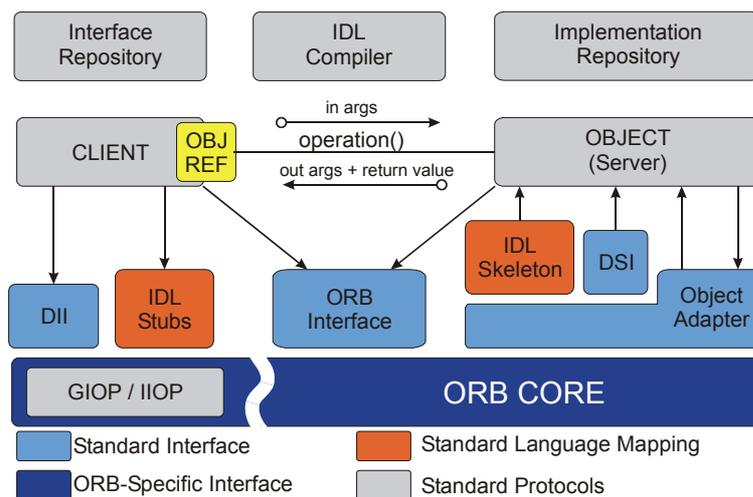
Für die Realisierung *Verteilter Systeme* bedarf es geeigneter Softwarearchitekturen und für diese geeignete Infrastrukturen (Middleware, Verteilungsinfrastruktur oder das Netzbetriebssystem). Für ein kognitives Assistenzsystem ist die Veranlagung zur Kooperation in solchen *Verteilten Systemen* vorzusehen. Sogenannte Client/Server-Systeme sind typische Vertreter von *Verteilten Systemen*.

Bei der Auswahl einer geeigneten Software-Infrastruktur gibt es einige prinzipiell verschiedene Ansätze, wobei jedoch keine allgemeine Empfehlung gegeben werden kann. Den meisten Ansätzen ist gemein, dass die Entwicklung einer Anwendung mit der

Spezifikation der Schnittstellen beginnt und danach die Implementierung der Server und Clients erfolgt. Für die hier vorliegende Arbeit fiel die Wahl auf CORBA (Common Object Request Broker Architecture). Bei CORBA handelt es sich um eine Spezifikation eines Standards zur Kommunikation in *Verteilten Systemen*. Es existieren hierfür zahlreiche Implementierungen.

### 5.1.2 COMMON OBJECT REQUEST BROKER ARCHITECTURE

Die Common Object Request Broker Architecture (siehe Abbildung 46) ist eine Spezifikation, die von der Object Management Group (OMG) entwickelt wird. Die OMG ist ein Konsortium der Firmen Digital, SunSoft, AmericanAirlines, Canon, Philips Telecommunications und Hewlett Packard. Sie wurde im Jahre 1989 mit dem Ziel gegründet, unter Verwendung der Objekttechnologie die Interoperabilität von Anwendungen in heterogen verteilten Umgebungen zu ermöglichen.



**Abbildung 46: CORBA – Common Objekt Request Broker Architektur**

Die verschiedenen separaten Standards der OMG umfassen eine Reihe von Komponenten, welche in der Object Management Architecture auf vier Kernkomponenten basieren. Zentraleinheit ist der Object Request Broker (ORB), der die Kommunikation zwischen den Objekten koordiniert. Der ORB selbst besteht aus diversen Komponenten. Hierzu zählen das Dynamic Invocation Interface (DII), das Dynamic Skeleton Interface (DSI) und der Object Adapter (OA), welche dem Entwickler in Form von Bibliotheken bereitgestellt werden.

### 5.1.3 IDENTIFIKATION VON KLASSEN IN CORBA

Mit der Identifikation von Klassen sind Überlegungen bezüglich der inneren Zustände, dem Verhalten und der Identität von CORBA-Objekten verbunden. Das Objektverhalten bestimmt, in welcher Art und Weise ein CORBA-Objekt auf Anfragen von anderen CORBA-Objekten reagiert und versinnbildlicht all das, was ein CORBA-Objekt tun kann. Eine Klasse beschreibt eine Menge von gleichartigen CORBA-Objekten, welche gemeinsame Merkmale, ein gemeinsames Verhalten, gemeinsame Beziehungen zu anderen CORBA-Objekten und eine gemeinsame Bedeutung haben. Hier soll zunächst die Darstellung der Schnittstellen zu den CORBA-Klassen erfolgen.

Die IDL bietet dem Entwickler die Möglichkeit, für die CORBA zunächst implementierungsunabhängige Datenstrukturen und Schnittstellen für die CORBA-Objekte zu spezifizieren. Für die Umsetzung des IDL-Skripts in eine Programmiersprache existieren je nach verwendetem ORB verschiedene Compiler. Diese setzen die Spezifikationen in Softwarecodes (z.B. C, C++, Java) um. Für die Gestaltung der Schnittstellen der CORBA-Objekte im ORB wurde hier in Anlehnung an die Sprechakttheorie (siehe Searle, 1969), welche für die Kommunikation in Multi-Agentensystemen bereits Verbreitung gefunden hat, ein Client-Server-Kommunikationskonzept (siehe Abbildung 47) realisiert.

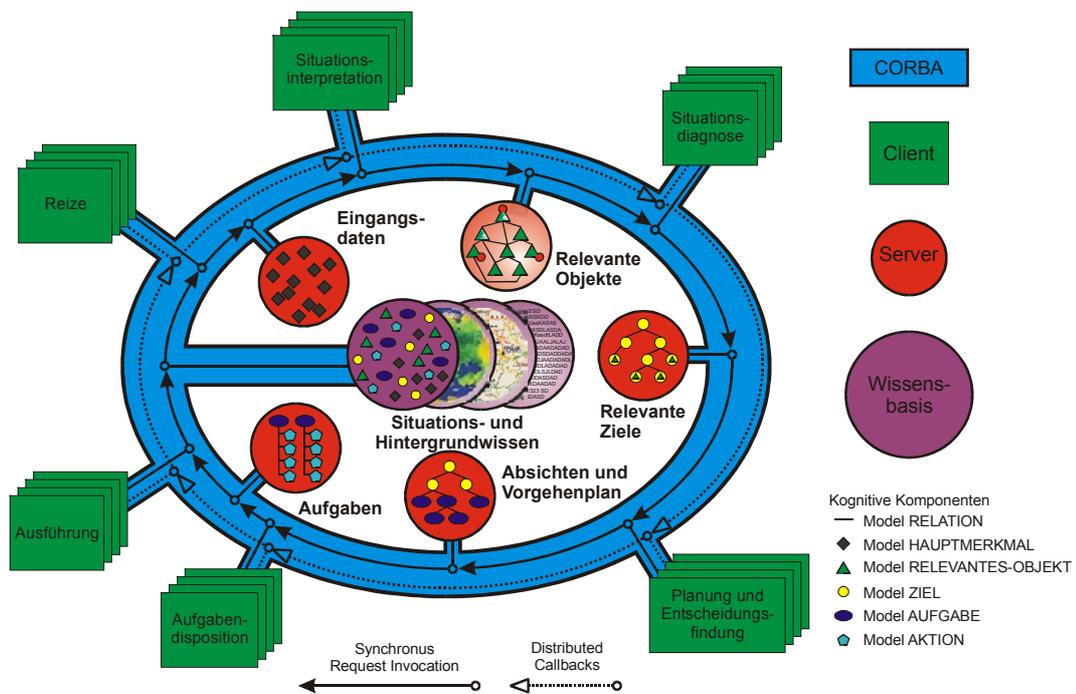


Abbildung 47: Client-Server-Kommunikationskonzept am Beispiel des kognitiven Prozesses (Walsdorf, 2000)

Sogenannte *cooperation-primitives* (Lux, 1993), welche aus *co-operation types* und *co-operation objects* bestehen, dienen zur Kommunikation zwischen den Situationselementen oder deren Repräsentationsformen bezüglich unterschiedlicher Informationen wie Ergebnisse von Funktionsaufrufen, Parametern oder anderen Wissensentitäten zwischen intelligenten Softwareagenten. Diese sind in Tabelle 2 dargestellt.

Object/Type	propose	accept	reject	refine	modify	inform	query
Intention	—	—	—	—	—	•	—
Goal	•	•	•	—	—	—	—
Plan	•	•	•	•	•	—	—
Commitment	•	•	•	—	—	—	—
Goal breakdown	•	•	•	•	•	•	•
Org. structure	•	•	•	•	•	—	—
Schedule	•	•	•	•	•	•	•
Task assignment	•	•	•	•	•	—	—
Resource allocation	•	•	•	•	•	—	—
Untyped Info	—	—	—	—	—	•	•

**Tabelle 2: Cooperation Primitives der Sprechakttheorie (Haugeneder, 1998)**

Durch die Umsetzung der Konzepte der Sprechakttheorie zur Kommunikation innerhalb der hier erstellten *zentralen, objektorientierten Situationsrepräsentation* ist eine klare und nachvollziehbare Kommunikation in der komplexen Implementierung möglich. Nicht zuletzt bietet die Typisierung der Kommunikationsschnittstellen ein Potential für die Erweiterung des Gesamtsystems. Die Umsetzung dieser Erkenntnisse in der IDL erfolgt in der Spezifikation durch

- CORBA-Strukturen und
- CORBA-Objekt-Schnittstellen

unter Berücksichtigung der für die Kommunikation zwischen den verschiedenen „mentalen“ Repräsentationsformen charakteristischen Bedürfnisse. Zunächst soll hier der Entwurf der *co-operation types* beschrieben werden. Zur Umsetzung der *zentralen, objektorientierten Situationsrepräsentation* sind zunächst Vorlagen für die Objekte der

- Hauptmerkmale,
- Relevanten-Objekte und
- Ziele

zu entwerfen.

### 5.1.3.1 CORBA-Strukturen

Die Relevanten-Objekte sind durch die CORBA-Struktur *Relevante-Objekte* und deren *Hauptmerkmale* beschrieben, die wiederum in einen informationstragenden (*Hauptmerkmal Wert*) und einen informations-beschreibenden (*Hauptmerkmal Beschreibung*) Teil zerfallen. Die Merkmalsinformation, eine real-wertige Größe, kann alternativ in einer Union als konstante (*Konstantes Hauptmerkmal*), als diskrete (*Diskretes Hauptmerkmal*) oder als kontinuierliche (*Kontinuierliches Hauptmerkmal*) Größe abgelegt sein. Zu deren Interpretation gibt die Merkmalsbeschreibung *Hauptmerkmal Beschreibung* Auskunft über deren Identifizierungsnamen und deren physikalische Maßeinheit. Die Modellierung einzelner Relevanter-Objekte erfolgt hier vereinfacht als reine Zustandsbeschreibung durch Merkmalsmengen (*Hauptmerkmale*). Der Aufbau der Templates für die Hauptmerkmale und Relevanten-Objekte ist in Abbildung 48 dargestellt.

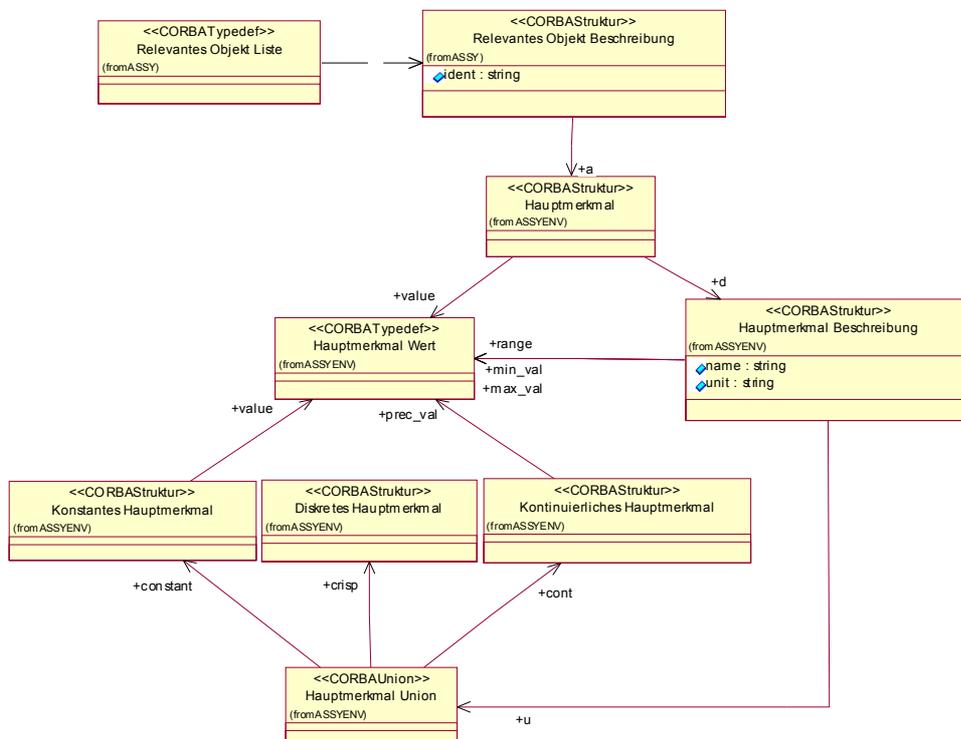
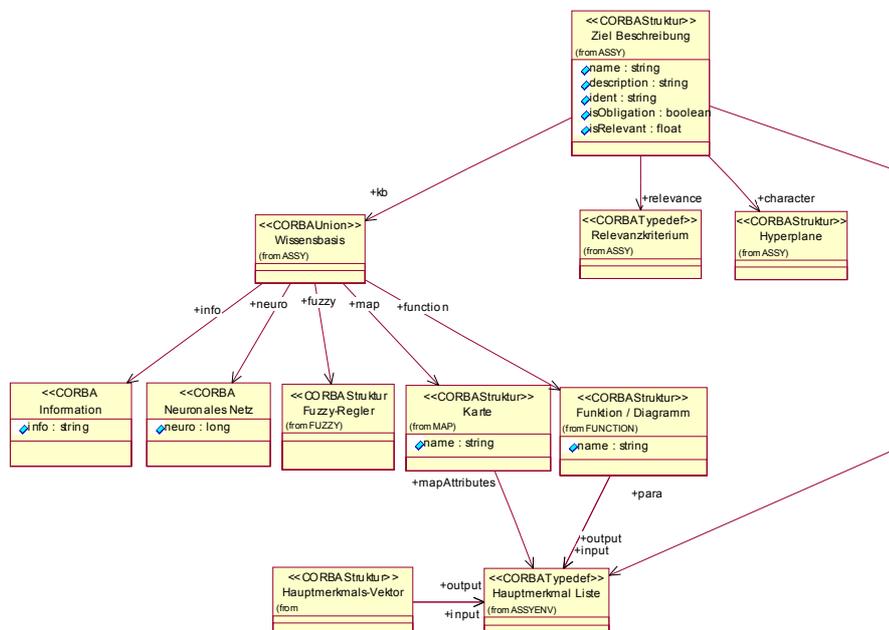


Abbildung 48: Struktur der CORBA-Vorlage für die Repräsentation eines Situations-Elements

Die Vorlage *Ziel* beinhaltet zunächst Merkmale, welche Aussagen über den Zustand des Ziels und über dessen Eigenschaften geben. Hierzu zählen ein Name zur Identifikation, eine Kurzbeschreibung zum Zwecke der Dokumentation, ein Merkmal zur Relevanz und Strukturen, welche das Inferenzwissen für die Bewertung von Situationen enthalten. Für diesen Inferenzmechanismus sind verschiedene Wissensformen implementiert worden, um den Besonderheiten der verschiedenen Zielarten gerecht zu werden. Bei den in dieser Arbeit umgesetzten Inferenzalgorithmen handelt es sich um

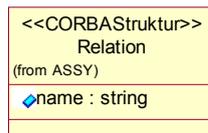
- Diagramme und algebraische Funktionen,
- Fuzzy-Logik,
- implizites Wissen aus Karten und
- freie Informationen

zur Bestimmung der Erfüllung einer Ausgangs-Situation gegenüber einer Ziel-Situation. Diese Inferenzmechanismen stellen den Kern eines Ziel-Modells dar und sind in einem eigenen Konzept verwirklicht. Die Modellierung der CORBA-Struktur *Ziel* basiert auf einfachen Basiskomponenten. Diese sind in ihrem Aufbau in Abbildung 49 dargestellt.



**Abbildung 49: Struktur der CORBA-Templates für die Repräsentation eines Ziel-Modells**

Die tiefere Bedeutung im Situationskontext erhält ein Situationselement durch Relationen zu anderen Situationselementen. Relationen attributieren die Beziehunghaftigkeit von Situationselementen untereinander. Relationen sind hier durch die CORBA-Struktur *Relation* modelliert. Diese sind in Abbildung 50 dargestellt. Relationen unterliegen hier einem einfachen Konzept und sind nur durch einen eindeutigen Namen in Textform charakterisiert.



**Abbildung 50: Struktur der CORBA-Vorlage für die Repräsentation einer Relation**

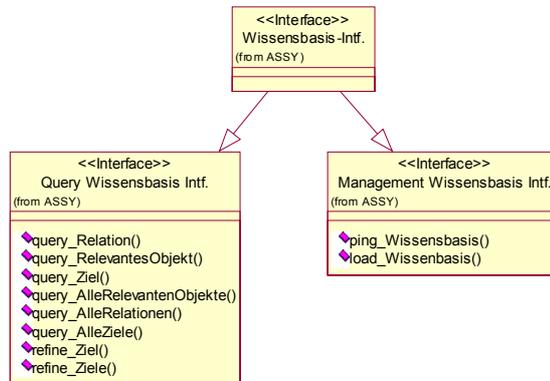
Die bis zu dieser Stelle spezifizierten CORBA-Strukturen stellen diejenigen Informationen dar, welche zwischen den Servern der unterschiedlichen „mentalen“ Repräsentationsformen kommuniziert werden.

### 5.1.3.2 CORBA-Schnittstellen zu den Servern

Neben der Spezifikation der kognitiven Komponenten *Relevante-Objekte*, *Ziel* und *Relation* ist die Akquisition, Repräsentation und Administration dieser Elemente im System entscheidend. CORBA stellt hierfür die Kommunikationskanäle bereit und unterstützt eine verteilte Client-Server-Applikation durch entsprechende Instrumentarien. Mit Hilfe der IDL werden *Interfaces* für die Server spezifiziert. Diese von einer Implementierung unabhängigen *Interfaces* dienen der Kommunikation zwischen den CORBA-Objekten. Server verwalten die Instanzen von CORBA-Objekten. CORBA-Interfaces können nach objekt-orientierten Entwurfsgundsätzen spezifiziert werden und besitzen damit die Möglichkeit, ihre Eigenschaften an abgeleitete *Interfaces* zu vererben.

Erster Schritt in dieser Arbeit ist die Definition der Schnittstelle zur Wissensbasis (*Wissensbasis Intf.*). *Wissensbasis Intf.* dient dazu, den Clients zur Laufzeit diejenigen kognitiven Komponenten bereitzustellen, die das kognitive System für die Abarbeitung des kognitiven Prozesses benötigt. Die Spezifikation der Schnittstelle *Wissensbasis Intf.* ermöglicht es den Clients, auf die dem System bekannten Modelle der kognitiven Komponenten zuzugreifen und diese für die Instanziierung in den

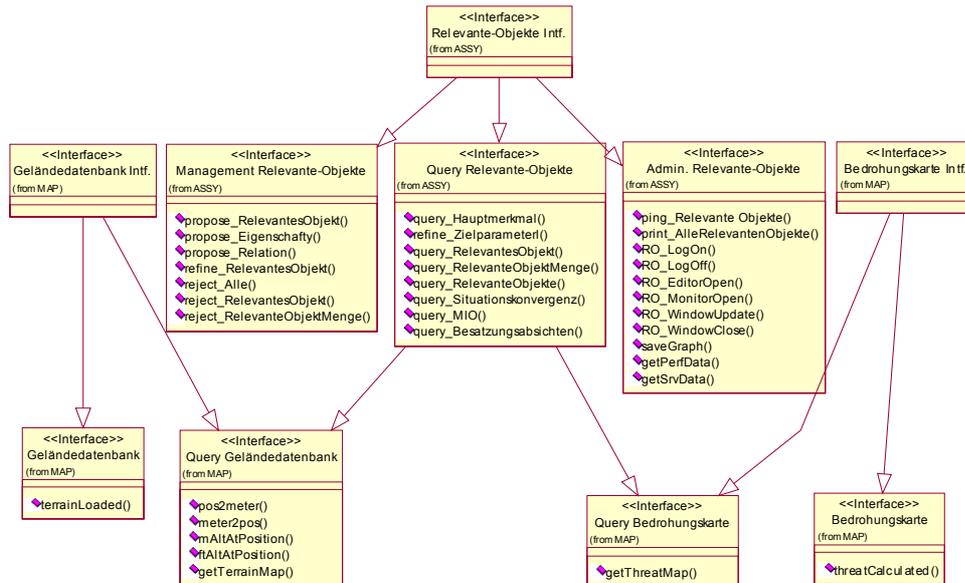
entsprechenden Repräsentationsstrukturen zu verwenden. Die Schnittstelle zur statischen Wissensbasis ist in Abbildung 51 dargestellt.



**Abbildung 51: Schnittstellenarchitektur des Servers zur Wissensspeicherung**

Die Schnittstelle *Wissensbasis Intf.* vererbt ihre Eigenschaften an die spezialisierten Schnittstellen *Query Wissensbasis* (Anfragen an die Wissensbasis) und *Management Wissensbasis* (Management der Elemente in der Wissensbasis).

Die Repräsentation für das dynamische Situationswissen, welche die Instanzen der Situationselemente in einer entsprechenden Repräsentationsform verwalten, sind ebenfalls durch CORBA-Server realisiert. Zunächst sollen der Server der relevanten Objekte und dessen Schnittstelle *Relevante-Objekte Intf.* vorgestellt werden (siehe Abbildung 52).

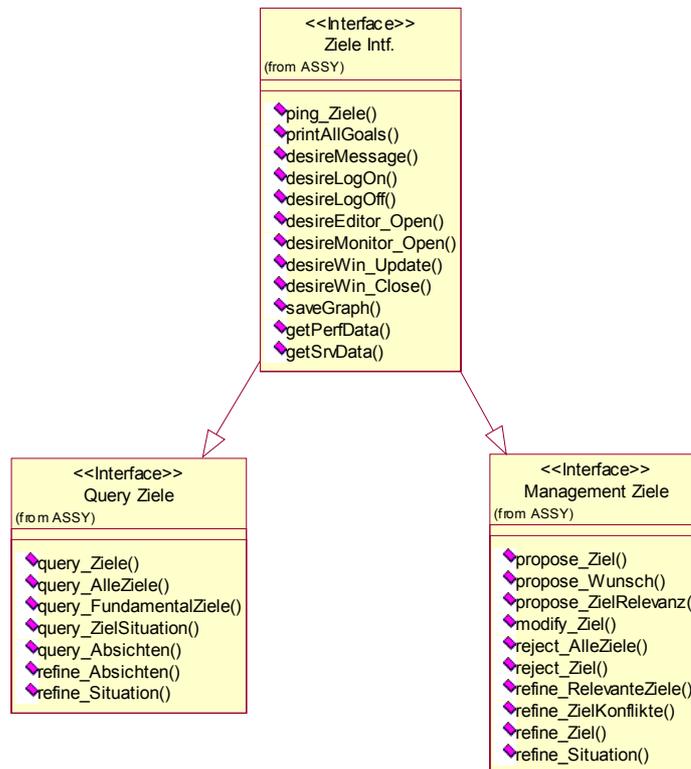


**Abbildung 52: Schnittstellen der Server zur Repräsentation der Lage**

Gleich dem *Wissensbasis Intf.* ist hier eine systematische Teilung der Kommunikation gemäß der Sprechakt-Theorie in Server-Administration und Client-Services vorgenommen worden. Die Schnittstelle *Management Relevante-Objekte Intf.* dient ausschließlich den Komponenten der Situationsinterpretation als schreibender Zugriff auf die Inhalte der Repräsentation der relevanten Objekte. Clients, welche auf diese Information zugreifen, nutzen die Schnittstelle *Query Relevante-Objekte Intf.* Anfragen bezüglich der momentanen, wahrgenommenen Lage können hierüber erfolgen. Die Schnittstelle *Admin. Relevante-Objekte* unterstützt die Überwachung und Analyse der Repräsentationsform zur Laufzeit.

Neben dem Entwurf der relationalen, objektorientierten Situationsbeschreibung existieren implizite Repräsentationsformen von implizitem Wissen. Vertreter impliziten Situationswissens ist z.B. Wissen, welches in Karten gespeichert ist. Geländeerhebungen in DTED (Digital Terrain Elevation Data) und Bedrohungskarten, welche auf Basis des bekannten Wissens über die Lage berechnet wurden, geben Beispiel für Komponenten des Hintergrundwissens. Schnittstellen hierfür sind *Geländedatenbank Intf.* und *Bedrohungskarte Intf.*, die den Zugriff auf Gelände- und Bedrohungskarten ermöglichen. *Geländedatenbank Intf.* und *Bedrohungskarte Intf.* stellen eine weitere Spezialisierung der Schnittstelle *Query Relevante-Objekte* dar.

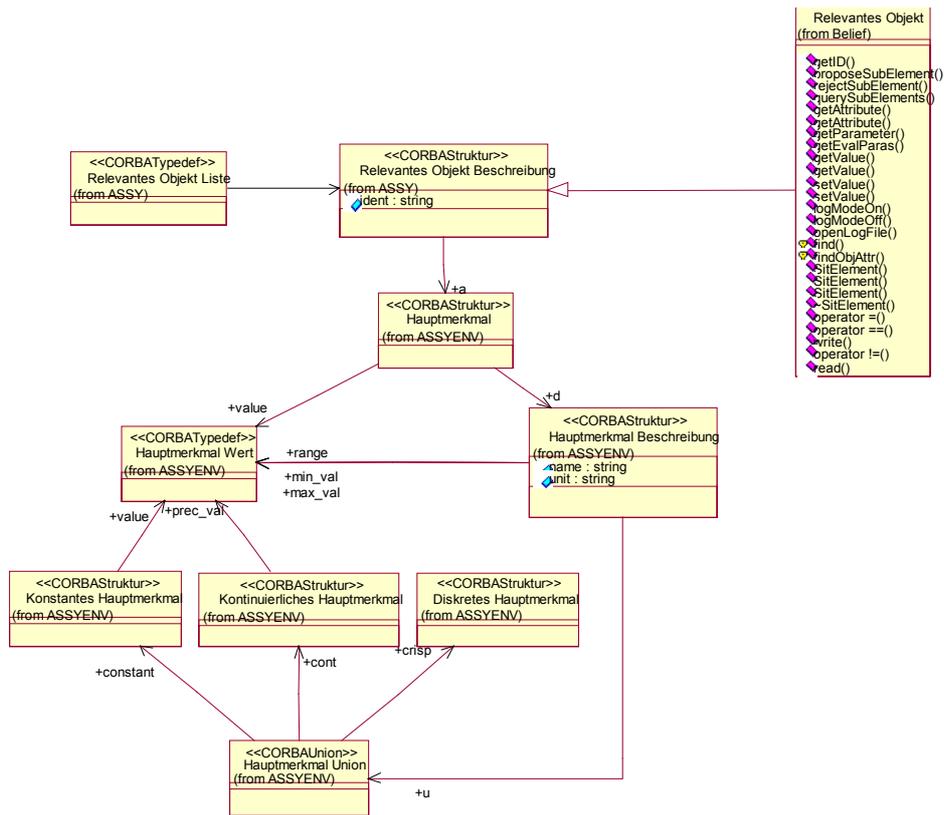
Für Zugriff und Administration der Ziele existiert das Interface *Ziele Intf.* Abgeleitete Schnittstellen sind *Query Ziele Intf.* und *Management Ziele Intf.* Die Gestaltung der Schnittstellen für den Server der Ziel-Repräsentation ist in Abbildung 53 dargestellt.



**Abbildung 53: Schnittstellenarchitektur *Ziele Intf.* des Servers zur Repräsentation der Ziele**

#### 5.1.4 UMSETZUNG DER SPEZIFIKATION IN C++

Die Umsetzung der Spezifikation erfolgt in der Programmiersprache C++. Der IDL Compiler generiert aus den beschriebenen CORBA-Strukturen und CORBA-Objekt-Schnittstellen Quellcode von C++-Strukturen und C++-Klassen mit den für die Nutzung in der CORBA benötigten Eigenschaften. Die erstellten C++-Strukturen werden durch C++-Klassen gekapselt, indem von den C++-Strukturen C++-Klassen abgeleitet werden. Die Kapselung der C++-Strukturen stellt diesen die nötigen Operationen für den geregelten Zugriff bereit und fördert den weiteren objektorientierten Entwurf des Gesamtsystems. Diese Spezialisierungen der CORBA-Strukturen *Relevante-Objekte*, *Ziele* und *Relation* werden in Abbildung 54, Abbildung 55 und Abbildung 56 vorgestellt.



CORBA-Spezifikation

C++-Implementierung

Abbildung 54: Instanziierung der CORBA-Struktur *Relevante-Objekte* (links) in einer C++-Klasse (rechts)

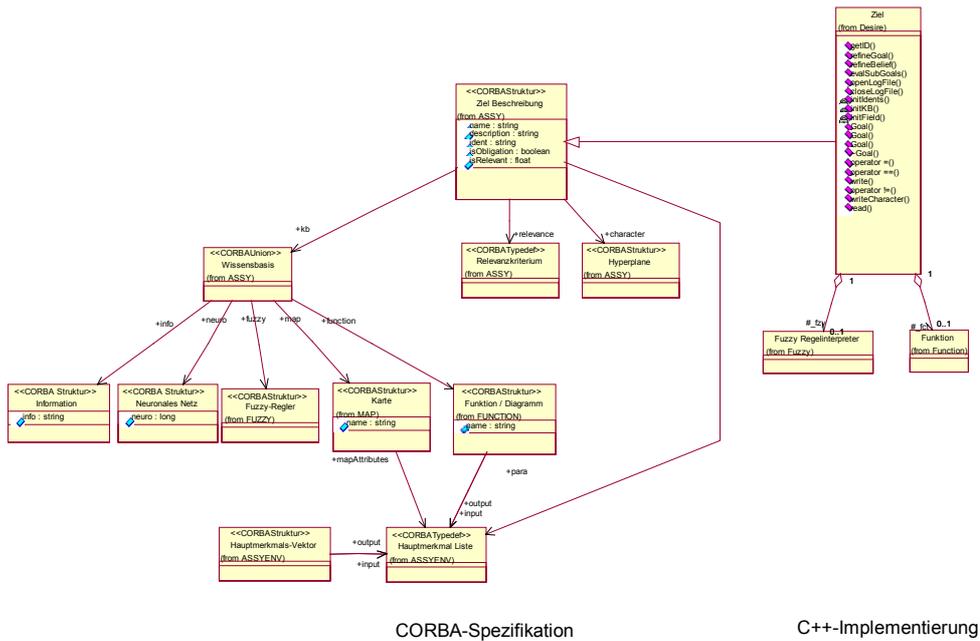


Abbildung 55: Instanziierung der CORBA-Struktur *Ziele* (links) in einer C++-Klasse (rechts)

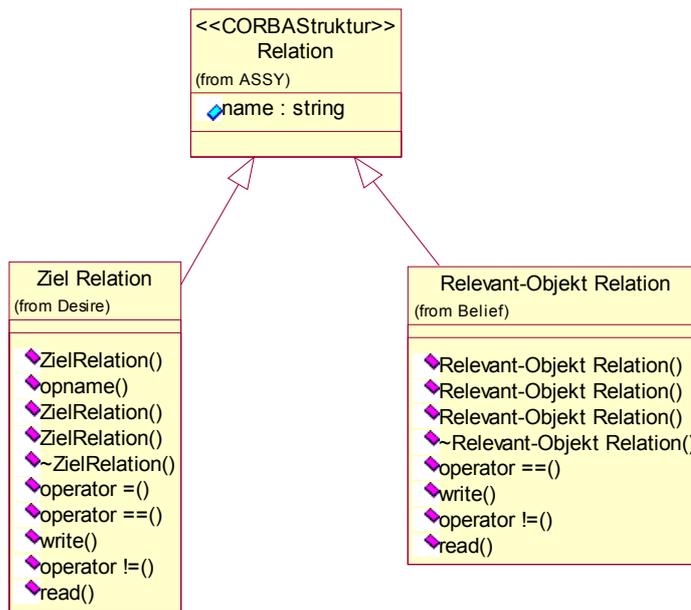


Abbildung 56: Instanziierung des CORBA-Struktur *Relation* in einer C++-Klasse

Für die Repräsentation der oben spezifizierten Modelle in einer objektorientierten Repräsentationsform sind CORBA-Server entworfen worden. Diese beinhalten zur Aufnahme der kognitiven Komponenten geeignete Verwaltungsstrukturen für das relationale Situationswissen. Sowohl für den CORBA-Server der relevanten Objekte als auch für den

CORBA-Server der Ziele wurde die parametrisierbare Klasse GRAPH verwendet. Diese verfügt bereits über Methoden für die Organisation und den Zugriff auf die kognitiven Komponenten. Aufbau und Inhalt dieser Server sind am Beispiel der Ziel-Repräsentation in Abbildung 57 dargestellt. Die Klasse GRAPH ist Teil der C++-Klassenbibliothek LEDA (siehe Kapitel 5.2.2).

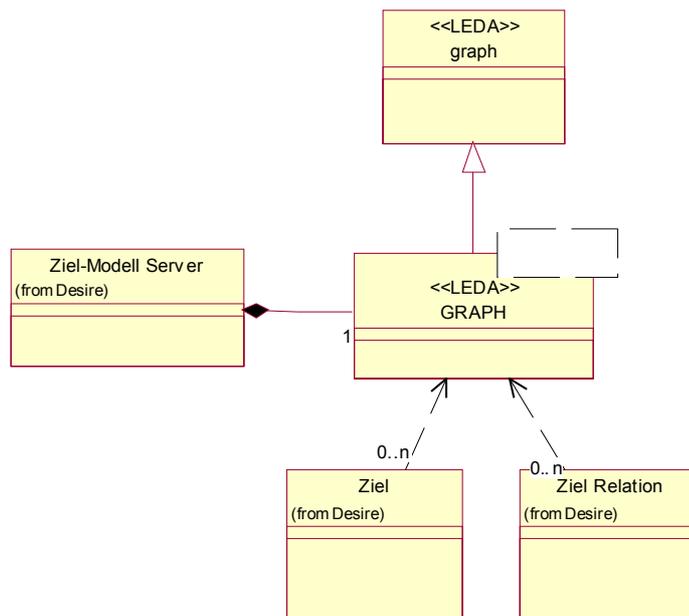
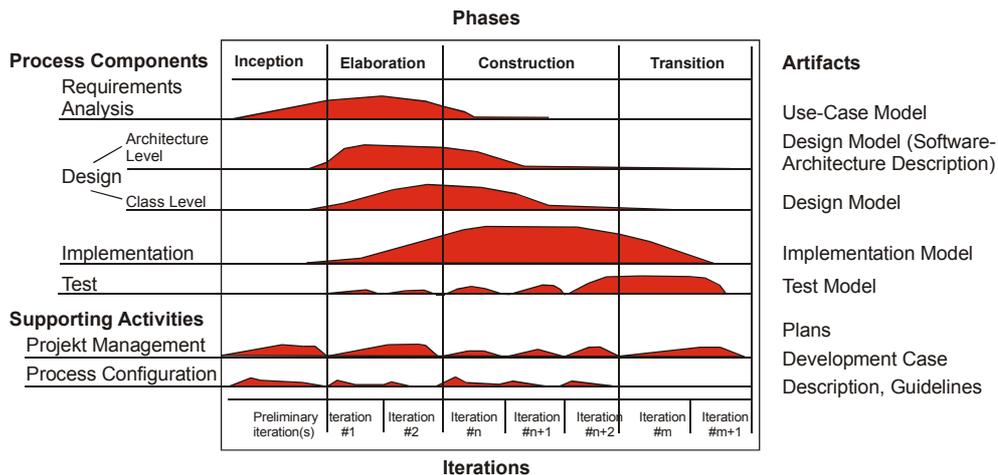


Abbildung 57: Architektur des Servers zur Repräsentation der Ziele

## 5.2 WERKZEUGE

In dieser Arbeit wurde das von der Firma RATIONAL Software Corporation entwickelte Designwerkzeug „Rose C++“ eingesetzt. „Rose C++“ bietet dem Entwickler Unterstützung in allen Entwicklungsphasen (siehe Abbildung 58).



**Abbildung 58: Rational Rose Entwicklungsprozess (Rational Rose C++ User Handbuch)**

Dabei werden von „Rose C++“ schwerpunktmäßig die Bereiche des iterativ/inkrementellen „life-cycle“ Prozess in den Phasen

- Analyse der Anforderungen,
- Systemanalyse,
- Entwurf,
- Implementierung und
- Test

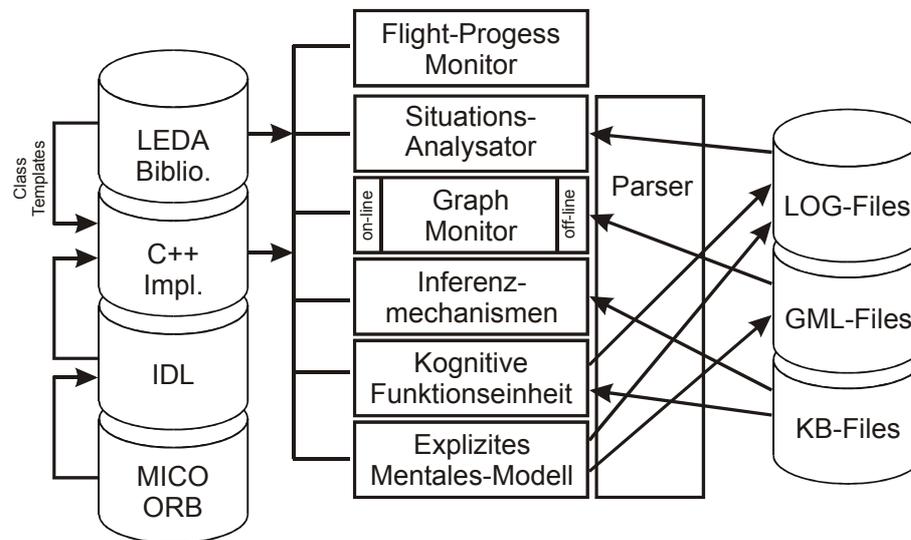
unterstützt.

Die sich an die Anforderungsanalyse anschließende Spezifikation eines Systems stellt den Bauplan für eine nachfolgende Implementierung dar. Hierzu wird zunächst eine Analyse der Anwendungsfälle („Use-Cases“) vollzogen. Use-Case-Diagramme beschreiben die Funktionalität eines Systems, wie sie einem außenstehenden Betrachter erscheinen würde. Use-Case-Diagramme stehen in der industriellen Entwicklung dem Kunden zur Überprüfung der Systemfunktionalität, dem Entwickler als Grundlage für die Erstellung der Komponenten und deren Wechselwirkungen, zur Verfügung und dienen der Dokumentation des Systems. An die Implementierung wurden folgende Anforderung gestellt:

- Nutzung der Rechner-Infrastruktur am Institut (SGI Workstations unter IRIX 6.5),
- hoher Anteil an kommerziellen Softwarekomponenten,
- Minimierung des Wartungsaufwands,

- Nutzung von Standards,
- Schnittstellen zur Programmiersprache C++ und
- Trennung von graphischen Benutzeroberflächen aus Integrations- und Echtzeitgründen.

Aus diesem Grunde wurden für die Implementierung frei verfügbare Softwarebibliotheken genutzt. Die Struktur der eingesetzten Software-Komponenten und Werkzeuge ist Abbildung 59 zu entnehmen.



**Abbildung 59: Eingesetzte Software-Komponenten und Werkzeuge**

MICO als CORBA Derivat stellt die Grundlage für die Verteilte Systemarchitektur dar, LEDA die Basis für die Implementierung in C++, insbesondere durch eine umfangreiche Palette an Datenstrukturen in Form von C++-Klassen. Diese beiden frei verfügbaren Komponenten sollen kurz Erläuterung finden.

### 5.2.1 MICO – OBJECT REQUEST BROKER

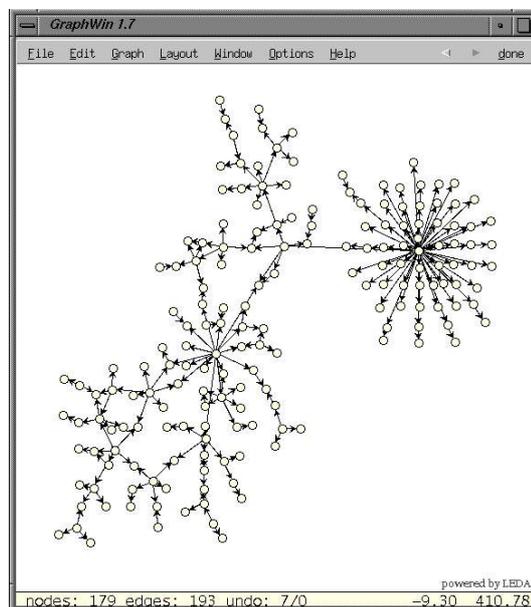
MICO ist ein Open-Source Projekt unter dem GNU Copyright zum Zwecke der Bereitstellung einer frei verfügbaren und vollständig verträglichen Implementierung des CORBA Standards. Die Implementierung von MICO in der Programmiersprache C++ basiert auf dem UNIX API Standard und benutzt ausschließlich weitverbreitete nicht-geschützte Software-Tools.

## 5.2.2 LEDA - BIBLIOTHEK

Die C++ Klassenbibliothek LEDA (Library of Efficient Data types and Algorithms) stellt alle diejenigen Komponenten für die Programmierung zur Verfügung, welche für den Programmereinsatz von komplexeren Datenstrukturen vonnöten sind. Insbesondere die Klassen, welche für die Repräsentation von relationalen Datenstrukturen (Graphen, Listen, Tabellen, etc.) benötigt werden, sind in LEDA in ausgesprochen effizienten, objektorientierten Algorithmen umgesetzt worden. Durch die Umsetzung von LEDA in einer C++-Klassenbibliothek als Compiler-unabhängige Form, kann hier eine große Anzahl an skalierbaren Datentypen genutzt werden. Neben der Sammlung an Datenstrukturen bietet LEDA zahlreiche Bausteine für die Visualisierung von Netzen, im Speziellen komfortable Darstellungsmöglichkeiten für Graphen. Aus diesen Gründen wurde LEDA für die Implementierung der *zentralen, objektorientierten Situationsrepräsentation* genutzt.

## 5.2.3 GRAPH-MONITOR

Der Graph-Monitor (GM) basiert auf den Komponenten zur Visualisierung von Graphen, wie sie von der LEDA-Bibliothek bereitgestellt werden (siehe Abbildung 60).



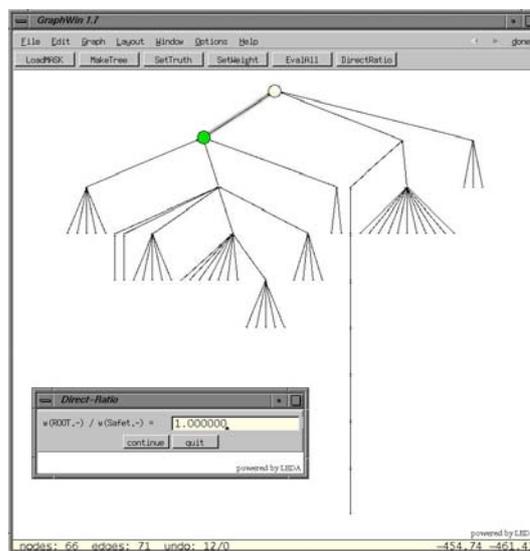
**Abbildung 60:** Graphische Oberfläche des „Graphen-Monitors“ (GM)

Der GM ermöglicht die Analyse und die Überwachung des Zustandes des relational repräsentierten Situationswissens, dessen Topologie und des inneren Zustandes der Situationselemente. Dabei greift der GM auf GML-Files (Graph Modelling Language) zu, die

das Gesamtbild der Situation beschreiben und auf einem Speichermedium (z.B. Festplatte) abgelegt sind. Der GM ist sowohl on-line als auch off-line einsetzbar.

#### 5.2.4 DIRECT-RATIO-BEDIENERSCHNITTSTELLE

Der systematischen Ermittlung der Ziel-Gewichte dient das *Direct-Ratio*-Verfahren (siehe Kapitel 4.3.2.2.3). Um dem Wissensingenieur eine intuitive Eingabeschnittstelle zur Verfügung stellen zu können, wurde eine graphische Nutzerschnittstelle geschaffen (siehe Abbildung 61). Mit Hilfe dieser ist es möglich, die Priorisierung der Einzelziele geordnet und systematisch durchführen zu können.



**Abbildung 61: Bedienerchnittstelle für das *Direct-Ratio*-Verfahren**

#### 5.2.5 FLIGHT-PROGRESS-MONITOR

Der Flight-Progress Monitor FPM (siehe Abbildung 62) dient der Überwachung des Versuchsablaufs. Hier können unterschiedliche Situationselemente und der Flugfortschritt über eine synthetische Kartendarstellung beobachtet werden.

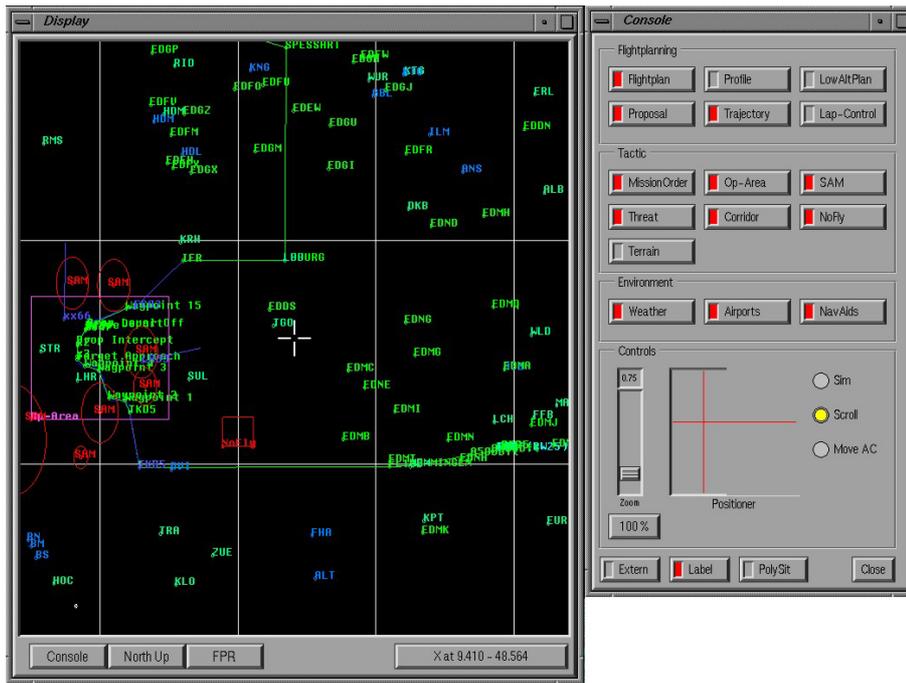


Abbildung 62: Graphische Oberfläche des „Flight-Progress-Monitor“ (FPM)

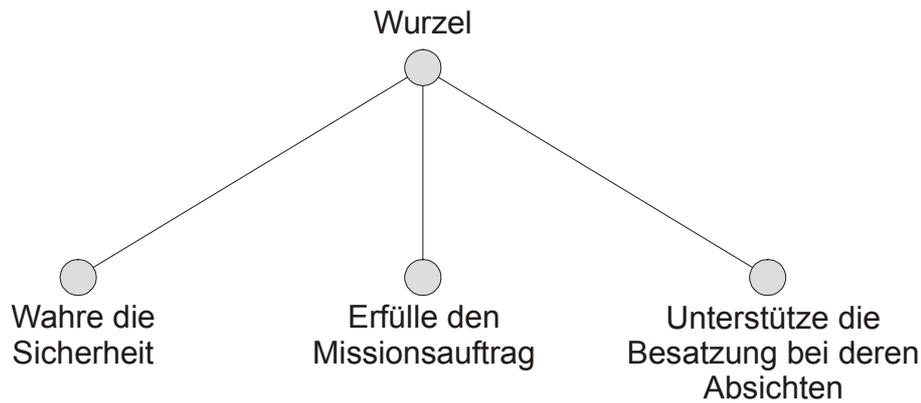
## 6 MODELLBILDUNG

Ziel der Modellbildung ist es, das in den vorherigen Kapiteln entwickelte Modell einer *zentralen, objektorientierten Situationsrepräsentation* mit dem Wissen über die Handlungsziele aus der Domäne der technischen Anwendung auszustatten. Das in dieser Arbeit entwickelte Modell der Handlungsziele soll einem kognitiven Cockpitassistenzsystem dienen, welches den Piloten eines militärischen Transportflugzeugs bei seinen Aufgaben während der gesamten Mission unterstützt. Zu diesem Zweck werden in Form einer *a-priori*-Analyse erste Richtlinien und Grundlagen des militärischen Transportfluges für einen strukturierten Wissenserwerb erarbeitet. Hierauf aufbauend werden mit Hilfe von Wissensakquisitionsverfahren die Prinzipien des militärischen Transportfluges erfasst, die dann in die Umsetzung des Modells der Handlungsziele fließen. Der Erwerb und die Repräsentation des Expertenwissens sind grundlegende Schritte für die Funktionstüchtigkeit und die daraus resultierende Unterstützungsleistung eines Modells der Handlungsziele. Die hieraus resultierenden Teilmodelle der relevanten Objekte und der Ziele werden beispielhaft vorgestellt. Die Ermittlung der die Ziele priorisierenden Gewichte und eine nachgeschaltete Sensitivitätsanalyse schließen die Modellierung ab.

Wie bereits in der theoretischen Vorbetrachtung dargestellt, werden Ziele in Fundamental- und Instrumentalziele klassifiziert und in einer hierarchischen Struktur repräsentiert. Hier sollen nun die in dieser Arbeit erstellten Strukturen erläutert und dargestellt werden. Die Struktur des hier umgesetzten Zielmodells ist durch die Anwendung in einem kognitiven kooperativen Cockpitassistenzsystem bestimmt und zerfällt in die Fundamentalziele

- Wahrung der Sicherheit,
- Erfüllung des Missionsauftrages und
- Unterstützung der Besatzung bei deren Absichten.

Hieraus ergibt sich das makroskopische a-priori-Zielmodell, welches der Wissensakquisition zu Grunde gelegt wird. Die Struktur dieses „high-level“-Modells ist in Abbildung 63 dargestellt.



**Abbildung 63: Modell der übergeordneten Ziele**

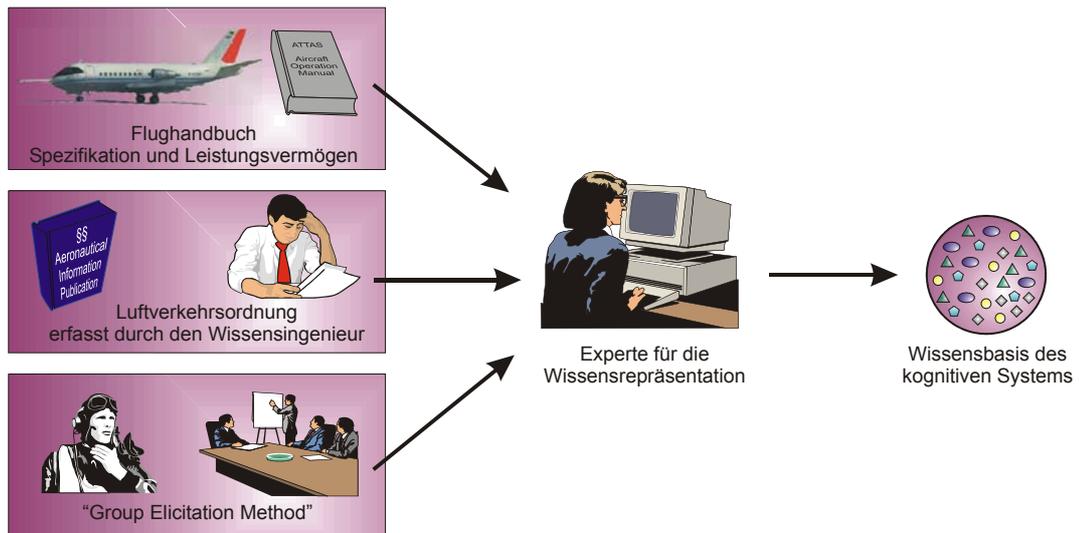
## 6.1 WISSENSAKQUISITION DER HANDLUNGSZIELE

Aus einer *a-priori*-Analyse der zu erstellenden Teilmodelle und den Betrachtungen der domänenspezifischen Aspekte ergeben sich vier Kategorien für die Wissensakquisition:

- Luftfahrzeug (Flugleistungen und –eigenschaften),
- Luftverkehrsordnung (zivil und militärisch),
- Missionsaufgaben und
- Aufgaben des Operateurs (Cockpitbesatzung).

Die Aufgabe im Rahmen der Wissensakquisition ist es, dieses Wissen zu ermitteln, zu strukturieren und für die Weiterverarbeitung in einem Rechner in geeigneter Form zu repräsentieren. Der Prozess der Wissensakquisition, wie er für den Aufbau der Modelle der Handlungsziele erfolgt, ist in Abbildung 64 dargestellt.

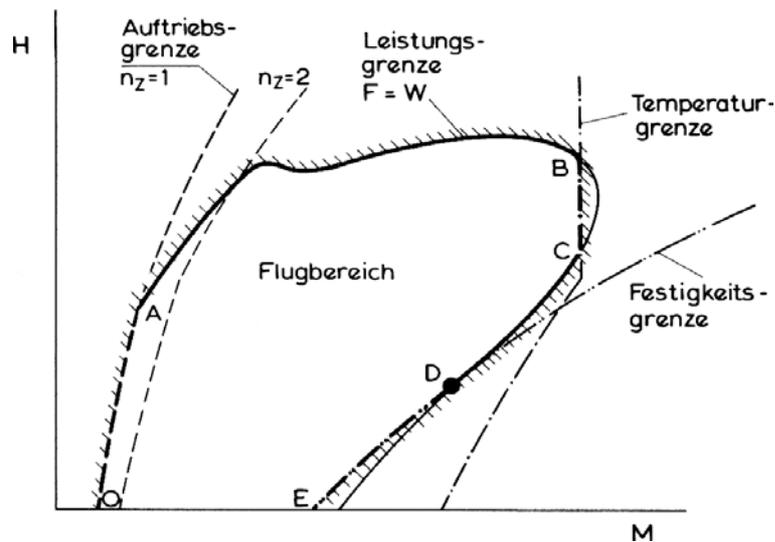
Die Wissensakquisition ermittelt Flugzeugparameter (Flugleistungen und Flugeigenschaften), Luftverkehrsordnung (zivil und militärisch) und militärtaktische Aufgaben der Flugzeugbesatzung.



**Abbildung 64: Prozess der Wissensakquisition**

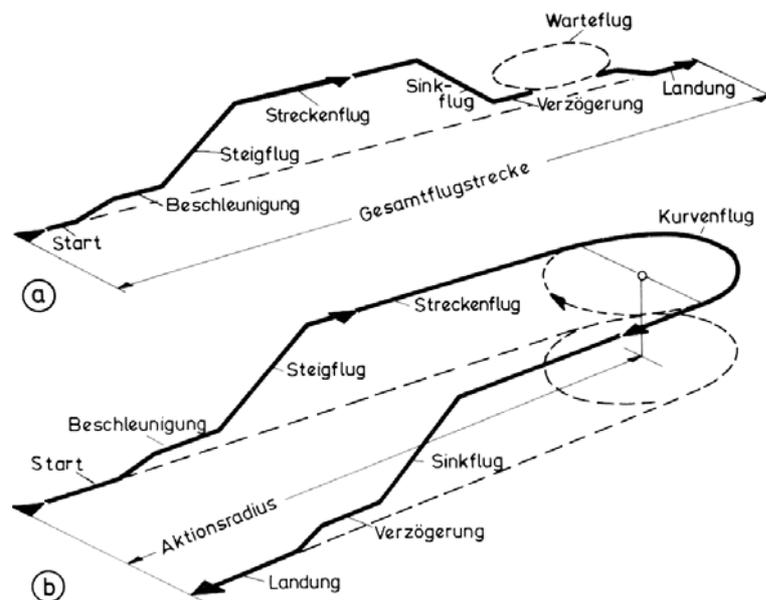
### 6.1.1 FLUGLEISTUNG UND –FÜHRUNG

Trotz des mehr und mehr wachsenden Anteils der *Managements der Flugzeugsysteme* stellt innerhalb der Flugführung die Aufgabe der *Flugzeugsteuerung* immer noch die vordringlichste Aufgabe dar. Die Betrachtung der Flugleistungen beginnt in der Flugmechanik mit einem idealisierten massegleichen Punkt, für den zunächst die Flugbereichsgrenzen ermittelt werden. Die minimale Fluggeschwindigkeit ist durch das Gleichgewicht von Auftrieb und Flugzeuggewicht bestimmt und wird als aerodynamische *Auftriebsgrenze* bezeichnet. Für die maximale Fluggeschwindigkeit ist das Gleichgewicht von Vortrieb und Luftwiderstand von Bedeutung. Sie wird durch die *Leistungsgrenze* des Antriebs beeinflusst. Die in Abbildung 65 dargestellte *Temperaturgrenze* ist nur für Hochleistungsflugzeuge relevant, welche aufgrund der hohen Fluggeschwindigkeit einen bedeutsamen Temperaturanstieg im Bereich des Staupunktes und der Grenzschicht der Strömung erfahren. Die *Festigkeitsgrenze* wird durch die Kurve konstanten maximalen Staudrucks beschrieben und stellt die Beanspruchungsgrenze der Flugzeugstruktur dar.



**Abbildung 65: Flugleistungsgrenzen im Höhen(H)-Machzahl(M)-Diagramm (Brüning, 1986)**

Für die Aufgabe der Flugzeugsteuerung ergeben sich jedoch bei Betrachtung des Flugverlaufs unterschiedliche Zielvorgaben. Diese zu identifizieren, geschieht durch die Dekomposition der Flugaufgabe. Die Dekomposition beginnt in der Analyse der Flugphasen. Flugphasen stellen typische Abschnitte einer Flugmission dar (siehe Abbildung 66) und können in unterschiedlicher Häufigkeit und Abfolge auftreten.



**Abbildung 66: Verläufe der Flugphasen für zwei Flugmissionen (Brüning, 1986)**

Für die Flugmechanik und der in ihr behandelten Flugleistungsbetrachtungen sind die Flugabläufe von besonderer Bedeutung, da hier die idealisierte, da statische, Betrachtung

eines massegleichen Punktes nicht mehr vorausgesetzt wird. Insbesondere in lang andauernden Flugabschnitten muss eine Integration über die Flugzeugmasse erfolgen.

Bei den in Abbildung 66 dargestellten zivilen Flugmissionen handelt es sich um die in der Flugmechanik typischerweise betrachteten Anwendungsfälle eines möglichst Ressourcen optimierten *Streckenfluges* (Abbildung 66a) und eines möglichst Zeit optimierten *Dauerfluges* (Abbildung 66b).

Beide Flugmissionen lassen sich in die typischen Flugphasen Rollen, Startphase, Streckenflug, Landeanflug und Landung zerlegen. Diese sind in Abbildung 67, Abbildung 68 und Abbildung 69 im Detail dargestellt.

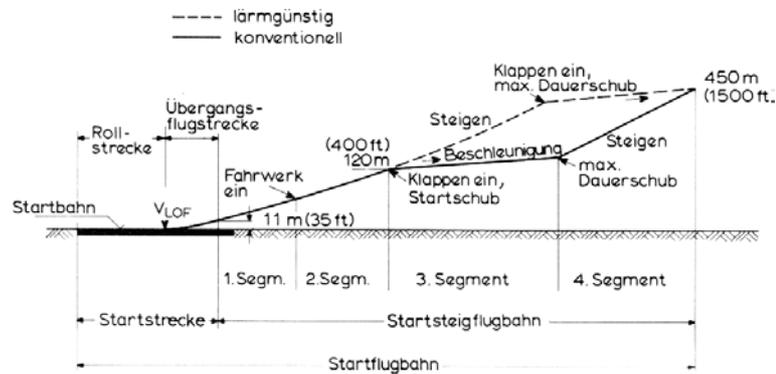


Abbildung 67: Flugleistungselemente in der Startphase (Brüning, 1986)

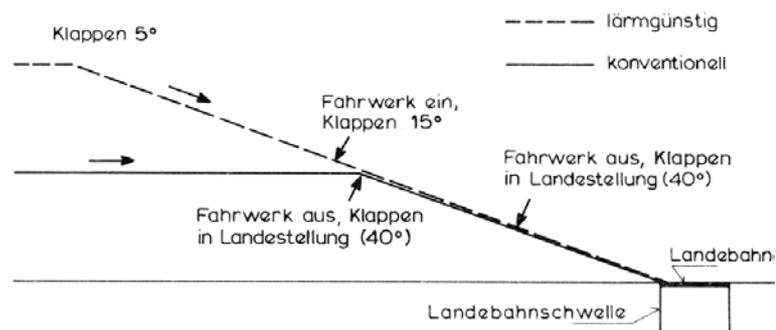
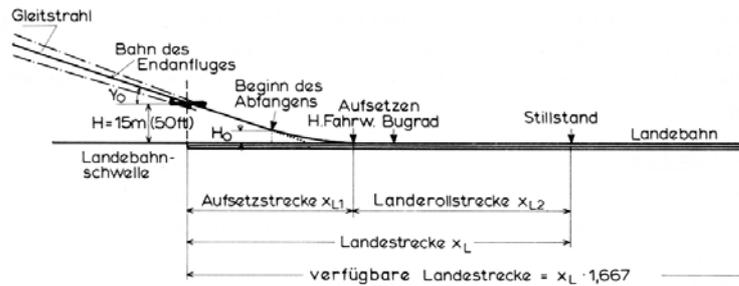


Abbildung 68: Flugleistungselemente im Landeanflug (Brüning, 1986)



**Abbildung 69: Flugleistungselemente bei der Landung**

## 6.1.2 LUFTVERKEHRSVORSCHRIFTEN

Die Luftverkehrsvorschriften ordnen den Luftverkehr und sind auf die nationalen Territorien beschränkt. Die Luftverkehrsvorschriften sind durch das Gesetz in Paragraphen niedergelegt und gliedern sich in Deutschland in die

- Luftverkehrsordnung (LuftVO) und deren Anhänge sowie die
- Durchführungsverordnung und deren allgemeine Bekanntmachungen.

Die Luftverkehrsordnung beschreibt die grundlegenden Vorschriften zur Führung eines Luftfahrzeuges. Hierbei sind sicherheitsrelevante Vorschriften (z.B. Einhaltung einer Mindesthöhe über Menschenansammlungen) von prozeduralen Abläufen (z.B. Einflug in eine Warteschleife) zu unterscheiden. Beide Themen sind in der Durchführungsverordnung detailliert festgelegt. Die Flugsicherheit betreffende Verordnungen sind in drei weitere Bereiche unterteilt:

- allgemeine Regeln,
- Regeln für den Sichtflug (VFR) und
- Regeln für den Instrumentenflug (IFR).

Für eine Repräsentation der Vorschriften in einem Modell muss zunächst eine Analyse hinsichtlich ihrer Strukturierbarkeit erfolgen. Diese Analyse wurde im Rahmen der vorliegenden Arbeit durchgeführt. Folgende Annahmen und Einschränkungen wurden dabei zu Grunde gelegt. Ausgeschlossen von der Analyse sind:

- Vorschriften den Kunstflug,
- der taktische Tiefflug,
- Notfallverfahren,

- Sonderverfahren im Sichtflug und
- Bestimmungen den Wechsel von Flugregeln betreffend.

#### 6.1.2.1 Analyse eines Paragraphen

Die Analyse der Vorschriften betrachtet zunächst die kleinsten Bausteine der Vorschriften, die Paragraphen. Ein Paragraph setzt sich aus den folgenden wesentlichen Bestandteilen zusammen:

- **Tatbestand:**  
Deklaration der Situation, in der die Vorschrift anzuwenden ist.
- **Ausnahmen:**  
Deklaration der Situationen, in denen vom Tatbestand abgewichen werden kann oder muss.
- **Rechtsfolge:**  
Deklaration der Bedingungen, welche eintreten müssen, damit die Vorschrift erfüllt ist.

Hieraus lassen sich für die Modellierung in einem kognitiven System die folgenden Sachverhalte ermitteln:

- Der Tatbestand und die Ausnahmen dienen der Feststellung der Relevanz und definieren damit diejenigen Situationen, in denen die Vorschrift Geltung hat.
- Die Rechtsfolge beinhaltet das Wissen über die Vorschrift und dient somit der Bewertung einer Situation hinsichtlich Erfüllung.
- Tatbestand und Ausnahmen der Vorschriften beschreiben Situationen, in denen die Vorschrift Gültigkeit erlangt.
- Neben rein extrinsischen Situationsmerkmalen können diese auch die Gültigkeit einer weiteren z.B. übergeordneten Vorschrift beinhalten. Dadurch ergeben sich Beziehungen zwischen den Paragraphen, welche neben den Paragraphen selbst einen wichtigen Teil der Vorschriften und somit des Modells darstellen.

Folglich beinhaltet ein Paragraph einer Luftverkehrsvorschrift die folgenden Komponenten:

- Relevanz,
- Bedingung für Erfüllung und

- Beziehung zu einer übergeordneten Vorschrift.

Vorschriften können nun sowohl anhand der Relevanz bezüglich der Situation klassifiziert werden, in der sie Gültigkeit haben, zum anderen können sie in einer hierarchischen Struktur abgebildet werden, welche sich aus den Beziehungen und damit Abhängigkeiten der Vorschriften untereinander ergibt.

### 6.1.2.2 Repräsentation einer Vorschrift

Die Aufbereitung der Luftverkehrsvorschriften auf Basis der in der Analyse gewonnenen Zusammenhänge bedarf zunächst noch der Beantwortung der Frage nach einer geeigneten Struktur. Ziel dieser ist es, die Akquisition des Vorschriftenwesens LuftVO durch eine geeignete Prozedur zu unterstützen und die Nutzung des gewonnen Wissens im Nachhinein bei der Implementierung zu erleichtern. Im Rahmen dieser Arbeit wurde eine auf *Frames* basierte Repräsentationsform gewählt. Die Nutzung von *Frames* gewährleistet eine strukturierte und damit vollständige Extraktion von relevanten Sachverhalten aus dem Gesetzestext. Die verwendeten *Frames* für *Relevanz*, *Bedingung für Erfülltheit* und *Beziehung zu einer übergeordneten Vorschrift* werden nachfolgend kurz vorgestellt. Neben den für den Sachverhalt spezifischen *Slots* besitzt jeder *Frame* zum Zweck der internen Verwaltung *Slots* für

- Primärschlüssel,
- Kurzname und
- Kommentar.

Der Primärschlüssel ist der ein-eindeutige Bezeichner. Anhand dieses Schlüssels kann der *Frame* identifiziert und von anderen unterschieden werden. Die Nomenklatur des Schlüssels setzt sich aus zwei Großbuchstaben und einer fortlaufenden Nummer zusammen. Der Schlüssel ist nachfolgend in Tabelle 3 erläutert:

	Identifier 1	Identifier 2	Identifier 3
Primärschlüssel	R (Relevanz)	C (Clearances) F (Flugphase) L (Luftraumstruktur) P (Parameter) T (Traffic) W (Wetter)	
	V (Vorschrift)	A (allgem. Vorschrift) V (VFR – Vorschrift) I (IFR – Vorschrift)	1 ... N
	B (Bedingung)	A (Absetzen einer Last) K (Kommunikation) E (Instrumenteneinstellungen) P (Parameter) W (Wetter)	

**Tabelle 3: Verschlüsselung der Frames durch einen Primärschlüssel**

Der Kurzname beschreibt den Frame im Klartext, und der Kommentar kann erläuternde Textauszüge aus der Vorschrift selbst enthalten.

### 6.1.2.3 ‚Frame Vorschrift‘

Im Frame Vorschrift (siehe Abbildung 70) sind alle eine Vorschrift beschreibenden Sachverhalte abgelegt.

Vorschrift
Primärschlüssel
Kurzname
<b>Relevanz</b>
<b>Regel</b>
Übergeordnete Vorschrift
Inhalt
Quelle
Kommentar

**Abbildung 70: Frame „Vorschrift“**

Hierzu gehören:

- **Relevanz:**  
Die Beschreibung der relevanten Situationen ist im ‚Frame Vorschrift‘ über einen Primärschlüssel referenziert. Dieser verweist auf einen untergeordneten ‚Frame Relevanz‘.

- **Regel:**  
Die Bedingung für Erfülltheit einer Vorschrift ist ebenfalls über einen Primärschlüssel referenziert und in einem eigenen ‚Frame Regel‘ abgelegt. Hier kann auch auf untergeordnete Vorschriften verwiesen sein, die zur Gesamterfüllung der Regel beitragen. Logische Ausdrücke formulieren hier Meta-Regeln, womit eine universellere Beschreibung der Vorschriftenregel möglich ist.
- **Übergeordnete Vorschrift:**  
Die hierarchische Beziehungshaftigkeit zwischen den Vorschriften ist über die Referenzierung in Form von Schlüsselidentifikatoren abgelegt.
- **Inhalt:**  
Zu Dokumentationszwecken ist hier in stark zusammengefasster Form der Inhalt der Vorschrift in Klartext wiedergegeben.
- **Quelle:**  
Der Ursprung der Vorschrift wird hier genannt. In der Regel ist dies der Paragraph.
- **Kommentar:**  
Der Wortlaut des Paragraphen ist hier in Form des Gesetzestextes wiedergegeben.

#### 6.1.2.4 ‚Frame Relevanz‘

Der ‚Frame Relevanz‘ (siehe Abbildung 71) beinhaltet die Beschreibung derjenigen Situation, in der die Vorschrift Gültigkeit erlangt.

Relevanz
Primärschlüssel
Kurzname
Kriterium
Kommentar

Abbildung 71: ‚Frame Relevanz‘

Hierzu gehört:

- **Kriterium:**  
Beschrieben werden hier die Kriterien für den situativen Kontext.

### 6.1.2.5 ‚Frame Regel‘

Hier ist der eigentliche Inhalt (Rechtsfolge) abgelegt.

Regel
Primärschlüssel
Kurzname
Inhalt
Kommentar

Abbildung 72: ‚Frame Regel‘

Hierzu gehört:

- **Inhalt:**  
Neben den Situationsmerkmalen der Eingangsgrößen ist die Regel hier abgelegt.

### 6.1.2.6 Beispiel

Anhand der Beispiele „Geschwindigkeitsbegrenzung unterhalb FL100“ und „Mindestflugsicht im Sichtflug“ nach §28 I LuftVo in Verbindung mit Anlage 5 LuftVo wird nachfolgend das Resultat der Wissensakquisition für die Luftverkehrsordnung erläutert. Hierbei wird die Hierarchie der relevanten Flugregeln aufgebaut, wobei mit der Wurzel, also der obersten Vorschrift begonnen wird. Diese Wurzel-Vorschrift „allgemeine Flugregeln“ stellt sich, wie in Abbildung 73 dargestellt, dar.

Vorschrift Vx1 Allgemeine Flugregeln
Primärschlüssel: {VX1}
Kurzname: {"Anwendung Flugregeln"}
<b>Relevanz:</b> {ANY_SITUATION == TRUE}
<b>Regel:</b> {VA1 AND (VV1 OR VI8)}
Übergeordnete Vorschrift: {}
Inhalt: {"Oberste Regel für die Anwendung der Flugregeln"}
Quelle: {"§4 LuftVO"}
Gesetzestext: {"(1) Der Betrieb ..."}

Abbildung 73: Vorschrift „Allgemeine Flugregeln“

Aus dem Slot Relevanz ist zu ersehen, dass diese „Wurzel-Vorschrift“ immer, also in jeder Situation, Gültigkeit hat. Die Regel für die Erfülltheit basiert auf der logischen Verknüpfung von drei weiteren Vorschriften: VA1, VV1 und VI8.

Vorschrift VV1 ist in Abbildung 74 dargestellt und soll weitere Erläuterung finden.

<b>Vorschrift VV1 VFR Flugregeln</b>
Primärschlüssel: {VV1}
Kurzname: {"Visual Flight Rules"}
<b>Relevanz:</b> {RF6 == FALSE}
<b>Regel:</b> {VV2 AND VV3 and VV4 and VV5}
Übergeordnete Vorschrift: {VX1}
Inhalt: {"Oberste VFR Flugregel"}
Quelle: {}
Gesetzestext: {}

**Abbildung 74: Vorschrift „Sichtflugregeln“**

Der Primärschlüssel VV1 klassifiziert diese Vorschrift über den Bereich der VFR bzw. Sichtflugregeln. Vorschrift VV1 ist dann von Relevanz, wenn die Kriterien, abgebildet im Frame RF6 (siehe Abbildung 75), nicht erfüllt sind.

<b>Relevanz RF6 Instrument Flight</b>
Primärschlüssel: {RF6}
Kurzname: {"Instrument Flight"}
Kriterium: {FLUGREGELSATZ == IFR}
Kommentar: {"Das LFZ wird ..."} 

**Abbildung 75: Relevanz „Instrument Flight Rules“**

Hieraus ergibt sich eine Negation der Relevanzkriterien. Das Relevanzkriterium RF6 bezieht sich auf die Flugphase, insbesondere hier auf die Durchführung des Fluges nach Instrumentenflugbedingungen. Damit ergibt sich für die Relevanz von VV1: VV1 ist relevant, wenn der Flug nicht nach Instrumentenflugbedingungen durchgeführt wird.

Für die Erfüllung der Vorschrift VV1 ist u.a. die Vorschrift VV2 zu erfüllen (siehe Abbildung 76).

<b>Vorschrift VV2 Lufträume, allgemein</b>
Primärschlüssel: {VV1}
Kurzname: {"Lufträume, allgemein"}
<b>Relevanz:</b> {ANY_SITUATION == TRUE}
<b>Regel:</b> {VV6 OR VV7 OR ...}
Übergeordnete Vorschrift: {VV1}
Inhalt: {"Luftraumnutzung bei VFR. Vgl. Anlage 5 LuftVO ..."}
Quelle: {"Anlage 5 LuftVO"}
Gesetzestext: {"Luftraum Klasse A: ..."}

**Abbildung 76: Vorschrift der „Lufträume“**

Vorschrift VV2 formuliert die Luftraumreglementierung. Sie ist in jeder Situation relevant und damit immer gültig. Ihre Erfülltheit hängt von der Erfüllung untergeordneter Vorschriften ab, u.a. von Vorschrift VV14 (siehe Abbildung 77).

<b>Vorschrift VV14 Luftraum G</b>
Primärschlüssel: {VV14}
Kurzname: {"Luftraum G"}
<b>Relevanz:</b> {AC-Pos WITHIN Airspace-G}
<b>Regel:</b> {VV20 AND BW6 AND ...}
Übergeordnete Vorschrift: {VV2}
Inhalt: {"Anlage 5 LuftVO. ..."}
Quelle: {"Anlage 5 LuftVO"}
Gesetzestext: {"Luftraum Klasse G: ..."}

**Abbildung 77: Vorschrift „Luftraum GOLF“**

Die Relevanz dieser Vorschrift ergibt sich aus der Situation, dass sich das Flugzeug, welches dieser Vorschrift zu folgen hat, innerhalb eines Luftraums der Klasse GOLF befinden muss. Die Regel selbst erreicht Erfülltheit, wenn u.a. die Vorschrift VV20 und die Regel BW6 erfüllt sind. Bedingung BW6 ist in Abbildung 78 dargestellt.

<b>Relevanz RW6 Minimale Flugsicht</b>
Primärschlüssel: {BW6}
Kurzname: {"Minimale Flugsicht"}
Kriterium: {FLUGSICHT >= 1,5km}
Kommentar: {"Gilt für Flächen-LFZ ..."}

**Abbildung 78: Bedingung „Minimale Flugsicht“**

In BW6 ist eine Bedingung formuliert, welche meteorologische Bedingungen zur Grundlage hat. Mit der VV20 (siehe Abbildung 79) gelangt man auf die unterste hierarchische Ebene des Beispiels.

<b>Vorschrift VV20 Geschwindigkeitsbeschränkung</b>	
Primärschlüssel:	{VV20}
Kurzname:	{"Geschwindigkeitsbeschr."}
<b>Relevanz:</b>	{(RC2 ==FALSE) AND ...}
<b>Regel:</b>	{BP15}
Übergeordnete Vorschrift:	{}
Inhalt:	{"In den Lufträumen ..."} 
Quelle:	{"Analge 5 LuftVO"}
Gesetzestext:	{}

**Abbildung 79: Vorschrift „Geschwindigkeitsbeschränkung“**

Hier findet man eine Reihe von Relevanzkriterien, u.a. RC2 (siehe Abbildung 80).

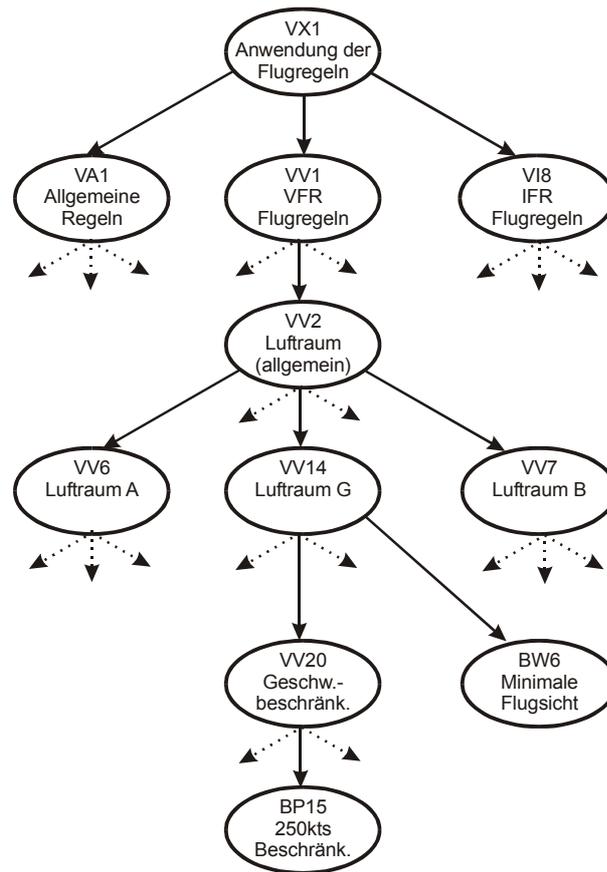
<b>Relevanz RC2 ATC Freigabe für 250kts</b>	
Primärschlüssel:	{RC2}
Kurzname:	{"ATC-Freigabe für 250kts"}
Kriterium:	{RECEIVED CLEARANCE 250KTS}
Kommentar:	{"Die zuständige FVK-Stelle ..."} 

**Abbildung 80: Relevanz „ATC Freigabe für 250kts“**

<b>Regel BP15 250kts Beschränkung</b>	
Primärschlüssel:	{BP15}
Kurzname:	{"250kts-Beschränkung"}
Inhalt:	{IAS <= 250kts}
Kommentar:	{"250kts Beschränkung gilt ..."} 

**Abbildung 81: Regel für „250kts Beschränkung“**

Hieraus ergibt sich eine hierarchische Vorschriftenstruktur (siehe Abbildung 82).



**Abbildung 82: Beispiel für eine Hierarchie von Vorschriften**

Die Flugleistungen und Manöver in den zivilen Flugphasen sind um die militärischen Verfahren in den taktischen Flugphasen zu erweitern.

### 6.1.3 MILITÄRISCHE TRANSPORTFLUGVERFAHREN

Da militärische Transportflugverfahren nur unzureichend dokumentiert sind, hat die Wissensakquisition direkt über den Experten zu erfolgen. Wie bereits von (Hart, 1986) erarbeitet, ist es möglich, Wissen durch Fragebögen, Datenaufzeichnung und Beobachtung der Experten zu gewinnen. Zudem stellt die Befragung des Experten selbst eines der wichtigsten Elemente dar, nicht nur alleine aus dem Grunde, um den Experten frühzeitig in den Entwicklungsprozess einzubinden und ein für den Nutzer sinnvolles und anwendbares System zu entwickeln.

(Hart, 1986) empfiehlt für die Wissensakquisition den offenen Dialog zwischen Entwickler und Experte und gibt hierfür folgende Empfehlungen:

- **Vorbereitung**

Der Wissensingenieur sollte sicherstellen, dass Zeit und Ort für die Befragung in

Einvernehmen mit allen Beteiligten sind, und dass Unterlagen und Hilfsmittel bereit stehen. Jeder Teilnehmer sollte über Sinn und Zweck der Befragung und seine Aufgabe darin informiert sein.

- **Dokumentation**

Experten werden schnell gelangweilt und verärgert, wenn sie sich in ihren Aussagen wiederholen müssen. Daher sollte die Dokumentation des Wissens gut vorbereitet sein und entsprechendes Material (Schreibblock, Audio-/Video-Aufzeichnung) bereitstehen. Oftmals erweist sich auch eine zusätzliche Person zur Protokollierung als hilfreich.

- **Spezifische Problemstellungen**

Vage Fragestellungen führen zu Abschweifungen, Wiederholungen und unstrukturierten und damit schwierig zu analysierenden Informationen. Spezifische Problemsituationen sind für einen Experten einfacher zu erfassen.

- **Einfühlungsvermögen**

Der Experte sollte nicht dazu gezwungen werden, Informationen zu geben, die ihm fremd oder unbekannt sind, was nicht bedeuten soll, dass der Experte bei komplexen Aufgaben ohne Nachdruck die Befragung durchführen sollte.

- **Werkzeuge**

Dem Experten sollte die Möglichkeit gegeben werden, sein Wissen mit dem ihm bekannten Werkzeug darzustellen. Insbesondere in frühen Phasen der Akquisition sollten keine dem Experten fremden Instrumentarien verwandt werden.

### **6.1.3.1 Expertenbefragung mittels der “Group-Elicitation-Method”**

Zur Akquisition von Expertenwissen wird für das hier zu erstellende Modell der Handlungsziele die von (Boy, 1996) entwickelte „Group-Elicitation-Method“ verwandt. Diese Gruppenbefragung nutzt die Technik des semi-formal strukturierten Interviews und basiert auf einem sechsstufigen Verfahren (Boy, 1996):

- Methodisch unterlegte Einführung der Experten in die Brainstorming-Befragung durch den Wissensingenieur und Aufstellen von Thesen,
- Brainstorming-Befragung der Experten,
- Reformulierung und Zusammenfassung der gesammelten Ergebnisse,

- Bewertung der einzelnen Statements,
- Priorisierung und
- Analyse der Ergebnisse.

Die Formulierung von Thesen dient der Einführung von Experten in das Thema der Befragung und basiert auf dem *a-priori*-Wissen des Wissensingenieurs, welches er sich durch die Analyse der Flugzeugparameter, der Luftverkehrsordnung und der militärtaktischen Verfahren erworben hat. Hierbei werden aus Sicht des Wissensingenieurs der Zweck und die Zielrichtung sowie mögliche Annahmen bezüglich der Resultate der Befragung präsentiert. Diese Sensibilisierung dient der Vermittlung des Zwecks der Befragung. Daneben vermittelt sie den Experten das nötige Hintergrundwissen, welches sie für die Phase des Brainstormings benötigen. Die Akzeptanz, Motivation und Sensibilisierung der Experten bezüglich der ihnen gestellten Aufgabe stellt einen, wenn nicht gar den wichtigsten Gegenstand einer Expertenbefragung dar. Neben der Schaffung eines ausreichenden Situationsbewusstseins gehören zu diesem ersten Schritt auch die Vorgabe von Randbedingungen, die für die spätere Befragung wichtig sind. Die Inhalte solcher Rahmenbedingungen sind variabel und hängen von dem erforderlichen Fachwissen ab.

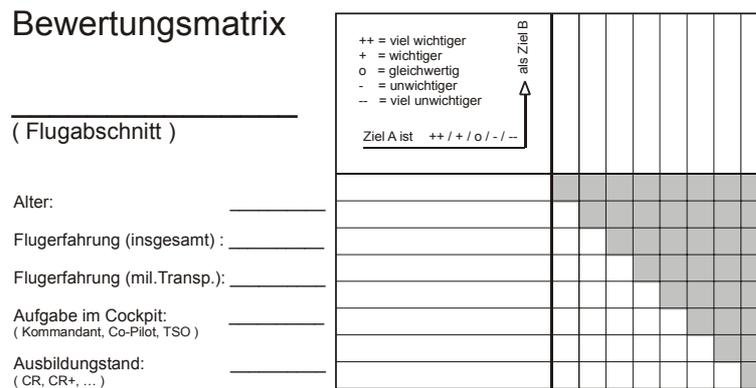
Das Brainstorming findet in stiller Einzelarbeit statt. Hierbei sammeln die Experten auf einem Blatt Papier zunächst ihre subjektiven Aussagen bezüglich der im ersten Schritt aufgestellten These. Die während des Brainstormings gesammelten Statements können je nach These einen sehr allgemeinen, aber auch einen speziellen Charakter besitzen. Ist die für das Nennen von Aussagen eines Experten bestimmte Zeit verstrichen, wird das Formular mit den gesammelten Aussagen an den Tisch-Nachbarn weitergereicht. Im nachfolgenden Zyklus besteht nun die Aufgabe für die Experten darin, die bereits gesammelten Aussagen zu kommentieren, bzw. neue Aspekte, die sich durch das Studium der ihnen vorliegenden Aussagen ergeben, zu dokumentieren. Der Verfahrenschritt des Brainstormings endet, sobald jeder Experte sein Blatt, welches ihm zu Anfang ausgehändigt wurde, wieder in Händen hält. Der Vorteil dieser Vorgehensweise liegt darin, dass sich die Experten nicht durch die Argumente der übrigen Teilnehmer beeinflussen lassen, wie es häufig bei Diskussionen der Fall ist.

Die während des Brainstormings gesammelten Statements werden den Teilnehmern im Anschluss präsentiert. Die gesammelten Aspekte werden dann offen in der Gruppe analysiert, diskutiert und strukturiert. Hierbei sind folgende Modifikationen möglich:

- Zusammenfassung bereits existierender Punkte,
- Hinzufügen neuer Punkte und
- Auftrennen bestehender Punkte in neue Unterpunkte.

Abschluss findet diese Phase in der Dokumentation der gesammelten Aussagen.

Die Bewertung der gesammelten Statements geschieht zum Zwecke einer Priorisierung bezüglich der Wichtigkeit. Dies geschieht über eine Bewertungsmatrix (siehe Abbildung 83).



**Abbildung 83: GEM Bewertungsmatrix**

In die Randspalte und die obere Randzeile der Bewertungsmatrix werden die gesammelten Aussagen eingetragen. Diese werden nun über eine fünf-wertige Bewertungsskala ( $-2=$ viel unwichtiger,  $-1=$ unwichtiger,  $0=$ gleichwertig,  $+1=$ wichtiger,  $+2=$ viel wichtiger) gegeneinander gewichtet.

Bei der Erstellung einer Rangfolge der im Brainstorming gesammelten Statements liegt das Hauptaugenmerk auf der *mittleren Priorität* (MP) der einzelnen Gesichtspunkte.

$$MP = \frac{\sum W_i}{N_{A_i}}$$

mit  $N_{A_i}$  = Anzahl der Teilnehmer  
 und  $\sum W_i$  = Summe über Wertungspunkte

Die MP kann dabei Werte zwischen +100 und -100 annehmen. Hieraus ergibt sich z.B. eine MP von +100, wenn alle Teilnehmer der Befragung eine Aussage gegenüber allen anderen

Aussagen mit +2 bewertet haben. Die MP wird dabei aus dem Mittelwertsatz gewonnen. (Boy, 1996) empfiehlt, der Bewertung die *Abweichung von der MP* oder die *Stabilität* hinzuzuziehen. Diese Werte beziehen sich auf die Standardabweichung von der MP. Die Standardabweichung drückt die Konvergenz der Gruppe bezüglich des Ergebnisses aus.

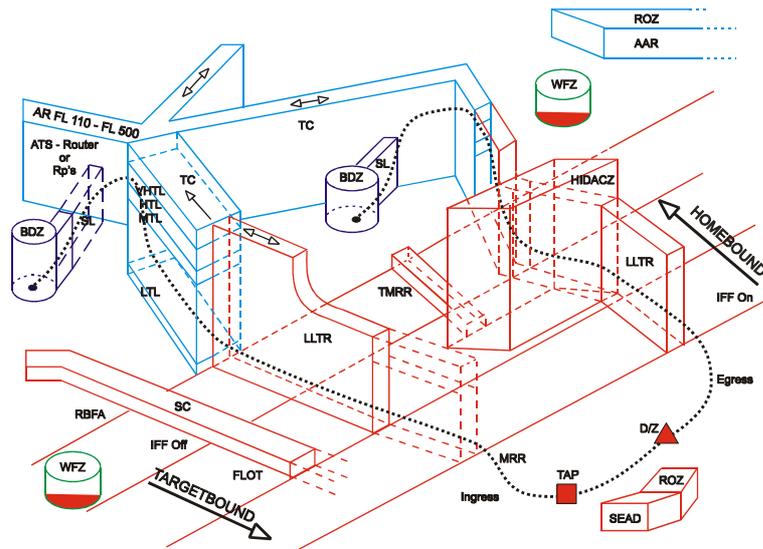
In dieser Arbeit wurde GEM zur Analyse und Strukturierung der Handlungsziele während einer taktischen Transportmission herangezogen. Hierzu wurde zunächst der militärtaktische Flug in seine typischen Flugphasen unterteilt:

- Start,
- Marschflug unter IFR Flugbedingungen,
- Transit IFR- taktischer VFR via taktischen Korridor,
- Tiefflug,
- Absetzen der Last,
- Tiefflug,
- Transit taktischer VFR – IFR via taktischen Korridor,
- Marschflug unter IFR Flugbedingungen und
- Landung.

Bei der Analyse lag das Hauptaugenmerk auf denjenigen Flugabschnitten, welche innerhalb des Operationsgebietes und damit unter feindlicher Bedrohung liegen. Zunächst sollen jedoch die Charakteristika einer militärischen Transportmission erörtert werden.

### **6.1.3.2 Grundlagen des militärischen Transportfluges**

Der militärische Transportflug (siehe Osswald, 1994) unterscheidet sich von einem zivilen Verkehrsflug dadurch, dass hier Mensch und/oder Material in ein taktisches Operationsgebiet verlegt werden. Die Missionen können sowohl zu Friedenszeiten als auch im Verteidigungsfall erfolgen. Sie erstrecken sich von humanitären Einsätzen bis zum Absetzen von Krisenreaktionskräften in feindlichen Gebieten. Ein taktisches Szenario wird durch eine militärische Kontrollinstanz koordiniert. Hieraus resultieren grundlegende Änderungen des Luftraums gegenüber einem zivilen Einsatzspektrum (siehe Abbildung 84).



**Abbildung 84: Taktische Szenarienelemente einer militärischen Transportmission**

Die militär-taktische Luftraumstruktur, welche mehrmals täglich durch einen Luftraumkoodinierungsbefehl (ACO = Airspace Coordination Order) geändert wird, ist in drei Bereich unterteilt:

- **Annäherungszone**  
 Erstreckung vom hinteren Luftraum mit zivilen Regularien bis zum hinteren Rand der Verteidigung (RBFA = Rear Boundary of Forward Area).
- **Übergangszone**  
 Erstreckung vom hinteren Rand der Verteidigung (RBFA) bis zum vorderen Rand der Verteidigung (FLOT = Forward Line of Own Troops). Den Luftangriffs Kräften soll hier ein möglichst sicheres und unerkanntes Eindringen in das feindliche Gebiet ermöglicht werden. Hierzu werden Übergangskorridore in den Tiefflug (LLTR = Low Level Transit Routes), Routen mit minimalem Bedrohungsrisiko (MMR = Minimum Risk Routes) für den bidirektionalen Überflug des vorderen Randes der Verteidigung und Sicherheitskorridore eingerichtet.
- **Identifikationszone**  
 Erstreckung vom vorderen Rand der Verteidigung (FLOT = Forward Line of Own Troops) bis zur elektronischen Erkennung-An-Linie (IFF-ON = Identifikation Friend Foe ON). Die Identifizierung zurückkehrender Luftangriffskräfte erfolgt hier über elektronische Mittel.

Vor Flugantritt wird der Besatzung im Rahmen eines Briefings die momentane taktische Lage bekanntgegeben. Hierzu gehören:

- **Air-Tasking-Order**

Dieses NATO weit standardisierte Formular enthält in seiner monatlichen, wöchentlichen oder täglichen Veröffentlichung die geplanten Einsätze des Tages und wird zur Übersicht für die Durchführung des täglichen Luftkrieges erstellt.

- **Intelligence-Briefing**

Die von den aufklärenden Streitkräften gewonnenen Informationen werden hierin bekanntgegeben und gemäß der geplanten Flugmission in eine Karte eingetragen.

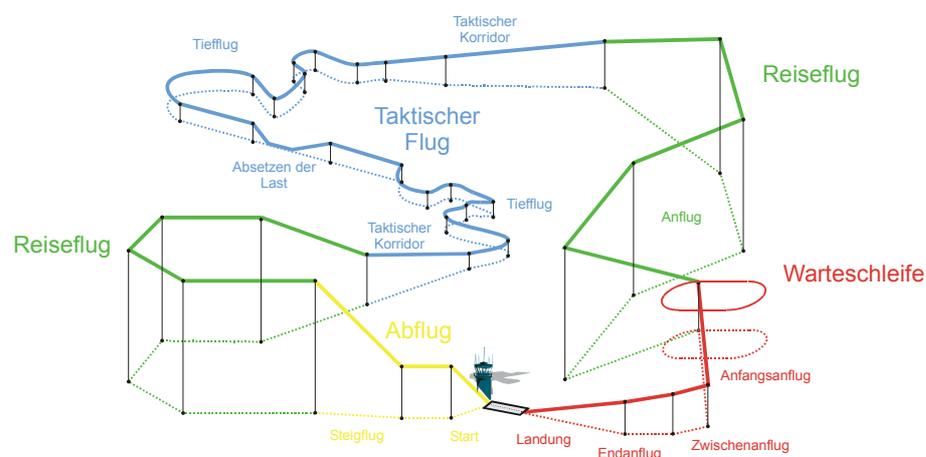
Die Nachrichtenlage umfasst drei Bereiche:

- Air-Order-of-Battle (Verteilung und Belegung der Flugplätze)
- SAM-Order-of-Battle (FLAK und FLARAK Lage)
- Electronic-Order-of-Battle (Lage der Anlagen zur elektronischen Kampfführung)

- **Combat-Crew-Concept**

Die Aufgaben, die es an Boden vor dem Flug zu erledigen gilt, werden festgelegt, und die Zeiten für deren Durchführung werden bekannt gegeben.

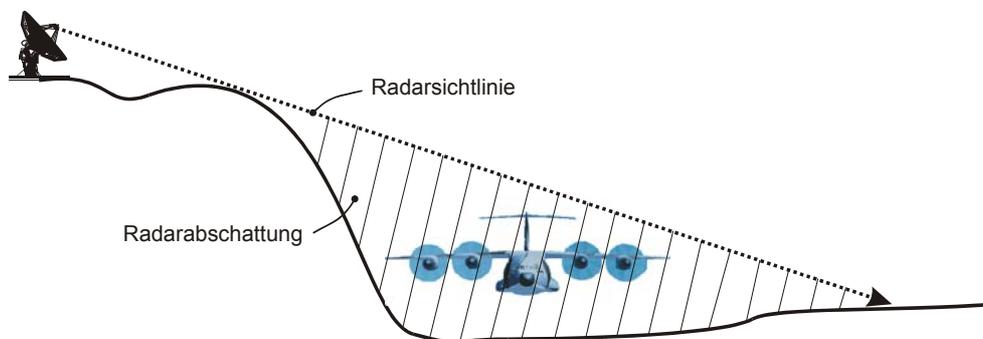
Die Durchführung eines taktischen Transportfluges, wie er in dieser Arbeit angenommen wurde, erfolgt gemäß dem in Abbildung 85 dargestellten Flugprofil.



**Abbildung 85: Missionsprofil einer militärischen Transportmission**

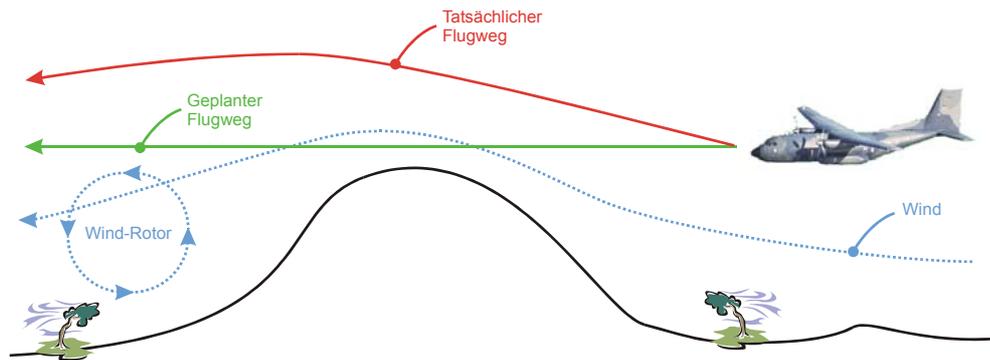
Der Hin- und Rückflug zum Operationsgebiet erfolgt unter IFR Flugregeln (jedoch unter militärischer Luftaufsicht), der operationelle Anteil im taktischen Gebiet nach Sichtflugverfahren.

Ein weiteres Kriterium, welches mit der Durchführung eines militärtaktischen Transportfluges einhergeht, sind die Rahmenbedingungen für die erfliegenen Flugleistungen. Wird im zivilen Flug nach Instrumentenflugregeln geflogen, erfolgt der militärische Flug teils unkontrolliert, oft nach Sichtflugregeln und in Bodennähe. Hinzu kommt, dass dem militärischen Transportflieger umfangreichere und gewichtigere Kommunikations- und Identifizierungsaufgaben zufallen. Die Identifizierungsverfahren (z.B. aus Codetafeln) dienen bei der Kontaktaufnahme mit der Bodenstelle der Authentisierung. Die elektromagnetische Signatur des Flugzeugs stellt im taktischen Flug eine entscheidende Rolle dar, da sie das Aufspüren durch den Feind entscheidend beeinflusst. Aus diesem Grunde erhält die Besatzung durch den Nachrichtenoffizier vor dem Start einen EMCO-Status (Emission-Control), der Richtlinien über den Gebrauch der mitgeführten Sender an Bord umfasst. Zusätzlich wird darin die Art des Funkverkehrs angegeben, ob dieser taktisch, zivil oder gar nicht zu erfolgen hat. Bei den militärtaktischen Flugverfahren handelt es sich um Manöver und Verhaltensweisen, wie sie während des Fluges im feindlichen Operationsgebiet anzuwenden sind. Hierzu zählt das Unterfliegen der Bedrohung (siehe Abbildung 86).



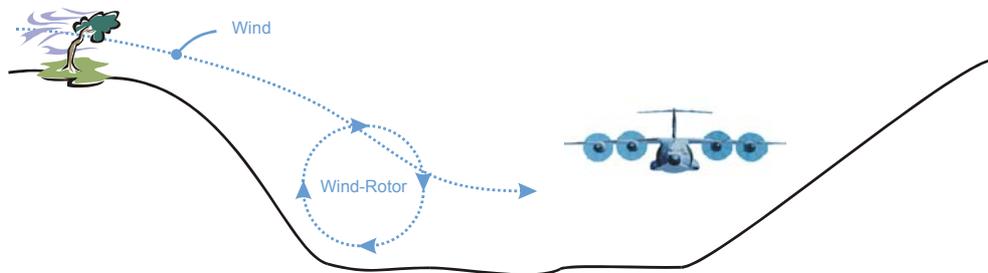
**Abbildung 86: Bedrohungsangepasste Flugwegplanung**

Die Berücksichtigung von Luv- und Lee-Effekten, welche besonders beim bodennahen Flug auftreten (siehe Abbildung 87), stellen weitere wichtige Verfahren dar.



**Abbildung 87: Berücksichtigung der Luv-/Lee-Gebiete bei der Flugwegplanung**

In ausgeprägten Tallagen müssen Turbulenzen, welche durch Abwindfelder (siehe Abbildung 88) entstehen, im Flugverlauf Berücksichtigung finden.



**Abbildung 88: Berücksichtigung von Hangturbulenzen bei der Flugwegplanung**

Die Aufgaben an Bord eines militärischen Transportflugzeugs sind im Combat Crew Concept definiert und verteilen sich auf eine 5-köpfige Besatzung, bestehend aus

- Kommandant,
- Copilot,
- Navigator,
- Bordtechniker und
- Luftlademeister.

Die den einzelnen Besatzungsmitgliedern zugeordneten Aufgaben sind in der nachfolgenden Tabelle 4 aufgestellt.

<b>Crew Coordination Concept</b>	<b>Boden</b>		<b>Luft</b>
<b>Kommandant</b>	Studiert die ATO Koordiniert Unterstützung am Boden und in der Luft Bereitet Einsatzbesprechung vor und erarbeitet Täuschungspläne für den Feind		Überwacht taktische Lage Regelt Funkverkehr
<b>Copilot</b>	Kartenstudium Flugroutenplanung Abwurfplanung	Berechnet Leistungsdaten Überträgt Flugroute in zweite Karte	Steuert Luftfahrzeug
<b>Navigator</b>		Berechnet Startzeit Berechnet Abwurfpunkt	Koppelnavigation Zeitmanagement (mit Bordmechaniker) Transponder-Verfahren Luftraumüberwachung
<b>Bordtechniker</b>	Erarbeitet für den Fall einer Notlandung einen Verhaltensplan. Überprüft Luftfahrzeug		Überwacht Triebwerkinstrumente Zeitmanagement (mit Navigator) Luftraumüberwachung Authentisierungsverfahren
<b>Luftlademeister</b>			Luftraumbeobachtung Sicherheit im Laderaum Absetzen der Last

**Tabelle 4: Crew-Coordination-Concept eines militärischen Transportflugzeugs**

Der Erstellung der Fragebögen wurde zunächst eine a-priori-Analyse bezüglich der sequentiellen Durchführung und eine aufgabenbezogene Betrachtung einer militärischen Transportmission vorangestellt. Die flugphasenbezogene Unterteilung dient auch der GEM zum strukturellen Ansatz. Wie jedoch bereits erwähnt, soll sich die direkte Expertenbefragung auf das Wissen konzentrieren, welches unveröffentlicht und nur schlecht strukturiert ist. Hier werden diejenigen Flugphasen analysiert, welche den militärtaktischen Flug während einer Transportmission charakterisieren (siehe auch Abbildung 85). Dies sind der Transitflug durch einen Korridor, der Flug in geringer Bodennähe zum Absetzpunkt und das Absetzen der Last.

Hierbei wurden mit Unterstützung eines Experten die für eine solche Mission relevanten Flugphasen und die dort auftretenden Flugführungsaufgaben ermittelt. Erste Hypothesen bezüglich der erwarteten Ziele und Teilziele wurden aufgestellt. Diese dienen dem Wissensingenieur zur Formulierung der Fragestellungen für die einzelnen GEM-Sitzungen. Die während der a-priori-Analyse gewonnen Erkenntnisse sollen hier dargestellt werden.

### 6.1.3.3 Taktischer Korridor

Das Aufstellen von Hypothesen beginnt in Analogie zum Flugverlauf mit dem taktischen Korridor. Als Hypothese für ein Fundamentalziel wurde hier das Einhalten der militärischen Luftraumordnung angenommen, welches in weitere Instrumentalziele zerfällt (siehe Tabelle 5).

Bedeutung	Passieren der RBFA (Rear Boundary of the Forward Area) und der FLOT (Forward Line of the Own Troops)
Hypothese Fundamentalziele	Vorgegebene militärische Luftraumstruktur beachten
Hypothese Instrumentalziele	Laterale Grenzen des Korridors beachten Vertikale Grenzen des Korridors beachten Zeitplan für Durchflug durch den Korridor einhalten

**Tabelle 5: Ziel-Hypothesen für die Flugphase „Taktischer Korridor“**

### 6.1.3.4 Taktischer Tiefflug

Dem Durchfliegen des taktischen Korridors folgt der Tiefflug durch das Operationsgebiet. Die *a-priori*-Analyse des militärischen Transportfluges zeigt, dass insbesondere die Vermeidung von Bedrohungsstellungen durch das Ausnutzen von natürlichen Geländeformationen anzustreben sind. Diese sind in Form von Hypothesen in die Konzeption der Wissensakquisition für diesen Flugabschnitt eingeflossen. Die sich hieraus ergebenden Annahmen für die Instrumentalziele sind in Tabelle 6 aufgeführt.

Bedeutung	Flug unter militärtaktischen Bedingungen und feindlicher Bedrohung
Hypothese Fundamentalziel	Flugzeug und Beladung sollen möglichst sicher, unter Einhaltung des Zeitplanes und unter idealer Ausnutzung der Tarnung und des Geländes zum Absetzpunkt gelangen. Das fliegerische Risiko ist dabei so gering wie möglich zu halten
Hypothese Instrumentalziele	Flugweg frei von Gelände halten Flugweg frei von Hindernissen halten Abstand zu Hindernissen wg. Radar-Signatur minimieren Elektromagnetische Abstrahlung minimieren Thermische Abstrahlung minimieren Flug über offenem Gelände ( See, Ebene, etc.) vermeiden Zeitplanung einhalten Kraftstoffressourcen managen Bodenbedrohung (SAMs, etc.) um- bzw. unterfliegen Vorgegebene militärische Luftraumstruktur beachten

**Tabelle 6: Ziel-Hypothesen für die Flugphase „Taktischer Tiefflug“**

### 6.1.3.5 Absetzen der Last

Ziel der militärischen Transportmission ist das Ausliefern einer Last im Operationsgebiet, woraus sich direkt die Hypothese für das Fundamentalziel in diesem Flugabschnitt ergibt.

Diese Annahmen sind in Tabelle 7 zu finden.

Bedeutung	Taktisches Absetzmanöver von Lasten über der Abwurfzone
Hypothese Fundamentalziel	Ladung im vorgeschriebenen Gebiet abgesetzt
Hypothese Instrumentalziele	Minimale Gefährdung, sowohl für sich selbst, als auch für die Last Zeitminimales Absetzen wg. potentieller Feindbedrohung Vorgegebene militärische Luftraumstruktur beachten

**Tabelle 7: Ziel-Hypothesen für die Flugphase „Absetzen einer Last“**

Der Analyse der Flugphasen folgt eine Studie bezüglich potentieller übergeordneter Ziele für die Flugführungsaufgaben, welche sich als eine den Flugphasen untergeordnete Kategorie darstellen. Diese unterteilen sich in die fünf Aufgabenkategorien:

- Flugzeugsteuerung,
- Navigation,
- Kommunikation,
- Systemüberwachung und
- Lageerfassung und taktische Planung.

Hierbei werden zunächst den Aufgabenkategorien untergeordnete Ziele identifiziert.

Das Aufstellen von Ziel-Hypothesen für die Aufgabenkategorien der Flugführungsaufgabe beschränkt sich auf diejenigen Handlungsziele, welche speziell für die taktische Fliegerei zu verfolgen sind. Die Flugführungsaufgaben stellen die Instrumentalziele dar und sind bereits durch die Analyse der zivilen Flugführungsziele identifiziert. Sie sind in Tabelle 8 aufgeführt.

Hypothese Instrumentalziele Flugzeugsteuerung	Vorgegebene/vorgeplante Flughöhe einhalten Vorgegebene/vorgeplante Geschwindigkeit einhalten Vorgegebene/vorgeplante Kurse einhalten Berücksichtigung von eskortierenden Verkehrsteilnehmern Betriebsgrenzen einhalten Prozeduren gemäß Checklisten durchführen
Hypothese Instrumentalziele Navigation	Einhalten der vorausberechneten Flugzeiten Reduzieren der Flugstrecke bei Verspätung Suchen von Auffangpunkten Ständige Aktualisierung der Streckenberechnung Anzeigen der Abweichung vom geplanten Flugweg Vorschlagen von Kurskorrekturen Einhalten der militärischen/zivilen Korridore
Hypothese Instrumentalziele Kommunikation	Authentisierung mit Boden-/Luftkoordinierungsstelle EMCON Status einhalten Störsender-Management („Chattermark Procedures“) IFF (Identification-Friend-Foe) Procedures einhalten
Hypothese Instrumentalziele Systemüberwachung	Kraftstoffverbrauchsbegrenzung einhalten Ausreichende Systemfunktionalität gewährleisten
Hypothese Instrumentalziele Lageerfassung und Taktische Planung	Wärmestrahlung durch Leistungsminimierung reduzieren Meiden offener Geländeflächen Radarabstrahlung minimieren Minimalen Hindernisabstand einhalten Minimale Flughöhe einhalten Fluglärm reduzieren Ein-Anflug-Abwürfe erreichen Kurze Zeiten einhalten Militärische Korridore einhalten

**Tabelle 8: Ziel-Hypothesen für die taktischen Anteile der Flugführungsaufgabe**

#### 6.1.3.6 Durchführung

Die Wissensakquisition für die Handlungsziele im taktischen Tiefflug erfolgte in enger Zusammenarbeit mit der 1./Lufttransportgeschwader 61, Landsberg, im Zeitraum Oktober 1999 bis März 2000. Der lange Zeitraum für die Durchführung ergab sich insbesondere durch die eingeschränkte Verfügbarkeit der vier Teilnehmer. Bei der Auswahl wurde Wert auf eine breite Streuung gelegt. Hierbei standen Alter, Flugerfahrung, Aufgabe im Cockpit und Ausbildungsstand im Vordergrund. Die Daten der vier Teilnehmer sind Tabelle 9 zu entnehmen.

	Alter	Flugerfahrung gesamt	Flugerfahrung milit. Transporter	Position Cockpit	Taktischer Ausbildungs- stand
Teilnehmer 1	28 a	1020 h	800 h	Co-Pilot	Combat Ready+
Teilnehmer 2	36 a	3000 h	2700 h	Aircraft Commander	Combat Ready
Teilnehmer 3	38 a	4400 h	3400 h	Aircraft Commander	Combat Ready+
Teilnehmer 4	40 a	4500 h	4000 h	Tactical Systems Officer	Combat Ready+
<b>Mittel:</b>	<b>35,5 a</b>	<b>3230 h</b>	<b>2725 h</b>	–	–

**Tabelle 9: Versuchspersonen an der GEM**

Die Expertenbefragung erfolgte an drei Terminen:

**Termin 1:** „Ein-/Ausflug in das Operationsgebiet“

**Termin 2:** „Tiefflug zum/vom Zielgebiet“

**Termin 3:** „Absetzen der Last“

Hierbei wurden den Teilnehmern zunächst in einer kurzen Einführung Hintergrundinformationen zu der Befragung und die zu betrachtende Flugphase mitgeteilt. Für die Teilnehmer sowie für den Befragungsleiter wurde ein Zeitrahmen wie folgt vorgegeben:

- Briefing einer militärischen Lage und eines Missionsauftrages (30 Minuten)
- Sammeln bzw. Überarbeiten von Aussagen auf dem Fragebogen in Einzelarbeit (10 Minuten je Teilnehmer)
- Diskussion der gewonnenen Aussagen und Strukturierung der Ergebnisse (60 Minuten)
- Bewertung der Aussagen untereinander bezüglich Priorität (45 Minuten)

Ergebnis der Wissenserhebung sind die Handlungsziele während der taktischen Flugabschnitte. Hierbei handelt es sich zunächst um rein verbale Aussagen der Experten in Form von gewichteten Schlagwortlisten. Diese sind im Anhang 10.2 zu finden. Das Verfahren nach (Boy, 1996) hat sich für das hier vorliegende Problem bewährt, da es ein methodisch gut strukturiertes Vorgehen ermöglicht hat und auch von den beteiligten Besatzungen akzeptiert

wurde. Die gewonnenen Ergebnisse können für die nachfolgenden Schritte der Modellierung von Einzelzielen genutzt werden.

#### 6.1.4 REPRÄSENTATION DES WISSENS

Die Repräsentation des gewonnenen Wissens erfolgt gemäß Tabelle 10 und basiert auf den in Kapitel 4 erarbeiteten Grundlagen.

Wissen Ursprung	Funktionen und Diagramme	Fuzzy Logik	Karte	Freie Wissensform	(Neuronales Netz)
Dokumentation	●	–	–	●	–
Experte	●	●	–	●	–
Domäne	●	–	●	●	●

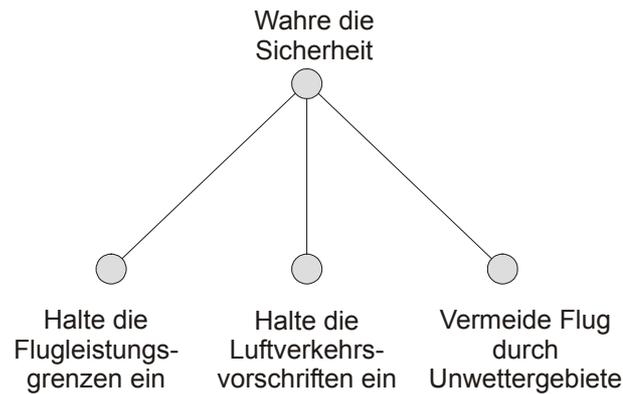
**Tabelle 10: Ursprung und Modellierung des Modells der Handlungsziele**

## 6.2 MAKROSKOPISCHE STRUKTUR DES MODELLS DER HANDLUNGSZIELE

Die aus der Wissensakquisition gewonnenen Handlungsziele münden in einer Zielhierarchie, die einen wesentlichen Teil des Hintergrundwissens des kognitiven Cockpitassistenzsystems darstellt. Nachfolgend soll der Aufbau dieses Modells diskutiert werden.

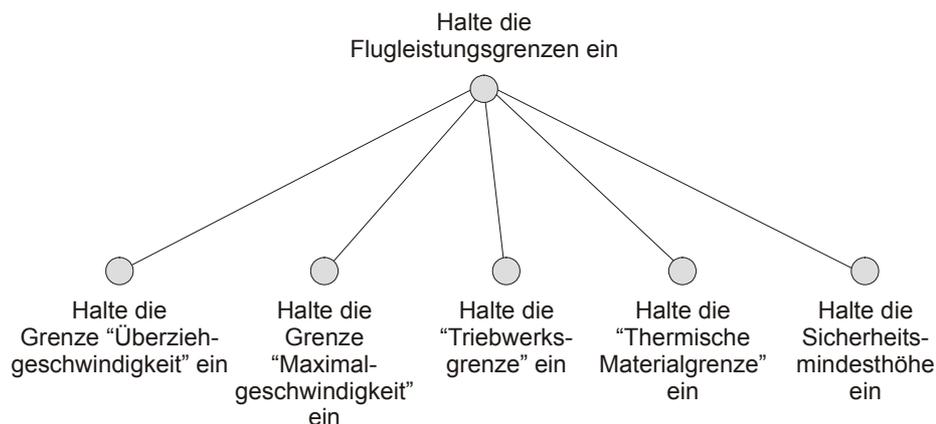
### 6.2.1 GEFÄHRDUNGSMODELL

Das *Gefährdungsmodell* repräsentiert diejenigen sicherheitsrelevanten Ziele der Flugführungsaufgabe, welche für die Durchführung des Fluges unter Gewährleistung des Bestandes von Material und Leben zu jedem Zeitpunkt erfüllt sein müssen bzw. deren Umsetzung vor die Erfüllung anderer Ziele gestellt werden muss. Es basiert auf den in Abbildung 89 dargestellten Zielen.



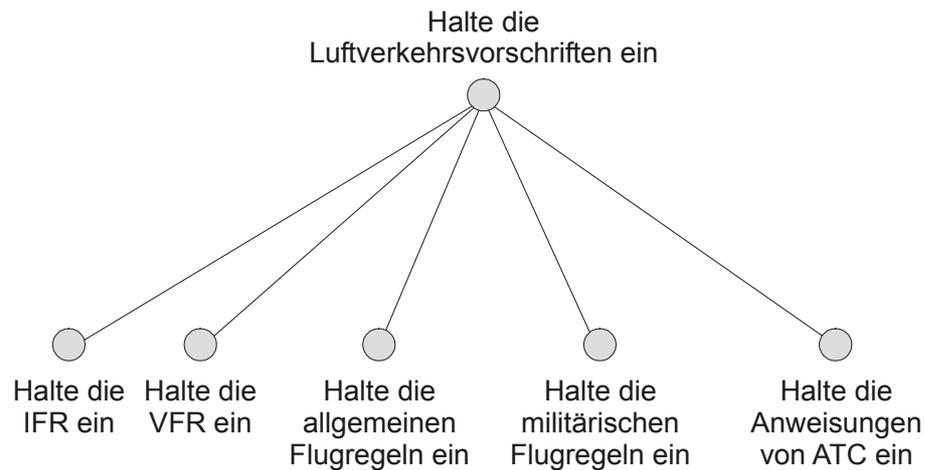
**Abbildung 89: Gefährdungsmodell**

Die Flugleistungsgrenzen einzuhalten, stellt eines der drei Teilmodelle des Gefährdungsmodells dar (siehe Abbildung 90). Es zerfällt in die Instrumentalziele, welche aus den Flugleistungsgrenzen der Überziehggeschwindigkeit, der Maximalgeschwindigkeit, der Triebwerksleistungsgrenze, der thermischen Materialgrenzen und der Sicherheitsmindesthöhe zusammengesetzt sind.



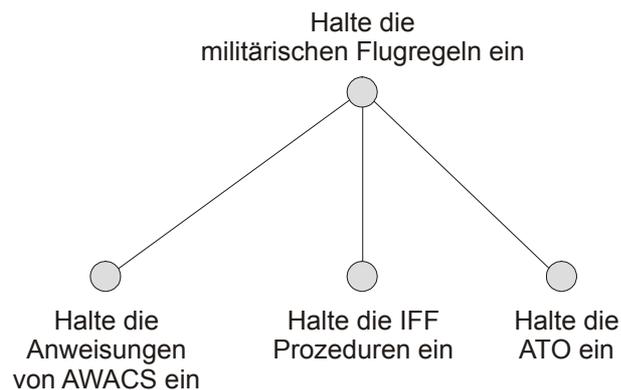
**Abbildung 90: Modell der Flugleistungsgrenzen im Teilmodell der Gefährdung**

Das Ziel der Einhaltung der Luftverkehrsvorschriften dient ebenfalls dem Ziel der Sicherheit. Neben den Vorschriften für Flüge unter Instrumenten- und Sichtflugbedingungen stehen die Allgemeinen Flugregeln (siehe Abbildung 91).



**Abbildung 91: Gefährdungsmodell der *Luftverkehrsvorschriften* im Teilmodell *Gefährdung***

Die für einen militärischen Transporter relevanten Vorschriften sind Abbildung 92 zu entnehmen.

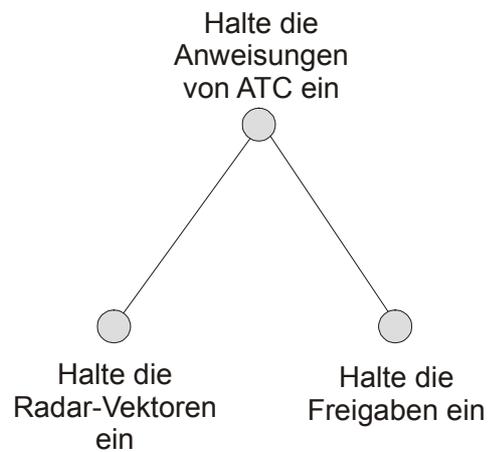


**Abbildung 92: Modell der *Militärischen Flugregeln* im Teilmodell der *Luftverkehrsvorschriften***

Die militärischen Flugregeln spielen insbesondere für den Flug im feindlichen Operationsgebiet eine entscheidende Rolle, dürfen aber nicht mit taktischen Zielvorstellungen verwechselt werden. Militärische Regeln ordnen vielmehr das Geschehen am Kriegsschauplatz und dienen der Koordination der Streitkräfte. Sie sind durch das Einhalten der Vorgaben von Führungsleitstellen (AWACS), den Instrumentarien zur Freund/Feind-Erkennung und die ‚Air-Tasking-Order‘ definiert.

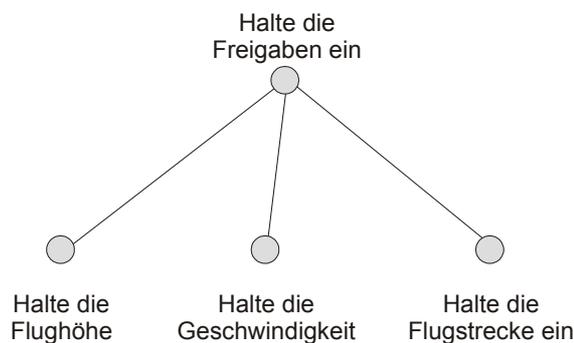
In den zivilen Flugabschnitten stellt die Flugsicherung (ATC) in Form von Flugverkehrs-anweisungen durch Freigaben und Vektoren die Verkehrsordnung sicher. Neben

außerplanmäßigen Radar-Vektoren ist es Aufgabe von ATC, koordinierte Freigaben für den geplanten Flugverlauf mitzuteilen (siehe Abbildung 93).



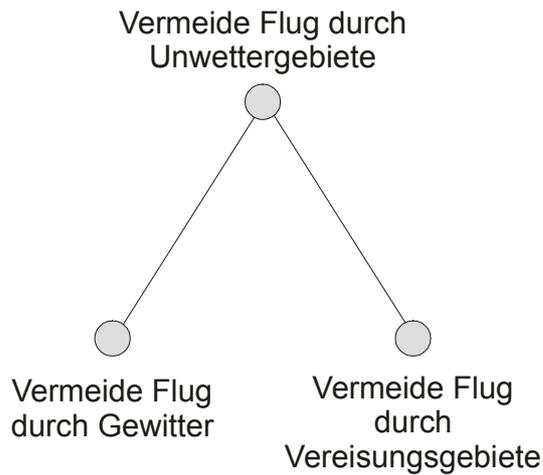
**Abbildung 93:** Modell der *ATC-Anweisungen* im Teilmodell der *Luftverkehrsvorschriften*

Freigaben betreffen den vorausliegenden Flugweg in gleichem Maße wie Geschwindigkeit und Flughöhe (siehe Abbildung 94).



**Abbildung 94:** Modell der *Freigaben* im Teilmodell der *ATC-Anweisungen*

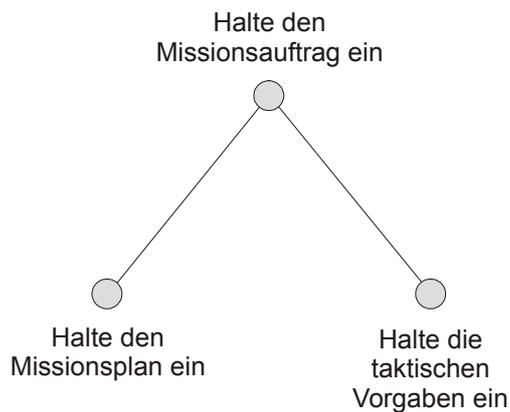
Dritter Teil des Gefährdungsmodells stellen Gefahren durch Unwetter dar. Hierzu gehören sowohl lokale Gewitter, welche durch heftige Turbulenzen und Niederschläge den Zustand des Luftfahrzeugs gefährden, als auch Vereisungsgebiete (siehe Abbildung 95).



**Abbildung 95: Modell *Unwetter* im *Gefährdungsmodell***

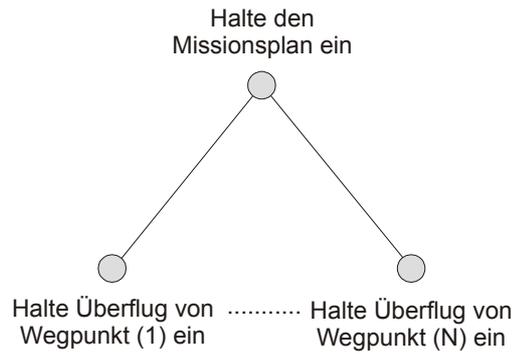
### 6.2.2 MODELL DES MISSIONSAUFTRAGES

Der Missionsauftrag (siehe Abbildung 96) stellt aus militärischer Sicht die der Besatzung aufgetragenen Ziele dar, welche es mit den zur Verfügung stehenden Ressourcen zu erreichen gilt.

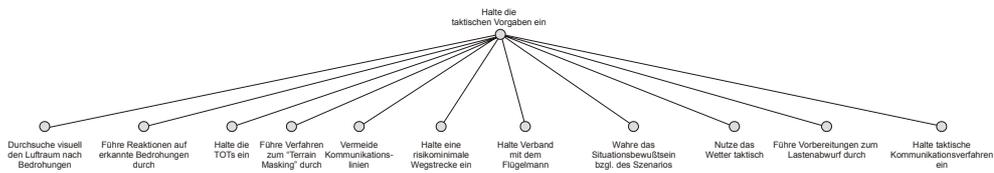


**Abbildung 96: Modell des *Missionauftrages***

Der Missionsauftrag ist in der Regel durch Einzelziele formuliert, die einer Reduzierung der Aufgabenkomplexität dienen. Militärische Missionsaufträge umfassen neben dem Missionsplan (siehe Abbildung 97), der sequenziell abzuarbeitende Vorgaben beinhaltet, auch taktische Richtlinien (siehe Abbildung 98).



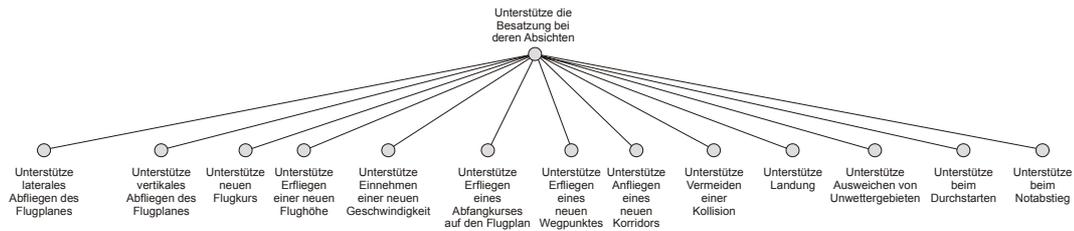
**Abbildung 97: Modell des *Missionsplanes* im Teilmodell des *Missionsauftrages***



**Abbildung 98: Modell des *taktischen Verhaltens* im Teilmodell des *Missionsauftrages***

### 6.2.3 MODELL DER BESATZUNGSABSICHTEN

Die Anforderung nach Kooperation bedarf einer Repräsentation der verfolgten Absichten der Besatzung. Hierzu sind diejenigen Ziele, welche zur Veranlagung eines kooperativen Verhaltens vonnöten sind, ebenfalls in der Modellierung zu berücksichtigen. Die Forschungsarbeiten am Institut für Systemdynamik und Flugmechanik haben unter anderem die Interpretation der verfolgten Pilotenabsichten zum Gegenstand gehabt. Das Ergebnis dieser Funktionalität, welche bereits im System CAMA Umsetzung gefunden hat, ist in das Modell der Handlungsziele in Form des Teilmodells *Unterstützung der Besatzung bei deren Absichten* eingeflossen. Hier sind neben den erklärten Absichten (Anweisungen der Operateure an das System bezüglich Flugwegplanung und allgemeiner Anweisungen) zwecks Verpflichtung des Assistenzsystems zur Zusammenarbeit auch die vom Assistenzsystem selbstständig erkannten Absichten repräsentiert (siehe Abbildung 99).



**Abbildung 99: Modell der *Besatzungsabsichten***

Exemplarisch ausgewählte Zielmodelle aus den oben genannten Kategorien sind in Aufgabe, Funktion und umgesetzter Wissensrepräsentationsform in Anhang 10.4 auf der mikroskopischen Ebene dargestellt und macht die Modellierung im Detail deutlich.

### 6.3 ERMITTLUNG DER ZIELGEWICHTE

Die Ermittlung der Zielgewichte erfolgt mit Hilfe des *DirectRatio*-Verfahrens. Es soll hier beispielhaft anhand ausgewählter Instrumentalziele, als auch für die Fundamentalziele: *Wahre die Sicherheit*, *Halte den Missionauftrag ein* und *Unterstütze die Besatzung bei deren Absichten* diskutiert werden. Qualitativ unterscheiden sich diese beiden Zielklassen durch ihren Einfluss auf das Gesamtsystem. Während sich die Instrumentalziele auf einen lokalen Definitionsbereich beziehen, haben die übergeordneten Fundamentalziele direkte Auswirkung auf das Verhalten des Gesamtsystems.

#### 6.3.1 INSTRUMENTALZIELE

Die Ziele, welche für die Einhaltung der Flugleistungsgrenzen ermittelt wurden, sind Teil des Gefährdungsmodells. Ziele des Gefährdungsmodells sind direkt verantwortlich für Erhaltung von Leib und Leben sowie des Materials. Sie sollen an dieser Stelle Beispiel für die Ermittlung der Zielgewichte der Instrumentalziele geben.

##### 6.3.1.1 Priorisierung

Die Priorisierung stellt die qualitative Ordnung der Zielmenge nach einer subjektiven Wichtigkeit dar. Hierdurch werden für den Gewichtungsvorgang bereits fundamentale Grundannahmen getroffen. Die Priorisierung der Flugbereichsgrenzen soll hier wie folgt getroffen werden:

- (1) Halte die Sicherheitsmindesthöhe ein
- (2) Halte die Grenze der Überziehgeschwindigkeit ein

- (3) Halte die Grenze der Maximalgeschwindigkeit ein
- (4) Halte die thermische Materialgrenze ein
- (5) Halte die Triebwerksgrenze ein

Diese Strategie zur Priorisierung ist auf Grund des Gefährdungspotentials und der Reaktionszeit zur Einleitung von Problemlösungsmaßnahmen entwickelt worden. Hieraus ergibt sich die Geländeannäherung als subjektiv wichtigstes Vermeidungsziel, da die Kollision mit dem Gelände ein hohes Risiko für Material und Besatzung darstellt und in der Regel eine kurze Reaktionszeit erfordert. Niedrigste Priorität erhält die Triebwerksgrenze, da sie weder ein hohes Gefährdungspotential darstellt, noch auf spontane Reaktionen angewiesen ist.

### 6.3.1.2 Quantifizierung

Die Quantifizierung soll hier an drei Beispielen diskutiert werden. Ausgehend von der Priorität der Einzelziele werden die Gewichte bestimmt, die sich auf Basis der *DirectRatio*-Methode ergeben (siehe Tabelle 11).

		Sicherheits- höhe	Überzieh- grenze	Maximal- Geschw.	Thermische Grenze	Triebwerks- Grenze
Bsp. (1)	Rating	3	2	1	2	
	Gewicht	1.0	0.66	0.44	0.33	0.11
Bsp. (2)	Rating	3	2	3	2	–
	Gewicht	1.0	0.81	0.54	0.36	0.09
Bsp. (3)	Rating	1.2	1.5	2.3	1.7	–
	Gewicht	1.0	0.78	0.48	0.28	0.13

**Tabelle 11: Ergebnisse des *DirectRatio*-Verfahrens für die Flugleistungsgrenzen**

### 6.3.2 FUNDAMENTALZIELE

Die Gewichtung der übergeordneten Instrumentalziele *Sicherheit*, *Missionsauftrag* und *Besatzungsunterstützung* sind wohl der das Systemverhalten am tiefgreifendsten beeinflussende Faktor. Aus diesem Grunde sollen an dieser Stelle zwei unterschiedliche Strategien für die Entwurfsgrundsätze und damit die Auslegung eines kognitiven Assistenzsystems betrachtet werden.

#### 6.3.2.1 Kollegiales Systemverhalten

Kollegiales Systemsverhalten soll durch die Höhergewichtung des *Missionsauftrages* gegenüber der *Besatzungsunterstützung* charakterisiert sein. Ein solches System ist in der Art

und Weise veranlagt, dass es die Erfüllung des *Missionsauftrages* vor die *Besatzungsunterstützung* stellt. Hieraus ergibt sich die folgende Zielpriorität:

- (1) Wahre die Sicherheit
- (2) Erfülle den Missionsauftrag
- (3) Unterstütze die Besatzung bei deren Absichten

Die Ergebnisse der Quantifizierung sind in Tabelle 12 dargestellt

		Sicherheit	Mission	Besatzung
Bsp. (1)	Rating	2	2	
	Gewichtung	1.0	0.6	0.2
Bsp. (2)	Rating	2	1	
	Gewichtung	1.0	0.5	0.25
Bsp. (3)	Rating	3	1	
	Gewichtung	1.0	0.4	0.2

**Tabelle 12: Ergebnisse des *DirectRatio*-Verfahrens für ein kollegiales System**

### 6.3.2.2 Kooperatives Systemverhalten

Das kooperative System stellt das Erfüllen der *Besatzungsunterstützung* vor die Erfüllung des *Missionsauftrages*. Hieraus ergibt sich die folgende Zielpriorität

- (1) Wahre die Sicherheit
- (2) Unterstütze die Besatzung bei deren Absichten
- (3) Erfülle den Missionsauftrag

Und eine Gewichtung der Einzelziele wie in Tabelle 13:

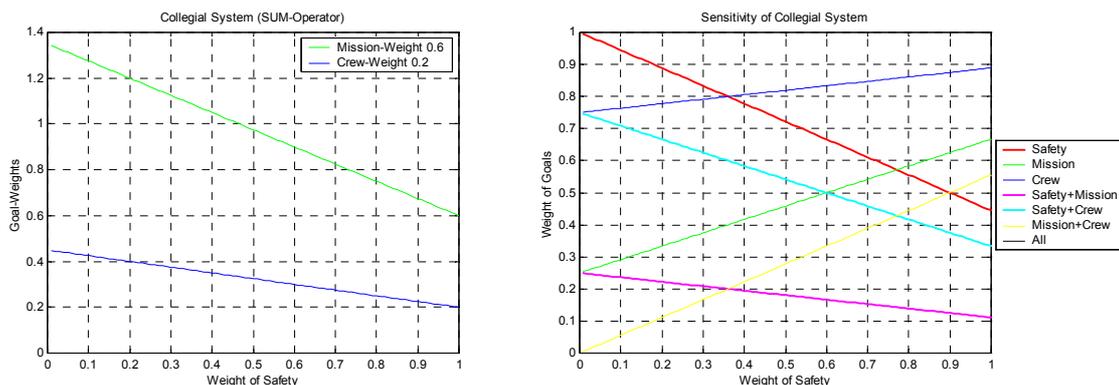
		Sicherheit	Besatzung	Mission
Bsp. (1)	Rating	2	2	–
	Gewichtung	1	0.6	0.2
Bsp. (2)	Rating	2	1	–
	Gewichtung	1	0.5	0.25
Bsp. (3)	Rating	3	1	–
	Gewichtung	1	0.4	0.2

**Tabelle 13: Ergebnisse des *DirectRatio*-Verfahrens für ein kooperatives System**

## 6.4 SENSITIVITÄTSANALYSE DER HANDLUNGSZIELE

Die für die Anwendung kritischen übergeordneten Fundamentalziele sollen hier einer Sensitivitätsanalyse anhand von drei Verhaltenskonfigurationen unterzogen werden. Sensitivitätsanalysen dienen dem Studium eines Kriteriums und der Beeinflussung durch dessen Nachbarkriterien. Insbesondere für die Verhaltensdefinition eines kognitiven Systems sind solche Verfahren mit besonderer Sorgfalt umzusetzen. Ziel ist es hier, eine ausreichende Sensitivität des Ziels *Wahre die Sicherheit* nachzuweisen.

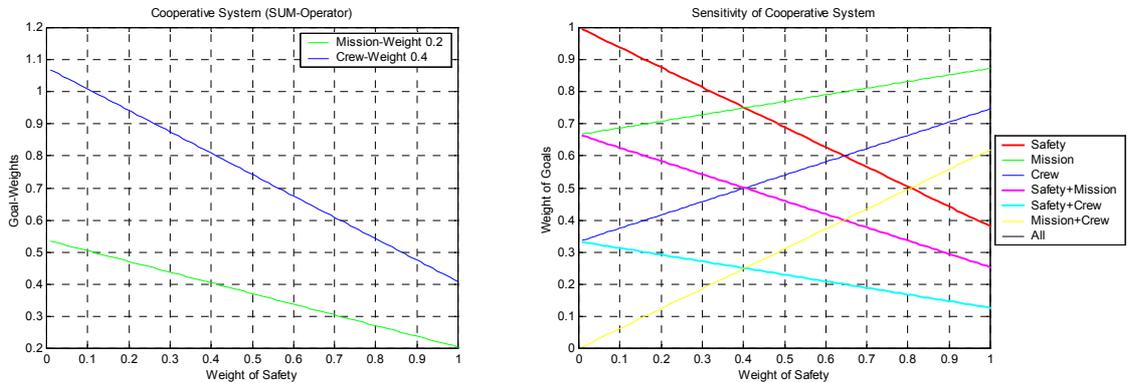
Aus den im rechten Teil der Abbildung 100 dargestellten Sensitivitätsdiagramms eines kollegialen Systems ( $w_{Mission} = 0.6$  und  $w_{Crew} = 0.2$ ) erkennt man, dass bei den zugrunde liegenden Gewichten von *Halte den Missionsauftrag ein* und *Unterstütze die Besatzung bei deren Absichten* die Sensitivitätsgrenze eine Erstreckung zwischen 0.36 (Konfliktsituation im Bereich Besatzungsunterstützung) und 0.9 (Konfliktsituation im Bereich Mission + Besatzungsunterstützung) besitzt. Dies bedeutet, dass eine Reduktion des Zielgewichts Sicherheit bis auf den Wert 0.9 ohne Verlust an ausreichender Prägnanz möglich wäre. Mit dem gewählten Wert von 1.0 für das Ziel Sicherheit ist diese Forderung ausreichend erfüllt. In Abbildung 100 und Abbildung 101 sind Sensitivitätsdiagramme sowohl für ein kollegiales als auch für ein kooperatives System unter der Annahme eines additiven Verknüpfungsoperators und der ermittelten Gewichte gemäß Bsp. (1) (siehe Tabelle 12) dargestellt.



**Abbildung 100:** Kollegiales System: Zielgewichte *Halte den Missionsauftrag ein* und *Unterstütze die Besatzung bei deren Absichten* in Abhängigkeit vom additiven Zielgewicht der *Sicherheit* (links) und Alternativen für die übergeordneten Ziele (rechts).

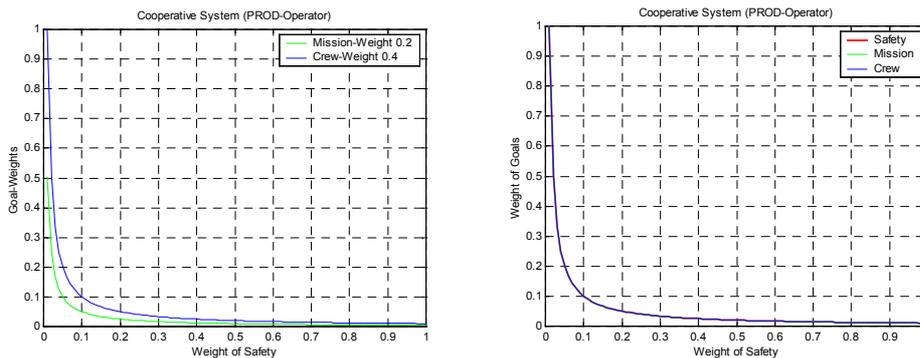
Für den Fall eines kooperativen Systems gemäß Bsp. (1) aus Tabelle 13 ( $w_{Crew} = 0.4$  und  $w_{Mission} = 0.2$ ) lassen sich Sensitivitätsgrenzen im Bereich 0.4 (Konfliktsituation: *Mission*) und

0.81 (Konfliktsituation: *Mission* und *Besatzungsunterstützung*) aufzeigen. Hieraus ergibt sich eine Grenze für das Zielgewicht der Sicherheit von 0.81.



**Abbildung 101: Kooperatives System: Zielgewichte der Halte den *Missionsauftrag ein* und *Unterstütze die Besatzung bei deren Absichten* in Abhängigkeit vom additiven Zielgewicht der *Sicherheit* (links) und Alternativen für die übergeordneten Ziele (rechts).**

Geht man an Stelle eines additiven von einem produktiven Verknüpfungsoperator aus, so ergibt sich die Sensitivitätstafel wie in Abbildung 102 dargestellt. Die direkte Abhängigkeit der Zielgewichte untereinander über einen multiplikatorischen Zusammenhang zeigt, dass eine vollständige Kompensation der Gewichte erfolgt. Anders ausgedrückt. Es liegt eine direkte Sensitivität der Einzelziele auf den Erfülltheitsgrad des Oberziels vor.

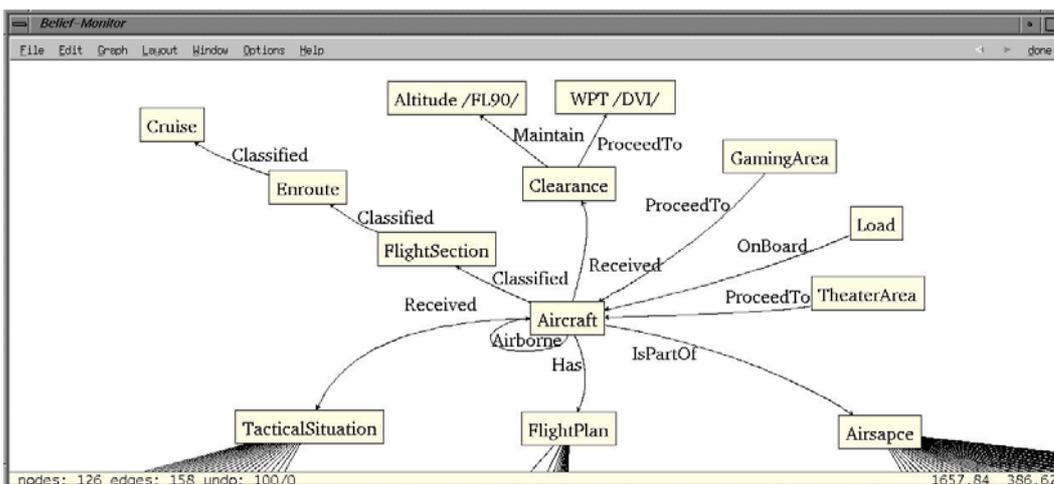


**Abbildung 102: Kooperatives System: Zielgewichte von Halte den *Missionsauftrag ein* und *Unterstütze die Besatzung bei deren Absichten* in Abhängigkeit vom produktivem Zielgewicht von *Wahre die Sicherheit* (links) und Alternativen für die übergeordneten Ziele (rechts).**

Daraus ergibt sich, dass die Sensitivitätsanalyse ein geeignetes Mittel darstellt, um auf Basis der ermittelten Zielehierarchie Untersuchungen bezüglich der Gewährleistung des Systemverhaltens auf Basis der Entwurfskriterien durchzuführen.

## 6.5 STRUKTUR DER RELEVANTEN OBJEKTE

Die Beschreibung derjenigen Situationselemente, welche konkreter (objekthafter) Natur sind (keine Datenbanken, Karten, etc.), erfolgt in einer relationalen Struktur. Diese Repräsentationsform ist durch Situationselemente und Relationen in einem Graphen realisiert, in dem die Knoten die relevanten Objekte und die Kanten die Relationen darstellen. Für die Umsetzung in dieser Arbeit wurde ein vereinfachtes Situationsmodell entwickelt, welches eine auf das eigene Flugzeug bezogene Situationsinterpretation umsetzt (siehe Abbildung 103).



**Abbildung 103: Modell der interpretierten Lage während des Reisefluges**

Der Teilprozess der *Situationsinterpretation* konzentriert sich in dieser Arbeit auf die für die Flugführung relevanten Bereiche. Hierzu zählen das eigene Flugzeug und sein Zustand bezüglich des Missionsauftrages, seine derzeitigen Flugphasen, der Luftraum und die Verkehrsinfrastruktur sowie die taktischen Elemente in Form der aufgeklärten Lage im Operationsgebiet. In Abbildung 103 ist der Zustand der Repräsentation der relevanten Objekte für den Reiseflug zum Operationsgebiet dargestellt. Im Mittelpunkt steht das AIRCRAFT, mit der Eigenschaft airborne. Die aktuelle Flugphase ist als FLIGHTPHASE->ENROUTE->CRUISE klassifiziert worden, und eine bereits erhaltene ATC-Freigabe bezüglich der Flughöhe und dem nächsten Funkfeuer (received CLEARANCE, maintain

ALTITUDE, proceed\_to WPT) ist Teil der momentanen Lage. Der aktuelle Flugplan ist im Subgraphen FLIGHTPLAN dargestellt. Die Luftrauminfrastruktur ist in dem Bereich AIRSPACE realisiert. Für den militärischen Transportflug ergeben sich die situationsrelevanten Klassifikationen proceed\_to THEATER und proceed\_to GAMING\_AREA sowie der Zustand der Fracht: LOAD on\_board. Trotz dieser recht geringen Modellierungstiefe lässt sich mit diesen Situationselementen eine gute Repräsentation der Lage realisieren.

## 7 ERPROBUNG

Das in dieser Arbeit entwickelte Modell der Handlungsziele repräsentiert innerhalb des kognitiven Prozesses dessen „mentalen“ Zustand in expliziter Form. Ziel ist es, neben den in Kapitel 2 genannten Anforderungen, durch die *zentrale, objektorientierte Situationsrepräsentation* den Zustand der realen Situation und das kognitive Verhalten möglichst vollständig und hinreichend genau bezüglich der verfolgten Handlungsziele abzubilden, um die Kooperation zwischen Mensch und Maschine zu verbessern. Die *zentrale, objektorientierte Situationsrepräsentation* soll besonders an die folgenden funktionalen Anforderungen angepasst sein:

- Bestimmung der Zielsituation,
- Konflikterkennung,
- Zustandsrepräsentation der Absichts- und Fehlererkennung und
- Unterstützung der Problemlösungskomponenten.

Diese Anforderungen stellen Leistungskriterien an den kognitiven Teilprozess der *Situationsdiagnose* dar, welche durch eine Integration der hier entwickelten Softwarekomponente in einem operativen Gesamtsystem zu beurteilen sind. Darüber hinaus lässt sich eine

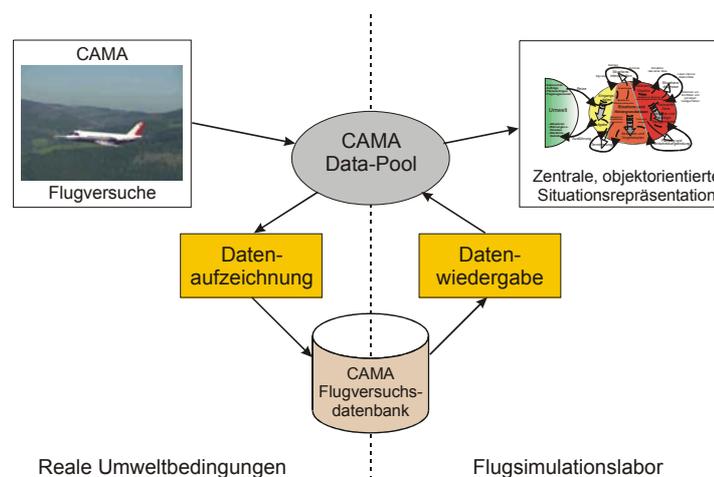
- Erklärbarkeit des kognitiven Systemverhaltens

durch die *zentrale, objektorientierte Situationsrepräsentation* erzielen. Ein *explizites* Modell ermöglicht darüber hinaus quantitative Analysen der Repräsentationsform. Hierfür sind Größen herzuleiten, die eine solche Untersuchung ermöglichen.

Zunächst soll jedoch auf die Belastung in Art und Größe eingegangen werden, die für die Beanspruchung der *zentralen, objektorientierten Situationsrepräsentation* in dieser Arbeit gewählt wurde, um eine realitätsnahe Erprobung des Konzeptes und des entwickelten Systems durchzuführen.

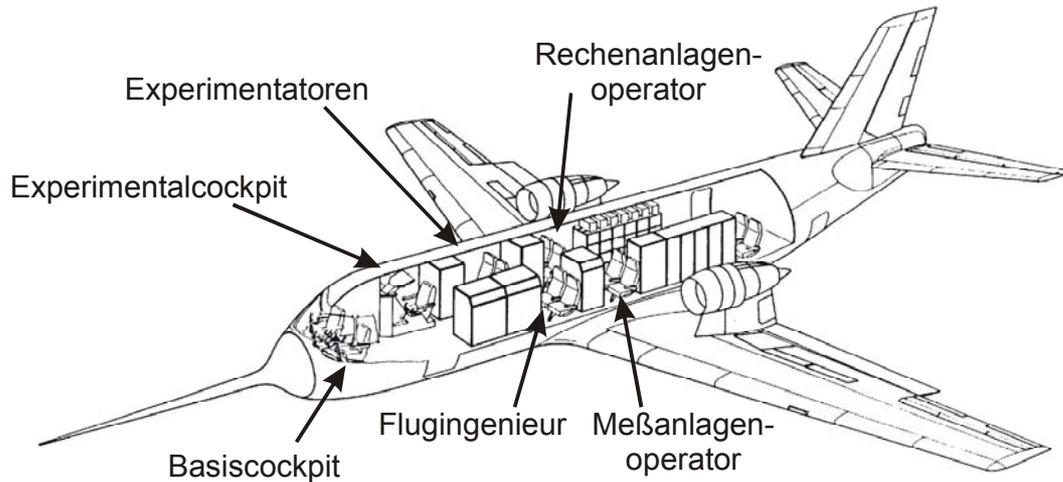
## 7.1 VERSUCHSSZENARIEN AUS FLUGVERSUCHSDATEN

Die ausreichende Leistungsfähigkeit einer Repräsentationsform ist insbesondere in Bezug auf die geplante Anwendung wichtig. Aufgabe der hier durchgeführten Erprobung ist es, das Konzept, den Entwurf von Modellstruktur und Einzelmodellen und die Umsetzung in der gewählten Implementierung an einem dem Zielsystem möglichst anwendungsnahen Fall zu testen. Das Verhalten des Modells der Handlungsziele in einem kognitiven Cockpitassistenzsystems für eine militärische Transportmission ist Gegenstand der Untersuchungen in dieser Arbeit. Im Rahmen der Entwicklung des kognitiven Cockpitassistenzsystems CAMA (siehe Kapitel 1.2.1.8) wurden zu dessen Erprobung im Jahre 2000 Flugversuche durchgeführt. Ziel der Flugversuchskampagnen war es, die während zweier Flugsimulationskampagnen im Jahre 1997 und 1998 gewonnenen Ergebnisse (siehe Walsdorf, 1998; Schulte, 1998) zu bestätigen und die Operabilität des Systems im Fluge unter Beweis zu stellen. Hierzu wurden jeweils fünf Flugmissionen im März und im Oktober 2000 von je 1.5 Stunden Dauer erfliegen. Diese Missionen entsprachen im Profil der militärischen Transportmission, wie sie in Kapitel 6 bereits vorgestellt wurde. Bei den Versuchsflügen kamen insgesamt vier Piloten des Lufttransportgeschwaders 61 aus Landsberg am Lech und zwei Testpiloten der Wehrtechnischen-Dienststelle aus Manching zum Einsatz. Die dabei gewonnenen Daten werden für die Erprobung des entwickelten Modells der Handlungsziele herangezogen, um dessen Verhalten unter Laborbedingungen bezüglich realen Umwelteinflüssen zu untersuchen. Die Datengewinnung aus den Flugversuchen und deren Nutzung für die Erprobung im Labor sind in Abbildung 104 dargestellt.



**Abbildung 104: Datennutzung aus den CAMA-Flugversuchen für die Erprobung des Zielmodells**

Als Versuchsträger wurde für die Flugversuche eine modifizierte VFW614, das Advanced Technology Testing Aircraft System ATTAS (siehe Abbildung 105) des Deutschen Zentrums für Luft- und Raumfahrt (DLR), eingesetzt.



**Abbildung 105: Flugversuchsträger ATTAS (Advanced Testing Aircraft System)**

Dieser Flugversuchsträger ermöglicht durch ein im vorderen Passagierraum integriertes Experimentalcockpit (siehe Abbildung 106) die Erprobung von Prototypensystemen unter realen Umweltbedingungen.



**Abbildung 106: ATTAS Experimentalcockpit während eines CAMA Versuchsfluges**

Durch die offene Systemarchitektur und die offene Schnittstellen des ATTAS können Systeme wie CAMA in das Experimentalcockpit integriert und im Fluge realitätsnahen Tests unterzogen werden.

Während der Flugversuche wurden die Eingangs-, Zustands- und Ausgangsdaten von CAMA aufgezeichnet. Die Datenaufzeichnung ermöglicht eine detaillierte objektive Analyse und Bewertung des Gesamtsystems und bietet online eine Wiedergabemöglichkeit des Systemverhaltens im Flugsimulationslabor des Instituts.

### 7.1.1 FLUGVERSUCHSABLÄUFE

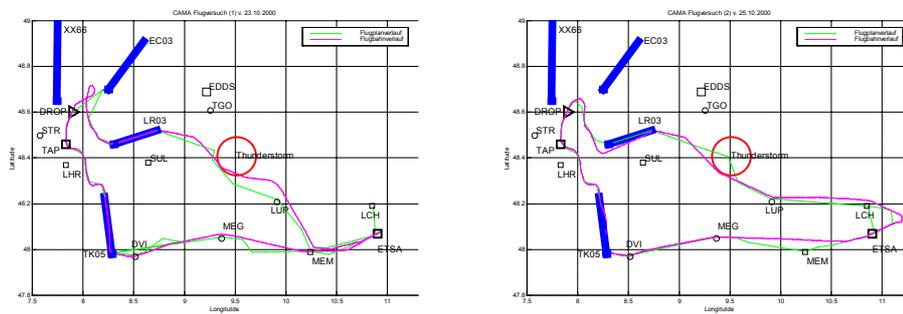
Nach Bekanntgabe des Missionsauftrages an die Besatzung und Einweisung des Experimentalpiloten in den Versuchsplan, erfolgte ein 1.5 stündiger Flug in den Schwarzwald, welcher als militärisches Operationsgebiet angenommen wurde. Die Flugphasen des Tieffluges fanden entlang des Kinzigtales statt. In der Nähe der Ortschaft Renchen wurde ein simulierter Lastenabwurf durchgeführt. Nach Beendigung der militärischen Flugabschnitte führte der Flugverlauf von Westen entlang der Kontrollzone des Flughafens Stuttgart über Laupheim wieder nach Landsberg. Während dieser Flüge wurde der Experimentalpilot mit zahlreichen realen wie simulierten Situationsereignissen konfrontiert (siehe Tabelle 14).

<b>Zeitpunkt</b>	<b>Situationsereignis</b>
IFR Legs inbound Op-Area	Data-Link-Simulation und Weitergabe aller Anweisungen von ATC durch den CAMA-Operateur an den CAMA-Experimentalpiloten
TACAN MEM outbound	Simuliertes Schließen des geplanten Zielflughafens EDDF (Frankfurt)
NOFLY-Area abeam	Simulierte Annäherung durch ein Flugzeug in gleicher Höhe
DROP outbound	Simulierte taktische Lageänderung
TK EC03 inbound	Simulierte Anweisung „new Destination EDDS (Stuttgart)“ durch AWACS (CAMA-Operateur)
IFR Legs outbound Op-Area	Data-Link Simulation und Weitergabe aller Anweisungen von ATC durch den CAMA-Operateur an den CAMA-Experimentalpiloten
IFR Legs outbound Op-Area	Simuliertes Gewitter
Approach ETSA (Landsberg)	Data-Link-Simulation und Weitergabe aller Anweisungen von ATC durch den CAMA-Operateur an den CAMA-Experimentalpiloten

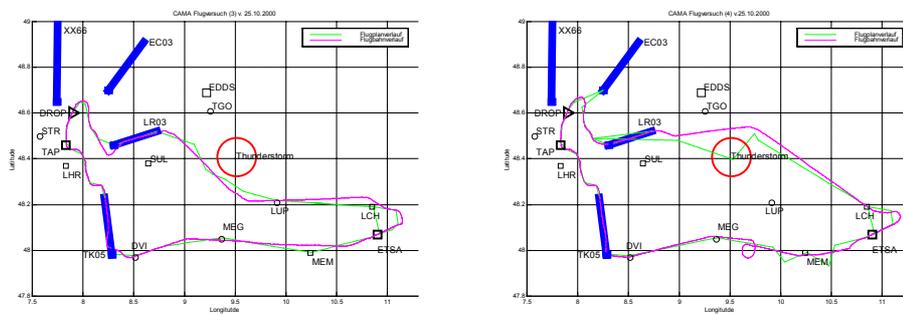
**Tabelle 14: Situationsereignisse während der Flugversuche**

Abbildung 107, Abbildung 108 und Abbildung 109 zeigen die Flugverläufe. Dargestellt sind alle für den Flugverlauf relevanten Szenarioelemente. Neben der geplanten Flugroute des

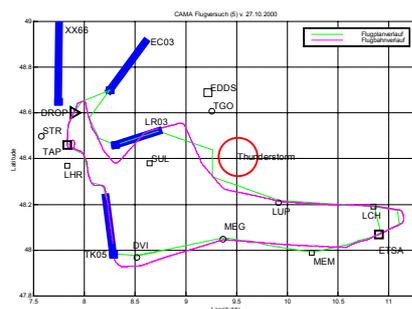
Flugplanes sind die tatsächlich erfolgene Trajektorie, die Luftraumstruktur (Funkfeuer, Flughäfen, etc.) sowie die taktischen Korridore des Operationsgebietes dargestellt.



**Abbildung 107: CAMA-Flugversuchsverläufe (links: Flug (1), rechts: Flug(2))**



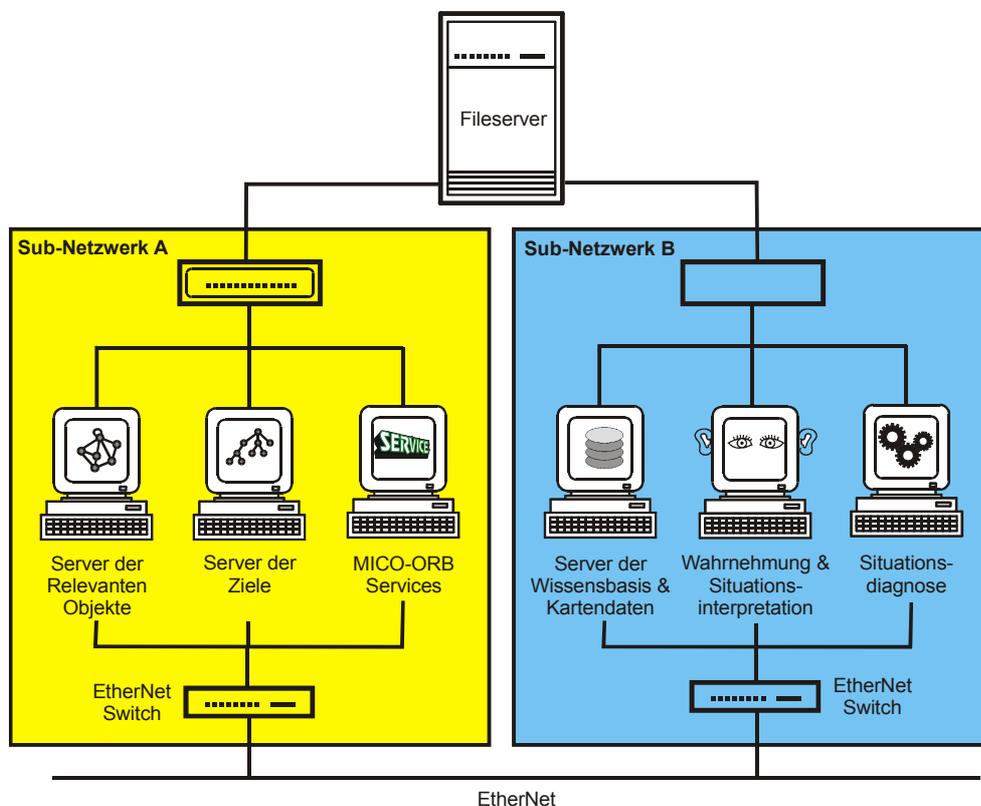
**Abbildung 108: CAMA-Flugversuchsverläufe (links: Flug (3), rechts: Flug (4))**



**Abbildung 109: CAMA-Flugversuchsverlauf (Flug 5)**

## 7.2 VERSUCHSAUFBAU IM LABOR

Für die dem Flugversuch nachfolgenden Untersuchungen am Modell der Handlungsziele im Labor wurde ein Rechnerverbund geschaffen, der eine den Auslegungskriterien an ein verteiltes System entsprechende Systemarchitektur bereitstellt. Es soll dabei gezeigt werden, dass die Implementierung des Zielmodells unter dem Vorbehalt des verwendeten CORBA-Derivats MICO den Ansprüchen einer heterogenen Anwendung in einem verteilten Rechnersystem gerecht werden kann. Hierzu wurden Workstations des Institutes für Systemdynamik und Flugmechanik in verschiedenen Datennetzen genutzt (siehe Abbildung 110).



**Abbildung 110: Versuchsaufbau und Rechnertopologie**

Bei den Rechnern handelt es sich um Silicon Graphics Workstations unter dem Betriebssystem IRIX 6.5. Bei der Auswahl der für die Server und Clients verwendeten Subnetze ist eine strikte Trennung zwischen der Administration der Server für die Situationsrepräsentationen (Sub-Netzwerk A), also den domänen-unabhängigen Softwareanteilen, und den funktionalen Applikationen (Sub-Netzwerk B), also den für die

Anwendung einer kognitiven Cockpitassistentz für den militärischen Transportflug relevanten Softwareanteile, realisiert worden.

Es wurden insgesamt fünf Versuche durchgeführt. Jeder Versuch reproduzierte den Ablauf eines Flugversuchs, wie er im Oktober des Jahres 2000 durchgeführt wurde. Ergebnisse der Versuchssequenzen sind nachfolgend dokumentiert. Bei der anschließenden Diskussion der gewonnenen Ergebnisse wurde Wert auf eine Trennung der Repräsentationsbereiche der relevanten Objekte und der Ziele gelegt, da diese von Charakter und Ressourcennutzung unterschiedlich sind. Hiermit soll die Eignung der *zentralen, objektorientierten Situationsrepräsentation* sowohl für den Bereich der relevanten Objekte als auch für den Bereich der Ziele gezeigt werden.

### 7.3 UNTERSUCHUNGSZIELE UND BEANSPRUCHUNG DES MODELLS

Das in dieser Arbeit entwickelte funktionale Leistungsspektrum des Modells der Handlungsziele wird hier in Bezug auf die Ergebnisse aus dem System CAMA erprobt und bewertet. CAMA stellt derzeit eines der wenigen kognitiven Cockpitassistentzsysteme dar, welches in Form eines Prototypensystems bereits umfangreichen Tests und Bewertungen durch Experten in Simulationskampagnen und Flugversuchen unterzogen wurde. Schwerpunkt der Erprobung liegt hier auf der zentralen Repräsentation der Handlungsziele und der damit explizit dargestellten Situation der Konflikte, der Fehler und der erkannten Pilotenabsichten, sowie der intentionalen Erklärungsfähigkeit des Systems auf der wissensbasierten Ebene.

Zunächst soll das Leistungsvermögen der *zentralen, objektorientierten Situationsrepräsentation* untersucht werden. Die Leistung einer relationalen Repräsentationsform ist durch geeignete qualitative und quantitative Größen zu definieren, um die Quantifizierbarkeit zur Bewertung der Repräsentationsleistung zu ermöglichen. Die funktionale Leistung des Zielmodells unter den im CAMA-Flugversuch aufgetretenen Beanspruchungen wird anschließend dargestellt.

#### 7.3.1 LEISTUNG DER RELATIONALEN REPRÄSENTATIONSFORM

Um die Leistung der relationalen Repräsentationsform zu erfassen, müssen hierfür Kriterien aufgestellt werden, die qualitative und quantitative Bewertungen ermöglichen.

### 7.3.1.1 Situationsverständnis

Für die Modelleigenschaften ergibt sich das qualitative Kriterium des *Situationsverständnisses (K1)*, welches auf alle Bereiche im relationalen Situationswissen des Modells der Handlungsziele angewendet werden kann. Eine direkte Bewertung der Richtigkeit der Situationsbeschreibung ist jedoch nur eingeschränkt möglich, da der Zustand der realen Situation einem subjektiven Empfinden unterliegt. Vielmehr gilt es, das Verständnis des Systems in Form von Inhalt, Aufbau und Semantik der repräsentierten Situation zu bewerten. Dies kann aus den oben beschriebenen Gründen nur anhand seiner Reaktionen auf die wahrgenommene Umwelt erfolgen.

### 7.3.1.2 Situationsumfang

Für die Bestimmung der Modelleistung müssen quantitative Merkmale definiert werden. Insbesondere die Repräsentation der Situation durch eine relationale Struktur ermöglicht hier gegenüber einer herkömmlichen, datenorientierten Repräsentationsform tiefere Untersuchungen. Zunächst lassen sich diese in *statische* und *dynamische* Zustandsgrößen unterteilen. Ausgehend von der Modellierungsbreite (d.h. der Anzahl der in der Wissensbasis abgelegten Modelle) kann zunächst die *Größe der repräsentierten Situation (K2)* durch die Anzahl der repräsentierten Knoten und Kanten im Graphen der relationalen Repräsentationsform definiert werden:

$$\binom{n}{r}$$

mit

$n$  : Anzahl der Situationselemente

$r$  : Anzahl der Relationen

### 7.3.1.3 Situationskomplexität

Aus den Größenwerten der relational repräsentierten Situation lässt sich die *Komplexität der Situation (K3)* ableiten. Die Komplexität soll hier auf der *maximalen Situationsvielfalt (K4)*

$$n_{\max}^2 \cdot r_{\max}$$

mit

$n_{\max}$  : Anzahl aller bekannten Situationselemente

$r_{\max}$  : Anzahl aller bekannten Relationen

basieren. Die Komplexität der momentanen Situation ist dann durch den Quotienten aus der *aktualen Situationsvielfalt* und der *maximalen Situationsvielfalt* definiert und beschreibt somit den *Grad der aktuellen Situation bezüglich der maximalen Situationsvielfalt*.

$$\frac{n \cdot r}{n_{\max}^2 \cdot r_{\max}} \cdot 100$$

mit

$n$  : Anzahl der Situationselemente  $\hat{=}$  Komplexität

$r$  : Anzahl der Relationen  $\hat{=}$  Kompliziertheit

$n_{\max}$  : Anzahl aller bekannten Situationselemente

$r_{\max}$  : Anzahl aller bekannten Relationen

#### 7.3.1.4 Situationsdivergenz

Neben den statischen sind die dynamischen Zustandsgrößen der Situation von besonderer Bedeutung. Hierzu soll die Veränderung des Situationswissens über die Zeit durch den Grad der *Situationsdivergenz* (K5) erfasst werden. Die zeitliche Veränderung des Situationswissens ergibt sich aus den für die Metrik der Situation entscheidenden Anteile (Inhalt, Struktur, Semantik). Die *Divergenz* sei hier die Änderung der Situationselemente, ihrer Relationen und die die Relationen beschreibenden Attribute.

$$\bar{S}_{Div} = \begin{pmatrix} Div_{Inhalt} \\ Div_{Aufbau} \\ Div_{Semantik} \end{pmatrix}$$

$$Div_{Inhalt} = \frac{N_t \circ N_{t+1}}{N_t} \cdot 100 \quad , N_t = \text{Situationselemente zum Zeitpunkt } t$$

$$Div_{Aufbau} = \frac{R_t \circ R_{t+1}}{R_t} \cdot 100 \quad , R_t = \text{Relationen zum Zeitpunkt } t$$

$$Div_{Semantik} = \frac{A_t \circ A_{t+1}}{A_t} \cdot 100 \quad , A_t = \text{Relationsattribute zum Zeitpunkt } t$$

(mit  $\circ$  = Operator mit dem Ergebnis "Anzahl der identischen Situationselemente")

### 7.3.1.5 Bandbreite des Merkmalsraumes

Implizite Repräsentationsformen können ebenfalls anhand quantitativer Größen bemessen werden. Für die Repräsentation durch Hyper-Karten stellt zunächst das Kriterium der *Bandbreite D* (K6) einen aussagekräftigen Wert dar. Es beschreibt den Wertebereich eines Hauptmerkmals.

$$D = p_{\max} - p_{\min}$$

### 7.3.1.6 Auflösung des Merkmalsraumes

Aus der Bandbreite des Merkmals kann nun die *Auflösung  $r_p$*  (K7) des Hauptmerkmals bestimmt werden. Hierzu sind die Bandbreite eines Hauptmerkmals durch die Anzahl der Diskretisierungsstufen *sets(p)* zu teilen. Die Auflösung des Hauptmerkmalraumes ist so ein Maß für den Umfang einer Merkmalsgröße und damit Ausdruck für die Größe der potentiell möglichen impliziten Situationsbeschreibung.

$$r_p = \frac{D}{\text{sets}(p)}$$

### 7.3.1.7 Komplexität des Merkmalsraumes

Aus der Anzahl *n* der in einer impliziten Situationsbeschreibung beinhalteten Hauptmerkmale *P* lässt sich mit Hilfe der *Auflösung* (K7) ein Maß für die *Komplexität C* (K8) bestimmen. Für

die Menge aller bekannten Hauptmerkmale  $N$  ergibt sich daraus die worst-case-complexity für die zu verarbeitende Situation eines kognitiven Systems.

$$C = \sum_{i=1}^N (n_p \cdot r_p)_i \quad , (\text{worst - case - complexity})$$

Die Kriterien, die für die qualitative Bewertung eines objektorientiertes Modell der Handlungsziele in relationaler Form herangezogen werden, sind in Tabelle 15 zusammengefasst.

	Eigenschaftsart	Bewertungskriterium	Teilmodell	Zeitverhalten	
Relationale Situationsrepräsentation	Qualitativ	Situationsbeschreibung durch Inhalt, Struktur und Aufbau	Lagerepräsentation (Inhalt, Struktur, Aufbau)	statisch	
			Zielrepräsentation (Inhalt, Struktur, Aufbau, Erfülltheit der Einzelziele)		
	Quantitativ	Situationsgröße	Lagerepräsentation $\begin{pmatrix} n_E \\ r_E \end{pmatrix}$		Zielrepräsentation $\begin{pmatrix} n_G \\ r_G \end{pmatrix}$
			Komplexität		
		Zielrepräsentation $\frac{n_G \cdot r_G}{n_{G_{\max}}^2 \cdot r_{\max}} \cdot 100$	Lagerepräsentation $\vec{S}_{E Div} = \begin{pmatrix} Div_{O_{Relv} Inhalt} \\ Div_{O_{Relv} Aufbau} \\ Div_{O_{Relv} Semantik} \end{pmatrix}$		
		Variation			Zielrepräsentation $\vec{S}_{G Div} = \begin{pmatrix} Div_G Inhalt \\ Div_G Aufbau \\ Div_G Semantik \end{pmatrix}$
Implizite Situationsrepräsentation	Quantitativ	Bandbreite	$D = p_{\max} - p_{\min}$	statisch	
		Auflösung	$r_p = \frac{D}{sets(p)}$		
		Komplexität	$C = \sum_{i=1}^N (n_p \cdot r_p)_i$	dynamisch	

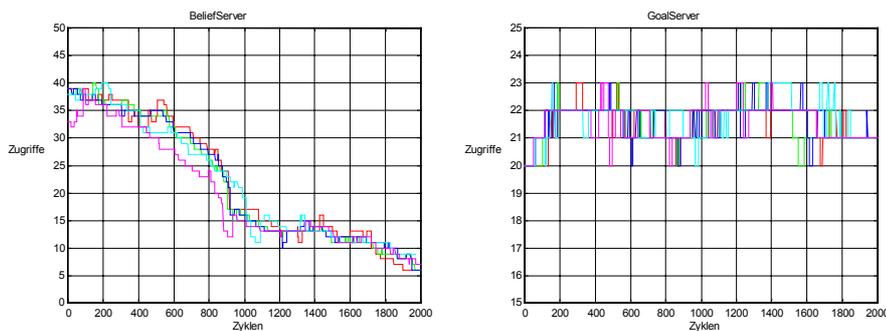
**Tabelle 15: Kriterien für die quantitative Bewertung einer relationalen Repräsentationsform**

### 7.3.1.8 Ergebnisse

Die Server stellen diejenigen Funktionen bereit, welche den kognitiven Teilprozessen (Clienten) den Zugriff und die Nutzung des repräsentierten Situationswissens ermöglichen. Die Systemleistung der Server ist durch die Zahl der verarbeiteten Zugriffe der kognitiven

Teilprozesse zur Laufzeit auf die jeweilige „mentale“ Repräsentation charakterisiert. Die Darstellung und Auswertung erfolgt getrennt für die Repräsentation der relevanten Objekte („BeliefServer“) und für die Repräsentation der Ziele („GoalServer“).

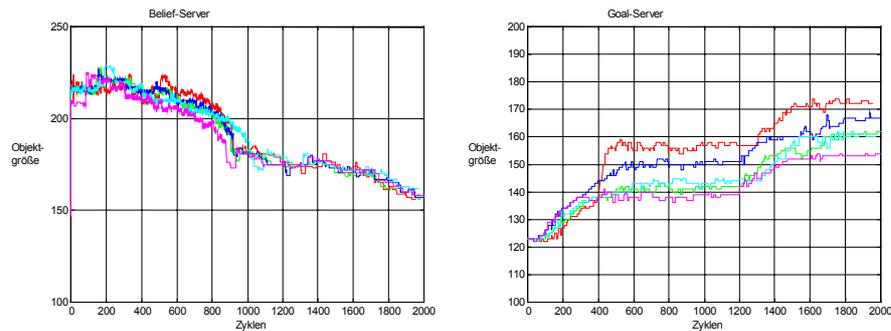
In Abbildung 111 ist die Anzahl der verarbeiteten Zugriffe auf den *BeliefServer* (links) und den *GoalServer* (rechts) über die Versuche dargestellt.



**Abbildung 111: Zugriffe während eines kognitiven Zyklusses auf *BeliefServer* (links) und *GoalServer* (rechts)**

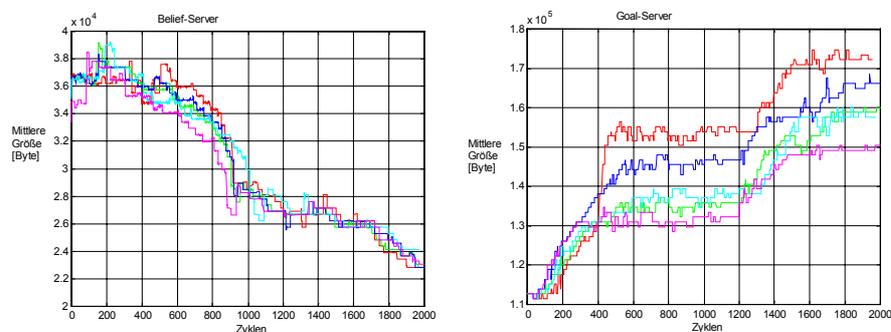
Die Zahl der Zugriffe bezieht sich hierbei auf einen kognitiven Zyklus von der *Situationsinterpretation* bis zur *Situationsdiagnose*. Die Anzahl der Zugriffe auf die Repräsentation der relevanten Objekte nimmt mit fortschreitender Missionserfüllung durch die Module der *Situationsinterpretation* ab. Während sich die Funktionen der *Situationsinterpretation* durch die *selektive Aufmerksamkeit* im Verlauf der Mission auf immer kleinere Situationsbereiche „konzentrieren“, erkennt man für die Repräsentation der Ziele eine relativ konstante Zugriffshäufigkeit über die gesamte Breite der repräsentierten Ziele. Dies lässt sich durch die fortschreitende Missionserfüllung und den stetigen strukturellen Aufbau der Zielhierarchie durch den Teilprozess der *Situationsdiagnose* erklären. Die Aufgaben der Zielextraktion und der Ermittlung der Zielkonflikte stellen Funktionen dar, welche während der gesamten Laufzeit aktiv ausgeführt werden.

Neben der reinen Häufigkeit des Zugriffs der Anwendung spielt der Umfang der kommunizierten Informationen eine wesentliche Rolle. Er drückt die eigentliche funktionale Leistung der Systemarchitektur aus und zeigt, inwieweit sich der gewählte Ansatz für die geplante Anwendung eignet. In Abbildung 112 sind das Kommunikationsaufkommen zwischen kognitiven Teilprozessen (Clients) und Repräsentationsform (Server) über die Versuchszeiten aufgetragen.



**Abbildung 112: Datenumsätze für *BeliefServer* (links) und *GoalServer* (rechts)**

Der Mittelwert der Größe aller kommunizierten Situationselemente ist über die Versuchsflüge in Abbildung 113 dargestellt.

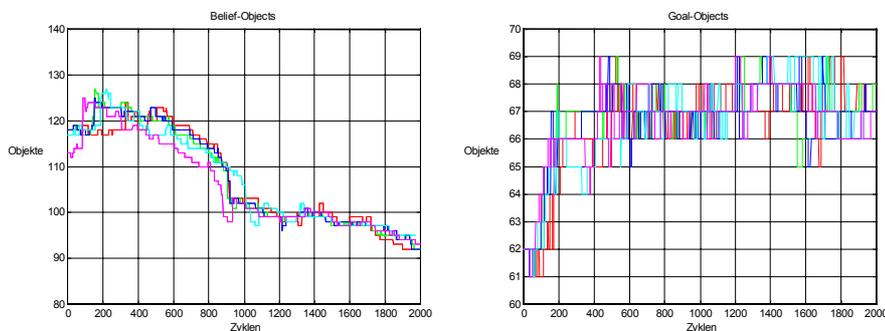


**Abbildung 113: Statistik bezüglich der „mittleren Größe“ der versandten Objekte**

Das Ergebnis für die Repräsentation der relevanten Objekte (*BeliefServer*) zeigt wieder eine zeitlich fallende Tendenz. Der Datenumsatz steht in Analogie zu den ermittelten Zugriffen auf den *BeliefServer* (siehe Abbildung 111). Die Repräsentation der Ziele (*GoalServer*) weist einen stufenweisen Anstieg auf. Diese ist durch das Flugversuchsszenario begründet (vergleiche Abbildung 107, Abbildung 108 und Abbildung 109). Während das erste Drittel des Fluges den Reiseflug zum Operationsgebiet mit relativ geringen externen Aufträgen und internen Zielvorgaben in Form von Besatzungsabsichten umfasste, stabilisierte sich die Zielsituation bis zum Einflug ins Operationsgebiet. Nach Verlassen desselben beginnt ein recht komplexer und dynamischer Situationsabschnitt (häufige Anweisungen durch ATC, zahlreiche Konfliktlösungen und erkannte Pilotenabsichten etc.).

### 7.3.1.8.1 Dynamik von Inhalt/Struktur/Semantik

Ausschlaggebend für die Beurteilung eines kognitiven Systems und die darin verwirklichten Konzepte ist die Situation, mit der das System konfrontiert wird. Je komplexer und komplizierter eine Situation ist, umso schwieriger fällt es dem System, diese zu verstehen. Der in dieser Arbeit erstellte Ansatz, die symbolische Repräsentation der Situation in einem technischen System zentral und objektorientiert darzustellen, ermöglicht aussagekräftigere Untersuchungen. Zunächst soll durch die Anzahl der die Situation repräsentierenden Situationselemente auf deren Komplexität geschlossen werden. Wie in Abbildung 114 (links) zu sehen ist, nimmt die Größe der erfassten Umwelt in Form der relevanten Objekte mit fortschreitendem Missionsfortschritt ab.

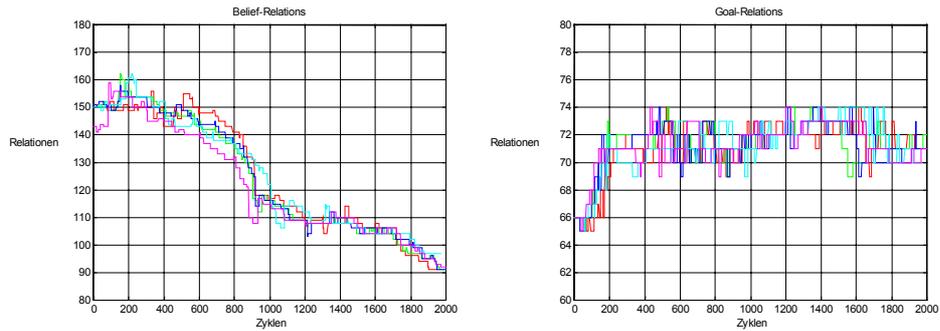


**Abbildung 114: Anzahl der Situationselemente (relevante Objekte (links) und Ziele (rechts))**

Dies resultiert aus der Veranlagung der *Situationsinterpretation* zur *selektiven Aufmerksamkeit*, welche durch die Zielsituation getrieben werden. Die Anzahl der Ziele über den Missionsverlauf zeigt demgegenüber eine stabilere Charakteristik.

### 7.3.1.8.2 Grad der Vernetztheit

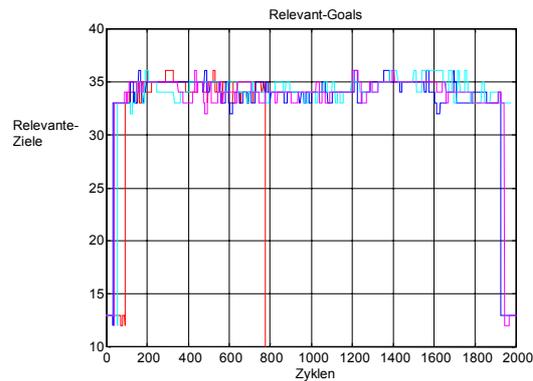
Der Grad der Vernetztheit der Situationselemente drückt die Kompliziertheit der Situation und damit die Leistung des Teilprozesses der *Situationsinterpretation* und der *Situationsdiagnose* aus. Die Vernetztheit einer relationalen Situationsrepräsentation ist durch die Anzahl der zwischen den Situationselementen bestehenden Relationen charakterisiert (siehe Abbildung 115).



**Abbildung 115: Vernetztheitsgrad (Anzahl der Relationen zwischen den relevanten Objekten und zwischen den Zielen)**

### 7.3.1.8.3 Relevanz der Ziele

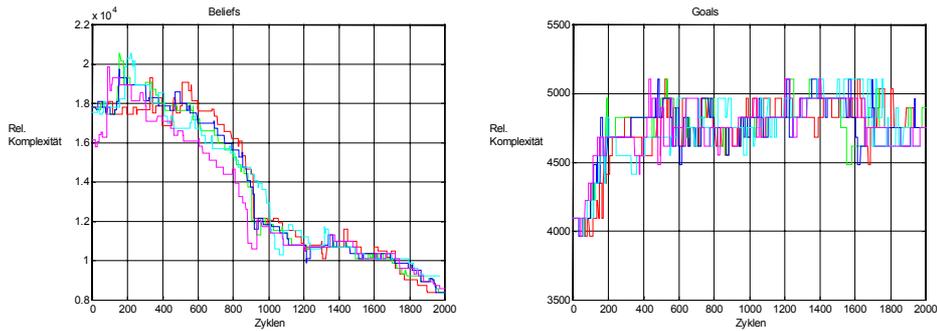
Neben der Vielfalt des Situationswissens spielt die Anzahl der relevanten Ziele für die Leistung der *Situationsdiagnose* eine zentrale Rolle. Wie in Abbildung 116 zu sehen, verhält sich die Anzahl der relevanten Ziele zwischen Start und Landung stabil im Bereich zwischen 33 und 37 Zielen.



**Abbildung 116: Relevanz der Ziele**

### 7.3.1.8.4 Komplexität der Situationsbereiche

Das Maß für die Beanspruchung des Gesamtsystems stellt unter anderem die Situationskomplexität dar. Sie drückt aus, welche Anforderungen an das kognitive System gestellt werden und welche Aufgaben von den Problemlösungskomponenten erfüllt werden müssen. Wie in Abbildung 117 dargestellt, ist auch hier der Verlauf der relativen Komplexität von relevanten Objekten und Zielen unterschiedlich.

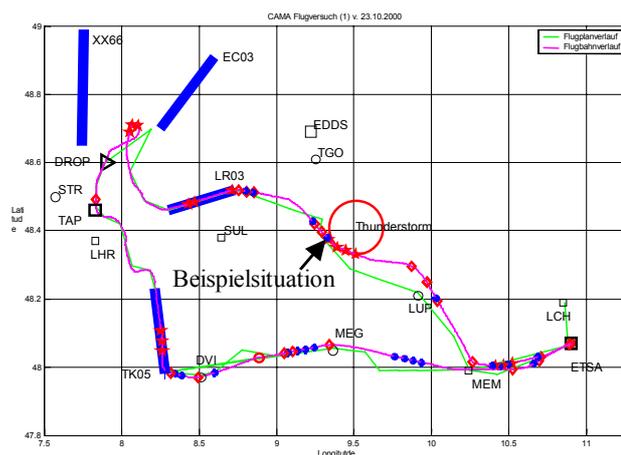


**Abbildung 117: Relative Komplexität von Lage- und Zielrepräsentation**

Das von den Modulen der *Situationsinterpretation* gelieferte mentale Bild der Situation verliert gegen Missionsende mehr und mehr an Komplexität. Hieraus kann geschlossen werden, dass diese Funktion dem Anspruch der *selektiven Aufmerksamkeit* genügt, da die Zahl der relevanten Ziele gegen ein Minimum strebt. Hierdurch fokussiert sich das Aufmerksamkeitsverhalten der die Situation interpretierenden Module zunehmend auf die kleiner werdende Menge der unerfüllten, relevanten Ziele.

### 7.3.2 ERPROBUNG DES MODELLS DER HANDLUNGSZIELE

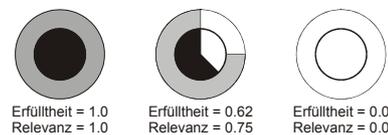
Zur Erprobung des Modells der Handlungsziele werden die Daten eines CAMA-Versuchsfluges herangezogen. Beispielhaft soll hier eine typische Situation aus einem Versuchsflug (siehe Abbildung 118) untersucht werden.



**Abbildung 118: CAMA-Flugversuch Nr. 3 vom 23.10.2000 zuzüglich erkannter Fehler (★) und Absichten (◆) des Piloten**

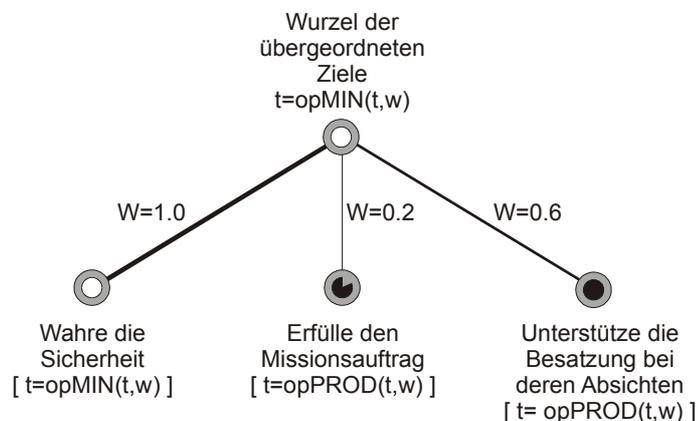
Die aus den Laborexperimenten gewonnenen Aufzeichnungen der Zielsituationen werden nachfolgend exemplarisch an einer charakteristischen Situation diskutiert. Der hier präsentierte Zielzustand entspricht der Flugphase zum Zeitpunkt des Umfliegens eines Gewitters (Position: 9,34°E 48,35°N).

Die Darstellung des Modells der Handlungsziele erfolgt durch die Zielhierarchie selbst. Hierbei sind neben der Topologie der Zielhierarchie die zu Grunde liegenden Gewichte für ein kooperatives System und die verwandten Operatoren für die Bestimmung der Erfüllung der Fundamentalziele aus den Ergebnissen der Instrumentalziele erfasst. Für den Erfülltheitsgrad und die momentane Relevanz wird die in Abbildung 119 gezeigten Darstellungsform gewählt.



**Abbildung 119: Darstellungsart für die Erfüllung (inneres Segment) und Relevanz (äußeres Segment) eines Ziels**

Die Gewichtung der Ziele fundiert auf den Ergebnissen der Wissensakquisition. Die Ergebnisse aus dem *DirectRatio*-Verfahren sind an den entsprechenden Stellen ausgewiesen. Die Diskussion der Ergebnisse soll mit den Fundamentalzielen beginnen. In Abbildung 121 sind die übergeordneten Ziele *Wahre die Sicherheit*, *Erfülle den Missionauftrag* und *Unterstütze die Besatzung bei deren Absichten* und deren zugeordnete Gewichtungen entsprechend der Ergebnisse aus der *DirectRatio*-Methode (vergleiche Tabelle 13 Bsp. 2) dargestellt.

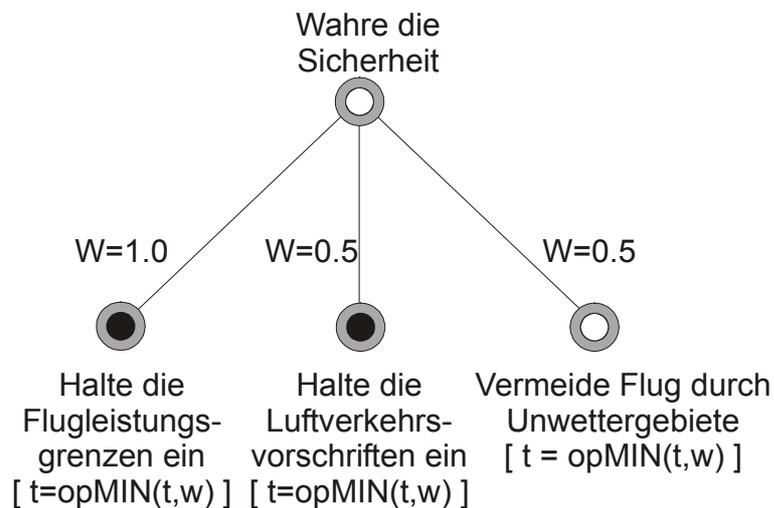


**Abbildung 120: Zustand der übergeordneten Ziele**

Der Zustand des Zielmodells zum Zeitpunkt des Gewitterumfliegens zeigt eine mangelnde Zielerfüllung im Gefährdungsmodell und eine teilweise Zielerfüllung des Missionsauftrages. Der resultierende Zustand des Gefährdungsmodells soll zunächst anhand der Instrumentalziele betrachtet werden.

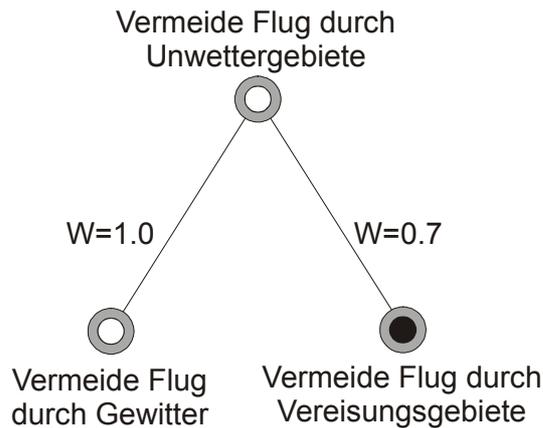
### 7.3.2.1 Zustände im Gefährdungsmodell

Der Zustand des *Gefährdungsmodells* setzt sich aus den Erfülltheitsgraden der Instrumentalziele *Halte die Flugleistungsgrenzen ein*, *Halte die Vorschriften des Luftverkehrs ein* und *Vermeide Flug durch Unwettergebiete* zusammen und wird über die zugeordneten Gewichtungen und, im Falle des Gefährdungsmodells mit Hilfe des Produktoperators ermittelt. Wie in Abbildung 121 zu sehen, wird für das Ziel *Vermeide Flug durch Unwettergebiete* ein Konfliktzustand festgestellt.



**Abbildung 121: Zustand der Ziele im Gefährdungsmodell**

Zur weiteren Erörterung des Zielzustandes muss das Instrumentalziel *Vermeide Flug durch Unwettergebiete* analysiert werden (siehe Abbildung 122).

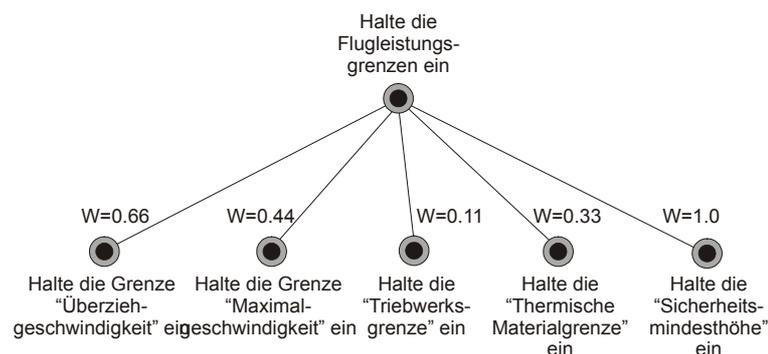


**Abbildung 122: Zustand der Ziele im Teilmodell *Unwetter***

Das Ziel *Vermeide Flug durch Unwettergebiete* wird erfüllt, wenn sich das Flugzeug in ausreichender Entfernung zu Gewitter- und Vereisungsgebieten befindet. Während hier das Ziel *Vermeide Flug durch Vereisungsgebiet* erfüllt ist, ist das Ziel *Vermeide Flug durch Gewitter* in der aktuellen Flugsituation verletzt. Ein Konflikt liegt vor. Dieser stellt die Ursache für die Konfliktsituation in den übergeordneten Zielhierarchien dar.

Der Vollständigkeit halber sollen diejenigen Teilmodelle, welche für die zu untersuchende Flugphase keinerlei Konfliktsituationen repräsentieren, in den nachfolgenden Abbildungen dargestellt werden.

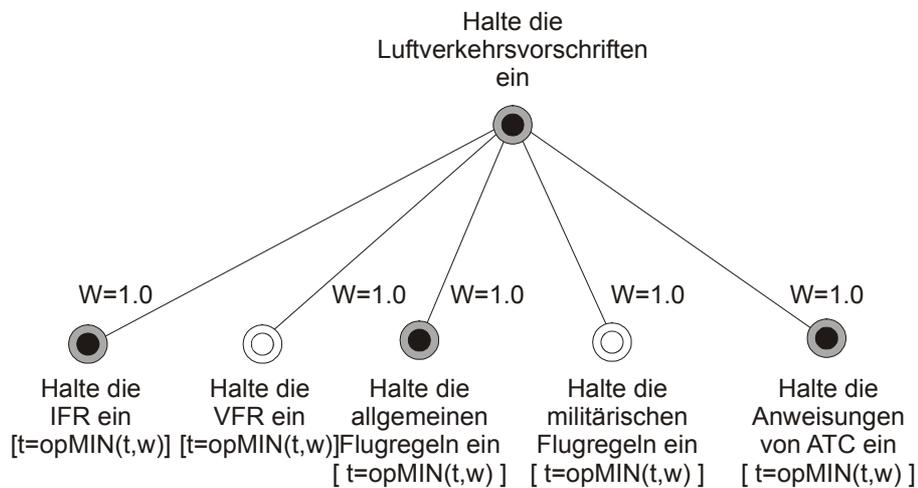
Abbildung 123 zeigt das Instrumentalziel *Halte die Flugleistungsgrenzen ein* mit den Gewichten gemäß Tabelle 11 Bsp (1).



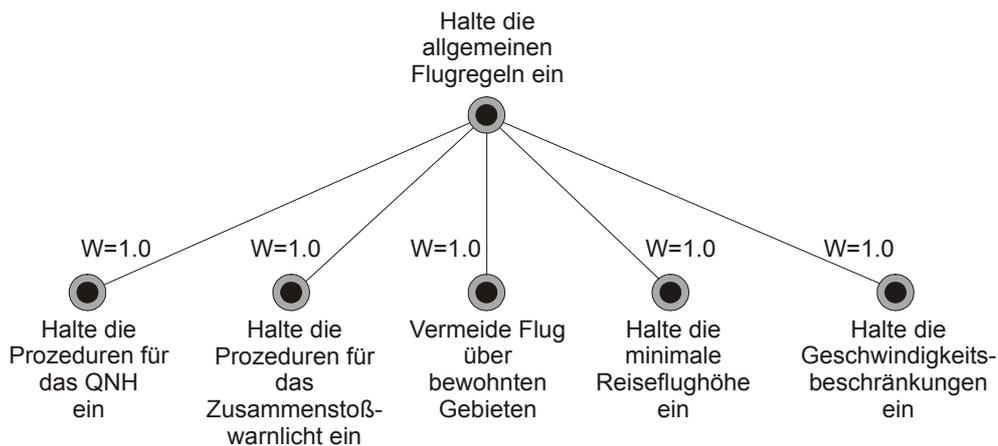
**Abbildung 123: Zustand der Ziele im Teilmodell *Flugleistungsgrenzen***

Des Zustand des Teilmodells der *Vorschriften des Luftverkehrs* zeigt als Besonderheit die Ziele *VFR* und *Militärische Flugregeln*, welche im gegenwärtigen Kontext der Situation keinerlei Relevanz besitzen. Nicht-relevante Ziele werden mit dem internen Zustand des

Erfülltheitsgrades 0 markiert. Die Teilmodelle der Allgemeinen Flugregeln und der Militärischen Flugregeln sind nachfolgend illustriert.

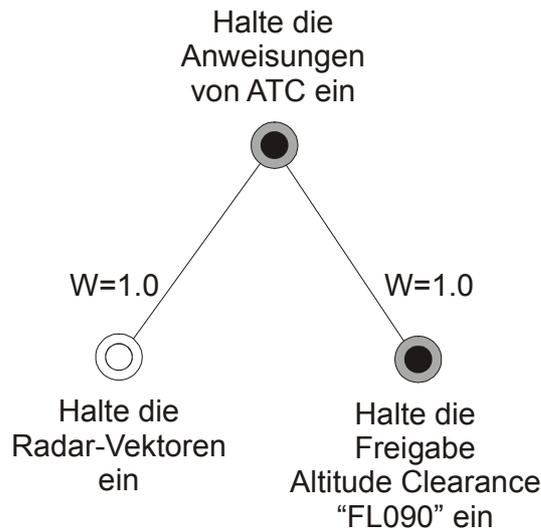


**Abbildung 124:** Zustand der Ziele im Teilmodell *Vorschriften des Luftverkehrs*



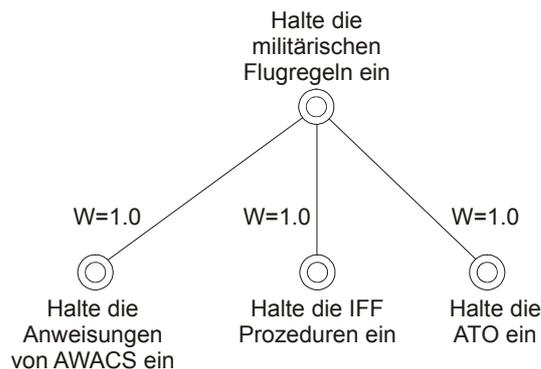
**Abbildung 125:** Zustand der Ziele im Teilmodell *Allgemeine Flugregeln*

Neben den Allgemeinen Flugregeln bestimmt das dynamische Teilmodell der ATC-Anweisungen die Gesamtsituation der übergeordneten Ziele entscheidend mit. Es ist in Abbildung 126 dargestellt.



**Abbildung 126: Zustand der Ziele im Teilmodell *ATC Anweisungen***

Die in Abbildung 127 illustrierten Ziele der militärischen Flugregeln besitzen während des hier betrachteten zivilen Abschnitts keine Relevanz und sind somit für den Zustand der Zielsituation zu vernachlässigen.



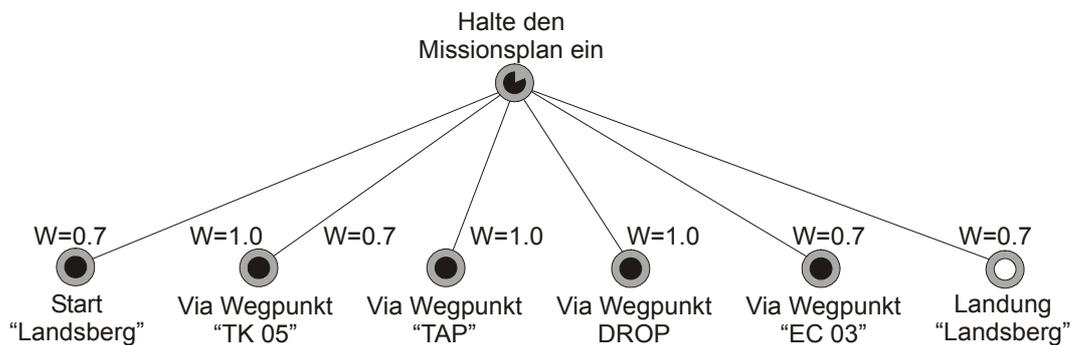
**Abbildung 127: Zustand der Ziele im Teilmodell *Militärische Flugregeln***

### 7.3.2.2 Zustände im Modell des Missionsauftrages

Das Teilmodell des Missionsauftrages zerfällt in die Fundamentalziele *Missionsplan*, mit Vorgaben über den Flugverlauf und durchzuführende Aktionen, und *Taktik*, mit den vorgeschriebenen Verhaltens- und Vorgehensweisen der Crew. Es ist in Abbildung 128 dargestellt. Während das Ziel *Missionsplan* eine partielle Erfülltheit besitzt, sind Ziele, die das militärtaktische Verhalten des Systems bestimmen, auf Grund der fehlenden Relevanz im Situationskontext unbedeutend.

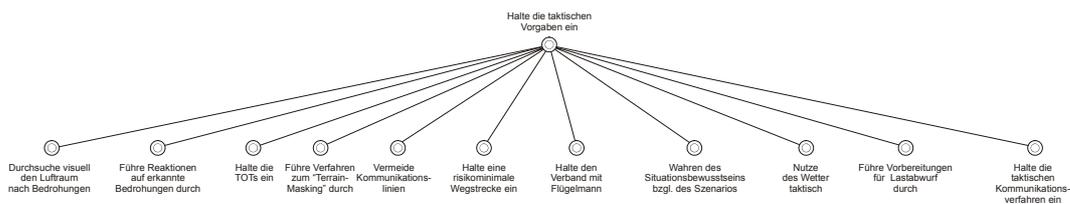
Das Fundamentalziel *Halte den Missionsauftrag ein* (siehe Abbildung 128) umfasst den dynamischen Anteil derjenigen Ziele, welche dem System von externen Quellen vorgeschrieben wurden.

Hierzu zählen die zu überfliegenden Wegpunkte im Missionsverlauf sowie für den militärischen Transport durchzuführende Aktionen (z.B. Absetzen der Last). Der Missionsplan ist in Abbildung 127 gezeigt.



**Abbildung 128: Zustand der Ziele im Teilmodell *Missionsplan***

Auch an dieser Stelle sei zur Vollständigkeit das Teilmodell der taktischen Ziele illustriert, welches jedoch aufgrund des Aufenthalts des Flugzeugs im zivilen Gebiet für die Situationsdiagnose keinerlei Bedeutung besitzt. Es ist in Abbildung 129 dargestellt.

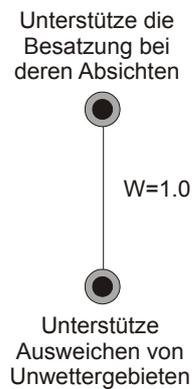


**Abbildung 129: Zustand der Ziele im Teilmodell *Taktik***

### 7.3.2.3 Zustände im Modell der Besatzungsabsichten

Insbesondere die für die Eigenschaft der Kooperation entscheidende Repräsentation der bekannten Besatzungsabsichten stellt einen wichtigen Anteil an der Repräsentation der Zielsituation dar. Erst durch dieses Teilmodell ist es dem kognitiven System möglich, kooperatives Verhalten der Besatzung gegenüber zu zeigen und angemessene Reaktionen zum

Zweck der Unterstützung bereit zu stellen. In Abbildung 130 ist die Situation dargestellt, welche die Repräsentation der erkannten Absicht *Ausweichen von Unwettern* verkörpert.



**Abbildung 130: Zustand der Ziele im Modell der Besatzungsabsichten**

Es sei an dieser Stelle nochmals darauf hingewiesen, dass Modelle von Absichten den Modellen von Handlungszielen weitgehend ähnlich sind, jedoch der Erfülltheitsgrad einer Absicht als Zugehörigkeit eines Verhaltensmusters bezüglich eines erwarteten individuellen Verhaltens verstanden werden muss. Das in Abbildung 130 illustrierte Modell nutzt hier ein vereinfachtes Absichts-Modell, was den Erfülltheitsgrad der Absicht *Ausweichen von Unwettern* erklärt.

## 8 ZUSAMMENFASSUNG

Die für kognitive Assistenzsysteme entscheidende *zentrale Repräsentation des Situationswissens* ist für den Bereich der Handlungsziele durch das hier konzipierte Modell umgesetzt worden. Der beschrittene Weg einer Repräsentationsform auf Basis von Relationen und Objekten und Implementierung einer verteilten Client-Server Architektur hat die Bewährungsprobe in der Funktion der Situationsanalyse eines kognitiven Cockpitassistenzsystems bestanden.

Damit wurde das zielgerichtete, kooperative Verhalten von Teilfunktionen des technischen Systems im Sinne der inneren Konsistenz verbessert. Die zentrale Repräsentation der Handlungsziele mit den darin abgebildeten Zielkonflikten ist darüber hinaus dem kognitiven Teilprozess der Situationsinterpretation dienlich, indem es der Umsetzung der Forderung nach *selektiver Aufmerksamkeit* und effizienter Nutzung der Aufmerksamkeitsressourcen genügt. Die sich aus dem gewählten Konzept einer *zentralen, objektorientierten Situationsrepräsentation* ergebende Reduktion der Softwarekomplexität resultiert durch die Reduzierung des Entwicklungs- und Diagnoseaufwandes in einer Verminderung der Entwicklungskosten. Durch die symbolische Darstellung von Wissen in relationaler Form ist der Abgleich der existierenden mentalen Situationsbilder zwischen Mensch und Maschine möglich. Eine Steigerung der Effektivität des Teams Mensch-Maschine durch die verbesserte Kooperationsfähigkeit ist damit erzielbar. Die Kommunikation über die explizit repräsentierten Handlungsziele ermöglicht die Erklärungsfähigkeit des Systemverhaltens. Durch die direkte Wechselwirkung der Handlungsziele mit den kognitiven Teilprozessen der Situationsinterpretation und der Planung und Entscheidungsfindung kann eine Aussage über das Gesamtsystemverhalten getroffen werden. Damit steht nun eine Fähigkeit des Systems bereit, die dem Operateur das Hinterfragen des Gesamtsystemverhaltens ermöglicht. Der universelle Ansatz zeigt Potentiale auf, die auch in anderen kognitiven Teilprozessen Anwendung finden könnten.

Aus den mit dieser Repräsentationsform gesammelten Erfahrungen ist abzusehen, dass die relationale Situationsrepräsentation ein hohes Maß an Flexibilität und Erweiterbarkeit mit sich bringt, welche durch starre Datenstrukturen nicht erzielt werden kann. Die Nutzung einer universellen Repräsentationsform für die Analyse von bereits existierenden kognitiven Systemen ist im Rahmen der Repräsentation der relevanten Objekte mittels der Ergebnisse der

Module zur Situationsinterpretation des Systems CAMA gezeigt worden. Die von einer Domäne unabhängige universelle Repräsentation der Situation durch kognitive Komponenten bietet zahlreiche Anwendungsmöglichkeiten, welche weit über die Umsetzung in einem Cockpitassistenzsystem hinaus gehen (siehe Putzer, 2001a/2001b).

Auch nach einer gelungenen Umsetzung einer *zentralen, objektorientierten Situationsrepräsentation der Handlungsziele* erscheinen weitere Untersuchungen lohnenswert. Auf Probleme im Konfliktmanagement interner Zielkonflikte und die Repräsentation von *Strategien* konnte in dieser Arbeit nicht eingegangen werden. Der Wunsch, unterschiedliche *Strategien* zu verfolgen, scheint offensichtlich über die Umsetzung von dynamischen Zielgewichten zu führen. Ein weiteres Aufgabengebiet stellen die *unklaren Ziele* dar. Wie ist mit Zielen zu verfahren, für die der Grad der Erfülltheit bisher nur unzureichend oder gar nicht in einem technischen System bestimmt war? Das Erlernen unklarer Ziele in Erweiterung des hier realisierten Modells könnte zu einer weiteren, deutlichen Verbesserung der *zentralen, objektorientierten Situationsrepräsentation der Handlungsziele* und damit des kognitiven Assistenzsystems führen.

## 9 LITERATURVERZEICHNIS

AAIC Report 96-5 .....

Alderfer, C. P.: *Existence, relatedness, and growth. Human needs in organizational settings.* Free Press, New York. 1972.

Barwise, J. and Perry, J.: *Situationen und Einstellungen - Grundlagen der Situationssemantik.* Walter de Gruyter, Berlin, 1987.

Billings, C. E.: *Aviation automation - the search for a human-centred approach.* Erlbaum, Mahwah, NJ, 1997.

Billings, C. E.: *Human centred automation: a concept and guidelines.* NASA Technical Memorandum 103885, Moffett Field, CA, August 1991.

Boy, G.: *The group elicitation method: An introduction.* In N. Shadbolt, K. O'Hara, and G. Schreiber, editors, *Advances in Knowledge Acquisition*, 9th European Knowledge Workshop, EKAW '96, Nottingham, United Kingdom, pages 290-305, May 1996.

Brachmann, R. J.: *On the Epistemological Status of Semantic Networks.* In *Associative Networks*, edited by N. V. Fiedler, 3 – 50. New York: Academic Press. 1979.

Brenner, W., Zarnekow, R., and Wittig, H.: *Intelligente Softwareagenten.* Springer, Heidelberg, 1998.

Brüning, G., Hafer, X., Sachs, G.: *Flugleistungen.* Springer Verlag Heidelberg. 1986

Champigneux, G. and Joubert, T.: *Copilote Electronique Project.* In 4th Joint GAF/RAF/USAF Workshop on The Human Electronic Crew: The Right Stuff ?, Kreuth, Ge, 1997.

Champigneux, G., Gaudry, Y., Gilles, A., Nory, P., d'Orso, M.: *Vers une Intelligence Artificielle embarquée pour l'assistance au pilote: le Copilote Electronique.* Conférence IA et Défense AVIGNON 1989.

Doran und Franklin, 1997

Dörner, D.: *Problemlösen als Informationsverarbeitung.* Kohlhammer, Studententext: Standard Psychologie, 1979.

Dudek, H. L.: *Wissensbasierte Pilotenunterstützung im Ein-Mann-Cockpit bei Instrumentenflug.* PhD thesis, Institut für Systemdynamik und Flugmechanik, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München, Germany, 1990.

Eisenführ, F., Weber, M.: *Rationales Entscheiden.* Springer Verlag Heidelberg, 1993.

Endsley, M. R. and Kiris, E. O.: *The out-of-the-loop performance problem and level of control in automation.* *Human Factors*, pages 381-394, 1995.

Federal Aviation Administration: *Human Factors Team Report on: The Interface between Flight-Crews and Modern Flight Deck Systems,* June 18 1996.

Funk, K., Suroteguh, C., Wilson, J., and Lyall, B.: *Flight deck automation and task management.* In *Proceedings of the IEEE 1998 International Conference on Systems, Man and Cybernetics.* San Diego, CA., October 1998.

Funk, K.: *Cockpit task management: Preliminary definitions, normative theory, error taxonomy, and design recommendations.* *The International Journal of Aviation Psychology*, pages 271-285, 1, 4 1991.

- Gerlach, M.: *Schnittstellengestaltung für ein Cockpitassistenzsystem unter besonderer Berücksichtigung von Spracheingabe*. PhD thesis, Institut für Systemdynamik und Flugmechanik, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München, Germany, 1996.
- Gibson, aus Guski"’Wahrnehmung”
- Hart, A.: *Knowledge Acquisition for Expert Systems*. Kogan Page Ltd, London, 1986.
- Haugeneder, H. und Steiner, D.: *Co-operation Agents: Concepts and Applications*. In: N. R. Jennings and M. J. Wooldridge (eds.) *Agent Technology: Foundations, Applications, and Markets*. Springer Verlag, 1998.
- Johannsen, G.: *Mensch-Maschine-Systeme*. Springer Verlag, Heidelberg, 1993.
- Karbach, W. und Linster, M.: *Wissensakquisition für Expertensysteme: Techniken, Modelle und Softwarewerkzeuge*. Hanser Verlag, München, Wien. 1990.
- Klix, F.: *Information und Verhalten*. Hans Huber, 1971.
- Kupferer, R.: *Rotorcraft Pilot’s Associate: From Concept to Flight Demonstration*. AHS-The Vertical Flight Society, 55<sup>th</sup> Annual Forum & Technology Display, Montreal (Quebec) Canada. May/Roto 25-27, 1999.
- Leontjew, A. N.: *Tätigkeit, Bewußtsein, Persönlichkeit: Studien zur Kritischen Psychologie*. Köln, 1982
- Lux, A, de Greef, P., Bomarius, F., Steiner, D.: *A generic framework for human-computer co-operation*. In: M. Huns, M. Papazoglou, G. Schlageter (Eds.) *International Conference on Intelligent and Co-operative Information Systems*. IEEE Computer Society Press, 89-97. 1993.
- Mag, W.: *Entscheidung und Information*. Verlag Franz Vahlen GmbH, München, 1977.
- Maslow, A.H.: *A theory of human motivation*. Psychological Review, 50, 370-396, 1943
- Maslow, A.H.: *Motivation and personality*. Harper, New York. 1954.
- McBryan, B., Joy, M.: *Rotorcraft Pilot’s Associate Hierarchical Planning*. AHS-The Vertical Flight Society, 55<sup>th</sup> Annual Forum & Technology Display, Montreal (Quebec) Canada. May 25-27, 1999.
- Minsky, M. L.: *Mentopolis*. Ernst Klett Verlag für Wissen und Bildung GmbH, Stuttgart. 1985.
- Neisser, U.: *Cognition and Reality. Principles and implications of cognitive Psychology*. Freeman, San Francisco. 1976.
- Newell, A.: *Unified Theories of Cognition*. Harvard University Press. Cambridge, Massachusetts, 1990.
- Onken R.; Walsdorf A.: *Assistance Systems for Vehicle Guidance: Cognitive Man-Machine Cooperation*. BASYS 2000, 4th IFIP/IEEE International Conference on IT for Balanced Automation Systems 27.-29. September 2000 Berlin, Deutschland
- Onken, R.: *Funktionsverteilung Pilot - Maschine: Umsetzung von Grundforderungen im Cockpitassistenzsystem CASSY*. In DGLR-Tagung des Fachausschusses Anthropotechnik, 1993.
- Onken, R.: *Knowledge-based cockpit assistance*. In The Role of Intelligent Systems in Defence, St. Hugh's College, Oxford, March., 1995.
- Onken, R.: *The assessment of situation awareness and workload for certification purpose. ???*, 25./26. March 1997.

- Onken, R.: *The cockpit assistant system CASSY as an on-board player in the ATM environment*. In US/Europe ATM R&D Seminar, Paris, 16.-20. June 1997.
- Orasanu, J. M.: *Decision-making in the cockpit*. In E. L. Wiener, B. G. Kanki, and R. L. Helmreich, editors, *Cockpit Resource Management*. Academic Press, San Diego, 1995.
- Prevôt, T.: *Maschinelles Flugplaner für Verkehrsflugzeuge als autonomes und kooperatives System*. PhD thesis, Institut für Systemdynamik und Flugmechanik, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München, Germany, 1995.
- Rasmussen, J., Pejtersen, A. M., and Goodstein, L. P.: *Cognitive Systems Engineering*. Wiley, New York, 1994.
- Rasmussen, J.: *Modelling distributed decision making*. In J. Rasmussen, B. Brehmer, and J. Leplat, editors, *Distributed Decision Making: Cognitive Models for Cooperative Work*, chapter 5. Wiley, New York, 1991.
- Rasmussen, J.: *Skills, rules and knowledge; signals, signs, and symbols, and other distinctions in human performance models*. In IEEE Transaction on Systems, Man and Cybernetics, volume SMC-13, pages 257-266, 1983.
- Reason, J.: *Human Error*. Cambridge University Press. 1996.
- Reimer, U.: *Einführung in die Wissensrepräsentation*. B. G. Teubner Stuttgart, 1991.
- Richter, M. M.: *Prinzipien der künstlichen Intelligenz*. B. G. Teubner Stuttgart, 1989.
- Romahn, S. E.: *Wissensbasierte Unterstützung bei der Benutzung komplexer technischer Systeme - angewendet auf die Arbeit von Piloten mit dem Flight Management System*. PhD thesis, RWTH Aachen, Fakultät für Maschinenwesen, 1997.
- Rosenstiel von, L., Molt, W. Rüttinger, B.: *Organisationspsychologie*. Verlag W. Kohlhammer, Stuttgart, 1995.
- Ruckdeschel, W.: *Modellierung regelbasierten Pilotenverhaltens mit Petrinetzen*. PhD thesis, Institut für Systemdynamik und Flugmechanik, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München, Germany, 1997.
- Schank, R. C. and Childers, P. G.: *Die Zukunft der künstlichen Intelligenz*. DuMont, Köln. Deutschland, 1986.
- Schulte, A. and Stütz, P.: *Evaluation of the cockpit assistant military aircraft CAMA in simulator trials*. In NATO RTO System Concepts and Integration Panel Symposium. Sensor Data Fusion and Integration of the Human Element, Ottawa, Canada, 13.-17. September 1998.
- Searle, J. R. *Speech Acts*. Cambridge University Press, 1969.
- Sternberg, R. J.: *Intelligence, information, processing, and analogical reasoning: The componential analysis of human abilities*. Erlbaum. Hillsdale, NJ, 1977.
- Walsdorf A.; Onken R.: *Cognitive Man Machine Operation: Modelling Operators' General Objectives and its Role within a Cockpit Assistant System*. IEEE International Conference on Systems, Man, and Cybernetics 8.-11. Oktober 2000, Nashville TN, USA.
- Walsdorf A.; Onken R.: *Intelligent Crew Assistant for Military Transport Aircraft*. NATO System Concepts and Integration Panel Symp., Sensor Data Fusion and Integration of the Human Element. Ottawa, Canada, 14-17 Sept. 1998 , pp 1-1 - 1-7.
- Walsdorf A.; Putzer H.; Onken R.: *The Cognitive Process and its Application within Cockpit Assistant Systems*. ITSC '99, IEEE/IEEEJ/JSAI Intern. Conf. on Intelligent Transportation Systems, Tokyo Japan, October 5-8, 1999
- Weizenbaum ...

Wickens, C. D.: *Engineering psychology and human performance*. Harper Collins, New York, NY, 1991.

Wilensky, R.: *Planning and Understanding*. Addison-Wesley Publishing Company, London, GB, 1983.

Wittig, T.: *Maschinelle Erkennung von Pilotenabsichten und Pilotenfehlern über heuristische Klassifikation*. PhD thesis, Institut für Systemdynamik und Flugmechanik, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München, Germany, 1994.

Wysotzki, F. and Geibel, P.: *Skriptum: Methoden der KI - Maschinelles Lernen*. WS 98/99. Technical report, Institut für Angewandte Informatik, TU Berlin, 1998.

# 10 ANHANG

## 10.1 UML KOMPAKT

Auszug aus

**UML kompakt**

von Günter Wahl

Erschienen im OBJEKTSpektrum 2/1998

Alle Rechte vorbehalten

Die Notation der UML umfasst Diagramme für die Darstellung der verschiedenen Ansichten auf das System, vergleichbar mit Bauplänen für Häuser. Auch hier gibt es z.B. einen Grundriss, einen Lageplan, verschiedene Außenansichten und Werkpläne für die Handwerker. Für eine spezielle Aufgabe ist meist eine Diagrammart besser geeignet als andere.

Insgesamt umfasst die UML

Use-Case-Diagramme,

Klassendiagramme,

Interaktionsdiagramme,

Package-Diagramme,

Zustandsdiagramme,

Aktivitätsdiagramme und

Implementierungsdiagramme.

Hinzu kommen noch in jedem Diagramm zahlreiche Stilelemente mit verschiedenen Beschriftungsarten. Insgesamt ist die UML also eine recht umfangreiche Sprache. Dies hat den Vorteil, dass mit der UML viel ausgedrückt werden kann, allerdings auch den Nachteil, dass die Sprache unübersichtlich und schwierig wird. Aber keine Angst! Alles ist eine Frage des Vorgehens und der Detailliertheit, in der ein System dargestellt werden soll.

Zum Vorgehen ist soviel zu sagen, dass bei weitem nicht alles modelliert werden soll. Ein Modell ist eine Ansicht der Realität, die den Zweck hat, einen bestimmten Sachverhalt darzustellen. So zeichnet beispielsweise ein Statiker ein Kräftediagramm, um die Kräfte in einem Dachbalken an einer bestimmten, wichtigen Stelle zu berechnen. Er zeichnet nicht für alle Balken an allen denkbaren Stellen Kräftediagramme – das wäre vollkommen überflüssig. Auch verwendet er ein Kräftediagramm und nicht die Außenansicht, die vielleicht schöner wäre, aber für diesen Zweck vollkommen ungeeignet ist.

Sich zu beschränken, ist eine Möglichkeit, die Komplexität, die sowohl der Methode als auch dem Softwaresystem innewohnt, zu beherrschen. Worauf sollte man sich beschränken? Hierauf eine allgemeine Antwort zu geben, ist nicht möglich. Der Modellierer muss über genügend Erfahrungen und Kenntnisse sowohl über den Fachbereich, als auch über sein Handwerkszeug – in unserem Fall die UML – verfügen.

Glücklicherweise bieten die UML und verschiedene Methoden, wie etwa die OMT, ein Grundrepertoire an, mit dem es möglich ist, alle praktisch wichtigen Problemstellungen in 90 Prozent der Fälle darzustellen. Nur selten werden weiterführende Stilelemente eingesetzt. Die Einsatzgebiete und Eigenschaften der Diagramme habe ich in Tabelle 1 zusammengefasst. Dabei habe ich die Phasen, für welche die Diagramme im allgemeinen besonders gut geeignet sind, angegeben; im Einzelfall können sich Abweichungen hiervon ergeben.

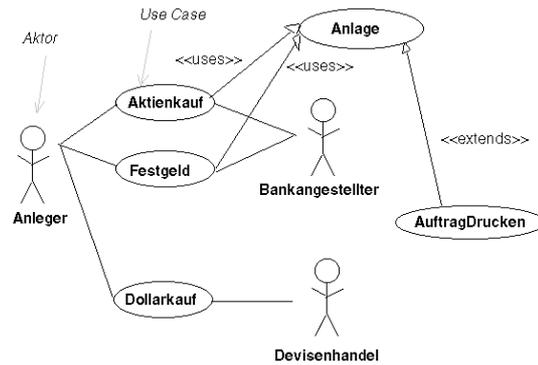
<b>Diagramm</b>	<b>Phase</b>	<b>Einsatzgebiet</b>
<b>Use-Case</b>	Anforderung, Festlegung, Erstellung, Übergabe	Geschäftsprozesse, allgemeine Einsatzmöglichkeiten
<b>Klassendiagramm</b>	Festlegung, Erstellung	So gut wie überall, das Klassendiagramm ist das wichtigste Diagramm der UML.
<b>Interaktionsdiagramm</b>	Anforderung, Festlegung, Erstellung, Übergabe	Zeigt den Nachrichtenfluss und damit die Zusammenarbeit der Objekte im zeitlichen Ablauf.
<b>Sequenzdiagramm</b>		Zeitliche Aufrufstruktur mit wenigen Klassen
<b>Kollaborationsdiagramm</b>		Zeitliche Aufrufstruktur mit wenigen Nachrichten
<b>Package-Diagramm</b>	Erstellung	Groborientierung, in welchem Modul welche Klasse zu finden ist. Aufteilung in Unterprojekte, Bibliotheken, Übersetzungseinheiten.
<b>Zustandsdiagramm</b>	Anforderung, Festlegung, Erstellung, Übergabe	Darstellung des dynamischen Verhaltens.
<b>Aktivitätsdiagramm</b>	Festlegung, Erstellung	Bei parallelen Prozessen und anderer Parallelität, Geschäftsprozesse.
<b>Implementierungsdiagramm</b>	Festlegung, Erstellung, Übergabe	Besonders für die Darstellung von verteilten Anwendungen und Komponenten; allgemein: Darstellung von Implementierungsaspekten (Übersetzungseinheiten, ausführbare Programme, Hardwarestruktur)
<b>Komponentendiagramm</b>		Zusammenhänge der Software
<b>Deployment-Diagramm</b>		Hardwareaufbau

**Tabelle 1: Einsatzgebiete und Eigenschaften der verschiedenen Diagramme**

Im folgenden beschreibe ich die verschiedenen Diagramme und gebe aus meiner Sicht Hinweise, Kommentare und Anregungen über das Einsatzgebiet. Wo es mir angebracht erscheint, habe ich mit Kritik nicht zurückgehalten. Einzelne Eigenschaften und Elemente werden als Durchbruch in der Geschichte der Softwareentwicklung angepriesen, sind aber Insidern längst bekannt.

### **Use-Case**

Soviel gibt es über die Use-Cases gar nicht zu erzählen – sie sind schnell erklärt, aber ungemein nützlich. Die Use-Cases wurden von I. Jacobson eingeführt (vgl. [Jac92]). Sie beschreiben das Zusammenwirken von Personen mit einem System. Personen werden als *Aktoren* bezeichnet. Die Beschränkung der Aktoren auf Menschen kann verallgemeinernd auch aufgehoben werden, und die Aktoren können damit auch andere Systeme sein. Handelt es sich bei den Aktoren um Personen, so müssen diese nach ihrem Rollenverhalten im Bezug auf das System und nicht nach den Menschen unterschieden werden. Ein Beispiel zeigt Abbildung 131. Hier kann z. B. der Bankangestellte auch selber ein Anleger sein.



**Abbildung 131 Use-Case „Geld anlegen“**

Unter einem Use-Case wird eine typische Handlung verstanden, die ein Benutzer mit dem System ausführt, z. B. „Aktienkauf“. In das Use-Case-Diagramm werden die Use-Cases als Ellipsen eingezeichnet. Use-Cases können beliebig kompliziert und umfangreich sein. Verbindungen zwischen den Use-Cases und den Aktoren werden durch Linien hergestellt. Damit wird angezeigt, welche Aktoren an dem entsprechenden Use-Case beteiligt sind. Im Use-Case-Diagramm gibt es zwei weitere Verbindungstypen: <<uses>> und <<extends>>. Der Verbindungstyp <<uses>> wird verwendet, wenn zwei oder mehr Use-Cases einen ähnlichen Aufbau haben und Verhalten durch Kopieren wiederholt dargestellt werden muss. Dadurch ist es möglich, Verhalten von den Use-Cases in einen separaten Use-Case zu verlagern, und <<uses>>- Verbindungen zu den entsprechenden Use-Cases zu zeichnen. In Abbildung 132 ist dies bei Anlage und Festgeld der Fall. Der Verbindungstyp <<extends>> klammert auch Verhalten aus, jedoch ist die Zielsetzung eine andere. Bei <<extends>> wird das Verhalten erweitert. Dies ist beispielsweise bei AuftragDrucken der Fall. AuftragDrucken stellt dem Use-Case Anlage zusätzliche Funktionalität zur Verfügung – hier z. B. das Drucken eines Auftrags (beachten Sie dabei bitte die Pfeilrichtung).

### Klassendiagramm

Klassendiagramme sind der zentrale Bestandteil der UML und auch zahlreicher objektorientierter Methoden. Wie die Klassen ermittelt werden, darüber gibt die UML keine Auskunft; hierfür gibt es andere Techniken, z.B. CRC-Karten (Abk. für „Class, Responsibility and Collaboration“) oder die Substantiv-Technik (vgl. [Rum91]). Die UML beschreibt lediglich die Notation und die Semantik:

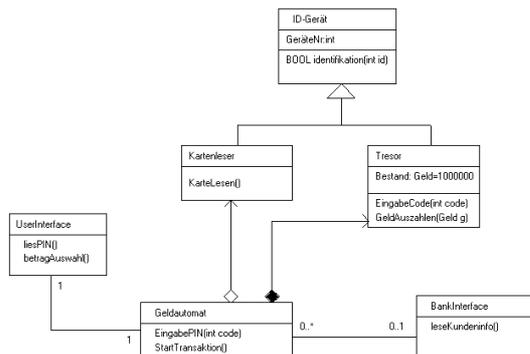
*Objekte:* Konzept, Abstraktion oder Gegenstand mit klarer Abgrenzung und präziser Bedeutung, z. B. der „rote Apfel“.

*Klassen:* Gruppen von Objekten mit ähnliche Eigenschaften, z.B. „Äpfel“.

Die strenge visuelle Unterscheidung zwischen Objekten und Klassen entfällt in der UML. Objekte werden von den Klassen dadurch unterschieden, dass ihre Bezeichnung unterstrichen ist, sonst sind die Symbole gleich. Auch können Klassen und Objekte zusammen im Klassendiagramm auftreten.

### Elemente und Darstellung des Klassendiagramms

Das Klassendiagramm beschreibt die statische Struktur der Objekte in einem System sowie ihre Beziehungen untereinander. Abbildung 132 soll diesen Sachverhalt anhand eines Beispiels für ein gesamtes Klassendiagramm verdeutlichen. Das Bild zeigt ein – zugegebenermaßen stark vereinfachtes – Klassendiagramm einer Bank mit einer automatischen Geldausgabe an einem Geldautomaten.



**Abbildung 132: Klassendiagramm "automatische Geldausgabe"**

Die *Klasse* ist das zentrale Element; sie wird als Rechteck dargestellt (z. B. Geldautomat). Die gefundenen Klassen werden durch Linien miteinander verbunden. Diese Linien stellen die Elemente *Assoziation*, *Aggregation*, *Komposition* und *Vererbung* dar. Die Assoziation stellt eine allgemeine Beziehung zwischen zwei Klassen dar – über die Realisierung wird dabei nichts ausgesagt. Eine besondere Assoziation ist die Aggregation, die durch eine Raute an der Linie dargestellt wird (z. B. zwischen Kartenleser und Geldautomat). Sie gibt an, dass eine Klasse Kartenleser in der Klasse Geldautomat „enthalten“ ist (Ist-Teil-von-Beziehung). An einer Assoziation können noch Multiplizitäten, d. h. Zahlen oder Zahlbereiche, angegeben werden. Diese bestimmen die Anzahl der Objekte, die miteinander in Beziehung stehen. Beispielsweise ist an einem Geldautomat genau ein UserInterface vorhanden. Die Komposition ist eine stärkere Form der Aggregation, die durch eine ausgefüllte Raute dargestellt wird (siehe Tresor). Bei der Beziehung Komposition handelt es sich um ein physikalisches Enthaltensein. In C++ können Aggregation und Komposition wie folgt realisiert werden:

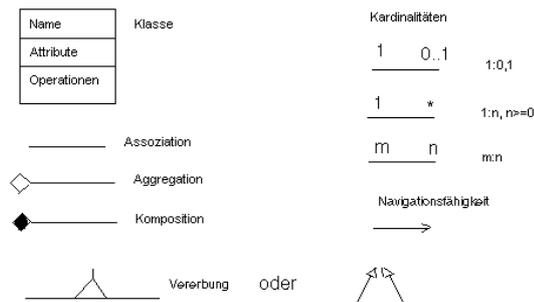
```

class Geldautomat
{
    // ...
    // Aggregation durch einen Zeiger realisiert:
    Kartenleser* pKartenleser;
    // Tresor ist ein Datenmember:
    Tresor tresor;
}
  
```

Jede Assoziation kann eine Richtung besitzen; hierfür wird ein Pfeil am Ende der Assoziation angebracht. Zugriffe können dann nur in Pfeilrichtung erfolgen. Man verwendet deshalb den Begriff der Navigationsfähigkeit (engl. Navigability). Eine Assoziation mit Pfeil kann als Zeiger in einer Programmiersprache betrachtet werden.

Die Vererbung stellt eine Verallgemeinerung von Eigenschaften dar – sie wird auch als Spezialisierung und Generalisierung oder "Ist-ein"-Beziehung bezeichnet. Z. B. *ist* ein Auto ein Fahrzeug. Ein Fahrzeug hat generelle Eigenschaften eines Autos, und ein Auto spezialisiert die Eigenschaften von Fahrzeugen.

Die wichtigsten Elemente eines Klassendiagramms sind nochmals in Abbildung 133 dargestellt. Elemente für eine erweiterte Modellierung finden sich in [Rat97]. In Abbildung 133 sind für die Vererbung zwei gleichwertige Schreibweisen angegeben.



**Abbildung 133: Elemente eines Klassendiagramms**

Ausführlicher betrachtet hat eine Klasse den in

Abbildung 134 gezeigten Aufbau - nun können Sie das Beispiel aus Abbildung 132 etwas besser verstehen:



**Abbildung 134: Aufbau einer Klasse: Beispiel und allgemeiner Aufbau**

Der Tresor hat einen Anfangsbestand von 1.000.000. Eine Klasse besteht aus *Attributen* und *Methoden*. Attribute stellen die Eigenschaften der Objekte einer Klasse dar und bilden den Datenbestand der Klasse. Was mit diesen Daten getan werden kann, legen die Methoden fest. Methoden sind die aus anderen Sprachen bekannten Funktionen.

Wer auf die Daten zugreifen kann, wird durch die *Sichtbarkeit* der Daten bestimmt. Sichtbarkeiten können frühestens nach der Festlegungsphase eingeführt werden. *private* bedeutet, dass nur die Methoden dieser Klasse Zugriff haben, und *protected* heißt, dass auch Objekte der abgeleiteten Klassen Zugriff haben. *public* besagt, dass ein Zugriff jederzeit möglich ist. Die Sichtbarkeit kann sowohl für Attribute, als auch für Methoden angegeben werden.

## Interaktionsdiagramm

Es gibt zwei Arten von Interaktionsdiagrammen:

- Sequenzdiagramme und
- Kollaborationsdiagramme.

Beide beschreiben die zeitlichen Abläufe, d.h. Aufrufsequenzen. Ausgehend von Fallbeispielen – den *Szenarios* – können Interaktionsdiagramme erstellt werden. Deshalb werden die Interaktionsdiagramme auch *Szenariodiagramme* genannt. Beim Erstellen der Diagramme konzentriert man sich auf die „wichtigsten“ Fälle (primäre Szenarios). Anschließend werden die Sonderfälle mit einbezogen und eventuell weitere Diagramme erstellt (sekundäre Szenarios).

Bei meiner täglichen Arbeit sind die Interaktionsdiagramme – noch vor den Klassendiagrammen – die für mich wichtigsten Diagramme. Sie werden häufig erstellt und dienen sowohl als Diskussionsgrundlage, als auch als Grundlage für die Implementierung und Dokumentation.

## Sequenzdiagramme

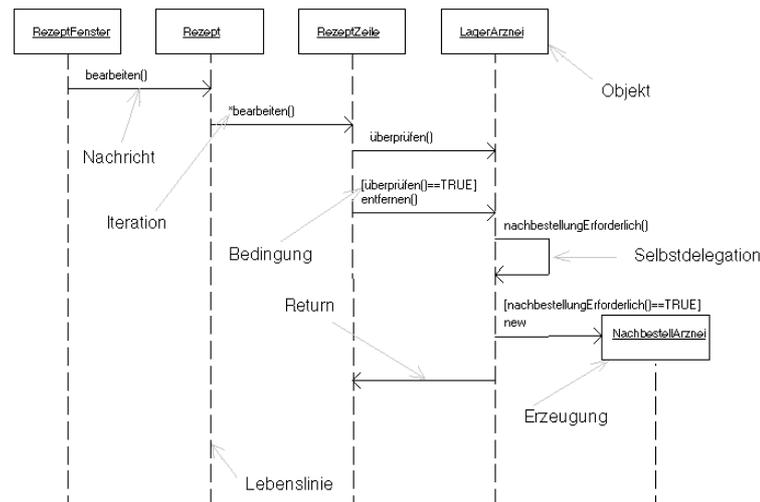
Doch zunächst zu den Sequenzdiagrammen. Als Beispiel für die Erklärung der Interaktionsdiagramme soll die Adler-Apotheke dienen. In dieser Apotheke werden die Rezepte der Kranken in einen Computer eingegeben. Die Software ist wie folgt aufgebaut:

In einem Eingabefenster RezeptEingabe werden die Rezepte eingegeben. Dieses Fenster sendet die Nachricht bearbeiten an das Rezept.

Das Rezept sendet die Nachricht bearbeiten an jede RezeptZeile von Rezept.

Jede RezeptZeile überprüft das Objekt LagerArznei wie folgt: Der Rückgabewert TRUE bedeutet, dass die RezeptZeile die entsprechende Menge der LagerArznei aus dem Lager entfernt hat. Anderenfalls ist die Menge der LagerArznei unter den Mindestbestand für eine Nachlieferung gefallen, und eine Nachlieferung dieser LagerArznei wurde veranlasst.

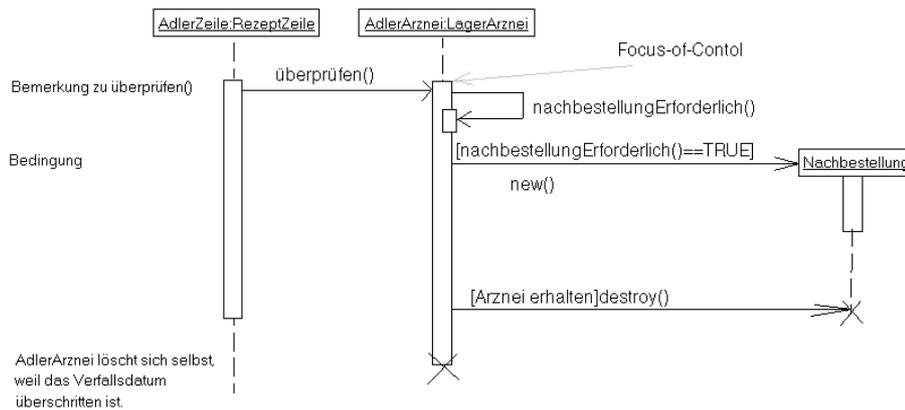
Eine Darstellung als Sequenzdiagramm zeigt Abbildung 135.



**Abbildung 135: Sequenzdiagramm „Adler Apotheke“**

Die Zeitachse verläuft von oben nach unten. Objekte sind als Rechtecke am oberen Ende von gestrichelten Linien – den „Lebenslinien“ – dargestellt. Am linken Rand können noch Kommentare stehen. Zwischen den Lebenslinien wird der Austausch von Nachrichten durch Pfeile dargestellt. Eine Nachricht kann an eine Bedingung geknüpft sein, die in eckigen Klammern angegeben wird. Optional können noch Rücksprünge eingetragen werden – im allgemeinen verschlechtern diese aber die Lesbarkeit des Diagramms. Einige andere Details des Diagramms, wie etwa die Erzeugung von Objekten und die Iteration, erklären sich von selbst. Überhaupt wurde bei der UML viel Wert auf einfache Verständlichkeit gelegt.

Die Sequenzdiagramme wurden – um ihre Aussagekraft zu erhöhen – um einige nützliche Notationselemente erweitert; das Ergebnis ist in Abbildung 136 dargestellt.



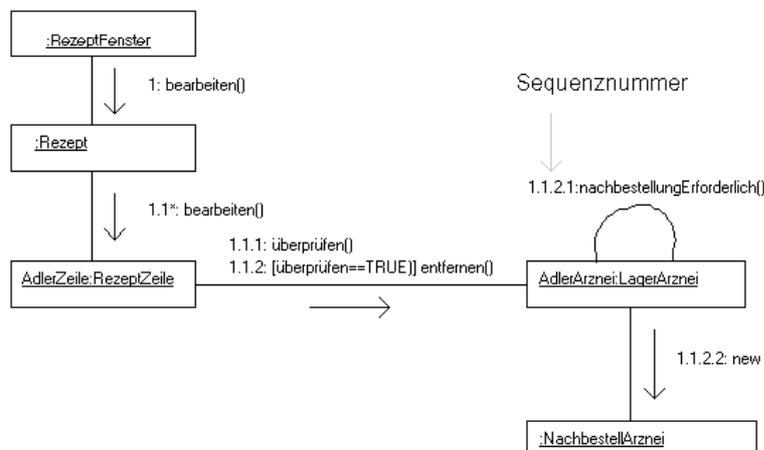
**Abbildung 136: Sequenzdiagramm mit Focus-of-Control, Erzeugen/Löschen und Bedingungen**

Der *Focus-of-Control* (das langegezogene Rechteck) gibt an, wo eine Aktivität ausgeführt wird. Ist ein Objekt ohne Aktivität vorhanden, wird dies durch eine gestrichelte Linie angezeigt. Objekte werden durch einen Pfeil auf das Rechteck erzeugt, ein Löschen wird durch ein Kreuz dargestellt. Über den Pfeilen stehen die Operationen, die ausgeführt werden. In eckigen Klammern können Bedingungen angegeben werden, wann die Operation ausgeführt wird. Ruft das Objekt eine Operation von sich selbst auf, etwa `nachbestellungErforderlich()`, dann zeigt der Pfeil auf die Lebenslinie zurück. Zusätzlich können am linken Rand noch Bemerkungen zu den einzelnen Operationen angegeben werden. Version 1.1 der UML bietet bei Bedarf noch einige weitere, feinere Notationselemente.

### Kollaborationsdiagramm

Das Kollaborationsdiagramm und das Sequenzdiagramm beinhalten die gleichen Informationen und unterscheiden sich lediglich in der Darstellung. Die automatische Umwandlung eines Sequenzdiagramms in ein Kollaborationsdiagramm und umgekehrt ist möglich. Bei vielen Klassen und wenigen Nachrichten sind Kollaborationsdiagramme übersichtlicher als Sequenzdiagramme. Sind wenige Klassen und viele Nachrichten vorhanden, so ist das Sequenzdiagramm besser geeignet.

Ausgehend von unserem Beispiel, der Adler-Apotheke, ergibt sich für das vorher vorgestellte Beispiel der Bearbeitung von Rezepten das Kollaborationsdiagramm nach Abbildung 137. Das Kollaborationsdiagramm ist nicht nur ein wichtiges Hilfsmittel beim Entwurf und der Analyse, es hat mir auch schon oft bei der Fehlersuche geholfen. Während der Fehlersuche zeichne ich ein Kollaborationsdiagramm - damit lassen sich die Aufrufstruktur und das Zustandekommen eines Fehlers gut verfolgen.



**Abbildung 137: Kollaborationsdiagramm für „Adler Apotheken Rezepte“**

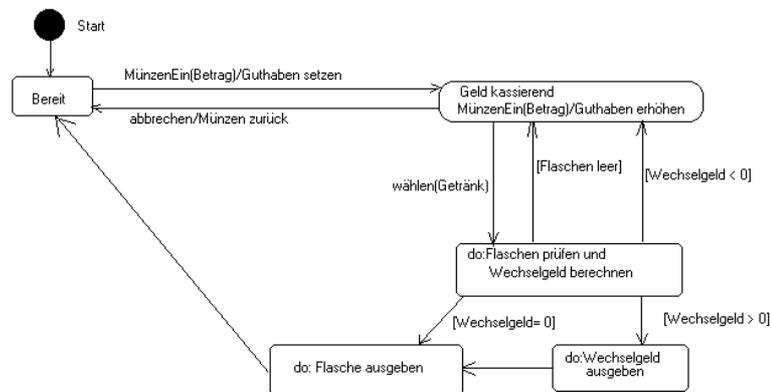
## Zustandsdiagramm

Aus den Interaktionsdiagrammen können Zustandsdiagramme entwickelt werden. Alle Szenarios definieren ein bestimmtes Zustandsdiagramm genauer. Die Szenarios sind so zu wählen, dass das Zustandsdiagramm einer Klasse vollständig definiert wird.

Theoretisch kann das Verhalten jeder Klasse durch ein Zustandsdiagramm beschrieben werden. In der Praxis ist das etwas einfacher. Nur für Klassen mit interessantem Verhalten werden Zustandsdiagramme angefertigt. Zustandsdiagramme sind nicht erforderlich für Klassen, die nur als Schnittstelle dienen und das Verhalten weiter delegieren. Das Gleiche gilt für Klassen, die nur Werte zwischenspeichern.

Ein interessantes Verhalten weisen unter anderem all diejenigen Klassen auf, deren reale Objekte in der Umgangssprache mit „Automat“ bezeichnet werden, wie z. B. Geldautomaten, Getränkeautomaten, automatische Aufzüge oder automatische Garagenöffner. Aus meiner täglichen Praxis kenne ich noch weitere Klassen mit interessantem Verhalten – Beispiele sind Klassen, die Übertragungsprotokolle realisieren, sowie Klassen, die Benutzeraktionen abarbeiten (z.B. Kennworteingabe).

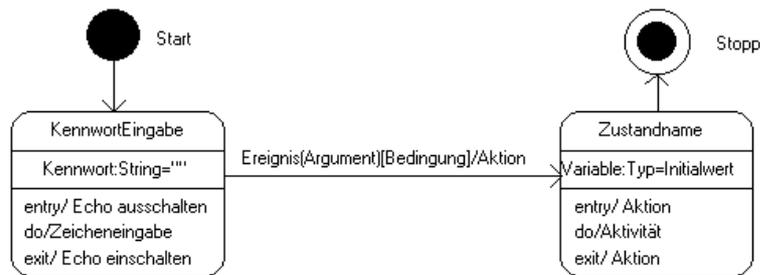
Im folgenden soll das Beispiel eines Getränkeautomaten, aus dem verschiedene Getränke entnommen werden können, behandelt werden (der Automat ist in Abbildung 138 wiedergegeben).



**Abbildung 138: Zustandsdiagramm für einen Getränkeautomaten**

Rechtecke mit abgerundeten Ecken stellen die Zustände dar; zwischen jedem Nachrichtenaustausch im Sequenzdiagramm eines Objekts kann ein Zustand liegen. Durch das Eintreffen von Ereignissen kann ein anderer Zustand erreicht werden, was durch die Pfeile angedeutet wird, die eine Beschriftung für das auslösende Ereignis haben. Die Pfeile stellen die Übergänge dar. In den Zuständen werden Operationen mit einer gewissen Zeitdauer ausgeführt, die Aktivitäten genannt werden. Im Gegensatz dazu wird die Zeitdauer von Aktionen mit Null angenommen. Aktionen sind Operationen, die an den Übergängen ausgeführt werden. Angezeigt werden die Aktionen durch einen Schrägstrich vor dem Namen. Selbstverständlich können Übergänge auch an Bedingungen, die wieder in eckigen Klammern stehen, geknüpft werden.

Aus meiner Sicht (und auch nach [You96]) ist die Unterscheidung zwischen Aktivitäten und Aktionen akademisch. Jede reale Operation hat eine Zeitdauer – ob diese für eine Aufgabe relevant ist, hängt vom Einzelfall und nicht von der Darstellung in einem Diagramm ab. Den allgemeinen Aufbau eines Zustandsdiagramms zeigt Abbildung 139.



**Abbildung 139: Zustandsdiagramm Symbole**

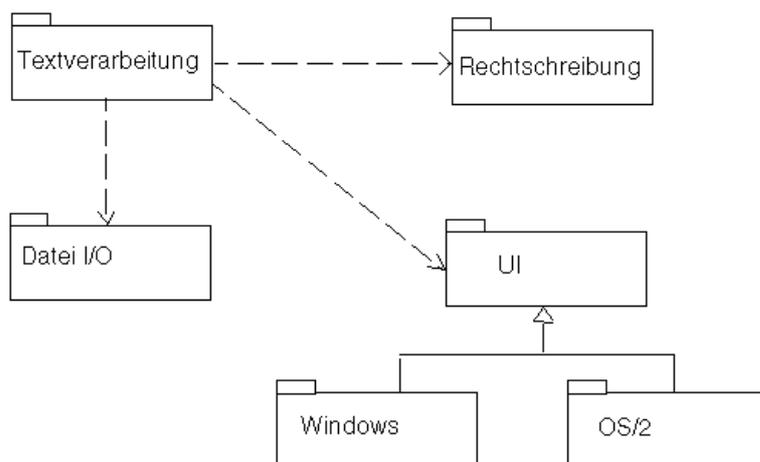
Die Wörter *entry*, *do* und *exit* sind reserviert und können als Bezeichnungen für Ereignisse nicht verwendet werden. *entry* gibt an, welche Aktion beim Eintritt in einen Zustand ausgeführt wird. Ähnlich bezeichnet *exit* die Aktion, die beim Verlassen des Zustandes ausgeführt wird. Und schließlich beschreibt *do* den Aufruf eines verschachtelten Zustandsdiagramms. Allgemein bezeichnet *do* eine andauernde Aktivität, die unendlich lange andauern kann.

Erweiterungen auf parallele Verarbeitung, Oberzustände usw. sind möglich und können in der Literatur unter „parallele Zustandsdiagramme“ (z. B. [Fow97], [Rat97], [Rum91]) nachgelesen werden. Falls Sie nicht sehr spezielle Probleme behandeln, reichen die hier vorgestellten Elemente für das Zustandsdiagramm bei weitem aus. Theoretisch ist ein Zustandsdiagramm auf eine Klasse bezogen. Wesentlich bedeutender finde ich es, dieses Konzept auch während der Analyse – unabhängig von speziellen Klassen – zu haben. Dann kann auf ein Blatt Papier ein Zustandsdiagramm, das sich nicht auf eine spezielle Klasse bezieht, gezeichnet und diskutiert werden. Später lassen sich daraus bei Bedarf Zustandsdiagramme für spezielle Klassen ableiten.

Zustandsdiagramme für die Formatprüfung von Benutzereingaben, z.B. die Uhrzeit, können sehr gut direkt gezeichnet werden. Aber auch Übertragungsprotokolle werden häufig über Zustandsdiagramme definiert. Nicht immer lässt sich ein Zustandsdiagramm direkt zeichnen, dann sind vorher geeignete Szenarios und gegebenenfalls Interaktionsdiagramme zu erstellen.

### Package-Diagramm

*Package*-Diagramme dienen der Strukturierung der verschiedenen Darstellungen. Damit werden Gruppen von Diagrammen oder Elementen zusammengefasst. Dies ist insbesondere wichtig für den Überblick über das System und die Aufteilung in verschiedene Übersetzungseinheiten. Ein Beispiel ist in Abbildung 140 zu sehen.



**Abbildung 140: Beispiel für ein Package-Diagramm**

Ein Package-Diagramm besteht im wesentlichen aus *Packages* (dargestellt durch große Rechtecke mit kleinem Rechteck links oben) und Abhängigkeiten (den gestrichelten Pfeilen). Eine Abhängigkeit gibt an, dass bei einer Änderung des Packages an der Pfeilspitze das Package am anderen Ende der gestrichelten Linie eventuell

geändert bzw. neu übersetzt werden muss. Ob eine Änderung bzw. Neuübersetzung erforderlich ist, muss im Einzelfall geprüft werden. Packages, die nicht durch eine Abhängigkeit verbunden sind, können bei der Änderung außer Acht gelassen werden. Packages können weitere Packages enthalten.

Packages geben zum einen einen guten Überblick über die Auswirkung von Änderungen und ermöglichen zum anderen einen Überblick über das Gesamtsystem. Insbesondere für größere Systeme ist eine solche Strukturierung wichtig.

In Abbildung 140 muss die Applikation Textverarbeitung auf Änderungen überprüft werden, wenn sich das Package Datei I/O, UI oder Rechtschreibung ändert. Das Package UI soll hier eine abstrakte Klasse darstellen, die lediglich als Schnittstelle dient. Damit ist es möglich, dass Änderungen in den Packages Windows und OS/2 keine Auswirkungen auf das Package Textverarbeitung haben. Aus diesem Grund ist hier eine Generalisierung eingezeichnet (vgl. Klassendiagramme).

### Aktivitätsdiagramm

Aktivitätsdiagramme sind ganz neu; sie stellen eine Mischung verschiedener, bekannter Darstellungstechniken dar. Grundlage waren unter anderem Zustandsdiagramme, Flussdiagramme und Petrinetze. Da Aktivitätsdiagramme noch neu sind, liegen auch noch keine umfangreichen Erfahrungen vor. Ihre weitere Entwicklung und Bedeutung wird sich erst erweisen, sonst werden sie das gleiche Schicksal wie die Petrinetze erleiden - bekannt, theoretisch untersucht, in der Praxis aber selten verwendet.

Petrinetze sind ihrem Wesen nach für parallele Prozesse und die Modellierung von Arbeitsabläufen (Workflow-Modellierung) geeignet, allerdings nur, solange die Problemstellung nicht zu komplex wird oder sich die Problemstellung reduzieren lässt. Umfangreiche Zusammenhänge sind mit Petrinetzen nur schwer darstellbar. Für die Aktivitätsdiagramme gilt das gleiche.

Ich möchte die Aktivitätsdiagramme deshalb nicht überbetonen und hier nur eine kurze Darstellung geben. Grundelemente dieser Diagrammart sind:

- Aktivitäten,
- Transitionen,
- Synchronisationslinien und
- optionale „Swimlanes“ (Schwimmbahnen).

Aktivitäten sind Zustände, in denen Vorgänge ablaufen. Transitionen erfolgen automatisch am Ende der Aktivitäten; sie werden durch Pfeile dargestellt. Synchronisationslinien werden durch Striche dargestellt und schalten, wenn alle Eingangstransitionen vorhanden sind. „Swimlanes“ teilen ein Aktivitätsdiagramm so ein, dass die Bereiche, die sie abgrenzen, einzelnen Klassen zugeordnet werden können.

Ein Beispiel für ein Aktivitätsdiagramm zeigt Abbildung 141. Zusätzlich können noch Bedingungen eingeführt werden, welche die Transitionen näher bestimmen.

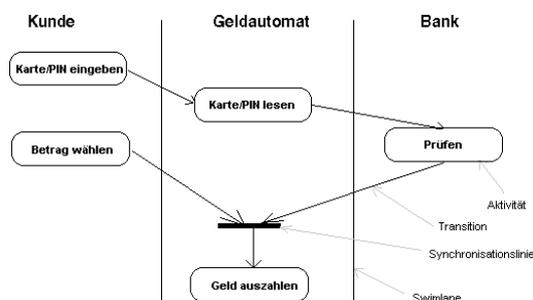


Abbildung 141: Beispiel für ein Aktivitätsdiagramm

## Implementierungsdiagramm

Implementierungsdiagramme zeigen Aspekte der Implementierung. Diese umfassen die Codestructur und die Struktur des Systems zur Ausführungszeit. Insbesondere für verteilte Anwendungen und Komponentensoftware ist das wichtig. Von den Implementierungsdiagrammen gibt es zwei Formen:

Komponentendiagramme und  
Deployment-Diagramme.

Komponentendiagramme zeigen die Struktur des Codes und Deployment-Diagramme die des Laufzeitsystems. Mit zunehmender Bedeutung der verteilten Anwendungen und Komponentensoftware werden diese Diagramme an Bedeutung gewinnen. Bis jetzt habe ich für Projekte nur wenige Implementierungsdiagramme gezeichnet. Eine Ausnahme hiervon bilden Diagramme, die wichtige Übersetzungsabhängigkeiten zeigen.

### Komponentendiagramm

Das Komponentendiagramm (siehe Abbildung 143) zeigt die Abhängigkeiten unter den Softwarekomponenten, genauer die Abhängigkeiten zwischen Quellcode, Binärcodekomponenten und ausführbaren Programmen. Einige dieser Komponenten existieren nur während des Übersetzungsvorgangs, einige nur während des Linkens, andere zur Ausführungszeit und wieder andere die ganze Zeit über. Im Komponentendiagramm haben die Darstellungen nur Typencharakter, im Gegensatz zum Deployment-Diagramm, wo sie zu Instanzen werden (d.h. die Bezeichnungen werden unterstrichen).

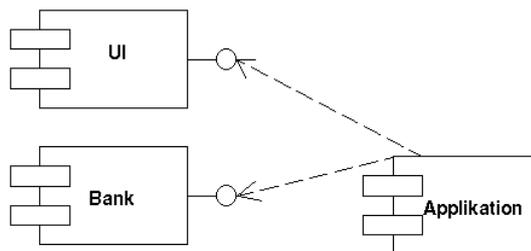


Abbildung 142: Beispiel für ein Komponentendiagramm

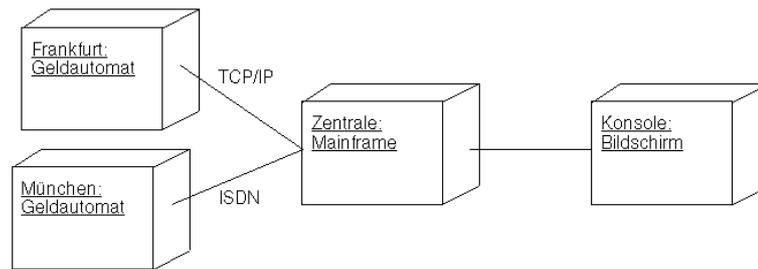
Die Komponenten werden als drei ineinander verschachtelte Rechtecke gezeichnet; ihre Schnittstellen sind Striche mit Kreisen am Ende. Dadurch können die verschiedenen Schnittstellen der Komponenten dargestellt werden. Das Diagramm enthält ferner Abhängigkeiten in Form von gestrichelten Pfeilen.

Sie werden sich fragen, wo der Unterschied zu den Package-Diagrammen liegt. Der Unterschied besteht darin, dass die Package-Diagramme ein allgemeines Mittel für die Strukturierung darstellen, während die Komponentendiagramme Implementierungsaspekte zeigen. In der Praxis entspricht eine Komponente einem Package, aber nicht jedes Package einer Komponente.

Die Abhängigkeiten eines Package-Diagramms, das in ein Komponentendiagramm umgewandelt werden kann, sind genau die gleichen wie die im Komponentendiagramm.

### Deployment-Diagramme

Zur Darstellung der Hardware werden Deployment-Diagramme verwendet. Diese besitzen sogenannte *Knoten*, die eine Verarbeitungs- oder Hardwareeinheit darstellen. Knoten werden als Quader gezeichnet. Unter den Knoten existieren *Verbindungen*: Dabei handelt es sich um die physikalischen Kommunikationspfade, die als Linien eingezeichnet werden (ein Beispiel zeigt Abbildung 143).



**Abbildung 143: Beispiel für ein Deployment-Diagramm**

In das Deployment-Diagramm können noch die Komponenten und ihre Abhängigkeiten eingezeichnet werden. Somit erhalten Sie einen Überblick, wo die Komponenten ausgeführt werden.

### Stereotypen

Die Erklärung des Begriffes „Stereotyp“ möchte ich nicht versäumen. Früher oder später werden Sie im Zusammenhang mit der UML auf diesen Begriff stoßen. Falls Sie dann nachfolgende Erklärung nicht mehr in Erinnerung haben, können Sie diese zumindest nachlesen.

*Stereotypen* sind Mechanismen, die es erlauben, die UML zu erweitern. Der Begriff wird nur selten ins Deutsche übersetzt, und überhaupt sind Erklärungen dafür recht selten, obwohl der Begriff „Stereotyp“ häufig verwendet wird. Übersetzungen aus dem Lexikon ergeben Begriffe wie „Druckvorlage“, „Muster“, „Musterbeispiel“, „Vorbild“. Stereotypen haben in der UML die Bedeutung „so ähnlich etwa wie“. Stereotypen können eigene Piktogramme (Icons) haben und werden in eckigen Klammern dargestellt, z. B. <<uses>>. Sie erlauben eine weitere Einteilung der Klassen, Abhängigkeiten, Assoziationen etc. So ist z. B. eine Einteilung der Klassen in Schnittstellen-, Kontroll- und Entitätenobjekte (irgendwelche Dinge) möglich.

### Literatur des Anhangs „UML-Kompakt“

[**Boo94**] G. Booch, Object-oriented Analysis and Design with Applikations, 1994 Benjamin/Cummings Publishing Company, 1994

[**Fow97**] M. Fowler, UML Distilled, Applying the Standard Object Modeling Language, 2nd Printing, Addison-Wesley, 1997

[**Jac92**] I. Jacobson, Object-Oriented Software Engineering, A Use Case driven Approach, Addison-Wesley, 1992

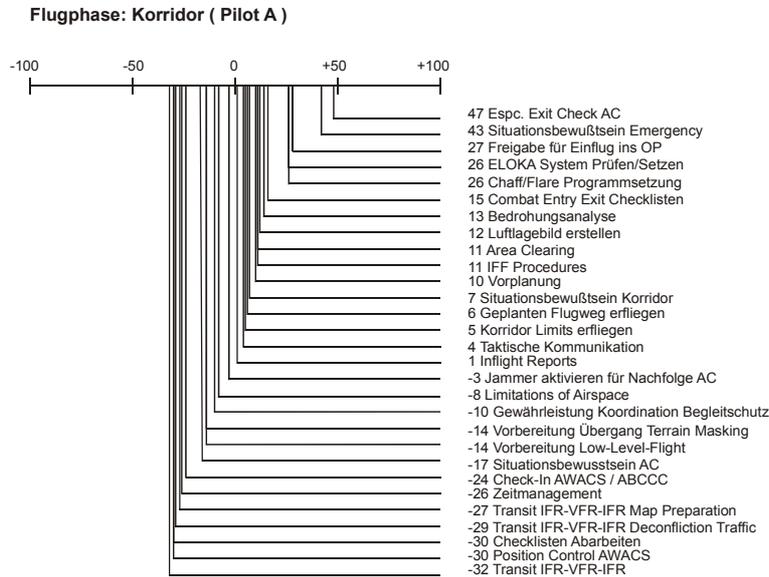
[**Rat97**] UML Unified Modeling Language, Version 1.0 und 1.1 (siehe: <http://www.rational.com>)

[**Rum91**] J. Rumbaugh, Objektorientiertes Modellieren und Entwerfen, Hanser Verlag 1993 (Titel der amerikanischen Ausgabe: Object-Oriented Modeling and Design, Prentice Hall, 1991)

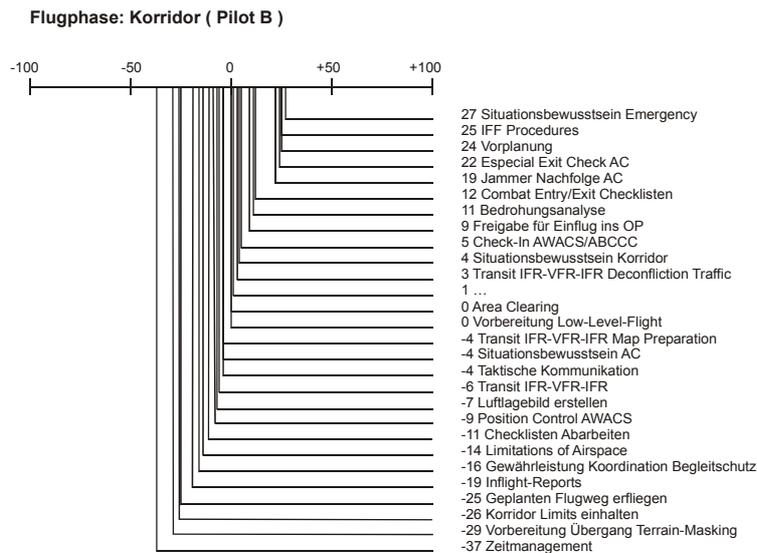
[**You96**] E. Yourdon et. al, Mainstream Objects, Prentice Hall, 1996

**Dipl. Inform. Günter Wahl (E-Mail-Adresse: [GWahl@t-online.de](mailto:GWahl@t-online.de)) ist Softwareingenieur bei der Firma Robert Bosch, wo er bereits zahlreiche Projekte betreut und realisiert hat. Sein Schwerpunkt liegt auf dem Gebiet der objektorientierten Methoden und Programmierung.**

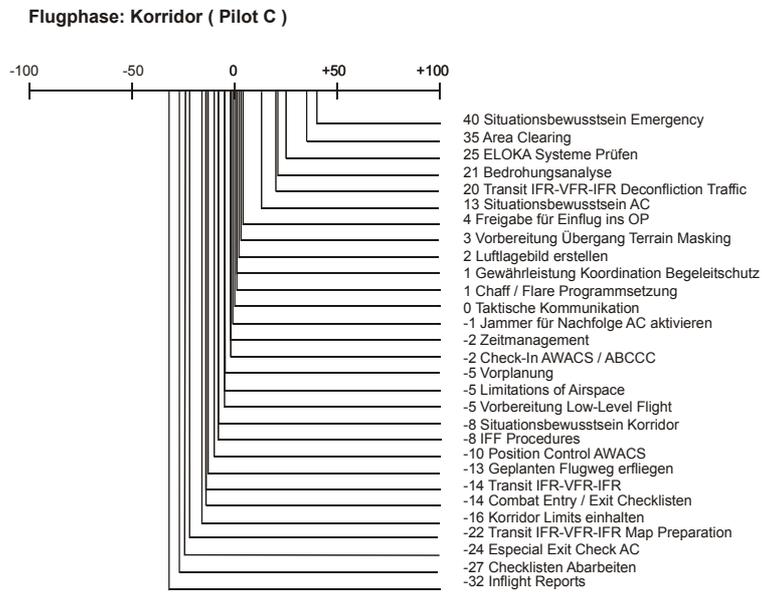
## 10.2 EINZELERGEBNISSE DER GROUP ELICITATION METHOD



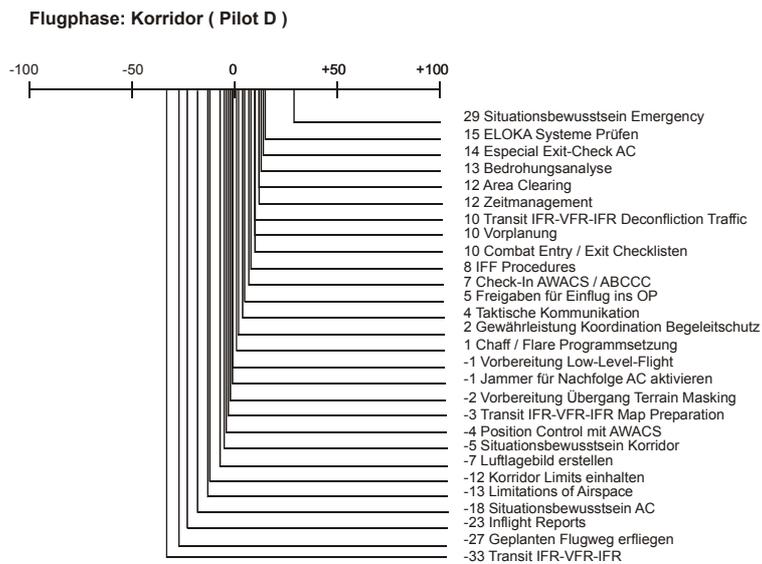
**Abbildung 144: Ergebnis Pilot A zur Flugphase Korridor**



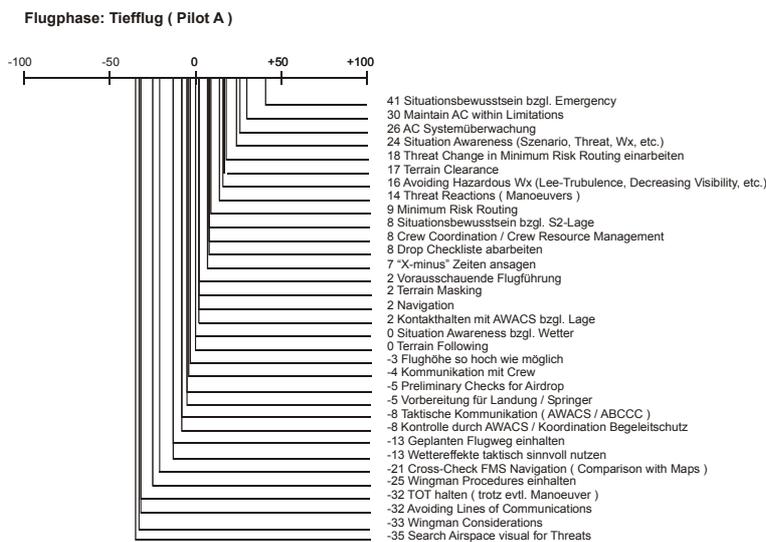
**Abbildung 145: Ergebnis Pilot B zur Flugphase Korridor**



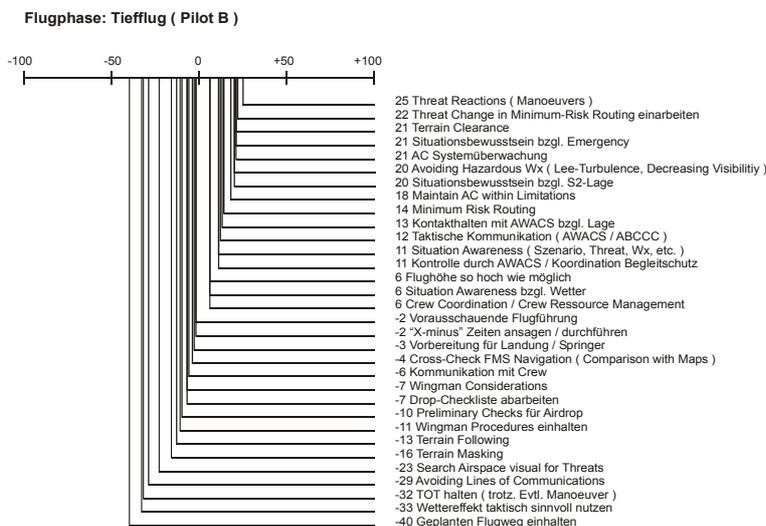
**Abbildung 146: Ergebnis Pilot C zur Flugphase Korridor**



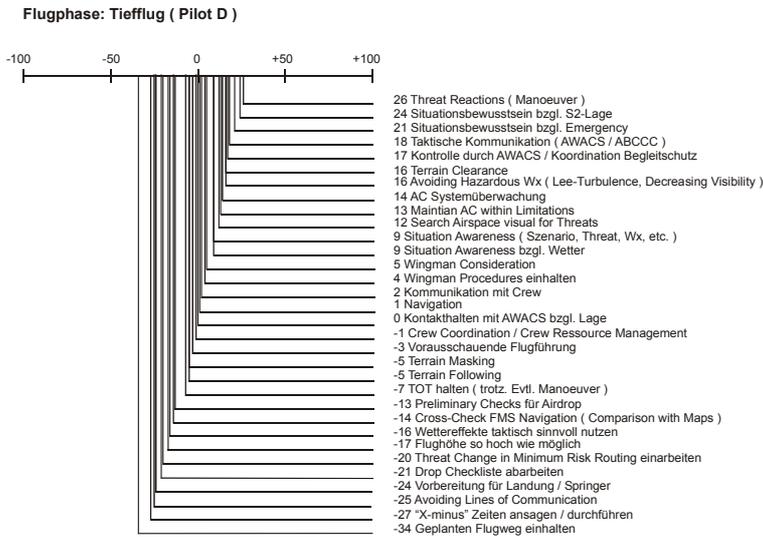
**Abbildung 147: Ergebnis Pilot D zur Flugphase Korridor**



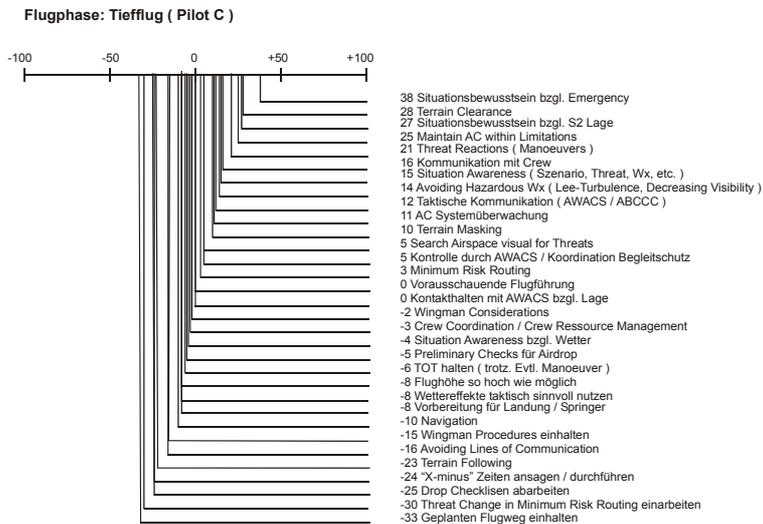
**Abbildung 148: Ergebnis Pilot A zur Flugphase Tiefflug**



**Abbildung 149: Ergebnis Pilot B zur Flugphase Tiefflug**

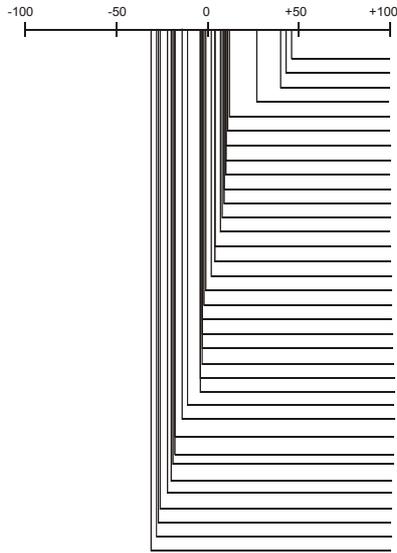


**Abbildung 150: Ergebnis Pilot D zur Flugphase Tiefflug**



**Abbildung 151: Ergebnis Pilot C zur Flugphase Tiefflug**

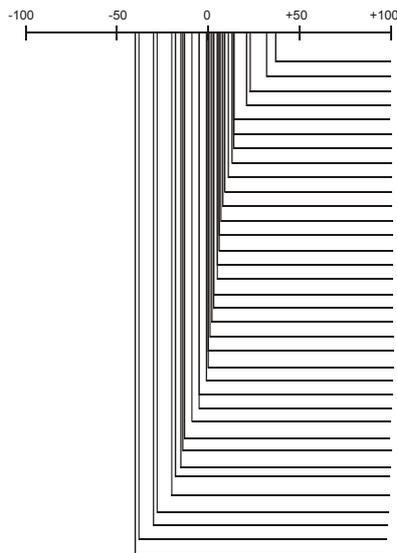
**Flugphase: Lastenabwurf ( Pilot A )**



- 46 Reaktionen auf Threats
- 43 Drop-Zone Hindernisfreiheit
- 40 Terrain Clearance
- 27 Identifizierung Drop-Zone
- 12 Abflug ( schneller Konfigurationswechsel, Speed-Up )
- 11 Navigation ( Routing ), Threat after Drop
- 10 Run-In TAP / IP bis PI as short as possible
- 10 Briefing PF: Cross-Track Distance, Commandet Altitude
- 10 Tactical Communication ( Listening Watch Threat )
- 9 Escape ( Track, Speed, Formation Assembly )
- 9 Communication mit Drop Zone ( Clearance, Guidance )
- 8 Speed Reduction as late as possible
- 7 Terrain Masking
- 4 AC Steuerung ( Checklisten, Procedures, Limitations )
- 4 Escape Procedure ( Recover / Klappen / Rampe )
- 2 Einhalten der Limits ( Klappen, Fahrwerk, Rampe, etc. )
- 1 Bestmögliches Absetzen der Last
- 2 Vorbereiten der Last
- 3 Einhalten der Procedures / Konfigurationsänderungen
- 3 Präzises Absetzen der Last
- 3 Drop-Checklisten Abarbeiten
- 3 Notverfahren briefen
- 4 Kompletter Loadmaster Check
- 4 Evtl. Routenänderungen mit AWACS klären
- 11 Update Drop Data (FMS: Wind, QNH, Temperature)
- 14 Absetzgeschwindigkeit einnehmen / halten
- 18 Einhalten Anflugkurs ohne Steuerbewegungen
- 18 Vorgeplante Route zum Korridor einhalten
- 19 Sinken / Steigen auf Absetzhöhe
- 20 Abflug ohne Ablenkung ( Radio, Gespräche, etc. )
- 22 PNF: Cross-Check Carp ( FMS ) with Map
- 26 Situationsbewußtsein bzgl Escort
- 27 Reduction Procedure "x-3"
- 28 Zeitmanagement TOT
- 31 Search Airspace visual for Threats

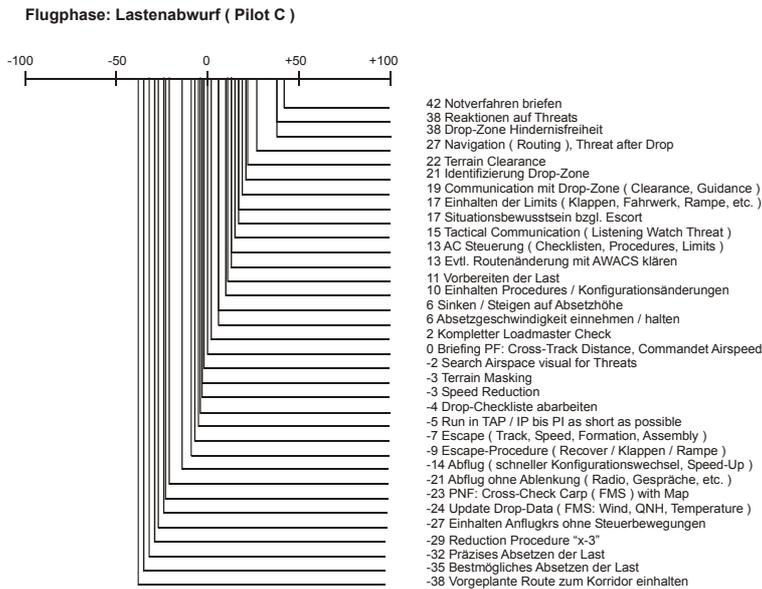
**Abbildung 152: Ergebnis Pilot A zur Flugphase Lastenabwurf**

**Flugphase: Lastenabwurf ( Pilot B )**

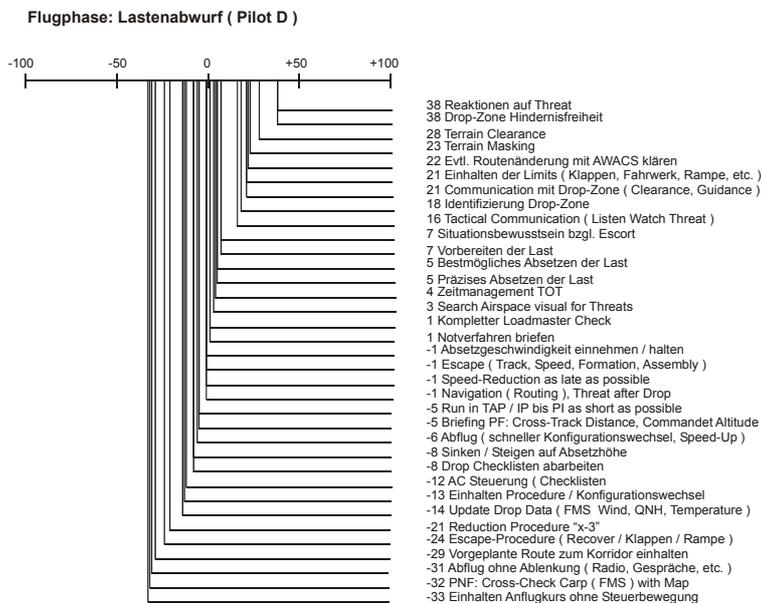


- 37 Reaktionen auf Threats
- 32 Drop-Zone Hindernisfreiheit
- 23 Notverfahren briefen
- 21 Terrain Clearance
- 14 Identifizierung Drop-Zone
- 14 Kompletter Loadmaster Check
- 14 Drop-Checkliste abarbeiten
- 13 AC Steuerung
- 11 Einhalten der Limits ( Klappen, Fahrwerk, Rampe, etc. )
- 9 Vorbereiten der Last
- 8 Navigation ( Routing ), Threat after Drop
- 7 Speed Reduction as late as possible
- 6 Terrain Masking
- 6 Tactical Communication ( Listening Watch Threat )
- 5 Escape-Procedure ( Recover / Klappen / Rampe )
- 5 Präzises Absetzen der Last
- 3 Abflug ( schneller Konfigurationswechsel, Speed-Up )
- 3 Briefing PF: Cross-Track Distance, Commandet Altitude
- 2 Run in TAP / IP bis PI as short as possible
- 1 Evtl. Routenänderung mit AWACS klären
- 0 Absetzgeschwindigkeit einnehmen / halten
- 0 Escape ( Track, Speed, Formation, Assembly )
- 1 Einhalten Procedure / Konfigurationsänderungen
- 5 Bestmögliches Absetzen der Last
- 5 Communication mit Drop-Zone ( Clearance, Guidance )
- 9 Situationsbewußtsein bzgl. Escort
- 13 Search Airspace visual for Threats
- 14 Reduction Procedure "x-3"
- 15 PNF: Cross-Check Carp (FMS) with Map
- 18 Sinken / Steigen auf Absetzhöhe
- 20 Update Drop Data ( FMS: Wind, QNH, Temperature )
- 28 Vorgeplante Route zum Korridor einhalten
- 30 Zeitmanagement TOT
- 38 Abflug ohne Ablenkung ( Radio, Gespräche, etc. )
- 40 Einhalten Anflugkurs ohne Steuerbewegung

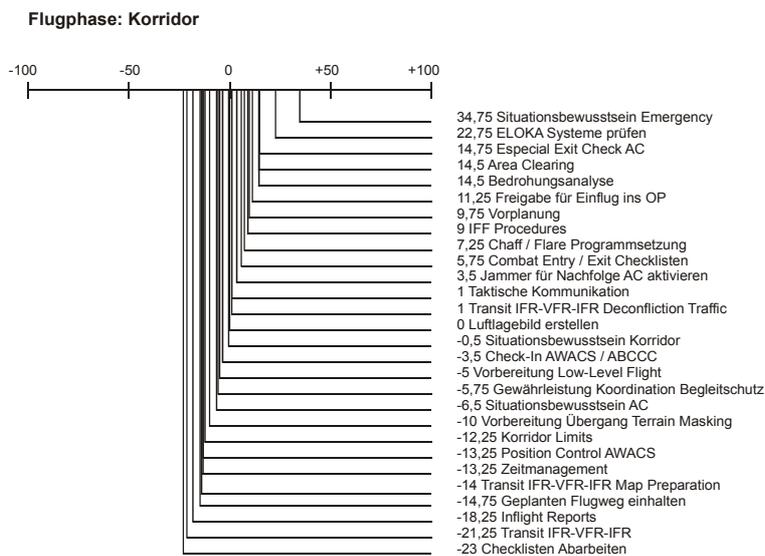
**Abbildung 153: Ergebnis Pilot B zur Flugphase Lastenabwurf**



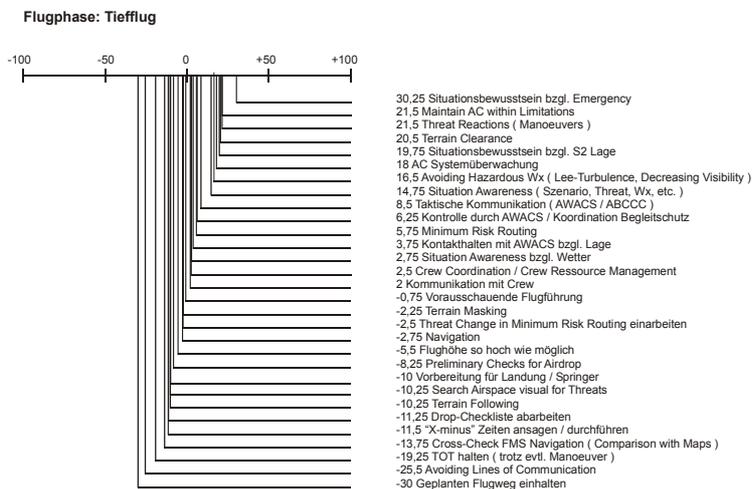
**Abbildung 154: Ergebnis Pilot C zur Flugphase Lastenabwurf**



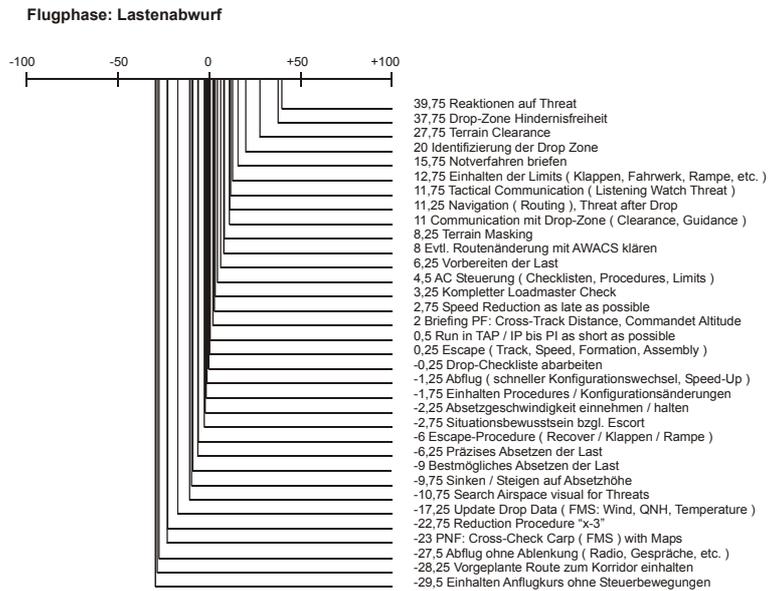
**Abbildung 155: Ergebnis Pilot D zur Flugphase Lastenabwurf**



**Abbildung 156: Gesamtergebnis für die Flugphase Korridor**



**Abbildung 157: Gesamtergebnis für die Flugphase Tiefflug**



**Abbildung 158: Gesamtergebnis für die Flugphase Lastenabwurf**

## 10.3 SPRACHE ZUR WISSENSREPRÄSENTATION

```

Knowledge-Model ::= Relation-Knowledge
                  Situation-Object-Knowledge
                  Goal-Knowledge
                  Action-Knowledge

Relation-Knowledge ::= [RELATION Name-of-Relation]+
Name-of-Relation ::= Name

Situation-Object-Knowledge ::= Situation-Element+
                              | Situation-Object+
Situation-Element ::= ATTRIBUTE Situation-Type
                    Situation-Attribute
Situation-Object ::= SIT_OBJECT Situation-Type
                    Situation-Parameters
Situation-Type ::= Name
Situation-Attribute ::= Attribute-Types ;
Situation-Parameters Knowledge ::= ELEMENTS { Situation-Parameter-
Knowledge }
Situation-Parameter-Knowledge ::= Situation-Parameter+
Situation-Parameter ::= -> Name
                       | -> Name [ Float ]
                       | <-> Name
                       | <-> Name [ Float ]
Attribute-Types ::= Continous-Attribute Attribute-Unit
                  | Crisp-Attribute Attribute-Unit
                  | Constant-Attribute Attribute-Unit
                  | Copy-of-Attribute
Copy-of-Attribute ::= Name
Attribute-Unit ::= Name
Continous-Attribute ::= CONTINUOUS Description-of-
Continous-Value
Description-of-Continous-Value ::= [ Float , Float ; Float ]
Crisp-Attribute ::= CRISP Description-of-Crisp-Value
Description-of-Crisp-Value ::= [ Crisp-Set-Knowledge ]
Crisp-Set-Knowledge ::= Float
                       | Crisp-Set-Knowledge , Float

```

```

Description-of-Constant-Value ::= [ Constant-Value ]
Constant-Attribute Value      ::= CONSTANT Description-of-Constant-Value
Constant-Value                ::= Float

Goal-Knowledge                 ::= [ Description-of-Goal
                                Weight-of-Goal
                                Relevance-of-Goal
                                Body-of-Goal]+
                                | [ Description-of-Goal
                                Weight-of-Goal
                                Relevance-of-Goal
                                Goal-Parameters
                                Body-of-Goal]+

Description-of-Goal           ::= GOAL Name-of-Goal
                                | DESIRE Name-of-Goal

Name-of-Goal                  ::= Name
Type-of-Goal                  ::= Name
Weight-of-Goal                ::= WEIGHT Float
Relevance-of-Goal            ::= RELEVANT_IF Situation-Tuple+
Situation-Tuple              ::= < Name : Name , Name >
                                | < Name : Name >
                                | < Name >
                                | < Name , Name >

Body-of-Goal                  ::= Fuzzy-Operatoror-of-Goal Sub-Goals
                                | Goal-Inference-Knowledge

Fuzzy-Operatoror-of-Goal     ::= FUZZY-OPERATOR Name
Goal-Input-Parameters        ::= FUZZY-INPUTS goalIpt
goalIpt                      ::= Name+
Goal-Parameters              ::= SITUATION-PARAMETERS
                                Goal-Parameter-Knowledge

Goal-Parameter-Knowledge     ::= Goal-Parameter+
Goal-Parameter               ::= Name [. Name]*
Sub-Goals                    ::= SUBS Subgoal
Subgoal                      ::= Name+
Goal-Inference-Knowledge     ::= INFERENCE FUZZY_RULES { Fuzzy }
                                | INFERENCE FUNCT { Algebra-Function
}

```

```

Neuronal-Net }
| INFERENCE NEURONAL-NETNAL {
| INFERENCE MAP { Map }
| INFERENCE INFO { Info ; }
| INFERENCE { TBD }
Action-Knowledge ::= [Description-of-Action Action-Parameters]+
Description-of-Action ::= ACTION Name-of-Action Type-of-Action
Name-of-Action ::= Name
Type-of-Action ::= Name
Action-Parameters ::= PARAMETER Action-Parameter
Action-Parameter ::= Name+
Algebra-Function ::= Name-of-Function Function-Parameter
| GLEICH { Term-of-Function }
Name-of-Function ::= Name
Function-Parameter ::= ( Function-Variable-List )
Function-Variable-List ::= Function-Variable
| [, Function-Variable-List]*
Function-Variable ::= Name
| Name [ Float , Float ]
Term-of-Function ::= TABLE Table
Table ::= List-of-Points ;
List-of-Points ::= Point [, List-of-Points]*
Point ::= [Function-Fuzzy-Inputs Function-Output]
| [ Defaultvalue -> Float ]
Function-Fuzzy-Inputs ::= Float
| { Float , Float }
| Function-Fuzzy-Inputs , Float
| Function-Fuzzy-Inputs , { Float ,
Float }
Function-Output ::= Float
Info ::= Name+
Fuzzy ::= Name-of-Controller
| Fuzzification
| Fuzzy-Operator
| Defuzzification
| Fuzzy-Rule-Definition

```

		Name-of-Controller
		Fuzzification
		Defuzzification
		Fuzzy-Rule-Definition
		Name-of-Controller
		Fuzzification
		Fuzzy-Operator
		Fuzzy-Rule-Definition
		Name-of-Controller
		Fuzzification
		Fuzzy-Rule-Definition
Controller	::=	<b>CTRL_NAME</b> ( <i>Name</i> )
Fuzzification	::=	Fuzzy-IO Fuzzy-Set-Definition
Fuzzy-IO	::=	<b>FUZZY-INPUTS</b> Fuzzy-Inputs <b>OUTPUTS</b> Fuzzy-Outputs
		<b>FUZZY-INPUTS</b> Fuzzy-Inputs <b>OUTPUTS</b> Fuzzy-Outputs <b>EXTERNALS</b> Fuzzy-
Externals		
		<b>FUZZY-INPUTS</b> Fuzzy-Inputs error
Fuzzy-Inputs	::=	( Fuzzy-Input-Names )
Fuzzy-Input-Names	::=	Fuzzy-Input-Name [, Fuzzy-Input- Name]*
Fuzzy-Input-Name	::=	<i>Name</i>
		<i>Name</i> [ <i>Float</i> , <i>Float</i> ]
Fuzzy-Outputs	::=	( Fuzzy-Output-Names )
Fuzzy-Output-Names	::=	<i>Name</i> [, <i>Name</i> ]*
Fuzzy-Externals	::=	( Fuzzy-Externals-Names )
Fuzzy-Externals-Names	::=	Fuzzy-External-Name   Fuzzy-Externals-Names , Fuzzy-External-Name
Fuzzy-External-Name	::=	<i>Name</i>   Fuzzy-External-Name . <i>Name</i>
Fuzzy-Set-Definition	::=	<b>SETS</b> <i>Name</i> { Fuzzy-Set-Definitions }
		Fuzzy-Set-Definition <b>SETS</b> <i>Name</i> { Fuzzy-Set-Definitions }
Fuzzy-Set-Definitions	::=	<i>Name</i> = Fuzzy-Set
Fuzzy-Set		Fuzzy-Set-Definitions <i>Name</i> =

```

Fuzzy-Set ::= TRAPEZ Quadrupel
           | TRIANGLE Triangle
           | GREAT Great
           | LESS Less
           | GAUSS Gaussia
           | BELL Bell
           | DOUBLESIGMOID Double-Sigmoid
           | CRISP_POINT Singlet
           | CRISP_INTERVALL Crisp-Intervall
           | CRISP_GREAT Crisp-Great
           | CRISP_SMALL Crisp-Less

Quadrupel ::= ( Number , Number , Number ,
Number )

Triangle ::= ( Number , Number , Number )

Great ::= ( Number , Number )

Less ::= ( Number , Number )

Gaussia ::= ( Number , Number , Number )

Bell ::= ( Number , Number , Number )

Double-Sigmoid ::= ( Number , Number , Number )

Singlet ::= ( Number )

Crisp-Intervall ::= ( Number , Number )

Crisp-Great ::= ( Number )

Crisp-Less ::= ( Number )

Number ::= Float
         | Name [. Name]*

Fuzzy-Operator ::= [GLOBALDEF : Operator-Definition
]+

Operator-Definition ::= AND = AND-Definition
                       | OR = OR-Definition
                       | GAMMA = GAMMA-Definition
                       | AKKOP = ACCUMULATOR-Definition
                       | IMPLOP = IMPLICATION-Definition
                       | DEFUZZYOP = Defuzzyfication-

Definition
Defuzzification ::= [Name :
                    DEFUZZYOP = Defuzzyfication-
Definition]+

AND-Definition ::= MINIMUM

```

		<b>PRODUKT</b>
OR-Definition	::=	<b>MAXIMUM</b>
		<b>SUMME</b>
GAMMA-Definition	::=	<b>PRODSUM</b>
		<b>MINMAX</b>
ACCUMULATOR-Definition	::=	<b>MAXIMUM</b>
		<b>SUMME</b>
IMPLICATION-Definition	::=	<b>MINIMUM</b>
		<b>PRODUKT</b>
Defuzzyfication-Definition	::=	<b>MOM</b>
		<b>MAXIMUM</b>
Fuzzy-Rule-Definition	::=	<b>REGELN</b> { Rules }
Rules	::=	Rule+
Rule	::=	if proposition <b>THEN</b> instruction <b>ENDIF</b>   if proposition <b>THEN</b> instruction <b>ELSE</b> instruction <b>ENDIF</b>
if	::=	<b>PRINT_RULE IF</b>
		<b>IF</b>
then	::=	<b>THEN</b>
else	::=	<b>ELSE</b>
proposition	::=	Name <b>GLEICH</b> Name   Name <b>UNGLEICH</b> Name   ( proposition )   <b>NOT</b> proposition { add_bed_opcode   proposition <b>OR</b> proposition   proposition <b>AND</b> proposition   proposition <b>GAMMA</b> { Number }
proposition		
instruction	::=	assign   Rules   assign Rules   Rules assign
assign	::=	Name <b>GLEICH</b> Name   Name <b>GLEICH</b> Name [ Number ]   assign Name <b>GLEICH</b> Name   assign Name <b>GLEICH</b> Name [ Number ]

```

Neuronal-Net ::= Neuronal-Net-Input-List
                Neuronal-Net-Class-List
                Neuronal-Net-Rule-List

Neuronal-Net-Input-List ::= Neuronal-Net-Input-Desc
                Neuronal-Net-Input-Nodes

Neuronal-Net-Input-Desc ::= FUZZY-INPUTS Float
Neuronal-Net-Input-Nodes ::= Neuronal-Net-Fzy-Variable+
Neuronal-Net-Fzy-Variable ::= Neuronal-Net-Var-Head
                [ Neuronal-Net-Fzy-Sets ]
                | Neuronal-Net-Fzy-Variable

Neuronal-Net-Var-Head ::= { Name } ( Float , Float ) Number
Neuronal-Net-Fzy-Sets ::= Neuronal-Net-Set-Descr+
Neuronal-Net-Set-Descr ::= Neuronal-Net-Smaller-Set
                | Neuronal-Net-Greater-Set
                | Neuronal-Net-Triangle-Set
                | Neuronal-Net-Trapez-Set

Neuronal-Net-Smaller-Set ::= SMALLER Name [ Float Float ]
Neuronal-Net-Greater-Set ::= GREATER Name [ Float Float ]
Neuronal-Net-Triangle-Set ::= TRIANGLE Name [ Float Float Float
]
Neuronal-Net-Trapez-Set ::= TRAPEZ Name [ Float Float Float
Float ]
Neuronal-Net-Class-List ::= Neuronal-Net-Class-Desc
                Neuronal-Net-Class-Node

Neuronal-Net-Class-Desc ::= CLASSES Number
Neuronal-Net-Class-Node ::= Name+
Neuronal-Net-Rule-List ::= Neuronal-Net-Rule-Desc
                Neuronal-Net-Rule-Agg-Func
                Neuronal-Net-Rule-Node

Neuronal-Net-Rule-Desc ::= RULES Number
Neuronal-Net-Rule-Agg-Func ::= AGG_ALGEBRA-FUNCTION Number
Neuronal-Net-Rule-Node ::= [Neuronal-Net-Rule-Param
                Neuronal-Net-Rule-Class
                Neuronal-Net-Rule-Weight]+

Neuronal-Net-Rule-Param ::= [ Neuronal-Net-Para-List ]
Neuronal-Net-Para-List ::= Number

```

```

| Neuronal-Net-Para-List Number
Neuronal-Net-Rule-Class ::= { Number }
Neuronal-Net-Rule-Weight ::= ( Float )

Map ::= Map-Type Map-Rules
Map-Type ::= TYPE ( MapDescr ) ;;
Map-Descr ::= Name DPUNKT Map-Attributes IS Map-
Attributes
Map-Attributes ::= Map-Attribute [, Map-Attribute]*
MapAttribute ::= Name
| Name [ Float , Float ]
| Name [ DELTA ]

Map-Rules ::= RULES ( MapRule ) ;
Map-Rule ::= Float PLUS Name GLEICH Float
| Name PLUS Float GLEICH Float
| Float MINUS Name GLEICH Float
| Name MINUS Float GLEICH Float
| Float DIV Name GLEICH Float
| Name DIV Float GLEICH Float
| Float PROD Name GLEICH Float
| Name GLEICH Float

```

## 10.4 BEISPIELE ZUR MIKROSKOPISCHEN STRUKTUR VON ZIELMODELLEN

### ZIEL „EINHALTEN DER FESTIGKEITSGRENZE“

Ein Grossteil der sicherheitsrelevanten Ziele lassen sich durch das Studium der Flugleistungsgrenzen ermitteln. Diese werden im Höhen-Machzahl-Diagramm dargestellt (siehe Abbildung 159) und umfassen Auftriebs-, Leistungs-, Temperatur- und Festigkeitsgrenze. An dieser Stelle soll die Festigkeitsgrenze diskutiert werden. Die diese Grenzen bestimmenden physikalischen Gleichungen lautet:

$$M_{\max,q} = \sqrt{q_{\max} \frac{2}{\kappa \cdot p(H)}}$$

$M_{\max,q}$ :	Festigkeitsgrenze
$q_{\max}$ :	maximaler Staudruck
$\kappa$ :	Adiabatexponent
$p$ :	Luftruck

Sie begrenzt denjenigen Bereich, der die sicheren Grenzwerte für die Belastung der Flugzeugstruktur aufgrund von mechanischer Beanspruchung durch Kräfte und Momente nicht übersteigt. Da die Belastung der Struktur proportional zum Staudruck ist, kann die Festigkeitsgrenze durch eine Kurve konstanten maximalen Staudrucks

$$q = \frac{\rho}{2} V^2$$

$q$ :	Staudruck
$\rho$ :	<i>Luftdichte</i>
$V$ :	Umströmgeschwindigkeit

dargestellt werden:

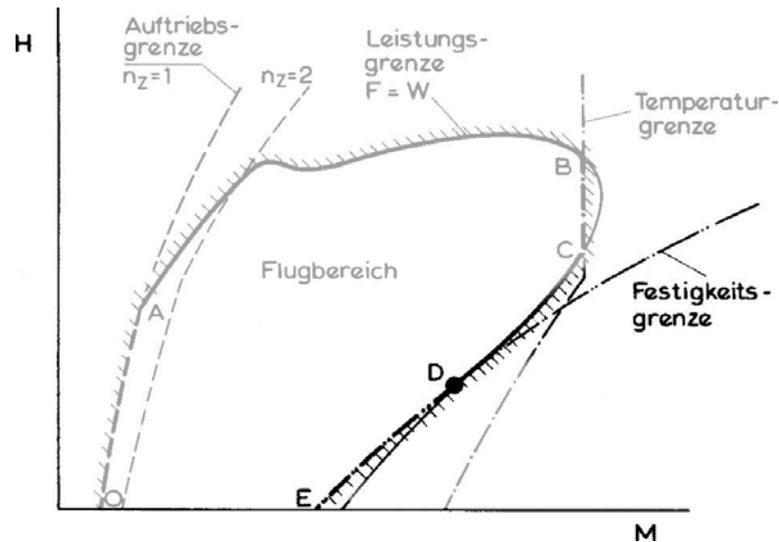


Abbildung 159: Flugleistungskurve (aus Brüning, 1990)

### Wissensrepräsentation

Die Umsetzung des oben dargestellten Ziels erfolgt für das hier entwickelte *explizite mentale Modell* mittels der Konzepte aus der Wissensrepräsentationssprache. Das in der Wissensbasis abgelegte Modell der Festigkeitsgrenze ist nachfolgend abgedruckt. Die Beschreibung des Ziels besteht aus dem Bezeichner **GOAL** und dem Namen **OverSpeed**. Sicherheitsrelevante Ziele wie das diskutierte erhalten die höchste Priorität (**WEIGHT**) 1.0. Die Bestimmung der Gesamtheit der Gewichte wird später erläutert. Das Relevanzkriterium für die Festigkeitsgrenze ist **<airborne: Aircraft>**, d.h. sobald sich das Flugzeug in der Luft befindet erfolgt das Verfolgen des Ziels.

Der vorangestellten Ziel-Beschreibung folgt das Inferenzwissen, hier in Form einer Tabelle zur Interpolation der Zielerfülltheit. Das in dieser Arbeit umgesetzte Gefährdungsmodell ist für die Sicherheitsgrenzen einer VFW 614 ausgelegt. Die Umsetzung der Flugleistungsgrenzen in ein Zielmodell basiert auf den Parametern und Bewegungsgleichungen des Simulationskern des im Flugsimulator des ISF betriebenen VFW614-Modells. Die Auslegung des Simulationskern ist für Flugmissionen im unteren Luftraum konzipiert. Aus diesem Grunde wurde der Einfluss der Flughöhe auf die Festigkeitsgrenze als vernachlässigbar eingestuft. Die damit höhenunabhängige Festigkeitsgrenze basiert zwei Situationsmerkmalen: **FLAPS** (Klappenstellung) und **IAS** (Angezeigte Fluggeschwindigkeit).

```

GOAL      OverSpeed
WEIGHT    1.0
RELEVANT_IF <airborne: Aircraft>
KBASE FUNCTION {

    Truth( Flaps,Ias ) = {
    TABLE [default -> 0.0],
    [-6,160 -> 1.0],[ -6,170 -> 1.0],[ -6,180 -> 1.0],[ -6,190 ->
1.0],
    [-6,200 -> 1.0],[ -6,210 -> 1.0],[ -6,220 -> 1.0],[ -6,230 ->
1.0],
    [-6,240 -> 1.0],[ -6,250 -> 1.0],[ -6,260 -> 1.0],[ -6,270 ->
1.0],
    [-6,280 -> 1.0],[ -6,290 -> 1.0],[ -6,300 -> 1.0],[ -6,310 ->
1.0],
    [-6,320 -> 1.0],[ -6,330 -> 1.0],[ -6,340 -> 1.0],[ -6,350 ->
1.0],
    [-6,360 -> 1.0],[ -6,370 -> 1.0],[ -6,380 -> 0.5],[ -6,390 ->
0.0],
    [-6,400 -> 0.0],

    [ 1,160 -> 1.0],[ 1,170 -> 1.0],[ 1,180 -> 1.0],[ 1,190 ->
1.0],
    [ 1,200 -> 1.0],[ 1,210 -> 1.0],[ 1,220 -> 1.0],[ 1,230 ->
0.5],
    [ 1,240 -> 0.0],[ 1,250 -> 0.0],[ 1,260 -> 0.0],[ 1,270 ->
0.0],
    [ 1,280 -> 0.0],[ 1,290 -> 0.0],[ 1,300 -> 0.0],[ 1,310 ->
0.0],
    [ 1,320 -> 0.0],[ 1,330 -> 0.0],[ 1,340 -> 0.0],[ 1,350 ->
0.0],
    [ 1,360 -> 0.0],[ 1,370 -> 0.0],[ 1,380 -> 0.0],[ 1,390 ->
0.0],
    [ 1,400 -> 0.0],

    [ 5,160 -> 1.0],[ 5,170 -> 1.0],[ 5,180 -> 1.0],[ 5,190 ->
1.0],
    [ 5,200 -> 1.0],[ 5,210 -> 0.5],[ 5,220 -> 0.0],[ 5,230 ->
0.0],
    [ 5,240 -> 0.0],[ 5,250 -> 0.0],[ 5,260 -> 0.0],[ 5,270 ->
0.0],
    [ 5,280 -> 0.0],[ 5,290 -> 0.0],[ 5,300 -> 0.0],[ 5,310 ->
0.0],
    [ 5,320 -> 0.0],[ 5,330 -> 0.0],[ 5,340 -> 0.0],[ 5,350 ->
0.0],
    [ 5,360 -> 0.0],[ 5,370 -> 0.0],[ 5,380 -> 0.0],[ 5,390 ->
0.0],
    [ 5,400 -> 0.0],

    [14,160 -> 1.0],[14,170 -> 1.0],[14,180 -> 1.0],[14,190 ->
1.0],

```

```

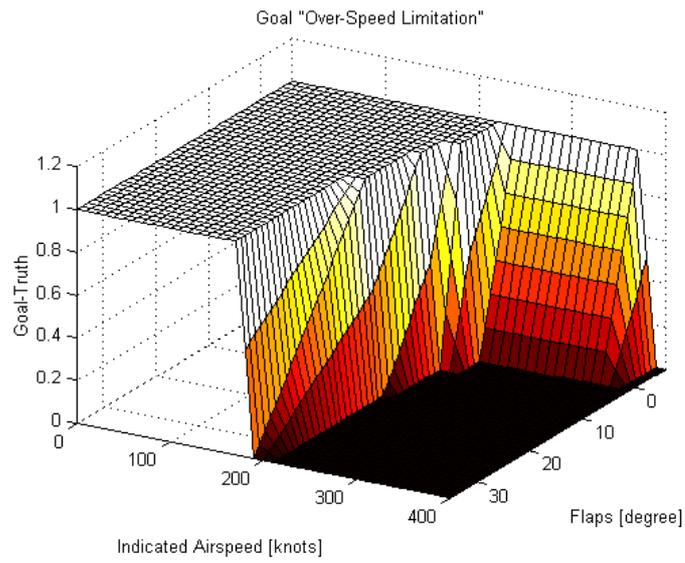
    [14,200 -> 0.5], [14,210 -> 0.0], [14,220 -> 0.0], [14,230 ->
0.0],
    [14,240 -> 0.0], [14,250 -> 0.0], [14,260 -> 0.0], [14,270 ->
0.0],
    [14,280 -> 0.0], [14,290 -> 0.0], [14,300 -> 0.0], [14,310 ->
0.0],
    [14,320 -> 0.0], [14,330 -> 0.0], [14,340 -> 0.0], [14,350 ->
0.0],
    [14,360 -> 0.0], [14,370 -> 0.0], [14,380 -> 0.0], [14,390 ->
0.0],
    [14,400 -> 0.0],

    [35,160 -> 1.0], [35,170 -> 1.0], [35,180 -> 0.5], [35,190 ->
0.0],
    [35,200 -> 0.0], [35,210 -> 0.0], [35,220 -> 0.0], [35,230 ->
0.0],
    [35,240 -> 0.0], [35,250 -> 0.0], [35,260 -> 0.0], [35,270 ->
0.0],
    [35,280 -> 0.0], [35,290 -> 0.0], [35,300 -> 0.0], [35,310 ->
0.0],
    [35,320 -> 0.0], [35,330 -> 0.0], [35,340 -> 0.0], [35,350 ->
0.0],
    [35,360 -> 0.0], [35,370 -> 0.0], [35,380 -> 0.0], [35,390 ->
0.0],
    [35,400 -> 0.0];
}
}

```

## Hyperkarte

Die Interpretation des Inferenzwissens für die maschinelle Verarbeitung in einem Computersystem ist Abbildung 160 dargestellt. Die Hyperkarte stellt das Kennfeld des Ziels für die zwei Merkmale ‚Klappenstellung‘ und ‚Angezeigte Fluggeschwindigkeit‘ dar. An dieser Stelle sei darauf verwiesen, dass die resultierenden Hyperkarten notwendigerweise auch hochgradig-dimensionale Kennfelder repräsentieren können, wenn dies aufgrund der Komplexität des im Inferenzwissen des Modells abgelegten Zusammenhangs von Nöten ist.

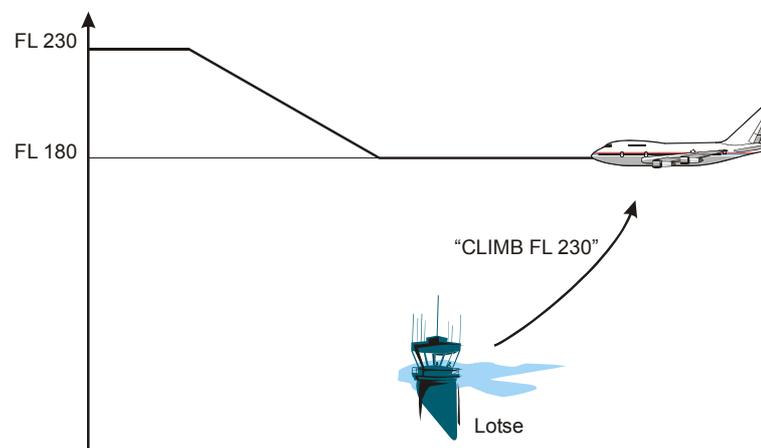


**Abbildung 160: Hyperkarte für das Einhalten der Festigkeitsgrenze**

## ZIEL „EINHALTEN DER HÖHENFREIGABE VON ATC“

Neben den im Aufbau überwiegend statisch parametrisierten Zielhierarchien des Gefährdungsmodells, existieren zahlreiche dynamische Modelle. Insbesondere der Missionsauftrag und die Freigaben der flugführenden Leitstellen (ATC und CCC) sind durch eine dynamische Baumstruktur und Parameter charakterisiert. Ein Vertreter dieser Gattung ist das Ziel, die von ATC vorgegebene Flughöhenfreigabe einzuhalten, welches nun erörtert werden soll.

Die für einen militärischen Transportflug typischen Flugabschnitte des Streckenfluges zum und vom Operationsgebiet erfolgen unter der Leitung übergeordneter Kontrollinstanzen. Diese tragen für eine geordnete Abwicklung des bereits vor Flugantritt bekanntgegebenen Flugplanes Sorge. Dieser Flugplan ist i.d.R. mit den aus den zivilen IFR-Flügen bekannten Flugplänen identisch, beinhaltet also bereits eine geplante Streckenführung, inkl. ‚Top-of-Climb‘ und ‚Top-of-Descent‘ Wegpunkten. Jedoch ist die Umsetzung dieser Planvorgaben aus dem Kontext des Fluges bedingt und muss von der ordnungsgebenden Instanz erst während des Fluges angewiesen werden. Bedingt es die Verkehrssituation im Luftraum, sind oftmals auch unplanmäßige Abweichungen vom Flugplan möglich.



**Abbildung 161: Flugverkehrsweisung für den Steigflug**

Die Flugverkehrsweisung zum verlassen einer aktuellen Flughöhe erfolgt vom Fluglotsen per Funk und Data-Link an das betreffende Luftfahrzeug (siehe Abbildung 161). Die Anweisung der Flughöhe erfolgt dabei in Flugflächen, welche einer um den Faktor 100 reduzierten Flughöhe in Fuß entsprechen. Flugflächen sind auf einen Referenzluftdruck von

1013 hPa bezogen. Dies dient der Eliminierung ortsgebundener Divergenzen des Luftdrucks während des Reisefluges.

### **Wissensrepräsentation**

Das obige Ziel ist zunächst durch eine reduzierte Priorität (WEIGHT = 0.7) charakterisiert, da es qualitativ den Gefährdungszielen in diesem Modell nachgestellt ist. Das Kriterium für die Relevanz stellt sich komplexer dar. Neben dem Zustand des sich-in-der-Luft-befindens <airborne: Aircraft> sind die wahrgenommene Höhenfreigabe <received: Aircraft, Clearance> und <isPartOf: Clearance, Altitude> Voraussetzung für eine Aktivierung des Ziels. Die für die dynamische Parametrisierung des Ziels nötigen Situationsmerkmale (SIT\_PARAMETER) sind anschließend spezifiziert. Diese müssen für die Instanziierung des Ziels der Repräsentation der wahrgenommenen Lage entnommen werden. Hier ist dies die von ATC zugewiesene neue Zielhöhe, welche durch eine Unter- und Obergrenze ergänzt wird. Diese Größen dienen dem Inferenzmechanismus neben dem für die Bewertung der Zielsituation herangezogenen Situationsmerkmal Altitude als Eingangsgrößen. Im Unterschied zu der interpolierbaren Tabelle wurde hier eine Wissensrepräsentation in Form einer Fuzzy-Logik verwandt. Die linguistischen Variablen werden durch die Terme in der SET Definition beschrieben. Der sich hieran anschließende Teil umfasst die Definition der Operatoren für Fuzzifikation und De-Fuzzifikation und die Regel zur Bewertung der Eingangsgrößen. Die Erfülltheit der eingehenden Merkmale ist dann durch den TRUTH-Wert gegeben.

```
GOAL      AltitudeClearance
WEIGHT    0.7
RELEVANT_IF    <airborne: Aircraft>
              <received: Aircraft , Clearance >
              <isPartOf: Clearance , Altitude >
SIT_PARAMETER Clearance.ClAltitudeMin
              Clearance.ClAltitude
              Clearance.ClAltitudeMax

KBASE FUZZY {
INPUTS(  Altitude)
OUTPUTS(  Truth )
EXTERNALS(      Clearance.ClAltitudeMin, Clearance.ClAltitude,
Clearance.ClAltitudeMax )
```

```

SETS Altitude
{
    klein      = KLEINER (    Clearance.ClAltitudeMin,
    Clearance.ClAltitude)
    mittel= DREIECK (    Clearance.ClAltitudeMin,
    Clearance.ClAltitude,
    Clearance.ClAltitudeMax)
    gross      = GROESSER(    Clearance.ClAltitude,
    Clearance.ClAltitudeMax)
}

SETS Truth {
    no      = DREIECK(-1.0,0.0, 1.0)
    yes     = DREIECK(0.0,1.0,2.0)
}

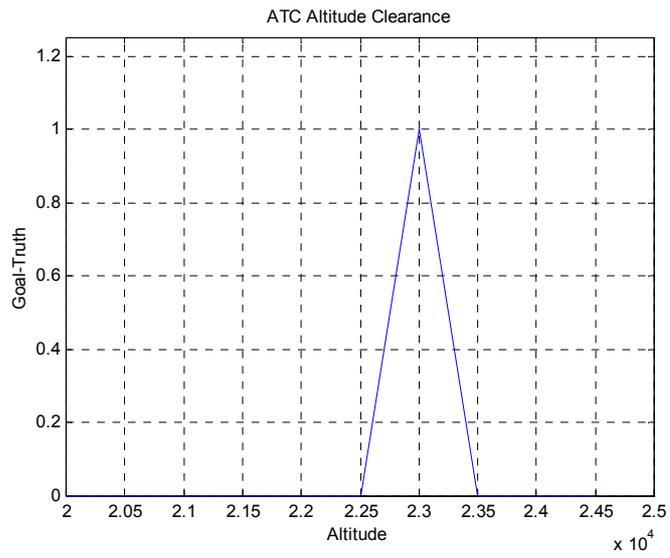
GLOBALDEF:      AND = MINIMUM
GLOBALDEF:      OR  = MAXIMUM
GLOBALDEF:      IMPLIKATION = PROD
GLOBALDEF:      AKKUMULATION = MAXIMUM
GLOBALDEF:      DEFUZZIFIZIEREN = MOM
GLOBALDEF:      GAMMA = MINMAX

REGELN {
    IF    Altitude= gross OR Altitude = klein
    THEN      Truth = no
    ELSE Truth = yes
    ENDIF
}
}

```

## Hyperkarte

Die Hyperkarte der Flugverkehrsfreigabe ist in Abbildung 162 für eine Freigabenhöhe in FL230 dargestellt. Durch die Toleranzen von  $\pm 500\text{ft}$  ergibt sich eine Charakteristik mit Flanken, d.h. der Erfülltheitsgrad steigt 500ft unterhalb der Zielhöhe von 0.0 auf 1.0 kontinuierlich an.



**Abbildung 162: Hyperkarte der Flugverkehrs freigabe**



