Universität der Bundeswehr München
Fakultät für Luft- und Raumfahrttechnik
Institut für Flugsysteme

Cooperation of Human and Artificial Intelligence on the Planning and Execution of Manned-Unmanned Teaming Missions in the Military Helicopter Domain: Concept, Requirements, Design, Validation

Ruben Strenzke, M. Comp. Sc.

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

Doktor-Ingenieurs (Dr.-Ing.)

eingereichten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. (habil) Markus Klein
1. Berichterstatter:	Univ.-Prof. Dr.-Ing. Axel Schulte
2. Berichterstatter:	Professor Jon Platts, PhD

Diese Dissertation wurde am 24. August 2018 bei der Universität der Bundeswehr München, 85577 Neubiberg, eingereicht und durch die Fakuötät für Luft-und Raumfahrttechnik am 26. Februar 2019 angenommen.

Tag der Prüfung: 14. Mai 2019

Singapur, 12. Oktober 2019

ABSTRACT

This dissertation provides *arguments in favor of the cooperation of human operators and an Artificial Intelligence (A.I.) based software* on the *in-flight planning and the execution of joint manned and unmanned military helicopter missions*. Furthermore, it presents *requirements concerning the design and performance of such an A.I. based cooperative software*.

In order to build the argumentation, a brief insight on the different cognitive advantages of the human and A.I. in general is given. After deriving a *concept of human-automation cooperation best described as "mixed initiative"* from this and describing the *design and implementation of a cooperative A.I. system prototype*, an *empirical analysis with subject matter experts* (SMEs) as test persons provides further arguments in favor of this approach.

The cooperation concept presented here is based on the assumption that it is hardly possible to know in advance, which subtask (or subproblem) is suited to be allocated to either of the two partners, human or automation. This is because both have different abilities, and each task or problem can have a different structure demanding specific problem-solving abilities. Hence, each of the partners in the mixed human-machine team should be able to take problem-solving initiative, i.e. the initiative for adapting or evolving the mission plan and start task execution. This constitutes the *mixed-initiative planning, replanning and plan execution approach,* which is the basis of the *Mixed-Initiative Mission Planner (MMP)* that has been designed and implemented by the author in a way that makes use of *symbolical A.I. technology (classical, domain-independent Automated Planning)*.

The empirical analysis has been performed as *full-mission simulation experiments with professional German Army Aviators* as SME test persons in *two main configurations: with support by the A.I. mission planning and execution component and without this support*. As this dissertation shows, the analysis has been validated by the SMEs to be representative. However, due to the small pool of eight test persons that makes statistical validation difficult, *the main evidence is based on their systematically collected subjective feedback* showing that *the advices and solutions given by the MMP were considered rather useful than hindering*.

Finally, *this dissertation is able to provide a basis for the requirements and the design of cooperative A.I. systems in this field of application*. This is achieved also by the systematic collection of subjective feedback concerning user requirements from the test persons after the experimental campaign. The feedback also includes information on alternative mission-planning work-flows and human-machine interface designs.

ACKNOWLEDGMENTS

I feel very thankful to Professor Dr. Axel Schulte for having received the opportunity to work on the on the scientific project MUM-T in the years from 2007 to 2011 and the possibility to develop this doctoral thesis from 2011 on, both under his supervision. He has always supported me in developing and applying my ideas and concerning my publications and speeches.

During such a long time one meets many people, works closely together with some of them and becomes inspired by outstanding individuals. First of all, I want to thank the whole team at the Institute of Flight Systems, which includes Professor Dr. Peter Stütz and his employees, for the good working atmosphere at the Institute of Flight Systems and their support during the whole time. It was an important period for me, professionally as well as personally, and it will always be.

Conducting complex experiments is only possible with a team effort. I especially thank Andreas Rauschert, Johann Uhrmann, Andreas Benzler and Felix Maiwald. Furthermore, I am deeply thankful to Stefan Brüggenwirth for the fruitful continuous discussion and the scientific cooperation. Although Florian Böhm was spending most of his time performing magic in the university's labs, it was a great pleasure to share the same office, and I thank him for that.

Finally, I thank all students that made valuable contributions to the Manned-Unmanned Teaming project, especially Nick Grünewald, Thomas Herbst, Michael Vohla and Hannes Klostermann. Each of them did a great job and played a part in supporting my research, which I herewith present.

Table of Contents

Table of Abbreviations

#	Number of subjects (valid answers / total answers)
A/C	Aircraft
AC-130	Attack Cargo aircraft 130 (by US manufacturer Lockheed)
AH-64D	Attack Helicopter 64 D (by US manufacturer Boeing)
aHuP	assumed Human Plan
ALADIN	Abbildende Luftgestützte Aufklärungsdrohne im Nächstbereich (UAV by
AP	Approach Point
AsP	Assistant system Plan
ATR	Aided Target Recognition
BMVg	Bundesministerium der Verteidigung (German Defence Department)
BO-105	Bölkow 105 (light utility helicopter by German manufacturer MBB)
C2	Command and Control
C3	Communications, Command and Control
CH-53	Cargo helicopter 53 (transport helicopter by US manufacturer Sikorsky)
cf.	confer
Config	Configuration
DARPA	Defense Advanced Research Projects Agency (USA)
DKC	Domain Knowledge Configuration
DLR	Deutsches Zentrum für Luft- und Raumfahrttechnik (German Aerospace
DP	Departure Point
EMT	Elektro-Mechanische Technologien (German UAV manufacturer)
EO	Electro-Optical
ESG	Elektroniksystem- und Logistik-GmbH (German aerospace services company)
FLOT	Forward Line of Own Troops
FOB	Forward Operating Base
GCS	Ground Control Station
GUI	Graphical User Interface
HALE	High Altitude, Long Endurance (UAV)
HMI	Human-Machine Interface
HOA	Helicopter Operation Area
HTN	Hierarchical Task Network
HuP	Human Plan

IAI	Intelligent Adaptive Interfaces
ISR	Intelligence, Surveillance and Reconnaissance
J-UCAS	Joint Unmanned Combat Air Systems
KZO	Kleinfluggerät Zielortung (small reconnaissance UAV by German Rheinmetall)
LOA	Level of Automation
Loc	Location
LoI	Level of Interoperability
LPG	Local search on Planning Graphs
LPG-td	LPG – timed initial literals and derived predicates
LRE	Launch and Recovery Element
LUNA	Luftgestützte Unbemannte Nahaufklärungs-Ausstattung (UAV by EMT)
M	Mean value
MALE	Medium Altitude, Long Endurance (UAV)
MCDU	Multifunctional Control and Display Unit
MCE	Mission Control Element
MCH	Modified Cooper-Harper (human workload rating scale)
Mgmt.	Management
MHDD	Multifunctional Head-Down Display
Mi-24	Mil 24 (multirole helicopter by Russian manufacturer Mil)
MiRA	Military Rotorcraft Associate (and also the name of the manned helicopter)
MiRA-T	MiRA – Teaming
MMP	Mixed-initiative Mission Planner
MOB	Main Operating Base
MUM-T	Manned-Unmanned Teaming
NASA	National Aeronautics and Space Administration (USA)
NATO	North Atlantic Treaty Organization
No.	Number
PC	Personal Computer
PDA	Personal Digital Assistant
PDDL	Planning Domain Definition Language
PF	Pilot Flying
PFD	Primary Flight Display
PIC	Pilot in Command

POI	Point of Interest
PPM	Planning Process Manager
RAH-66	Reconnaissance and Attack Helicopter 66 (prototype by Boeing and Sikorsky)
ReP	Reference Plan
Req.	Requirement
RPA	Rotorcraft Pilot’s Associate
RPG	Rocket-Propelled Grenade
RWR	Radar Warning Receiver
s	seconds (as a measure of time)
SAGAT	Situation Awareness Global Assessment Technique
SAM	Surface-to-Air-Missile
SD	Standard Deviation
SME	Subject Matter Expert
STANAG	Standardization Agreement
STCI	Simple Temporal Constraint Interface
SyP	System Plan
TN	Tactical Navigator
TLX	Task Load Index (human workload rating scale invented by NASA)
TUAV	Tactical UAV
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UBM	Universität der Bundeswehr Munich
UCAR	Unmanned Combat Armed Rotorcraft
UCAV	Unmanned Combat Aerial Vehicle
UH-1D	Utility Helicopter 1 Deutschland (Utility helicopter by US manufacturer Bell)
UI	User Interface
UO	UAV Sensor Operator
UP	UAV Pilot
US / USA	the United States of America
unrecc’d	unreconnoitered
VSM	Vehicle-Specific Module
WASM	Wide Area Search Munitions
WP	waypoint

X-45	experimental aircraft 45 (UCAV prototype by US manufacturer Boeing)
X-47	experimental aircraft 47 (UCAV prototype by US manufacturer Northrop G.)

Table of Tables

Table of Figures

1 Introduction

Unmanned Aerial Vehicles (UAVs) are becoming more and more popular for military as well as civil applications. This is due to the fact that, compared to manned vehicles, unmanned vehicles can be operated with reduced cost, reduced risk to human health or life, and they also often have longer mission endurance [Naval Studies Board 2005]. On board of the aerial system there is always some kind of automation, for example to stabilize the aircraft in flight. Furthermore, the ground-based part of system, which is interfacing with the human operator(s), normally also needs automated functions, such as waypoint (WP) optimization (preventing crashes with the ground) or the like.

It is important to point out that no complete unmanning takes place. Instead, no matter as how highly automatic or “autonomous” the UAV vendor praises the product - there is always a human stakeholder or a group of human stakeholders, wanting to make use of the mentioned advantages and the system’s capabilities as such. High degrees of automation of such UAV products (or UAS, Unmanned Aerial Systems) are in most cases competitive advantages, because they enable the UAV to perform more complex tasks, to perform faster, or to be controlled by less human personnel. With a really high degree of automation, one human operator might also become responsible for the control of multiple UAVs [Franke et al. 2005], which would further reduce the amount of cost-intensive human operators and, in addition, could save communication overhead between operators. This shows that in such systems, there always are a human part and an automation part. From this starting position, the question concerning how to divide labor between human and automation in a UAV mission arises. There are different points of view on that. A traditional approach in the development of systems like UAS is to implement state-of-the-art automation functions and possibly some novelties to cope with known problems and stay ahead of the competitors. [Hollnagel & Woods 2005] call this the *left-over principle* of automation. However, such an approach, which leads to automating more and more functions, can push the human operator into the role of a more or less passive monitor when finally using the highly automated system. Such configurations may not only be frustrating for the human operator, but can also lead to fatal accidents. The most frightening example is the false alarm of a Soviet satellite system, which was designed to warn of Western nuclear missile attacks [Hoffman 1999]. In 1983, this system had a false alarm as it interpreted a weather phenomenon, a reflection of the sun, to be specific, as such a missile launch. Only due to the correct and wise decision of the human operator, an escalation, which could easily have threatened the existence of our world, could be avoided. This was possible due to the fact that the human operator had some authority left in this highly automated setup and by means of his background knowledge he was able to infer that a single missile launch on the whole globe does not match any first-strike attack pattern. If he had instead confirmed the alarm to be a real missile launch, as the system suggested, this event would have been escalated up the chain of command, where severe decisions would have been to be made. One must imagine that in every step of the command chain, suchlike interpretations and decisions must be made very quickly and may therefore lead to cognitive overload.

In contrast to the traditional approach described before, *human-centered approaches* do not automate as much as possible, but try to optimize the cooperation of human and machine in such a way, that the human is not cognitively overloaded, and the machine follows the human's intentions [Steinhauser, Pavlas & Hancock 2009]. However, in complex and dynamic scenarios of high criticality, like the guidance of multiple military UAVs, situations may arise, in which mental overload is unavoidable, if the human shall be kept completely in the decision-loop. There can simply be too many decisions to be made in too short time, leading to the violation of deadlines, which can put lives and/or material to risk. Furthermore, it is not guaranteed that the human is always making the right decisions, which the machine should follow obediently. Humans make mistakes, especially when they are mentally overloaded or, at the other end of the spectrum, bored [Reason 1990]. The conclusion may be that to achieve the best outcome in such critical situations, the tasks should be allocated to both players in the human-machine mixed team according to their general strengths.[1]

However, this approach of task distribution between human and machine is problematic in case there is a task to solve with sub-tasks that are highly dependent on each other. An example for such a task can be the planning of a military mission, because decisions concerning details in the mission phases influence the constraints for the other phases [Strenzke & Schulte 2011b]. Another hurdle for allocating tasks according to general human or machine strengths is the individuality of a problem. By problem individuality the author means that a specific problem occurring in the real world might be very different from standard scenarios (if there is a definition of standard scenarios at all), although it is still of the same problem category. The problem category specifies *what has to done with an input pattern*. For example, the well-known Travelling Salesman Problem is defined by the task to find the shortest route between a

[1] This is actually an approach that has been around before the thought of human-centered automation [Fitts et. al 1951]. However, it becomes appealing again from time to time.

set of locations that all have to be visited at least once. The individual problems can be regarded as the input patterns, which can differ very much depending on the geographical distribution of the cities, such as an even distribution, few dense clusters of cities or all cities on one line. It becomes clear that the difficulty of solving the problem or the effort of optimizing the result very much depends on the structure and the size of the input pattern.

To explain this further, experiments using an *algorithm that automatically builds clusters of targets for planning UAV attack*s [Malasky et al. 2005] shall be considered. Malasky and others compared the performance of the algorithm with that of a human problem-solver and of a human-machine cooperative solution. Their results imply that for different individual problems of the same category, the relative performance of the algorithm and the human problem-solvers was quite different. [Malasky et al. 2005] also analyze the reasons for the differences in performance, which are to be found in pattern and complexity of the problem. For one problem, the algorithm performed better than the test persons, for another one they performed exactly equally, and for the remaining three the algorithm performed worse. Although the results cannot be statistically significant due to the small number of test persons, it makes sense to look at probable causes for the different outcomes – not from a statistical viewpoint, but with common sense.

A good example to explain the performance deviances by the characteristics of the target distribution, which differ from problem to problem, is described in the following. As Figure 1-1 shows, one of the problems includes a fence of surface-to-air-missile (SAM) launchers with overlapping threat radiuses. Far behind this wall, there were high-value targets, so that it did pay off for the UAVs to break through the wall of SAMs at some geographical point. This could be achieved by building a cluster containing high-value targets and one of the SAMs. It was possible for the test persons to come to this conclusion. However, in contrast, the clustering algorithm was not suited to solve this problem because of the spatial dislocation of the high-value targets and the high threat posed by the numerous SAM clusters that blocked the way to these. The algorithmic threat assessment was not intelligent enough to ‘understand’ that eliminating a single high-treat, low-value SAM-launcher would have paved the way to earning a lot of credit points associated to the high-value targets. The scenario where the algorithm performed better than the test persons was a large scenario with distributed groups of targets, where high-value targets were either unprotected or protected by only one SAM site.

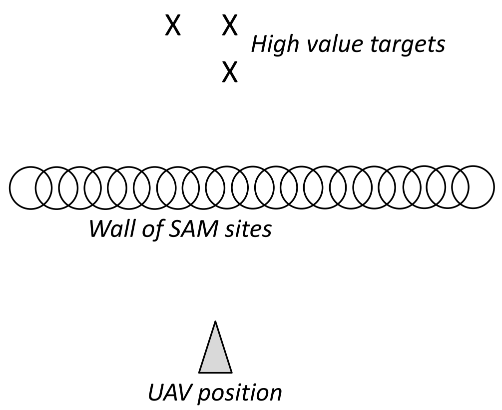

Figure 1-1: The wall problem showing low target-clustering algorithm- performance

Of course, it would be possible to solve the ‘wall problem’ described above by fundamental improvements in the algorithm, by spending more on processing power, by workarounds, or special case treatment and thereby again surpass human performance. However, if one continues to search for problems with poor machine performance after improving the algorithm, one might again find examples inside a task category that make either human or machine performance look bad.

The aspects of the work of [Malasky et al. 2005] described above lead the author to derive from that the following own conclusions that have an impact on how human-machine cooperation is regarded in this work:

1. In the unforeseeable real world (such as in military missions), there is a general problem in the approach to assign task types statically to either the human or the machine. This is because each individual task instance can lead to poor problem-solving performance, when either the human or the algorithm does not get along well with the problem pattern. It is important to point out that when dealing with the real world, the problem patterns are not known in advance.

2. Human-centered automation and adaptive automation are approaches to freely allocate individual tasks either to the human or the machine. However, all current approaches in these fields are also no solution to this, because they focus mainly on human workload optimization and never on the problem size or structure associated with a task instance. Assessing the problem size and structure could enable knowing *in advance* who can successfully work on a task instance, before allocating it to someone. It is important to point out that in time-critical environments trial-and-error is not a good approach for allocating and deallocating tasks.

After considering the conclusions number one and number two, a viable approach for the quick solving of complex real-world problems is the cooperative problem-solving by the human and the machine, in which both can initiate steps towards a solution. This enlarges the chance that either one of the two players in the mixed team is able to solve the problem quickly, or both deliver valuable inputs to a solution, thereby creating synergy. Especially for the latter case, the question of how cooperation can be optimized arises, i.e. how the interaction and the interface shall be designed to maximize cooperation benefits.

One such complex real-world problem is multi-vehicle military mission planning, which is the problem regarded in this dissertation. Here, it is necessary to focus on the question how the human and the machine can cooperate on finding, analyzing and solving a unique planning problem, instead of distributing tasks or problem categories amongst them beforehand. This leads to a third, relatively new approach besides distributing tasks either statically (traditional approach) or dynamically (human-centered or adaptive automation), which is called *mixed initiative*.

This dissertation covers the complete cycle of theoretical derivation of the human-machine cooperation approach, the design and implementation of the cooperative system as well as the scientific evaluation in realistic experiments. The remainder of this chapter first gives a brief description of this dissertation's research question. After that, the structure of the thesis is outlined.

1.1 Brief Description of the Research Questions

The application dealt with in this dissertation is the in-flight multi-UAV mission planning and replanning task in the military aviation domain. To be more specific, the task refers to the planning of a mission including a manned military transport helicopter and multiple reconnaissance UAVs in a so-called "*Manned-Unmanned Teaming*" setup. Due to mission order updates and changes in the tactical situation, this planning task has to be performed multiple times during the mission by the helicopter cockpit crew, specifically the pilot in command (PIC), who shall be supported in his/her mission planning, replanning and execution tasks by an assistant system. Although the research questions have been formulated in a rather generic way, they are examined in the concrete context given by the application described above.

The research questions of this dissertation are as follows:

1. *How can a human-machine cooperative system* for military Manned-Unmanned Teaming mission planning *be designed* in order to be helpful for the user(s)?

2. What are the *requirements of potential users* (Army Aviation pilots) concerning a *human-machine cooperative system* for military Manned-Unmanned Teaming mission planning?

These two questions shall be partially answered by development and evaluation of a mixed-initiative planning system prototype, which the author called *Mixed-Initiative Mission Planner* (*MMP*).

1.2 Thesis Structure

The structure of this dissertation is as follows. In the next chapter, the relevant background for this work is presented. This includes the current state of development concerning human-UAV interaction and recent research in this field as well as the topic of human-machine cooperative planning and the specific project, called Manned-Unmanned Teaming (MUM-T). The detailed description of this dissertation's research question concerning the mixed-initiative planning of a MUM-T mission is then also embedded into chapter 2. After that, in chapters 3 and 4, the MMP's requirements, its conception, design and implementation aspects are detailed in order to give answers to the abovementioned research question number one. Chapter 5 explains the experiments conducted for the evaluation of the MMP and the corresponding results. This includes the gathering of the user requirements concerning a mixed-initiative multi-UAV mission planning system (research question number two). And finally, the complete work is discussed in chapter 6, which also raises questions relevant for future research. With chapter 7, a summary of the complete content of this dissertation is given.

2 Background

This chapter provides the necessary background in order to understand and correctly interpret the main chapters 3, 4 and 5. It first describes the current state of development and recent research on human-UAV interaction. After that, the topic of human-automation interaction in general and especially collaborative mission planning is illuminated, which is the field the research question of this dissertation is primarily related to. Then, the research project MUM-T at the Universität der Bundeswehr Munich (UBM) is described, because this project is the application for the human-machine cooperation approach developed in this dissertation. Finally, the research question examined here is explained in detail.

2.1 Current State of Development Concerning Human-UAV Interaction

UAVs in use today are typically performing preprogrammed missions that can be manually altered in-flight by the UAV operators.[2] For example, waypoints can be activated or manipulated, and the altitude, speed and heading of the UAV can be controlled in an override mode. The UAV operation crew is working in a ground control station (GCS, for an example see Figure 2-1), and it consists of one to four, typically two to three human operators. The operator's standard roles are one of the following.

- UAV guidance and flight planning
- Payload operation and image processing
- Communications, Command and Control (C3)

The different operator roles, which have been mentioned above, are explained in the beginning of the remainder of this section. Then, the control of multiple UAVs by the same personnel, as far as it is possible with today's systems, is described. After that, a typical workflow of a UAV operation is given. Finally, the aspects of MUM-T and airborne UAV control stations, i.e. control stations aboard of manned aircraft are shown.

[2] In this dissertation, a UAV operator is considered to be the person or role that is responsible of operating one or more Unmanned Aerial Vehicles or one or more Unmanned Aerial Vehicle subsystems.

Figure 2-1: MQ-9 Reaper MALE UAV Ground Control Station [Wikipedia 2014]

2.1.1 UAV Operator Roles

The *UAV guidance and flight planning role* includes the task of planning the flight path for the mission, which depends on the goals that shall be met by using the UAV payload. These goals can be the reconnaissance of specified areas, the surveillance of certain objects, or weapons release. The flight path is planned before the mission starts, and it can be altered during mission execution as far as all modern UAVs (e.g. all of the UAV systems listed in Table 2-1) are concerned. Most UAVs are guided on a waypoint level, i.e. an ordered list of three-dimensional or four-dimensional[3] waypoints has to be generated (see “guidance modes” column in Table 2-1). During the operation, the UAV then automatically follows these waypoints to comply with the given order.

As an example, the Rheinmetall “KZO” tactical UAV (see Table 1), is guided on a level which is called *task-based* by the manufacturer. To be more specific, the flight-planning operator enters into the GCS ordered flyover, circle or sensor waypoints (i.e. areas or objects shall be targeted by the UAV’s onboard camera). The exact flight plan is then calculated by the GCS software and uploaded into the UAV. The upload of a new flight plan is also possible later on, when the UAV is already in the air.

[3] including time as fourth dimension

Table 2-1: Prominent examples of UAVs operational today with information on their GCS

UAV	Class	Producing country	No. of simultaneous UAVs per GCS	Guidance / Flight planning personnel	Payload operation / Image processing personnel	C3 personnel	Guidance modes
Northrop Grumman Global Hawk (MCE)	HALE	USA	3	1	1	2	WP-based
Northrop Grumman Global Hawk (LRE)	HALE	USA	3	1	-	1	?
Elbit Hermes 450 / 900	MALE	Israel	2	0,5-1	0,5-1	0-1	WP-based or manual
Rheinmetall KZO	TUAV	Germany	2	1	1	1	WP-based (“task-based”)
EMT LUNA	TUAV	Germany	1	1	1	0-1	WP-based or autopilot
Schiebel Camcopter S-100	TUAV (VTOL)	Austria	1	1	1	-	WP-based or manual
EMT ALADIN	MUAV	Germany	1	0,5	0,5	-	WP-based

Most UAV systems can also be operated in a manual mode, which allows a more direct interaction with the aircraft. Therefore, the UAV operator whose role is described here can also be called a “UAV pilot”.[4] Although all UAVs considered in Table 1 are able to follow a preplanned or replanned flight path without any further human intervention, in most cases the UAV operator crew includes a dedicated UAV pilot, who is busy with monitoring the flight of the aircraft and its system status. Only few UAV systems in the list, e.g. the Elbit Hermes 450 and 900 MALE (Medium Altitude, Long Endurance) UAVs as well as the EMT ALADIN Mini-UAV (MUAV) can be operated by a pilot who is responsible for the payload management at the same time. This may be typical for the MUAV class, but rather untypical or even non-existent for larger drones, i.e. Tactical UAVs (TUAV), MALE UAVs and HALE (High Altitude, Long Endurance) UAVs.

For these systems, there is normally a dedicated *payload operator*, who is concerned with the sensor and weapon systems and who also analyses the image data gathered by the onboard

[4] In multiple publications, the person who is responsible of steering the UAV is also called “UAV pilot” independent of the level of automation, i.e. even if there is no possibility for him/her to control the UAV in manual steering mode. Hence, this term could be considered to refer to a person or role that is responsible for the flight path of one or more Unmanned Aerial Vehicles.

sensors of the UAV. Furthermore, an additional (in some cases optional) *C3 operator* role exists for the mentioned larger UAVs. Such an operator can also examine the payload image data (live or playback), is normally responsible for communication with stakeholders and other relevant military units and can also bear the responsibility as commander of the complete UAV operations crew. In the case of the Global Hawk HALE UAV, which has the largest operator crew of all UAVs listed in Table 1, the C3 role is further distributed among a C2 (Command and Control) operator and a further operator who is responsible for communications. Another specialty here is that, in addition to the mentioned operator crew located in the Mission Control Element (MCE), there is another, smaller crew in the Launch and Recovery Element (LRE), responsible exclusively for the Global Hawk's launch and recovery phases.

2.1.2 Multi-UAV Control

In most cases, a UAV operator crew operates only one UAV, but with some GCS it is possible to operate multiple UAVs at the same time. However, the simultaneity can rather be imagined as the possibility to switch the dedicated control between two or three UAVs (cf. Table 2-1), while the "abandoned" UAVs are loitering or continuing with the planned flight path. There is no "UAV teaming" feature such as multiple UAVs executing a single commanded task together or integrated multi-UAV planning capability in today's GCSs or UAV flight computers. However, a lot of research work is going on in the field of multi-UAV guidance, as described later in chapter 2.2.

2.1.3 UAV Operation Workflow

Figure 2-2 shows the operational workflow of a UAV mission, using the KZO tactical UAV as an example. Boxes with rounded edges represent main states of the process, and sharp edges represent sub-states. Some sub-states can be reached either before or after UAV takeoff. After the mission order has been received by the operator crew, the system (vehicle and GCS) are set up for operation. The technical part of the mission preparation is left out here and instead the operational mission planning and execution tasks are in focus in the following. As already explained in chapter 2.1.1, the mission of the KZO UAV is planned on a waypoint level. The mission plan (flight and payload plan), which has been generated in the GCS, is then transferred to the UAV mission module, which is also referred to as Vehicle-Specific Module (VSM) by NATO Standardization Agreement (STANAG) 4586 [NATO 2010]. The mission module onboard the vehicle checks the plans and reports the corresponding result. If the checks were successful, the UAV is launched and it then follows the agreed plan. The flight of the UAV and the images generated by its payload are monitored by the crew as described in chapter 2.1.1. In case there is a change in the mission goal or a relevant situational change, mission replanning has to take place. This is performed by the UAV pilot / flight planner. The mission replanning loop (bold arrows in Figure 2-2) can be iterated multiple times before the mission is completed and the UAV lands. In case of MALE and HALE UAVs it also common that in lengthy missions, different crews take over the operation in shifts.

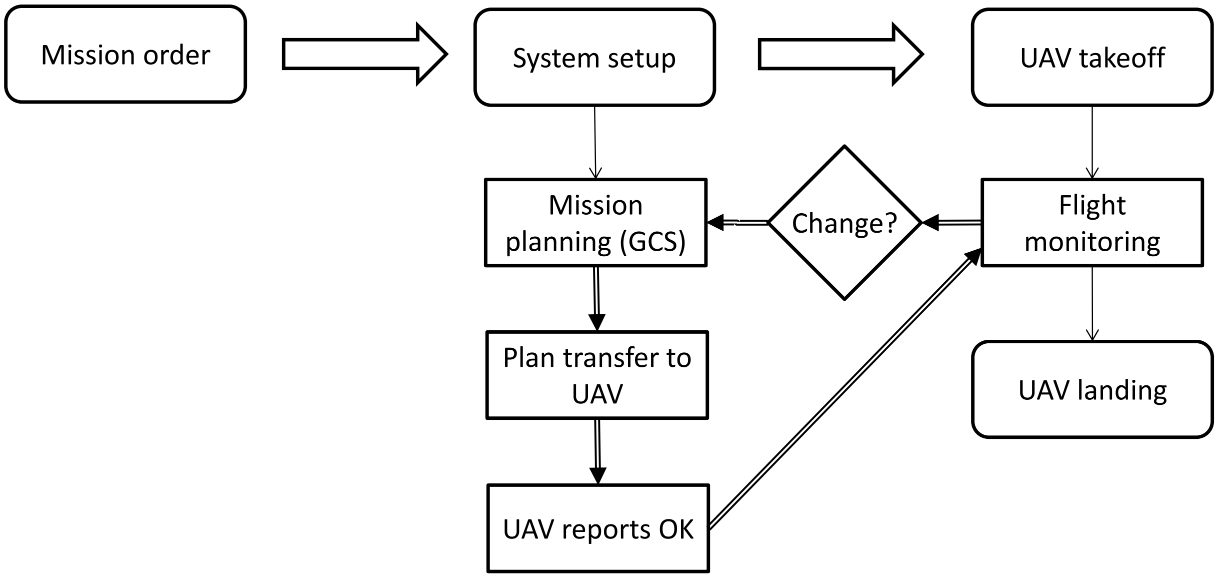

Figure 2-2: Operational workflow of a UAV mission, simplified adaptation of [Wohlers & Blohm 2007]

2.1.4 Manned-Unmanned Teaming and Airborne UAV Control Stations

The term "Manned-Unmanned Teaming" stands for the close operation of manned and unmanned aerial vehicles. It has been used for the cooperation of manned helicopters with MALE UAVs [Shelton 2011], TUAVs [Sperling & Kewley 2008], or rotorcraft UCAVs (Unmanned Combat Aerial Vehicles) [Maddock 2004]. The idea behind this is that a close cooperation will leverage the strengths of both concepts - that of the expensive manned vehicle with the intelligent (human) decision-maker onboard and that of the more or less expendable unmanned sensor and weapon carrier, which can fly out front in a more exposed way and provide an early detection of objects, which are then to be finally identified by the crew of the manned aircraft [Bergantz et al. 2002]. [Bergantz et al. 2002] summarize the results of US Army MUM-T experiments as follows: "As a team, MUM platforms are capable of achieving detection, classification, recognition, and identification at much greater ranges than either system could accomplish alone. […] By capitalizing on the strengths of MUM elements, the manned-unmanned team increases its overall effectiveness and enables mission accomplishment."

At a closer look on this form of the cooperation, the interoperability between the manned and unmanned vehicles becomes the decisive factor. The STANAG 4586 [NATO 2010], defines five Levels of Interoperability (LoI) in the context of control station to vehicle interoperation, which can in principle be applied to airborne control stations:

- LoI 1 means that the UAV's sensor data is transmitted to the manned aircraft indirectly,
- whereas in LoI 2 there is a direct data connection between the two vehicles.
- From LoI 3 onward, the crew of the manned vehicle is able to control the payload of the UAV, i.e. its sensors and weapons.
- From LoI 4 onward, the UAV's flight path can also be controlled by the crew,
- and LoI 5 additionally includes the ability to conduct takeoff and landing with the UAV.

Despite the large amount of ideas of operational MUM-T scenarios in the military community, the only approach that is currently led to an operational state is the teaming of the twin-seated AH-64D Apache Longbow attack helicopter with the MQ-1C Gray Eagle MALE UAV, which is a US Army effort [Shelton 2011]. To accomplish this, the AH-64D Block III version is able to carry a UAV data link called UTA (Unmanned Aerial Systems Tactical Common Data Link Assembly), which allows the Apache's CPG (Co-Pilot/Gunner) to control up to four UAVs. This includes UAV flight path control, sensor and weapons control, as well as sensor data display, which corresponds to LoI 4.

2.2 Recent Research on Human-UAV Interaction

As shown in the preceding subchapter, today a crew consisting of multiple operators is controlling usually a single UAV. Even if multiple UAVs are operated simultaneously, there are normally more human operators than unmanned vehicles. There are, however, research approaches to invert the operator-to-vehicle ratio in the future, e.g. [Franke et al. 2005]. In addition to this, in the context of MUM-T, research is conducted to change the location and the job role of the UAV operator in terms that he/she shall be at the same time a crew member in a helicopter [Durbin & Hicks 2009] [Kraay, Pouliot & Wallace 1998] or a fighter/attack aircraft [Wallis et al. 2002] [Cummings & Morales 2005].

In the following, an overview on relevant research work in the field of human-UAV interaction is given. A differentiation is made between the research topics *Manned-Unmanned Teaming*, *Multi-UAV Guidance*, *UAV Autonomy* and *Human-Machine Cooperation*, although these are deeply interconnected.

2.2.1 Manned-Unmanned Teaming

MUM-T stands for the joint missions of manned and unmanned aircraft, whereby the manned aircraft are in most cases helicopters. In order to optimally combine the use of the two different elements, research has been conducted concerning the question how the UAVs can be guided by the crew of the manned aircraft.

Before the MUM-T UAV control capability, mentioned in chapter 2.1.4, was integrated into the Apache helicopter, simulator experiments with single-UAV guidance by the Apache CPG on a waypoint level were conducted by the US Army Research Lab [Durbin & Hicks 2009]. Their experimental results showed lower workload for both the PF and for the CPG in comparison to a combat mission without UAVs. The reason for this is that the UAV was used for tasks that otherwise would have to be performed by the Apache helicopter itself, i.e. reconnaissance and attacking of targets. The displays and controls used for the completion of these tasks were the same as or at least similar to those that would have to be used for performing these tasks with the Apache itself. Therefore, the crew had only few additional tasks compared to a configuration without any UAV. In addition to that, the time criticality was reduced by using the UAVs flying high and far in front (e.g. 70 km) of the helicopter as opposed to using the helicopters own sensors and weapons in a low-level flight while being near to military threats.

Even before that, from 1997 up to 2002 the US Army conducted research in the control of multiple UAVs from the cockpit of the projected twin-seated RAH-66 Comanche armed light scout helicopter [Bergantz et al. 2002]. The simulator experiments were designed to generate, analyze and optimize tactics for MUM-T operations as well as to answer human-machine interface-related questions. The LoI and number of UAVs under control of the Comanche pilots were varied. In addition, Cognitive Decision Aid systems were introduced. These systems were based on the Rotorcraft Pilot's Associate (RPA) program. In these experiments, the UAVs took over an Intelligence, Surveillance and Reconnaissance (ISR) role, while the Comanche helicopter flew an Armed Reconnaissance. The goal was to seek out and destroy Scud surface-to-surface missile launcher vehicles, which were able to move. First, it was noticed that loading the UAV control task upon the helicopter crew increased their workload so much, that they were not able to accomplish the mission anymore due to task overload because of human-machine interface complexity; especially as far as the manual UAV sensor control task was concerned. The researchers then consolidated the Human-Machine Interfaces (HMIs) for the UAV with that of the helicopter and added ATR (aided target recognition) functionality. However, the crew's workload was still high, until a Cognitive Decision Aiding System was integrated into the helicopter cockpit. This system was based on functionality developed in the RPA program [Miller & Hannen 1999]. In this configuration, no degradation of crew performance by taking over UAV control tasks could be observed, and the overall mission effectiveness was increased. A further result of these MUM-T experiments was that with the help of the Cognitive Decision Aiding System, the manageable number of UAVs for one Comanche crew was determined to be two.

The German MUM-T research undertaking will be described in detail in chapter 2.5.

2.2.2 Multi-UAV Guidance

Although some GCS for UAVs operational these days are able to control multiple UAVs simultaneously, this is still done in a very limited way, as described in section 1.1.2. What is missing in today's implementations is some kind of UAV-UAV cooperation capability, sensor data preprocessing and sensor data fusion functionality, as well as a coherent user interface concept. Therefore, the guidance of multiple UAVs from a single GCS or airborne control station is subject to current research.

The guidance of multiple UAVs has been addressed in the *Playbook Approach* [Miller et al. 2004]. The underlying paradigm of this approach resembles the calling of plays as they are used for example in American football, wherein the operator is the trainer calling the plays and the UAVs are the players, acting according to the calls and their common knowledge about the plays in general. This Playbook™ Approach has been applied to scenarios where multiple heterogeneous UAVs had to solve a problem by taking over different roles (e.g. reconnaissance, target marking/lasing, and weapons usage) or by correct scheduling (e.g. one UAV performs surveillance, another takes over after some time and so forth). The Playbook™ Approach comprises a user front-end for (PC or PDA) as well as the planning tools which make the UAVs fulfill the task constraints given by the operator [Goldman et al. 2005]. These components incorporate the concept of variable autonomy [Miller et al. 2004]. This lets the operator specify either high-level tasks or detailed task instructions for multiple UAVs, which then work together in predefined parameterizable maneuvers actually called "plays". As shown in Figure 2-3, the tasks are ordered hierarchically, which is supported by the use of a Hierarchical Task

Network (HTN) planner as the computational backend of the system. As an example, the airfield denial task includes three phases (subtasks), which are ingress, strike and egress, and the strike phase consists of two parallel subtasks. Resources can be allocated to tasks either manually or automatically. The calling of plays feature might be used on-the-fly, i.e. in midst of a dynamic mission, as well [Miller et al. 2011].

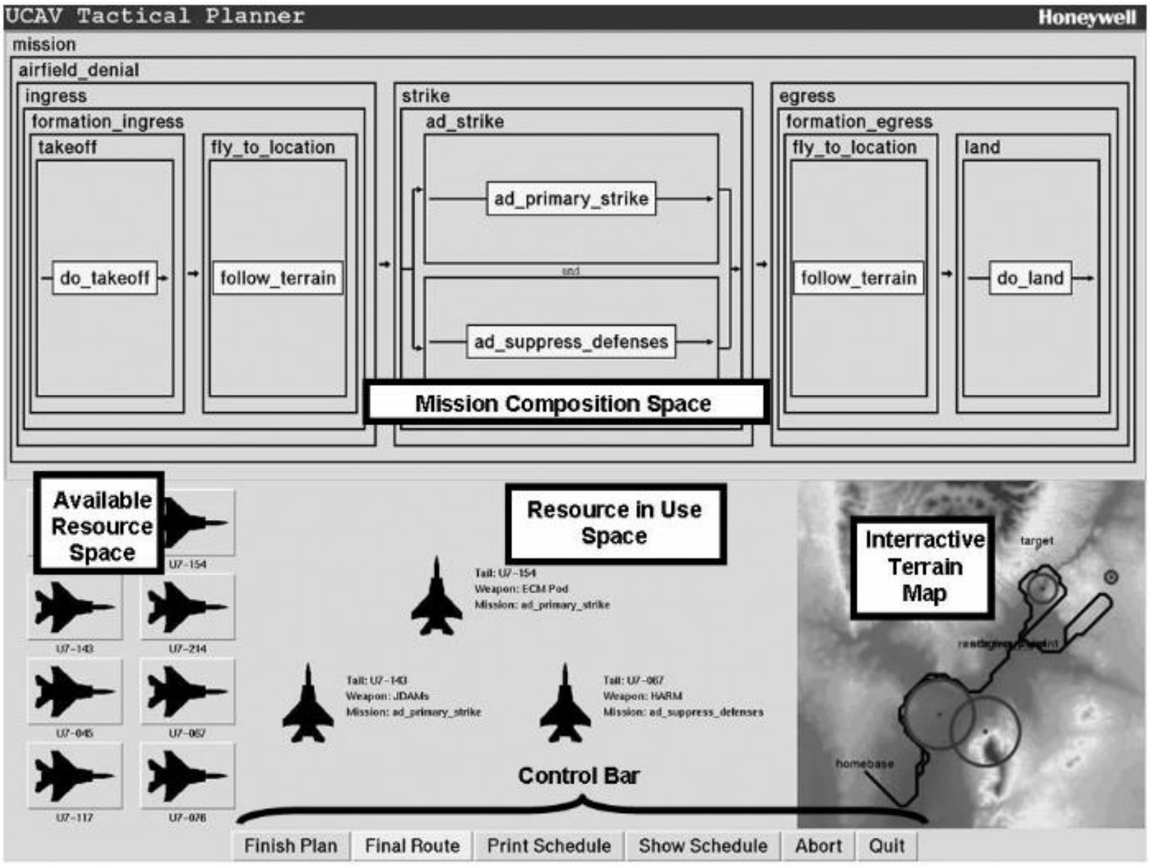

Figure 2-3: Hierarchical task planning user interface for multiple aircraft [Miller & Parasuraman 2007]

Another approach to multi-UAV guidance by a UAV operator crew was the use of so-called *Intelligent Adaptive Interfaces (IAI)* by [Hou & Kobierski 2006]. Intelligence in the operator interfaces should reduce workload (and thereby reduce manning) and enhance performance of military UAV systems. The application for IAI was a counter-terrorism scenario over open water with a UAV operator crew located in the tactical compartment in the back of a CP-140 maritime surveillance aircraft. The roles of the operators were:

- UAV Pilot (UP), responsible for the deployment, management, and control of the UAVs
- UAV Sensor Operator (UO), responsible for sensor management and data interpretation

- Tactical Navigator (TN), has the responsibility of the mission commander

In order to support the crew, the operator interfaces were dynamically configured by automation components called *IAI agents*, which also directly passed messages to the operators via the corresponding primary displays. These agents perform the following actions according to [Hou & Kobierski 2006]:

1. Gathering of all data concerning the status of the UAVs, their sensor tracks, and the current display configuration
2. Analysis of this data according to the rules that have been pre-defined by SMEs and derivation of any (symbolic) events that have occurred
3. Prioritization of these events by means of additional rules that have been pre-defined by SMEs
4. Execution of pre-defined tasks for each of the events that have occurred in the order of their prioritization

During trial runs, the IAI agents were able to support the operators in the following tasks identified by the SMEs in this context:

1. UAV Route planning: The IAI agent calculates the direct route to the nearest track that needs investigation and activates that route for the UAV.
2. UAV Route Following: in the IAI agent sets the UAV flight management to make it follow the track in close proximity in order to get into and stay in sensor range.
3. Screen Management: The tactical map is managed by the IAI agent according to pre-defined rules, whenever new high-priority events are identified, e.g. moving the map to a location of interest and zooming in or out.
4. Inter-crew Communications: The IAI agent reports any observation concerning sensor track relationships to all crew members via a specific message window, which relieves the crew of the need to make or confirm these observations.
5. UAV Sensor Management: The IAI agent takes over the management of the Electro-Optical (EO) sensor as soon as a track is in visual range. The sensor management includes controlling the sensor attitude in order to establish a stable lock on the moving target.

6. UAV Data Link Monitoring: The crew is informed by the IAI agent if there is a problem with the data link, which is made possible by letting the agent monitor the data link continuously.

[Hou & Kobierski 2006] conducted simulator-based experiments with CP-140 crews, which had the task to guide up to five heterogeneous UAVs in the mentioned counter-terrorism scenario. Two configurations were examined: crew support by IAI turned on vs. IAI turned off. The support by IAI referred to the six tasks mentioned above.

The third approach to multi-UAV guidance described here is the *Machinetta* machine-machine-teamwork system, which was used by [Lewis, Wang & Scerri 2006] to control a large number of UAVs with a small group of human operators. These UAVs are called Wide Area Search Munitions (WASMs). They are supposed to loiter or approach targets and then strike a target while destructing themselves. By means of their automated target recognition functionality, they are also able to identify and select targets on their own.

As a basis for their work, [Lewis, Wang & Scerri 2006] identified the difficulty of human operators to control a large number of unmanned vehicles (e.g. more than ten) due to the operators' cognitive abilities. Other problems, when deploying a large number of unmanned vehicles in the same geographic area, are friendly fire or strikes against targets that have already been destroyed. All these issues can be addressed by machine-machine cooperation for the achievement of common goals. A necessity for this are sophisticated communication methods between the vehicles.

In the simulation scenario used by [Lewis, Wang & Scerri 2006], an AC-130 gunship is escorted by up to 80 WASMs, which have to be guided by a human operator onboard the AC-130. The operator has an interface that displays the positions of the AC-130, the WASMs and the detected targets. In addition, it allows the operator to control either individual WASMs or complete teams by sketching their ingress path, patrol regions or referring to pre-defined areas of interest. Upon target detection by a WASM the operator is asked for attack decision. For each task specified by the operator either automatic or manual selection of WASMs for task execution is possible. The operator interface was evaluated to be easy to use by SMEs.

[Lewis, Wang & Scerri 2006] state that in other research programs heuristic methods have been developed, which make such machine-machine cooperation computationally feasible for operational systems. However, the problem of making the machine teams "responsive to their commander's intent" [Lewis, Wang & Scerri 2006] is not yet solved. A WASM team needs to identify situations in which human decision-making (meta-reasoning) is indicated, and in that case communicate with the operator. Experiments were conducted with a simulated operator who reacted to WASM team requests after different lag sizes (5 to 120 seconds). The WASMs were able to make autonomous meta-reasoning decisions in case the operator did not react in time. The simulated operator could also be configured concerning the quality ("proactivity") of its decisions. The experimental results showed that shorter response time as well as more proactivity both lead to better performance of the overall system.

2.2.3 UAV Autonomy

As a general definition, *autonomy* is the ability to control one's actions or behavior by him/her/itself [Gunderson & Gunderson 2004]. This can also be regarded as *local determination* and *social independence* [Beavers & Hexmoor 2003], i.e. determining one's behavior without external (social) influence, or self-government [Gunderson & Gunderson 2004] and self-directedness [Bradshaw et al. 2003].

In the author's view, for UAVs this means that some logic onboard the aircraft or in the GCS can either make *tactical decisions* on its own (high autonomy level), or prepare *tactical action alternatives* for the human operator (medium autonomy level). The term *tactical* is meant to describe every decision that is regarded to be located one or more levels above today's flight control and payload control functionality. For example, stabilizing a sensor turret in order to be locked on to certain geographic position is part of payload control today, whereas selecting the target position shall be regarded as a tactical decision, which can be automated in the future. It is important to note that over time, certain levels of autonomy may be regarded as "natural" and then, for example, the medium level of autonomy is considered to be even more sophisticated automation.

Often mixed up with the topic of autonomy is the concept of Levels of Automation (LOA), which will be described in detail in chapter 2.3.1. However, there is an important difference between what is meant by the term autonomy and what is described by the LOAs. In the LOAs, the technical system always reacts to a specific human input, i.e. a task given to the system. Autonomy, on the other side, can mean that the technical system initiates actions, i.e. generates tactical decisions, all by itself. As it will be explained in the following, a higher level of autonomy can be the basis for raising the effectiveness as well as the efficiency of UAVs, which leads to operational as well as economic benefits.

The effectiveness of a UAV depends on fast decisions, e.g. in the role of Close Air Support in a dynamic battlefield. Having a human in the decision loop may increase the reaction time into a non-acceptable range, leading to inability to fulfill planned tasks. In addition, the data link to the UAV might be disturbed due to physical limitations, technical problems, or malicious jamming [Billman & Steinberg 2007]. There can also be radio silence policies. In these cases, the UAV still has to act in a certain way (e.g. hover/loiter, self-destruct, return to base, or continue mission) - it cannot "do nothing". High autonomy levels are therefore suited to mitigate problems in case of data link unavailability.

The efficiency of a UAV is, among others, determined by the number of human operators needed for operating one vehicle. In theory, the higher the autonomy level of the UAV, the lower is the UAV operator's workload [Platts 2006]. Hence, the higher the autonomy level, the more UAVs can be controlled simultaneously by the UAV operator(s). A higher autonomy level would also lower the load on the data link [Platts 2006]. However, raising the autonomy level (i.e. introducing more automation into the human-machine system) can also produce counter-productive effects, such as the "Ironies of Automation" mentioned by [Bainbridge 1983] or [Prinzel 2003].

UAV autonomy could (in theory) be driven so far, that the user interface can be realized via fire-and-forget natural language, more or less eliminating further human-machine interactions. Such a scenario was envisioned by the DAPRPA and US Army's *UCAR (Unmanned Combat Armed Rotorcraft)* research program for MUM-T [Jameson et al. 2005]. Despite its early cancellation, sophisticated system concepts had been developed, as can be seen in [Jameson et al. 2005] as well as [Franke et al. 2006]. The UCARs were designed to fight and survive in low altitudes and were meant to follow strategic goals and cooperate with manned aircraft (like the AH-64D Apache Longbow) even more closely on a team level. This should enable an airborne operator to take advantage of multiple UCARs without burdening him/her too much with the UAV control task. By commanding the UCARs as a team the number of UAVs controllable by one operator is not supposed to be restricted [Maddock 2004]. Possible targets for the UCAR are transmitted to the operator who may authorize weapons release. It is envisioned that the operator can be located in an AH-64D, UH-60L Army Airborne Command and Control System or at a ground control station [Cantrell 2005].

Another US-American project, named *J-UCAS (Joint Unmanned Combat Air Systems)*, was initiated by the DARPA, the US Air Force and Navy to build UCAVs, which should be able to fulfill strike, Suppression of Enemy Air Defense (SEAD) and surveillance missions [DARPA

2009] with automatic decision-making on a tactical level while following tactical goals and cooperation between multiple UCAVs on a group level [Maddock 2004]. With the X-45 and X-47 prototypes were built, which were able to fly in formation with other UCAVs or manned combat aircraft [DARPA 2009]. A single operator shall be able to control up to four J-UCAS UCAVs [Maddock 2004].

2.3 Human-Automation Interaction

In the following, modern paradigms of human-automation interaction are explained. Starting with the Sheridan's supervisory control paradigm and the associated Levels of Automation, this chapter then gives a description of the Adjustable Automation and Adaptive Automation approaches. Afterwards, even more complex views on human-machine cooperation are depicted: Joined Cognitive Systems, Dual-Mode Cognitive Automation, and the mixed-initiative interaction paradigm.

2.3.1 Supervisory Control

Many tasks in the industry or the military are about controlling a system. For example, the temperature of a nuclear reactor has to be controlled, or the flight level of an airplane. Nowadays, in these control tasks, the human is supported by an automation, which makes the corresponding task easier and the system safer.

A common human-automation integration paradigm, in which the human "supervisor" first gives tasks to the automation and then monitors the course of action and the results, is called *supervisory control* [Sheridan 1992]. This is in contrast to manual control, where the human performs the usually continuous control task by him-/herself without the use of automatic control systems and monitors the outcome of his/her own actions.

An early and short definition of supervisory control is as follows:

"SUPERVISORY CONTROL: A hierarchical control scheme whereby a teleoperator or other device having sensors, actuators and a computer, and capable of autonomous decision-making and control over short periods and restricted conditions, is remotely monitored and intermittently operated directly or reprogrammed by a person." [Sheridan & Verplank 1978]

The reason [Sheridan 1992] mentions for the invention of supervisory control is that in the case of non-repetitive and unpredictable jobs, the machine cannot be preprogrammed (i.e. work fully automated) and then work without supervision. Human perception (monitoring), planning and control is needed when there is a certain *disturbance bandwidth* [Sheridan 1992]. This value describes the amount of disturbances occurring while trying to follow the initial work plan. The tasks that the human operator has in the supervisory control setup are shown as a state model in Figure 2-4. In addition to the figure used by [Sheridan 1992], the path to the manual control state has been added, which is described by [Sheridan 1992] as well. One of the main benefits of supervisory control is that the accuracy and reliability of machines are joining the cognitive capability and flexibility of the human operator [Sheridan 1992].

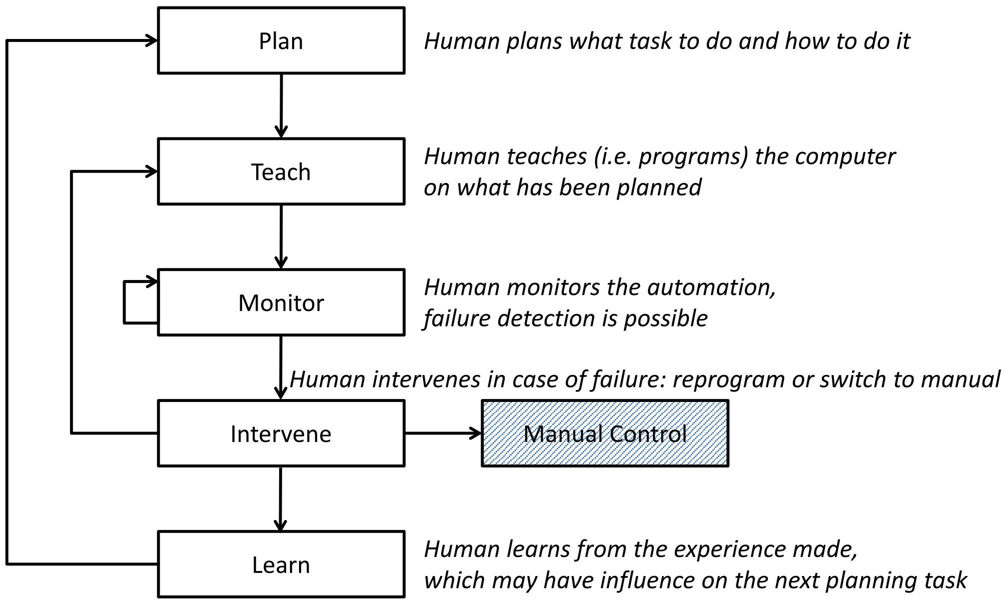

Figure 2-4: Tasks of the human operator in supervisory control - Adaptation from [Sheridan 1992]

A comparison between manual, supervisory and fully automatic control is depicted in Figure 2-5. As can be seen, manual control can be supported by an automation component already. In the other configurations, the automation can have more or less authority concerning in which way the task is executed. Looking from left to right the human's authority fades out more and more. In fully automatic control, the human only has a display to view the system state, but no means of control anymore, after he/she has given the task to the automated system.

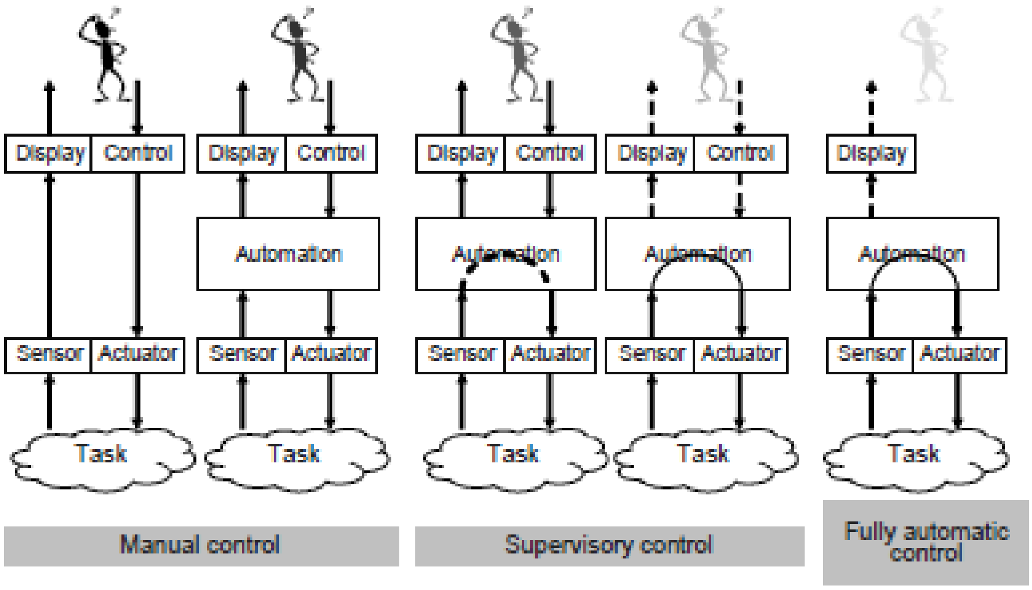

Figure 2-5: Stages of human machine dependency in supervisory control (Taken from [Hollnagel & Woods 2005], who made an adaptation of [Sheridan 1992][5])

A human-machine system design proposed by [Sheridan 1992] is depicted in Figure 2-6. In this context, [Sheridan 1992] stresses the analogy with the Rasmussen's scheme for human behavior [Rasmussen 1983], which categorizes human behavior into three layers, which are *skill-based*, *rule-based* and *knowledge-based* (see chapter 2.4.2 for details). In Sheridan's system design, the human is selecting the goal and planning the task execution on the knowledge-based behavioral layer. A so-called Human-Interactive Computer (HIC) interfaces to the human and aids him/her with the rule-based behavioral layer, which has stored procedures to control a work process. The interaction of the human supervisor with the HIC is defined by the Level of Automation (LOA). The HIC interfaces with multiple Task-Interactive Computers (TICs), which are responsible for the actual task execution and are normally physically located on the remote system (e.g. a telerobot), because this system needs to be controlled in real-time. The TICs correspond to Rasmussen's skill-based layer.

[5] Dashed line means most control loops are closed by the human, solid line means most control loops are closed by automation.

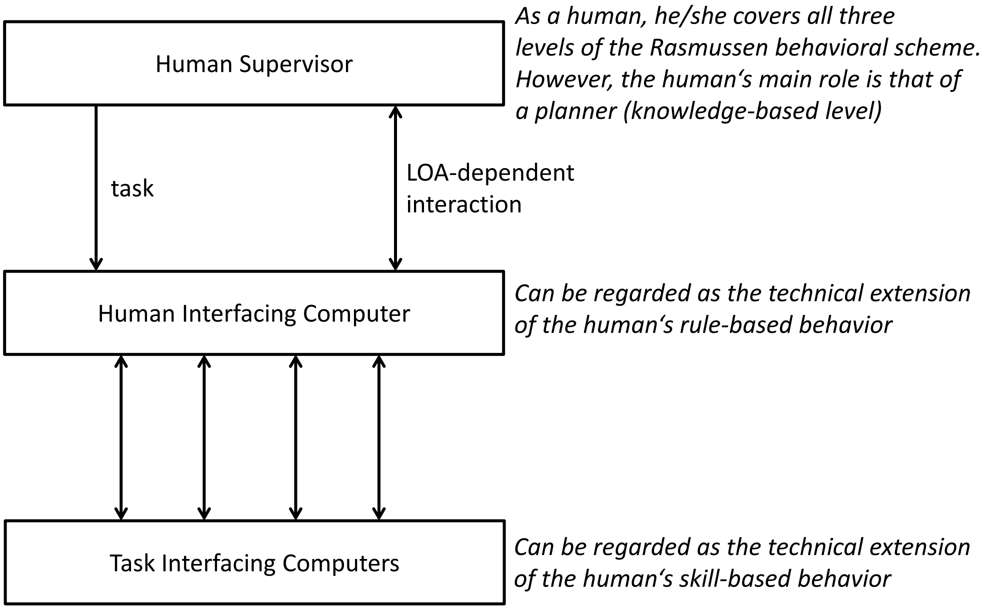

Figure 2-6: Human-machine system design for supervisory control (Information aggregated from multiple chapters of [Sheridan 1992])

The aforementioned LOAs can be used to classify the division of labor between human and automation. Originally ten such levels have been proposed by [Sheridan & Verplank 1978], which they did in the context of supervisory control. The revised list of LOAs [Sheridan 1992] is depicted in Table 2-2. It is important to note that the description of each LOA refers to the reaction of the technical system to a task given to it by the human supervisor.

Table 2-2: Sheridan's Levels of Automation [Sheridan 1992]

Automation Level	Automation Description
1	The computer offers no assistance: human must do it all.
2	The computer offers a complete set of action alternatives, and
3	narrows the selection down to a few, or
4	suggests one, and
5	executes that suggestion if the human approves, or
6	allows the human a restricted time to veto before automatic execution, or
7	executes automatically, then necessarily informs humans, or
8	informs him only if he asks, or
9	informs him after the execution if it, the computer, decides to.
10	The computer decides everything and acts autonomously, ignoring the human.

In the era of adjustable and adaptive automation (see below), system functions do not have to operate in one static mode, i.e. in one of the mentioned LOAs. Instead, they can switch the automation or cooperation mode dynamically, depending on situational demands [Steinhauser, Pavlas & Hancock 2009].

Another view on the LOAs is focusing more on the human-machine cooperation aspect, which applies only to the mid-level LOAs. The two relevant strategies here are "*management-by-consent*" and "*management-by-exception*" [Billings 1996]. The former means that the machine asks the human before acting/deciding, and the latter means that the machine acts/decides after allowing the human to veto for a certain time period. The similarities to Sheridan's LOAs are obvious (see also [Cummings & Mitchell 2007]).

The management-by-consent and management-by-exception strategies have been subject to many studies, also in the field of multi-UAV guidance [Ruff, Narayanan & Draper 2002] [Ruff et al. 2004] [Walliser 2011]. These studies indicate that management-by-consent, compared to management-by-exception, can lead to better operator situation awareness and trust[6] in the system. Which one of the two strategies leads to better overall human-machine system performance results, depends on factors like automation reliability, user trust, automation decision quality and many others, like, of course, the task setup for the human operator. It is also not a general question, but rather a situation-dependent one, as will be shown in the following subsections concerning the adjustable and adaptive automation paradigms.

2.3.2 Adjustable Automation

In human-machine systems, the functions to be executed by the overall system have to be either assigned to the human, or the machine. This allocation can take place at design time (static function allocation) or during run time (dynamic function allocation).

Static function allocation strategies distribute the tasks of the human and the machine at design time. There are different approaches that either maximize automation as such or try to minimize the economic costs. More human-focused strategies are to establish the lowest possible grade of automation or the static allocation of tasks in correspondence with the abilities that the human and the machine provide. The latter approach is based on the so-called Fitts' List (Fitts et al.

[6] Trust can be defined as "a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another" [Rousseau et al. 1998], whereby in this case the "another" is a technical system, towards which the human has positive expectations concerning its behavior. If these expectations have more weight than the fear of being "let down" by this system, the human has trust in it.

1951) and is also called MABA-MABA (Men Are Better At – Machines Are Better At) or compensatory principle (due to the compensation of human as well as machine weaknesses).

Dynamic function allocation is a reaction to the perception that “dynamic human issues such as selection, morale, motivation, fatigue, and monotony” [Hancock & Scallen 1996] have an impact on the performance of a human-machine system. With this insight, the allocation of tasks should not be static, but be handled dynamically instead [Hancock & Scallen 1996]. Systems designed according to this approach are able to dynamically distribute tasks among human and machine during run-time. One such system design approach is that of “adjustable automation” (sometimes also called “adaptable automation”).

Adjustable automation means that the human operator is able to adjust the grade of automation (e.g. in form of the LOAs mentioned above) for a specific function of the technical system. Examples for this are given by [Linegang et al. 2006] and [Billman & Steinberg 2007], who suggest that for controlling multi-UAV systems, there is the need for different interaction levels during operation, i.e. specifying tasks for the vehicles on either a low or a high level, depending on what the situation demands. A lower level of task specification is necessary in situations, where there is less predictability in the course of the mission and/or less predictability in the system behavior. Therefore, different interaction levels, which adjust the autonomy level of the system, may be necessary. This approach is also taken by the abovementioned Playbook™ Approach [Miller et al. 2005] for the mission planning phase: “The human user of such a system can express high-level mission goals or very specific mission plans - or anything in between” [Miller et al. 2005]. It is possible to realize such an interface by allowing the human operator to define constraints for the multi-vehicle system on different levels of a task hierarchy [Miller 2005]. Hence, in the author’s view, supervisory control can be regarded as mainly giving constraints to a semi-autonomous system or as constraining (and thereby adjusting) the autonomy of such a system. The system then generates solutions and/or acts inside these boundaries adjustable by the human operator, i.e. it is not adaptive in the sense that in is allowed to adjust its boundaries on its own. This makes sense, because the system possibly does not know of the exact goals the human operator pursues and/or the constraints he/she has in mind concerning the problem solution, unless he/she communicates these to the system (in form of constraints).

[Myers & Morley 2003] also performed interesting research in the field of adjustable automation. On the one hand, by the term “adjustable autonomy”, they refer to the dynamic definition of what the automation (or an agent built in software) shall perform autonomously.

On the other hand, "strategy preference" means the definition of how the automation shall act in its autonomy. Therefore, [Myers & Morley 2003] speak of *policy-based agent directability*. It is important to note that by the ability to set a preference, the human operator does not have to adjust the *level of involvement* [Myers & Morley 2003] with decision-making of the automated system in an ad-hoc manner, but is rather able to determine rules that apply to possible future events. Therefore, adjustable automation can be more than the delegation of tasks to the machine on different levels of detail or associating a LOA to every delegated task.

2.3.3 Adaptive Automation

In the previous section, the adjustable automation paradigm has been described, which allows the human operator to adjust the system's grade of automation (or that of any of the system's functions) on his/her own initiative. "Adaptive automation" goes one step further by allowing the adaptation of the task allocation through machine decision [Rouse 1976] [Hancock & Chignell 1987] [Scerbo 1996]. These decisions can depend on the task status on the one hand (e.g. in a UAV guidance task, a newly detected threat has an impact on the current flight plan of UAV). Or, on the other hand, they can be associated with the human operator's status (e.g. high workload or distraction of operator is detected by the system). In such situations, adaptive automation is meant to keep the human in the loop,[7] raise his/her situation awareness, keep his/her workload on an acceptable level, and thereby prevent errors in the overall human-machine system [Kaber et al. 2001]. These advantages of adaptive automation have also been shown in several studies, e.g. [Rouse 1977], [Hilburn, Byrne & Parasuraman 1997], [Kaber 1997], [Moray, Inagaki & Itoh 2000], or [Kaber & Endsley 2004]. However, the effectiveness of this approach always has to be scrutinized, and it is possible to conjure negative effects by applying it, as well, such as overtrust in the technical system and human skill decay [Hilburn et al. 1993].

Figure 2-7 was taken from [Endsley 1996]. It shows that the different tasks that need to be performed by the human-machine system can be handled at different LOAs. The third dimension is made up by the adaptive automation component, which follows rules prescribing when to switch the LOA for any task.

[7] One important aspect in human-automation integration is that the human should be kept "in the loop" of understanding system and environment states and in the loop of decision-making. Even if he/she transfers responsibilities to the automation (in particular in the sense of closed-loop control), the latter should inform the human about its decisions to guarantee his/her situational awareness, so that he/she is has all necessary information in order to intervene if necessary, and both partners can continue cooperating during the next problem situation.

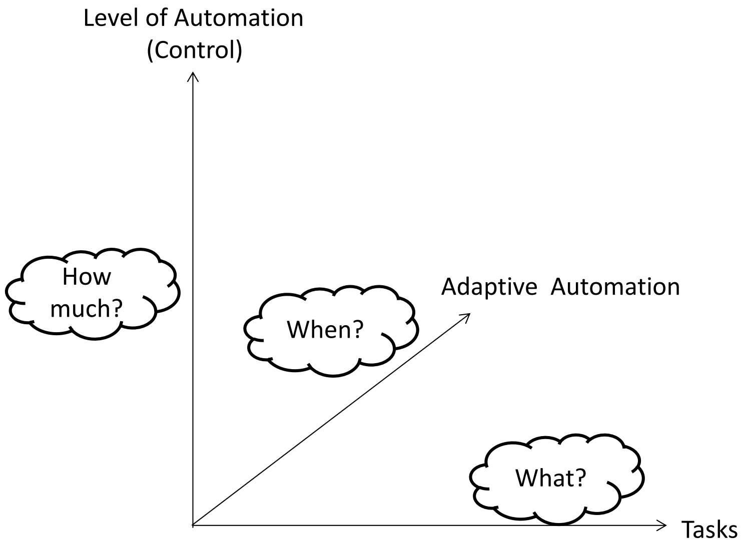

Figure 2-7: The three dimensions in the adaptive automation paradigm [Endsley 1996]

An important question associated with the adaptive automation approach is, how much authority or autonomy the adaptive machine shall have. [Scerbo 1996] and [Inagaki 2003] pointed out that in quickly changing environments with time-critical tasks, a fast and relatively autonomous decision by the automation may be necessary in order to prevent errors. The same can apply to situations, in which the human operator's workload is too high in order to react in time [Wiener 1989].

2.3.4 Joint Cognitive Systems and Cognitive Systems Engineering

The introduction and extension of automation may at a first glance increase effectiveness and efficiency of industrial or military systems. However, there are drawbacks associated with high levels of automation, which can be described with the term "ironies of automation" [Bainbridge 1983]. These include attention problems of human operators, who are now in the role of rather passive monitors. Due to the same reason, the operators have to face manual and cognitive skill degradation, leading to problems in understanding the process to control and misbehavior in case of manual takeovers [Hollnagel & Woods 1983]. Therefore, highly automated systems require intensive training efforts for the human operators [Bainbridge 1983]. Furthermore, the introduction and extension of automated functions in a human-machine system increases the system's complexity [Hollnagel & Woods 2005]. With higher system complexity, the task for the human operator normally also becomes more complex, which gives rise to additional human errors. These have to mitigated or prevented by adding more functionality (i.e. barrier functions and defenses) to the automation, again leading to a system complexity increase. The resulting "self-reinforcing complexity cycle" [Hollnagel & Woods 2005] is shown in Figure 2-8. In the end, the technical system might even become so complex that it is hardly manageable by the human operator(s).

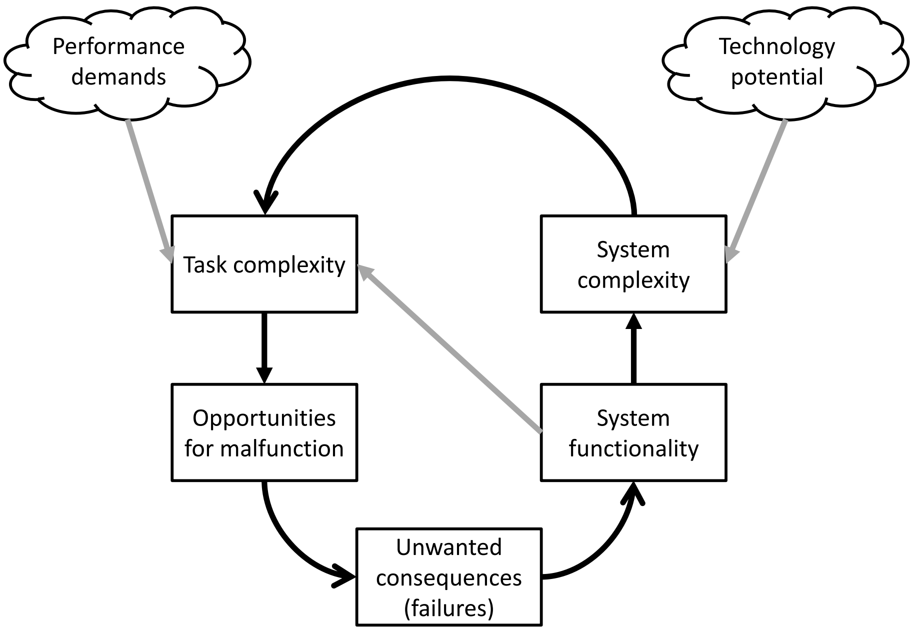

Figure 2-8: The self-reinforcing complexity cycle of technical systems [Hollnagel & Woods 2005]

The Cognitive Systems Engineering (CSE) approach [Hollnagel & Woods 1983] sets the stage for designing human-machine systems in such a way that human and automation collaborate on a cognitive level with the aim to mitigate the before-mentioned ironies of automation and the self-reinforcing complexity cycle. For this reason, in the CSE approach the human-machine system is viewed as a cognitive system as a whole, consisting of human(s) and technical system(s). This is why [Hollnagel & Woods 2005] also speak of a Joint Cognitive System (JCS). Therein, the human is seen as a cognitive system in accordance with the cognitive psychology paradigm, which means he/she is adaptive, has knowledge about him/herself and the environment, and he/she uses this knowledge for the planning and modification of intelligent actions [Hollnagel & Woods 1983]. Furthermore, a cognitive system is defined by producing goal-oriented behavior, performing symbol manipulation, and possessing knowledge of the world, which is used as heuristic (i.e. search-guiding) knowledge for planning and action selection. According to [Hollnagel & Woods 1983], the machine should be designed in such a way, that it is also a cognitive system, which works with an explicit model of the human operator and thereby effectively extends the human's mental capabilities. Cognitive coupling between the human and the artificial cognitive system can be achieved by shared cognitive models of the decision process, on which human-machine dialogs can be based ([Hollnagel &

Woods 1983] citing [Fitter & Sime 1980]). Viewing the human and the machine together creates a JCS. Putting the focus on the JCS instead of its components is important, because the designer is now able to analyze or optimize the performance of the complete human-machine system.

To briefly describe how JCS are related to CSE, [Hollnagel & Woods 2005] explain: "[…] the agenda of CSE is how we can design joint cognitive systems so that they can effectively control the situations where they have to function." Controlling means the timely and effective compensation of deviations between the actual and the desired state. The role of a JCS in a work environment therefore is to maintain a state of equilibrium, which means that it has to control the process, or, to be in control of the process. Pure feedback control is not sufficient, because it is always reactive, and is hence only able to re-attain the desired values in case of deviations. In order to maintain these values, additionally feedforward control is a necessity. The precondition for this is the ability to anticipate future process states and developments. Alternative developments have to be generated, characterized and evaluated with respect to the work objectives, and, finally, an action alternative leading to the desired development has to be selected by the JCS. According to [Hollnagel & Woods 2005], the generation of alternatives can be effectively supported by the machine, because computers are fast at calculations. As [Hollnagel & Woods 2005] put it, "Information technology can support generation of alternatives by supplementing the human ability for finding innovative solutions with the machine's single-mindedness and speed." However, machines are not considered to be good at the characterization and evaluation of alternatives.

In order to complete the definition of a JCS, it is important to note that a JCS can also consist of one cognitive system and one artifact, e.g. a (non-cognitive) tool or infrastructure, or of one cognitive system and one JCS [Hollnagel & Woods 2005]. Hence, JCS can be composed hierarchically.

The essential statement of [Hollnagel & Woods 2005] concerning the design of the automation is, that there should be a *coagency* (or *collaboration* [Hollnagel & Bye 2000]) between the human and the machine instead of a mere function allocation treating human and machine as independent entities.[8]

[8] Nevertheless, function allocation strategies, especially in the sense of adaptive automation, play an important role as well [Hollnagel 1999].

2.3.5 Dual-Mode Cognitive Automation

Whereas the supervisory control human-machine system setup described in chapter 2.3.1 only includes Rasmussen's skill- and rule-based behavioral layers into the automation, the Cognitive Automation approach [Onken & Schulte 2010] also puts the automation in charge of tasks located on the knowledge-based layer. This means, Cognitive Automation systems are able to plan, solve problems, make high-level decisions and act rationally in real-life situations that have not been foreseen during design time of the system [Schulte 2012]. As opposed to Cognitive Automation, conventional automation only has predefined solutions to predefined problems.

The human-machine system setup proposed by [Onken & Schulte 2010] is called *Dual-Mode Cognitive Automation*. It includes *Cognitive Agents* in two different roles: as a worker and as a tool [Schulte, Donath & Lange 2016]. . In the role of the tool the Cognitive Agent is integrated into the system as being controlled in supervisory control (*Hierarchical Control*) fashion by human workers or Cognitive Agent workers. In case a human worker and a Cognitive Agent worker both control the same tools, these two are in a *Heterarchical Control* relation [Schulte, Donath & Lange 2016]. A Cognitive Agent worker can be regarded as an *assistant system*, which assists the human operator/s.

In order to further explain the Dual-Mode Cognitive Automation approach, the concept of the *work system* and the related *work system analysis* [Onken & Schulte 2010] are laid out in the following. The work system, which is depicted in Figure 2-9, consists of two major elements. The first one is the *Worker*, which is the high-end decision component that pursues the overall work objective. The second element are the *Tools*, which are applied by the Worker to accomplish the work objective. Both are combined in order to achieve a certain work result (arrow going out to the right) on the basis of the given work objective (arrow coming from the left, e.g. from a supervising agency) while being constrained by environmental conditions (arrow coming from above, e.g. receipt of information or supplied resources). The Tools can be more or less automated artifacts, or even highly-automated Cognitive Agents. In both cases, the Tools have no knowledge of the overall work objective and simply perform the assigned subtasks that the Workers derived from the work objective. This relationship between the Workers and the Tools can be described by the supervisory control paradigm (cf. chapter 2.3.2).

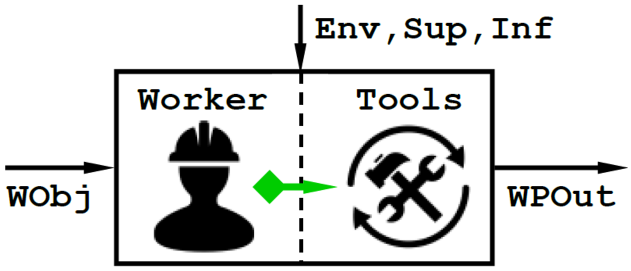

Figure 2-9: Work system, taking in the work objective, delivering work products [Schulte, Donath & Lange 2016]

As mentioned before, there are two ways to introduce Cognitive Automation into the work system. In the work system analysis, this can be either as part of the Workers or as a part of the Tools. In contrast to a Tool, a Worker knows and understands the work objective. The human Worker(s) and the Cognitive Agent Worker(s) can form a heterarchical team, which perform *Heterarchical Control*. Like the human operators, the Cognitive Agent Workers are able to derive necessary tasks from the common work objective as well as to delegate tasks to the available Tools, on their own initiative. As far as their authority is concerned, the main difference between humans and Cognitive Agent Workers is that Cognitive Agents in general, unlike the human operator, are not allowed to modify the overall work objective. This means that the human operator remains the highest authority in this work system, and the Cognitive Agents are not fully autonomous, because they cannot set their own objectives. Instead, the Cognitive Agents can be regarded as *semi-autonomous* systems. The combination of both types of implementations of Cognitive Agents in automation design is what [Onken and Schulte 2010] call Dual-Mode Cognitive Automation.

[Onken & Schulte 2010] have developed the Heterarchical Control approach as a decisive supplement to the supervisory control paradigm. It is not about substituting the supervisory control relationship between human and machine. Instead, Heterarchical Control means that the human is assisted by an additional automation component that in principle needs no supervision, but instead is able to take over supervision tasks itself and assist the human operator in a way similar to a human team member. As described above, this is made possible by the Cognitive Agent Worker's awareness of the overall work objective (including constraints), which can then be pursued by the Cognitive Agent on its own initiative [Onken & Schulte 2010]. It is important to note that in such a setup, from the human-automation integration standpoint, the human operator in charge is still fully kept in the supervisory control loop (e.g. controlling/guiding vehicles at a certain automation level in supervisory control). This means that the human operator is in principle able to handle all the tasks on his/her own. The cooperative assistant system is merely a workload moderator, whenever the situation demands for it. As recent experimental studies indicate, the contribution of such a cooperative assistant can be the prevention and/or correction of human erroneous actions [Theißing & Schulte 2016].

The Dual-Mode Cognitive Automation approach has already been used for implementing various research prototype systems, as described in the following chapters.

2.3.6 Mixed-Initiative Interaction

The CSE approach is focusing on human-machine coagency already, and, similarly, the Heterarchical Control approach is about the teaming of human and machine. In order to determine *how* such human-machine cooperation shall take place, it makes sense to look at the paradigm of mixed-initiative interaction. Although there is no “monolithic” mixed-initiative interaction paradigm, the many research works in this field deliver several ideas concerning what is technically possible and what are effective methods in human-machine cooperation.

In the mixed-initiative paradigm, human and machine are forming a mixed team. Both of the two partners may take initiative concerning either the interaction in dialogs or the working on tasks of the human-machine team [Cohen et al. 1998] [Strenzke & Schulte 2012b]. [Tecuci, Boicu & Cox 2007b] define the mixed-initiative paradigm as human and automation cooperating to achieve a common goal. This is similar to the Cognitive Automation approach described above. Furthermore, the mixed-initiative systems are designed either to accomplish goals that are unachievable by one of the partners on its own or to increase overall system effectiveness. It is important to note that [Tecuci, Boicu & Cox 2007b] do not speak of task allocation but instead of an interleaving of contributions by the human and the machine, which have different knowledge and different skills. These contributions are dynamic as far as their content and timing are concerned.

The idea of building mixed-initiative planning systems has been investigated since the 1990s [Burstein & McDermott 1996]. From then on, many systems following these principles have been developed for a wide range of domains [Tecuci, Boicu & Cox 2007a] [Ferguson, Hayes & Sullivan 2005] [Tecuci et al. 2003]. As will become clear in the following, the understanding of the term mixed-initiative varies widely between the different approaches and system implementations.

In order to categorize mixed-initiative systems, at first the term initiative has to be defined. [Cohen et al. 1998] present four theories concerning this definition, which are explained in the following. There are two underlying principles, one is seeing initiative as the control over the flow of conversation (human-machine dialog), and the other is seeing initiative as the control over the problem-solving task, respectively.

- The first of the four theories, theory #1, focuses on *initiative as dialog control*, i.e. who is leading the dialog has initiative.
- Theory #2 instead focuses on *initiative as control over a problem-solving task*, i.e. who is actively working on the problem-solving task has initiative.
- Theory #3 combines the aforementioned two theories and views *initiative as inducing problem-solving goals into the dialog*. The agent that defines the goals has initiative, while another agent may actually solve the problem associated with the goal. The latter agent would have initiative according to theory #2, but not according to theory #3.
- Finally, theory #4 builds upon theory #1 but adds the constructs of initiative strength and of conversation processes. The strength of initiative can be defined for every single utterance, even if the corresponding agent does not have initiative according to theory #1. The more distinct the utterance guides the problem-solving process, the more initiative strength it has. The analysis of conversation processes allows tracking different topics in a single conversation. Each topic is associated with one process. The processes may overlap during a conversation, and also multiple conversations (e.g. between more than two agents) may belong to the same process.

Based on the abovementioned definitions of initiative, [Strenzke & Schulte 2012a] have drawn a distinction between three main types of mixed-initiative interaction styles:

- *Dialog-focused systems*, which are based mainly on theories #1 and/or #4,
- *delegation-focused systems*, which are based mainly on theory #2 and
- *assistance-focused systems*, which are based mainly on theories #3 and #4.

It is important to note that these three styles can be intermixed in some way when designing a mixed-initiative system. As depicted in Figure 2-10, the delegation-focused approach is related to the supervisory control paradigm described in chapter 2.3.2, whereas the assistance-focused approach is related to the Heterarchical Control paradigm, which is described in chapter 2.3.6. Furthermore, the more dialog-focused a system is, the more negotiable is the problem-solving control. If it is not negotiable, it can be regarded as static or fixed.

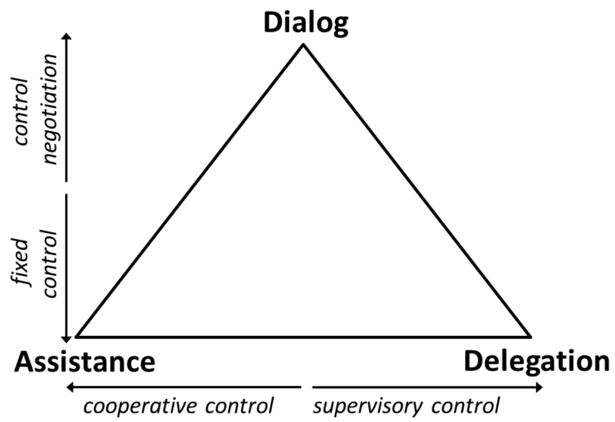

Figure 2-10: Different orientations of mixed-initiative systems

In dialog-focused systems the interaction is based on human-machine dialogs, in which either of the two agents can take initiative, e.g. in a more or less natural language-like dialog such as:[9]

- Human agent takes initiative (according to theory #1): “Let’s plan the route via Munich.”
- Machine agent responds “OK.” and starts calculating.
- Machine agent takes initiative (according to theory #1): “It would be faster to plan the route via Stuttgart thereby avoiding a traffic jam.”

The Rochester Interactive Planning System (TRIPS) [Ferguson & Allen 1998] is and a representative of this class of mixed-initiative systems.

In contrast to this, delegation-focused systems do not rely that much on dialogs, if at all. Instead the human agent delegates tasks to the machine agent, who may respond to the human (like “OK.” in the previous example). However, the understanding of initiative here is not based on leading the aforementioned dialog. Instead, initiative means initiative concerning the problem-solving task, i.e. the initiative of the machine is to begin, continue, or complete the planning

[9] This dialog is an adaptation of what is demonstrated by the TRIPS system [Ferguson & Allen 1998].

task or handed over from the human to the machine. Usually, in such systems the machine takes initiative in filling out the plan details. There is normally a strong focus of these systems on human-initiative as far as initiative theory #3 is concerned (i.e. human brings planning goals into play) and a strong focus of machine-initiative as far as theory #2 is concerned (i.e. machine is responsible of problem-solving). The Playbook™ Approach described in chapter 2.2.2 is an example for a delegation-focused mixed-initiative system.

Assistance-focused systems are characterized by the fact that, per design, the human is the main problem solver (i.e. having initiative according to theory #2) and the assisting automation (assistant system) is taking initiative in form of giving advices concerning problems or (sub)goals (i.e. having initiative according to theory #3), which is exactly the opposite role distribution compared to the aforementioned delegation style. When regarding advices more closely, theory #4 comes into play. It may be necessary to manage different conversation processes and/or to model advices of different of initiative strength, e.g. giving general advice that there is a certain problem vs. giving a specific proposal what the human is supposed to do. It is important to note that a highly automated assistant system can also take over tasks from the human, as described by the adaptive automation approach, which would correspond to initiative based on theory #2. The mixed-initiative system described by [Oates & Cohen 1994] can be regarded as assistance-focused.

[De Brun et al. 2008] also speak of initiative concerning the changing the LOA, which is in case of human initiative in coherence with the adjustable automation approach and in case of machine initiative in coherence with the adaptive automation approach. Raising the LOA leads to a shift of focus from dialogs and cooperation to machine-initiated problem-solving, i.e. the work upon the tasks delegated to the machine. Lowering the LOA of course has the opposite effect. Hence, an adaptive system is able to dynamically change its mixed-initiative focus during runtime.

2.4 Human-Machine Collaborative Planning

After having explained basics of Human-Machine Cooperation research, the next important background topic necessary to understand the main chapters (which are 3, 4, and 5) is that of human-machine collaborative planning. This subchapter first gives definitions of relevant terms in the area of planning and problem-solving. Then, the human and the machine's planning and problem-solving abilities are analyzed and compared.

2.4.1 Definitions

In the following, the terms problem, problem-solving, planning, and decision-making are defined for later use in this dissertation. There is no claim on universality of these definitions.

Definition of the Term "Problem"

Following the general and still accepted definition of [Duncker 1945], a problem exists when a living organism has a goal and does not know how this goal can be reached. This can also be regarded as having no *procedure* at hand to reach the goal [Hoc 1988]. A procedure is an ordered list of actions, which can include conditional branches. [Gilhooly 1989a] is even more specific by regarding a problem as a goal state that cannot be reached without *search*. Hence, if there is no procedure at hand by direct pattern-matching (from problem pattern to solution procedure), a procedure has to be found or generated by searching.

A problem consists of three components, which are the *initial state*, the desired *goal state*, and a set of *actions* to manipulate states [Reitman 1965]. These actions are also called *operators*. In addition to that, it is possible to define *constraints* as another component of a problem description [Simon 1964]. These constraints can refer to the goal state, intermediate states, or the operators [Hoc 1988].

Problems can be either *well-defined* or *ill-defined* [Reitman 1965]. For well-defined problems, all abovementioned components are completely defined. This is the case in the game of chess, for example. In contrast to this, in the real world, many problems are rather ill-defined. This means that the goal states are not completely defined, not all values of the variables in the initial state are known, and/or the effects of the operators are not fully predictable.

A further distinction between types of problems is drawn by the existence or the absence of an adversary. Again, one can take chess as an example for having an adversary. The problem becomes more complex in this case, because the finding of possible or likely actions of the adversary has to become part of the planning process.

Definition of the Term "Problem-Solving"

In this dissertation, problem-solving means the elimination of obstacles or closing of a gap in a plan in order to reach a desired goal [Betsch, Funke & Plessner 2011]. Problem-solving is a conscious, cognitive process [Betsch, Funke & Plessner 2011], during which first a representation of the problem is created, consisting of the components described in the previous subchapter [Ward & Morris 2005]. After creating the problem representation, either preplanning is conducted or a trial-and-error method is applied, or something in-between with preplanning and trial-and-error components.

Definition of the Term “Planning”

Planning is the creation of a plan consisting of list of actions. This process involves search. On the one hand, planning can be regarded as the solving of a planning problem, i.e. the working on a planning problem. On the other hand, planning is one step in many problem-solving processes, as described above. Planning is to be regarded as the creative part of problem-solving [Solem 1992].

Definition of the Term “Decision-Making”

Decision-making is the process of selecting one of multiple action alternatives with a decision for one alternative as the process outcome [Bertsch et al. 2011].

This process is tightly connected to that of planning. During planning, one has to make decisions, e.g. which path in the search tree to try out first, and after creating plans, one has to decide which plan is the best-suited one. Hence, previous planning can be the basis for decision-making [Hoc 1988].

Problem-solving and decision-making are also very tightly connected. A process can involve the solving of a decision problem, and during the solving of complex problems, decisions have to be made. According to [Solem 1992], decision-making is the uncreative part of problem-solving, whereas planning is the creative one (see above).

Definition of the Term “Rationality”

Rationality is described by [Russel & Norvig 2010] as follows: “A system is rational if it does the ‘right thing,’ given what it knows.” It then needs to be defined, how an alternative can be regarded as “the right thing”. According to [Simon 1955] economic human behavior equals rational behavior. This means, rationality is about the optimal decision, the optimal plan, the optimal problem-solving approach, and raising the expectations concerning success [Eisenführ, Weber & Langer 2010]. In addition, the decision process is only rational if a certain formal decision process is followed and there are no inconsistencies in the finding of the decision [Eisenführ, Weber & Langer 2010]. To be more specific, the consistency requirement for rational decision-making refers to respecting the laws of probability calculation[10] as well as the laws of transitivity[11] [Eisenführ, Weber & Langer 2010]. Furthermore, the measurement of

[10] E.g. if event or outcome b is a subcase of a, then the probability of b cannot be higher than that of a.
[11] E.g. if alternative a is to be preferred over alternative b, and b is to be preferred over c, then a has to be preferred over c as well.

rationality does not depend on the later outcome of the decision-making. However, on average, the approach of rational decision-making raises the probability of successful outcomes [Eisenführ, Weber & Langer 2010].

Because the human is oftentimes overcharged by acting rationally in case of complex problems [Eisenführ, Weber & Langer 2010], the prescriptive decision theory has been developed, which is a formalization of the common sense and can be regarded as a "manual" for rational decision-making. By means of this manual, problems difficult to grasp can be transformed or simplified through clearly defined steps of the decision analysis. It is important to note that decision-making is only to be regarded as rational if the effort for information gathering and information processing is justified [Eisenführ, Weber & Langer 2010]. This also means that losing too much time and/or resources during the decision-making process can change the conditions associated to the problem in a negative way.

2.4.2 Human Planning and Problem-Solving

Regarding the human as a problem solver is a huge field of research on its own inside the disciplines of psychology and cognitive sciences. Although many experiments have been made, only a few models concerning human planning and problem-solving have been developed, and these oftentimes are not empirically grounded. However, it is very common to distinguish between *procedures* (or *scripts*), which a human may apply to rather well-known, common situations and *plans*, which a human must generate in order to solve problems that he/she has not encountered before [Rasmussen 1983] [Hoc 1988]. Research that is more recent refers to human behavior and decisions that are not rational[12], but can nevertheless be efficient, especially in case of human experts. This is meant by the term Naturalistic Decision Making [Klein 2008].

Experts are better problem solvers than novices, because in comparison, they have more, more refined, and better proven *ready procedures* at hand [Hoc 1988]. In addition, they can keep a larger set of features of the problem situation in their working memory due to recognizing complex patterns and saving them as single elements, which are called *chunks*. This process is therefore called *chunking* [Chase & Simon 1973]. Chunking enabled experts to assess situations faster and more accurate, and enables them to match a well-fitting procedure to the problem pattern. Furthermore, chunking allows them, when they have to construct a plan, to be able to start planning at higher abstraction levels, and then drill down in a hierarchical manner. In case

[12] As [Hollnagel & Woods 1983] put it: "[…] man does not think as a calculus ratiocinator [...]".

a human has to deal with every single element of the problem (like every single vehicle on the battlefield or every chess piece on the table [Chase & Simon 1973]), his/her problem-solving performance is extremely degraded compared to the recognition of well-known patterns in combination with the application of well-known procedures. When speaking of well-known patterns and procedures, it becomes clear that the phenomenon of expertise is about *knowledge*.

Although chunking (or the absence of chunking) seems not reflected completely in Rasmussen's [Rasmussen 1983] model of human performance,[13] this model is in the author's view very well suited to explain human problem-solving processes, and it can be used to compare machine architectures to the human's manner of functioning. This is why in the following a brief overview of the Rasmussen Scheme is given.

Figure 2-11 shows multiple cognitive sub-functions, which are arranged in three layers. On the lowest of its three levels, the so-called *skill-based behavior*, the human performs unconscious (automatic) control subroutines. In the center, where the *rule-based behavior* is located, signs are recognized and rules (procedures) are triggered depending on the current state and the current task. These procedures, which sequence the aforementioned control subroutines,

"[…] may have been derived empirically during previous occasions, communicated from other persons' know-how [...], or it may be prepared on occasion by conscious problem-solving and planning. […] During unfamiliar situations, faced with an environment for which no know-how or rules for control are available [...], the control of performance must move to a higher conceptual level, in which performance is *goal-controlled* and *knowledge-based*." [Rasmussen 1983]

[13] Humans might need to plan on very low-level features (like every chess piece on the table when in case of a novice), and they might react to high-level symbolic problems by choosing a standard mid-level procedure.

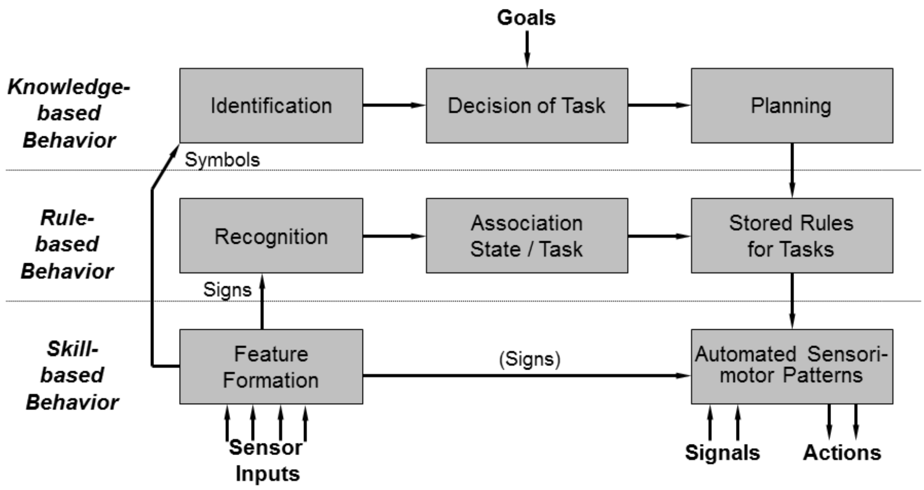

Figure 2-11: Rasmussen's scheme of human performance [Rasmussen 1983]

Another important aspect concerning human performance besides the ability to control on different levels (as in the Rasmussen scheme) is that the conditions under which the individual shall perform play a big role [Hollnagel & Woods 2005]. As examples, motivation, weariness and so forth can be mentioned.

2.4.3 Automated Planning and Problem-Solving

In order to explain the methods of automated problem-solving, i.e. the problem-solving by a machine, it makes sense to give an overview of its evolution first.

Early approaches of Artificial Intelligence (A.I.) concentrated on non-adversary well-defined problems (cf. chapter 2.4.1). The first prominent software that could solve some well-defined problems was the *General Problem Solver (GPS)*, developed in 1957 [Newell & Simon 1959]. It was able to perform problem reduction into subgoals by the so-called *means-ends analysis*, which worked backwards from the not yet reached goal state, searching for the best-suited method to move towards current state. Then, in 1971, another important approach was developed. The *Stanford Research Institute Problem Solver (STRIPS)* [Fikes & Nilsson 1971] was able to understand a model of actions that operate on the world state (and not on the problem itself as it was the case with GPS), which leads to a forward search. A robot knowing about its effectors (i.e. action possibilities) can use this kind of search process to solve well-defined problems. The invention of STRIPS was the kickoff for a long tradition of *classical, operator-based planning*. This can also be called *deliberative planning*, because the entity that is using such a planner is reasoning about its action alternatives and their corresponding outcomes. If costs are associated to actions, states, and/or events (or even event probabilities), something like machine *rationality* is achieved. Most of the classical planning systems use heuristic forward-search in the problem state space in order to find the action sequence solving the given problem.

Since 2002, action costs as well as durative actions, which allow temporal planning, are included in the Planning Domain Definition Language (PDDL) 2.1 standard [Fox & Long 2003], which is used by many developers of planning engines. In 2005, the PDDL3.0 standard [Gerevini & Long 2005] was proposed, which additionally includes soft constraints, i.e. constraints that can refer to states, which may only be violated at certain costs, as well as so-called *trajectory constraints*, i.e. constraints that do not refer to the goal state, but to intermediate states.

Between 1971 and 2002, two further developments can be regarded as important. First, whereas STRIPS was a *domain-independent planning* system (i.e. there were no domain-specific rules, when to prefer which action or which chain of actions), the discipline of *domain-configurable* planning systems arose in the mid-1970s. These systems do not only possess a world model, but also knowledge about how to solve problems in this domain, i.e. procedural knowledge. Hence, these systems are called *knowledge-based* planners. Because of their high performance, these systems are often used for real-world applications [Nau et al. 2005]. One example of a knowledge-based planning approach is Hierarchical Task Network (HTN) planning, in which tasks are divided into subtasks by the use of predefined knowledge concerning task divisibility until a sequence of atomic actions remains.

The second important development was *reactive planning* [Georgeff & Lansky 1987] [Firby 1987] [Kaelbling 1987]. This discipline arose through the need of real-time solving of simple problems in the field of robotics, in which deliberative planning is often neither applicable nor necessary [Brooks 1986]. Shortly after that, robotic scientists started to combine reactive planning with deliberative planning, which lead to *hybrid architectures* (often *three layer architectures*) [Gat 1992] [Gat 1998]. Thereby, they created machines that are able to react to unforeseen events in real-time (i.e. events that are not included in the agent's current action plan) while at the same time being able to reason about mid- or long-term goals and actions. For example, the Propice-Plan system (based on the Open Procedural Reasoning System / OpenPRS) is even able assemble reactive procedures by means of the deliberative planning layer, and then store these procedures for future use – exactly as described by the Rasmussen Scheme for (see above) [Despouys & Ingrand 2000]. This idea was already realized in 1975 by the HACKER system [Sussman 1975], but in contrast to Propice-Plan, it has not been used for solving real-world problems.

Other important approaches to the problem of planning in the real world included *hybrid planning*, which combines the advantages of classical and knowledge-based planning: flexibility and performance, respectively [Estlin, Chien & Wang 1997]. Furthermore, the solving of *probabilistic problems* became another research topic, which is very important for solving ill-defined real-world problems and for creating truly rational reasoning. In 2004, the Probabilistic PDDL (PPDDL) standard 1.0, which included probabilistic action effects and probabilistic distributions in the initial state, was released [Younes & Littman 2004]. It was extended by partial observability and renamed to Relational Dynamic Influence Diagram Language (RDDL) in 2011 [Sanner 2011]. More recent efforts of research lie in the integration of *planning and learning*. By solving a set of problems of a given domain, the planner may automatically learn to tune its parameter sets, find the regularities in the domain, or explore action outcomes or their probabilities [Vallati 2013]. Further steps that alleviate real-world planning problems are portfolio-based planning [Gerevini, Saetti & Vallati 2009] or the situation-dependent assemblage of algorithms and a modular planner [Jameson et al. 2005].

Figure 2-12 visualizes two main trends in automated planning. On the one hand, these are the generalization and flexibilization (e.g. triggered by the International Planning Competition's current focus on domain-independent planning). On the other hand, many real-world applications need not only the flexibility of a planning engine but also a lot of performance; Performance can even be the more important factor, depending on the specific needs of the application. As depicted in Figure 2-14, domain-configurable planners usually lack the flexibility of classical, domain-independent planning. For planning more complex missions with an acceptable response time, a hybrid approach of classical operator-based and HTN planning seems appropriate. Modular planners and dynamic algorithms-assembling are even more flexible and more performant approaches at the same time.

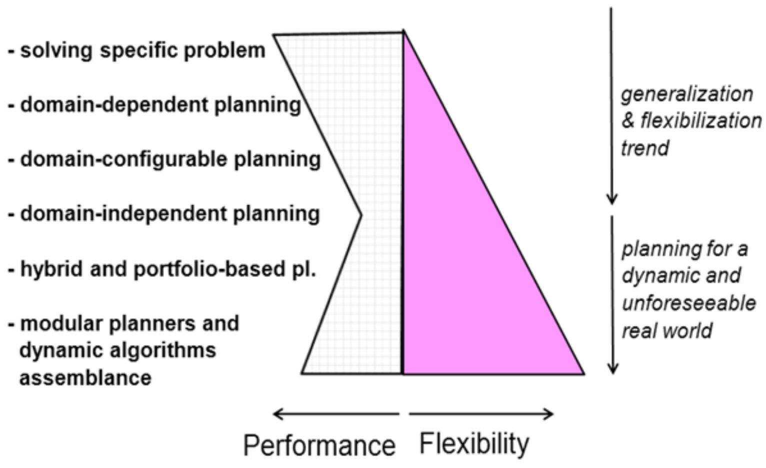

Figure 2-12: Performance and flexibility trends in automated planning [Strenzke & Schulte 2011a]

As can be seen from this brief overview concerning past and present trends in the discipline of automated planning, it is not possible to give a short and complete answer to the question, *how a machine is performing planning tasks*. What is common to all approaches is that the creativity in finding a solution is limited to the usage of atomic actions and methods which are usually statically defined in the planning domain. Another commonality is the dependence of the performance on the search-guiding heuristics, which are also in most cases static. Although today's machines can be regarded as uncreative, value lies in their ability to search for solutions systematically, to calculate times and costs quickly as well as correctly, and to decide perfectly rationally concerning the information that they have as input regarding the current situation and the model of the world.

2.4.4 Comparison of Human and Automated Planning and Problem-Solving

For the comparison of human and automated planning and problem-solving, the performance of the problem solver has to be considered. Because in the solving of planning problems, performance is no single-dimensional measure, the processes, strengths, and weaknesses of human and machine have to be compared.

It is common sense that *the human brain is a highly parallel working pattern-matching machine* [Kurzweil 2005]. Humans are also able to plan and act rationally, but this ability is subject to performance limitation through the *serial stream of consciousness*. Machines can become larger and faster concerning their memory and processing power, and will therefore soon exceed human planning capability in general, and they have done so already in many specific application fields, e.g. at chess. However, it is not so easy to achieve human-level pattern-matching capability in artificial cognition. *For very complex and ill-defined problems, human* *experts combine the two methods, pattern matching and serial rational thinking, and thereby achieve high problem-solving performance*. This leads to the conclusion that as long as machines are not more capable than humans in every aspect of the field of planning, systems that enable human and artificial agents to cooperate on the planning task are the most effective approach. [Hollnagel & Woods 1983] emphasize this with their statement that

"The [human] operator may eventually be found to be bad at any kind of work which can be described algorithmically, but this does not mean that a simple substitution of machine for man will improve the function of the total system."

For the approach of human-machine cooperative planning, where both partners should bring in their specific abilities, the obvious advantages of the machine are the *fast calculation of exact costs and times* during either plan generation or plan evaluation as well as the fast and exact application of deduction (see Figure 2-13 and [Strenzke 2011b]). In addition to that, automated planners are able to systematically search through large problem spaces, whereas the human brain is not suitable for this task [Gilhooly 1989b]. This is possible due to the quickly accessible long-term memory of the machine. Nodes in the problem space that have already been explored can be stored there, can be tagged with estimated costs, and can be retrieved again (in case of backtracking). In contrast, the human is limited in his/her performance by his/her dependence on external tools and the serial stream of consciousness. Hence, in order to minimize search effort, the human tends to retrieve a quickly available solution (e.g. an adaptation of a plan or procedure used before in a similar situation) [Gilhooly 1989b].

A human expert invests much time into the problem formulation [Reimann & Chi 1989]. Thereby they gain a deeper understanding of the facts and their relatedness as well as of applicable high-level schemata [Reimann & Chi 1989]. This enables experts to search for analogous problems and the corresponding solutions, either inside the same problem domain or even in other domains, which leads to far-reaching conclusion by analogy. The latter is possible because of the huge amount of background knowledge that humans gather throughout their life. This can be called intuitive reasoning rather than rationality. The search for analogous cases is completely different from the search through the state space of a problem. However, on the computer-side, there are also automated case-based planning systems [Veloso, Mulvehill & Cox 1997], which find similar problems and their solutions out of a database, but these are only used to find analogous cases inside the same problem domain – whereas the human can draw conclusions by analogy stretching over different domains. The detection of causal and semantic structures remains a human strength until today.

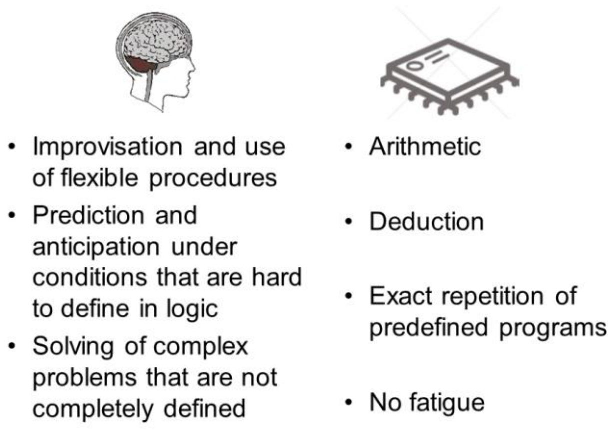

Figure 2-13: Selection of human and machine strengths [Strenzke 2011b] (based on [Hoyos 1990])

Another important difference between human and automated problem-solvers is the aspect of *rationality*. Whereas humans often do not decide or act rationally (as described above), *machines can be programmed to always decide and act rationally (in form of optimizing cost and risk)*. Nevertheless, for both human and machine the problem of *bounded rationality* has to be considered [Simon 1957]. This means that the problem solver is, for example, not able to know the values of all relevant variables, to have correct probability values for all action outcomes, or spontaneous events that could become relevant, and so forth. However, *due to different knowledge contents, knowledge representation, and knowledge processing methods, the human and the machine might again benefit from cooperating in solving the same problem.*

2.5 Manned-Unmanned Teaming Project

The German army is interested in performing MUM-T of manned helicopters with UAVs. In the envisioned scenario, these UAVs shall deliver real-time threat, environment, and terrain data directly to the helicopter crew, which is either made available in form of a video stream or as pre-interpreted and aggregated information. In a typical MUM-T mission, a group of manned helicopters is supported by one or more UAVs, which fly ahead and reconnoiter the planned helicopter routes, helicopter operation area (HOA), landing sites, target areas, and objects relevant for the mission and its safe execution (i.e. threats). The threat reconnaissance functionality shall include the localization, identification and tracking of ground vehicles as well as the localization of electromagnetic and infrared emitters. Furthermore, the UAVs shall be able to act as communication relays between arbitrary mission participants and have network-centric warfare abilities. As a constraint, the guidance of the UAVs should be possible from aboard a manned helicopter (e.g. NH90 or CH-53). Because there has not been specific research on MUM-T by the German Army so far, a primary study to make contact with this field and to analyze future technical possibilities was commissioned in 2007. One of the institutions that had been charged with this project was the Universität der Bundeswehr Munich (UBM). This dissertation is based on the work performed by the UBM for the German Army's MUM-T study.

In this chapter, first the organizational aspects of the MUM-T project are described. Then, the typical MUM-T operation scenario is detailed. After that, the general technical approaches delivered by the UBM are explained. Finally, the UAV operator's workstation and his/her tasks concerning the MUM-T mission execution are described.

2.5.1 Organizational Aspects of the Project

The project "Manned-Unmanned Teaming – Helicopter – Detached Sensor Platform" (MUM-T) was commissioned by the German Federal Ministry of Defense (BMVg) and the German Federal Office of Defense Technology and Procurement (BWB). The work upon the project started in 2007 and was completed in 2011. The UBM provided a staff of four research scientists and cooperated with the project partners Elektroniksystem- und Logistik-GmbH (ESG) and German Aerospace Center (DLR).

The research of the UBM in the MUM-T project focused on

- the examination of the aspect of multi-UAV guidance, which included UAV-UAV cooperation and human-machine cooperation (i.e. UAV operator assistance functions),
- the implementation of functional prototypes to demonstrate UAV-UAV cooperation and human-machine cooperation use cases in a simulated environment, and
- the implementation of selected use cases in a real-world environment with the UBM's fixed-wing and rotorcraft UAVs.

2.5.2 Operational Scenario

In the UBM's MUM-T reference scenario of operation, a manned transport helicopter is supposed to carry troops from a pickup zone (PZ) to an HOA containing two possible drop zones (DZ) (cf. Figure 2-14). In order to get there, the helicopter has to cross the forward line of own troops (FLOT) by the use of defined corridors. Each corridor has a specified time window in which it is open for crossing. Three rotorcraft UAVs fly in front of the helicopter with a lead time of about five minutes and a flight level that is significantly higher than that of the helicopter. The UAVs are taking over the job of preceding reconnaissance of the helicopter routes, the drop zone(s), and the troops' target, which is a warehouse. The more UAVs perform the reconnaissance of a route, the broader is their total sensor footprint (cf. Figure 2-15), thereby increasing safety for the manned high value asset. This reconnaissance information is made available to the helicopter cockpit crew as described below, which allows them to decide about the necessity of mission replanning, e.g. when a mission-relevant threat is discovered. For this operation, the envisioned Level of Interoperability (see chapter 2.1.4) is LoI 5.

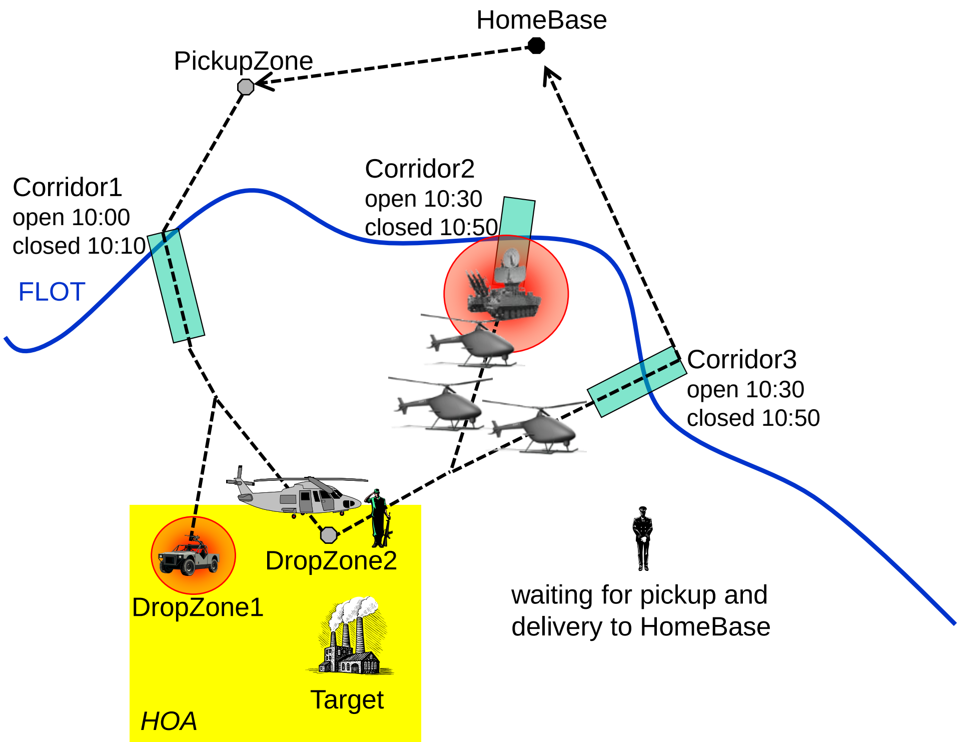

Figure 2-14: Manned-Unmanned Teaming mission scenario

To be more specific, the MUM-T setup of the UBM places the helicopter's Pilot-in-Command (PIC) in the responsibility to guide the UAVs and exploit the reconnaissance information gathered by these. He/she uses the data link that connects the helicopter with the UAVs to send commands (tasks) to these. The sensor data of the UAVs is returned to the helicopter cockpit, where the PIC is able to view and analyze the data. In the selected scenario, the UAVs are equipped with thermal infrared cameras, aided target recognition (ATR) functionality, and radar warning receivers (RWR). Thereby, they are able to transmit a camera live-stream, geo-referenced orthophotos, and photos as well as positions of objects that were categorized either as vehicles by ATR or as surface-to-air missile (SAM) sites by RWR.

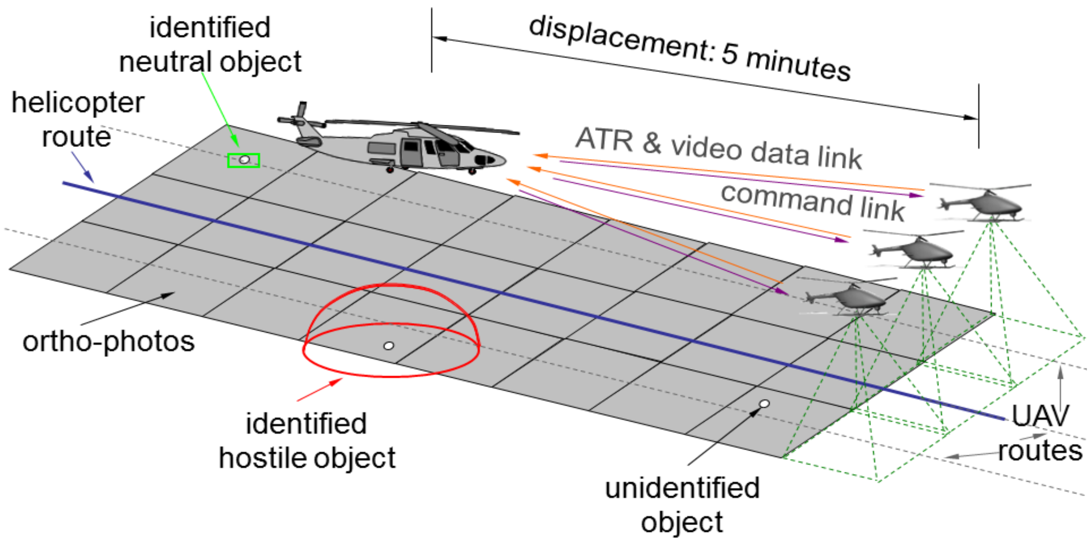

Figure 2-15: Manned-Unmanned Teaming operation concept (adapted from [Donath 2012])

The course of the MUM-T reference mission is shown in Figure 2-14 and described in detail in the following:

1. After receiving take-off clearance from the home base (HB) tower, the four aircraft start their mission.

2. The manned transport helicopter flies to the pick-up zone to load the troops and then proceeds to the primary ingress corridor.

3. The three UAVs head to this corridor directly and thereby gain an advance of approximately five minutes before the helicopter, which is in line with the MUM-T operation concept described above. In principle, they may also individually use an alternate ingress corridor.

4. The UAVs reconnoiter the helicopter route and its primary DZ for the troops, which is located inside the HOA. The manned helicopter is following with a delay of five minutes. The UAVs might find a threat (armed vehicle) near this DZ. In this case, the mission has to be replanned via the alternate drop zone, which also should be reconnoitered before the helicopter lands there.

5. Furthermore, the UAVs are supposed to reconnoiter the target of the troops as well as the troops' route to the target. After unloading the troops at a safe drop zone, the helicopter crew receives a follow-up mission order, which includes the transportation of one or multiple soldiers from hostile territory either to another location in hostile

territory or to the home base respectively. Figure 2-14 contains the example of a bailed-out pilot waiting to be transported to the home base.

6. Again, the helicopter routes and landing sites should be reconnoitered by the UAVs, because these have changed significantly.

7. As a final critical event, the UAVs may find the primary egress corridor blocked by a threat. When this happens, all aircraft need to be replanned towards the alternate egress corridor. After accomplishing the follow-up mission, all aircraft have to transit an egress corridor before they finally land again at the home base. The egress corridors have a limited opening time window.

It has to be emphasized that the scenario becomes challenging to the helicopter crew because the abovementioned events lead to the necessity of replanning the mission multiple times. In addition, due to the limited opening time windows of the corridors, a certain time pressure exists in the described reference mission.

2.5.3 Technical Approaches for Manned-Unmanned Teaming

For the MUM-T scenario described above, which includes the guiding of multiple UAVs by the PIC of the manned helicopter, the operator-to-vehicle ratio has to be inverted from *multiple operators, one UAV* to *one operator, multiple UAVs*. For this reason, the UBM has been conducting research on artificial cognitive systems that aid the UAV operator in coping with high work demands caused by multi-vehicle guidance and mission management in a dynamic military scenario. On the one hand, these systems can be deployed onboard of the UAVs to let them become semi-autonomous, cooperative, and guidable on a task-based level, which is more abstract than programming waypoints [Uhrmann & Schulte 2012]. On the other hand, the operator shall be supported in mission planning and UAV tasking by a cognitive assistant system [Rauschert & Schulte 2012] [Strenzke & Schulte 2011a].

2.6 Research Questions

The research questions first formulated in chapter 1.1, which will be detailed and put in context here, are the following:

1. *How can a human-machine cooperative system* for military Manned-Unmanned Teaming mission planning *be designed* in order to be helpful for the user(s)?

2. What are the *requirements of potential users* (Army Aviation pilots) concerning a *human-machine cooperative system* for military Manned-Unmanned Teaming mission planning?

These questions are researched in a specific context, which is the UBM's MUM-T project. This encompasses a single-operator multi-vehicle guidance setup, in which a manned transport helicopter is supported by multiple reconnaissance UAVs. The UAVs fly ahead of the manned helicopter and scan the helicopter routes and landing sites with sensor coverage as broad as possible. The problem that the UBM is addressing in the MUM-T application is to maximize overall human-machine system performance in the mentioned setup. In order to guarantee threat-safe helicopter routes and landing sites, the UAV operator has to assign reconnaissance tasks as well as other supporting tasks to the UAVs. In this setup, the UAV operator is at the same time the helicopter commander (pilot in command, PIC) who is located in the helicopter cockpit. More specifically, he/she has to assign about a dozen ordered tasks to each of the three UAVs at the beginning of the mission (first working under low time pressure) and later also has to maintain a workable plan of sufficient quality throughout the dynamic mission, in which the situation may change unexpectedly (then possibly requiring time-critical decision-making and replanning). This happens when a threat is detected that blocks a landing site or a designated flight corridor and when the mission goals are changed by a ground-based mission commander, i.e. a follow-up mission is commanded. There are certain time constraints concerning when ingress and egress corridors to/from hostile territory may be used, which limit the time frame of the overall operation and make the planning problem more difficult for the human operator.

With UAV guidance concepts that are in use today, the main problem in this configuration is the overwhelming task load for the PIC. Multi-UAV guidance is still a practically unsolved problem due to workload issues. Giving further tasks to the UAV operator, i.e. the responsibility for a manned helicopter, even worsens the workload situation. In order to keep the operator's workload inside an acceptable range and to maximize the overall system performance, he/she shall be supported in mission (re)planning and plan execution by an assistant system in a cooperative approach.

Hence, regarding question 1, to develop a cooperative mission planning system that is helpful, it shall operate in a way that it

- lowers the overall workload of the human operator,
- contributes positively to mission performance,

- generates solutions that the human operator understands and is aware of,
- generates solutions that have sufficient quality from the viewpoint of the human operator.

Question 1 can be expanded to ask, how a cooperative mission planning system can fulfill these four aspects.

Regarding question 2, the potential human operators‘ requirements can be categorized as follows:

- Requirements concerning plan management features
- Requirements concerning planning process duration
- Requirements concerning optimization goal priorities

To this end, the dissertation shall propose a concept concerning the cooperation of the UAV operator with an automated planning system (chapter 3), describe a design and the prototypical implementation of such technical system (chapter 4), evaluate the approach taken in human-in-the-loop simulator experiments with SMEs (German Army Aviation pilots), and gather further requirements in the abovementioned categories for such a system (chapter 5). In order to achieve this, for two configurations with assistant system enabled (mixed-initiative mode) and disabled (manual mode), the mission performance shall be objectively measured. In addition, the crew’s subjective workload and objective situation awareness shall be assessed. Furthermore, the SMEs shall subjectively evaluate the assisting functions and make statements concerning their planning strategies as well as their requirements for a planning system such as the MMP.

3 Manned-Unmanned Teaming System Concept

This chapter deals with the UBM's MUM-T system concept. It is divided into three sections, describing the overall system concept first, then the requirements for the Mixed-Initiative Mission Planner (MMP), and finally the MMP concept, which is being derived from these requirements.

3.1 Overall System Concept

In the following, the work system analysis is used to introduce Dual-Mode Cognitive Automation into the UBM's MUM-T setup.[14] This shall support the task performance of the cockpit crewmembers and to minimize operation errors.

As explained before, the work system itself consists of two major elements: first, the Worker is the deciding component of the work system, which also pursues the work objective as such, and second, the Tools, which are used by the Workers to accomplish the work objective and merely perform subtasks assigned to them (supervisory control paradigm). In order to understand these subtasks on a symbolical level and to interpret the current situation correctly, intelligent knowledge-based agents, i.e. Cognitive Automation, will be deployed for the Workers in the MUM-T setup. These Cognitive Agents, which are implementing the Cognitive Automation approach, will be installed onboard of the UAVs, i.e. on the side of the Tools (see the small robot heads on the right side in Figure 3-1). For more information about the UAV Cognitive Agent concept, see [Uhrmann 2013].

Considering the experimental results of [Uhrmann et al. 2009] and the ones reported in [Bergantz et al. 2002], it seems not possible to guide UAVs from the cockpit of a manned helicopter without assistance functions for the human operator.[15] Therefore, in the MUM-T setup, the PIC shall be supported by an assistant system, the so-called *MUM-T assistant system*, which is to be designed according to the Heterarchical Control paradigm. This means that the MUM-T assistant system has to be realized as a Cognitive Agent Worker. The combination of SCUs and OCUs in order to support the human operator leads to the Dual-Mode Cognitive Automation, as described in chapter 2.3.6.

[14] For background information concerning the work system analysis as well as the Cognitive and Cooperative approached, see chapter 2.3.6.

[15] "One very clear result from MUM IV is that cognitive decision aiding tools are absolutely necessary if the Objective Force concept continues to add remote weapons and sensors to unmanned systems." [Bergantz et al. 2002]

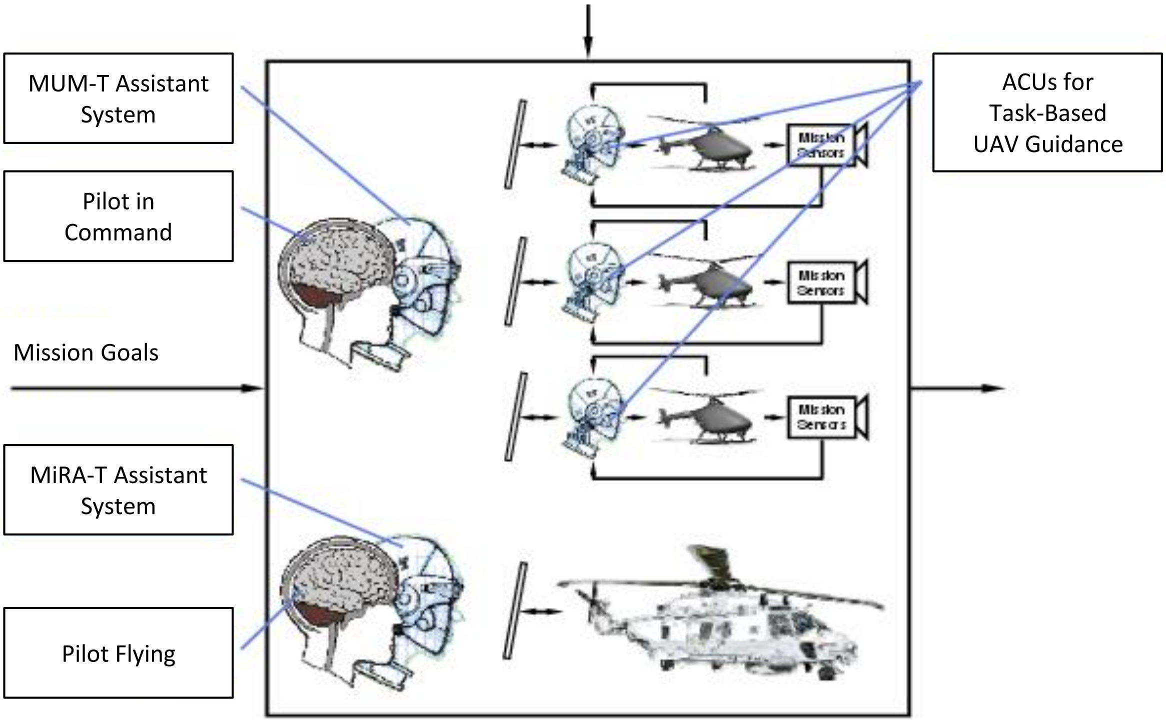

Figure 3-1: Work system analysis with Dual-Mode Cognitive Automation for MUM-T [Schulte 2012]

Related to the UBM's MUM-T system concept is the *MiRA-T (Military Rotorcraft Associate – Teaming)* research project, which was also processed at the UBM. MiRA-T is supposed to deliver a concept for and a prototypical implementation of an assistant system for the PF in the MUM-T setup. This means that in the UBM's Dual-Mode Cognitive Automation design for the combined MUM-T and MiRA-T, the PIC and the PF are each supported by a cognitive assistant system. Whereas the MUM-T assistant system will be described in the following, the MiRA-T assistant system is not specifically relevant for the work presented here. Details concerning the MiRA-T assistant system can instead be found in [Maiwald 2013] and [Strenzke et al. 2011]. Figure 3-1 shows the MUM-T plus MiRA-T work system configuration, which consists of a single work system. The PIC is clearly responsible for working upon the mission goals, and the PF also receives the mission briefing before the start of mission. Hence, the Workers include the PIC (upper human head) as well as the PF (lower human head). Both crewmembers are in control of the manned helicopter: the PF mostly via direct controls, e.g. flight stick, and the PIC exclusively via higher-level controls, e.g. Multifunctional Head Down Displays (MHDDs) and Multifunctional Control and Display Unit (MCDU).

The Cognitive Agent that shall support the PIC in the MUM-T setup is conceptually based upon the UBM's Cognitive Process model [Onken & Schulte 2010], which deals with mental concepts in the decision and behavior generation process of a human individual. These concepts are

- *environment model* (called *belief* when instantiated),
- *desire* (called *goal* when instantiated),
- *action alternative* (called *plan* when instantiated), and
- *instruction model* (called *schedule* when instantiated).

The remainder of this section first describes the concept for deployment of Cognitive Automation onboard of the UAVs in order to enable task-based multi-UAV guidance. After that, the MUM-T assistant system is explained.

3.1.1 Task-Based Multi-UAV Guidance Using Cognitive Automation

This subchapter gives a short overview concerning the Cognitive Automation approach for task-based multi-UAV guidance. For details, see [Uhrmann 2013] or [Uhrmann & Schulte 2012].

As described in chapter 2.1, the state of the art in UAV guidance is the definition of 3D or 4D (3D plus point in time) waypoints for single UAVs, either during mission planning before the start of the flight, or as a re-definition during mission execution. To relieve the human operator of workload, the UBM had to invest in research concerning two aspects of UAV guidance. The first one was *task abstraction*, which means that it shall be possible to command abstract tasks to the UAVs instead of waypoints. Details and in-between tasks left out by the operator shall be filled or added automatically (e.g. the UAVs can detect that, for example, the FLOT is not crossed correctly by the UAV's task agenda entered by the human operator, and that in this case a "cross-FLOT" task using the correct corridor has to be added at the correct position of the agenda). The second was the ability of multi-UAV guidance via a single operator interface. This also includes enabling the UAVs to cooperate on a task defined by the operator (e.g. the automatic coordination of the UAVs' flight patterns during a route reconnaissance task in order to optimize sensor footprint coverage). Because the human operator is giving tasks to the UAVs and then is able to monitor the mission progress, the human-automation integration approach taken for the UAV guidance in the MUM-T setup clearly is supervisory control (see chapter 2.3.2). The agents onboard of the UAVs need to be designed as SCUs (cf. Figure 3-1) and shall follow the Cognitive Process paradigm.

3.1.2 Cognitive Assistant System for UAV Operator

As already mentioned above, in addition to the SCUs onboard of the UAVs, the UBM's Dual-Mode Cognitive Automation concept for MUM-T includes the employment of an OCU as assistant to the PIC. Due to the fact that the MiRA-T assistant system already supports the helicopter crew in conducting the helicopter flight, the MUM-T assistant system has the role to support the PIC in guiding the UAVs. Therefore, the MUM-T assistant system is also called UAV operator assistant system in this dissertation.

Following the assistant system paradigms of [Onken & Schulte 2010], the assistant system shall initiate a dialog with the human operator only if it is found necessary to support him/her, i.e. if an error is detected by the system or an error is anticipated by the system. Otherwise, the assistant shall be passive and silent. In detail, the basic requirements for assistant systems given by [Onken & Schulte 2010] are the following:

> 1. "The assistant system has to be able to present the full picture of the work situation from its own perspective and has to do its best by own initiatives to ensure that the attention of the assisted human operator(s) is placed with priority on the objectively most urgent task or subtask." [Onken & Schulte 2010]
>
> 2. "If according to requirement 1 the assistant system can securely identify as part of the situation interpretation that the human operator(s) cannot carry out the objectively most urgent task because of overtaxing, then the assistant system has to do its best by own initiatives to automatically transfer this situation into another one which can be handled normally by the assisted human operator(s)." [Onken & Schulte 2010]
>
> 3. "If there are cognitive tasks, the human operator(s) is (are) principally not capable to accomplish, or which are too high risk or likely a cause of too high costs, these tasks are to be allocated to the assistant system or operation-supporting means, possibly a supporting cognitive unit." [Onken & Schulte 2010]

Now these basic conceptual requirements are to be applied to the MUM-T application. The aim of introducing an assistant system for the UAV operator in the MUM-T setup is to enhance the overall mission performance. An important aspect of this is the optimization of the support of the manned helicopter by the reconnaissance UAVs flying ahead. To grant this support, the human UAV operator has to task the UAVs intelligently, so that the helicopter routes are reconnoitered well and reconnoitered in time. The same counts for the drop zone(s) and the target of the ground troops.

First of all, the operator has to generate a good mission plan at the beginning of the mission. He/she is working under low time pressure in this situation, but one may also think of scenarios where even the initial mission planning is a time-critical issue. Then, during the execution of the mission, the operator has to maintain a good mission plan throughout the complete course of action, which most likely requires time-critical decision-making. This means, he/she has to react to unforeseen events quickly and thoroughly by replanning the mission according to the new requirements. For example, the primary landing site is found to be threatened, and therefore the alternate landing site has to be selected as new target for several tasks. In case the operator makes a mistake while entering his/her orders into the system, it is likely that he/she will notice the unwanted result quickly due to the feedback on the graphical display (which is described in the next section). Nevertheless, this leads to the necessity to edit or deleted UAV tasks, further increasing the time pressure. Because of the tight schedule of a military mission and the permanent threat to be detected and fired upon while flying low in hostile territory, a certain time pressure has to be considered. Due to the reasons given above, it makes sense to support the PIC in planning, replanning, and executing the MUM-T mission.

From a functional point of view, in the MUM-T scenario there is no difference between planning and replanning. [Rauschert 2013] has developed two *use cases of planning assistance* by the MUM-T assistant system. One is the appending of UAV tasks at the end of their agenda, in case that this task either fulfills a mission objective, prepares the fulfillment of a mission objective, or leads to a better reconnaissance coverage and thereby increases the helicopter's safety. The other is the (re)planning of the complete mission, which includes all UAVs as well as the manned helicopter. See also [Rauschert & Schulte 2012] for details. In both cases, the following three steps of assistance are taken by the automation, as proposed by [Onken & Schulte 2010]:

1. The first step is giving a notification to the human operator, e.g. “UAV1 needs follow-up task”.
2. The second is giving a proposal to the human operator, e.g. “Add task transit from Home Base to Pickup Zone for UAV1”.
3. The third is the execution of the proposed action by the automation followed by an acknowledgement, e.g. “Added task for UAV1”.

A complete example for such a human-machine dialog is given in the following:

- Assistant takes initiative: “UAV1 needs follow-up task”
- Operator presses “proposal” button
- Assistant proposes: “Add task transit from Home Base to Pickup Zone for UAV1”
- Operator presses “accept” button
- Assistant affirms: “Added task for UAV1”

The execution of the MUM-T mission is supported by two additional UAV-operator assistant system use-cases, which are not directly related to mission planning. These are, first, letting a UAV switch to its next task in case of a task that is not automatically ended by the automation onboard of the UAV, i.e. area reconnaissance or object surveillance; and second, supporting the human operator in the identification of a detected object (ground vehicle) by showing the corresponding photography on one of the operator’s displays. For the latter case, there is no automated takeover (step three) possible, because the target always has to be identified by the human operator. Apart from that, the stepwise increase of the LOA is designed as in the example mentioned above.

In order to achieve the LOA switching, the MUM-T assistant system shall be a knowledge-based system (i.e. a system which has a store of knowledge and apart from that procedures to work upon this knowledge) mainly holding knowledge about the modes of interaction with the human operator [Rauschert & Schulte 2012]. For the communication with the operator, the assistant system shall be able to instantiate a dialog or make an announcement via speech synthesis and the displaying of a message box in the task-based UAV-guidance graphical user-interface (GUI). Whenever appropriate, this message box includes a few buttons that allow the operator to invoke further aid by the assistant system or to either accept or reject its proposals.

3.1.3 Task Description for Human Operator

As described above, the cockpit crew of the manned helicopter consists of the *pilot flying (PF)* and the *pilot in command (PIC)*. The latter, who is responsible for the guidance of the UAVs, is the human operator in focus in this work. His/her workstation and the tasks he/she has to perform are described in the following.

In addition to being responsible for guiding the UAVs, the PIC is, as the job title suggests, the commander of the manned helicopter. Because this helicopter is the only one participating in the MUM-T reference mission, the PIC can furthermore be regarded as the mission commander. Hence, although the MUM-T research focuses strongly on UAV guidance issues, actually the planning of the overall mission as well as the helicopter flight path are the most important tasks for the PIC. Therefore, the PIC needs access to the mission management system of his/her helicopter. The interface and functionality of this mission management system is described in detail in chapter 3.2.3.

Besides his/her job as commander of the manned transport helicopter, in the UBM's MUM-T setup, the PIC is acting as the single operator of multiple UAVs (up to three to be specific as far as the MUM-T reference scenario is concerned). As tools he/she has a simple moving map-based planning interface (cf. Figure 3-3) allowing him/her to create, modify, and activate mission plans on a per-UAV basis. In the reference MUM-T mission, for the initial planning, it is necessary to allocate approximately 12 individual tasks to each of the UAVs, which results in about 36 UAV tasks in total[16]. These tasks are specified by a type (as listed below), a target location[17] and the designated UAV. Each task is inserted into a sequential agenda of the corresponding UAV, either at the end of it or at a position specified by the operator. Afterwards, the task is automatically connected with the preceding action and the successive (if there is any), i.e. routes between the task locations are automatically generated if necessary. It is not possible to specify anything like an execution or arrival time for any tasks. Hence, the UAV operator has the additional responsibility to check if the mission time constraints (e.g. ingress and egress corridor time windows) will be met by the plan. More details to the task-based guidance concept used in the MUM-T application can be found in [Uhrmann 2013] or [Uhrmann & Schulte 2012].

[16] As explained in chapter 3.2.3, the human operator can leave out tasks in between when entering the UAV plans, which will lead to automated task agenda completion by the UAVs' onboard automation. Hence, the UAV operator does not have to enter all the tasks manually during the planning or replanning of the MUM-T mission.

[17] Only target locations can be specified and no target objects. This has to do with the simplification that MUM-T reference scenario does not include moving objects.

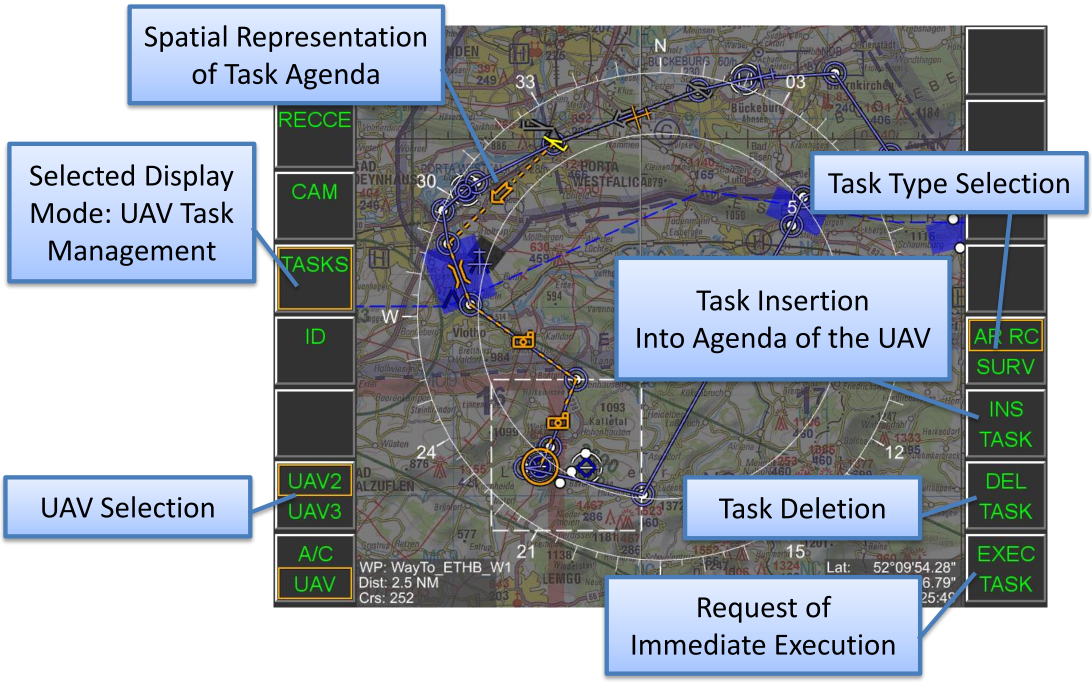

Figure 3-2: Map-based user interface for task-based UAV guidance [Strenzke et al. 2011]

The following UAV task types have been implemented for MUM-T:

- *Departure*: The UAV shall take off and follow a departure procedure.
- *Transit*: The UAV shall fly to a certain position, specified in symbolic mission terms, e.g. “drop zone alpha”. The UAV shall automatically avoid threats along the route.
- *Cross FLOT*: The UAV shall cross the FLOT by using a specified flight corridor.
- *Recce route*: The UAV shall gather sensor information (i.e. take overlapping orthophotos with its thermal camera) for a flight leg, which is defined by a start and a destination position (mission-relevant locations). The UAV shall coordinate itself with all other UAVs having the same recce route task to maximize sensor coverage.
- *Recce area*: The UAV shall gather reconnaissance information (i.e. take overlapping orthophotos with its thermal camera) for the area surrounding the specified location. This shall be accomplished by a flyover and circle loiter pattern.
- *Surveillance*: The UAV shall monitor an object continuously with the on-board sensor equipment (i.e. thermal video camera). This shall be accomplished by a circle loiter pattern
- *Landing:* The UAV shall fly an approach pattern towards a landing site.

In the following, the UAV behavior is explained in more detail. During the recce-route and recce-area tasks, the UAVs make orthophotos below them in such a way that a certain overlap exists between these. Thereby, a continuous area around either the helicopter route or a specified target is covered by the orthophotos (see Figure 3-4). As soon as a new photo is taken, the ATR of the UAV sensor management searches for hot objects in the photo, which have the size of a ground vehicle. In case there is such detection, the information about the location of the object is transmitted to all aircraft in the manned-unmanned team. In the manned helicopter, the map displays then show a yellow question mark at the corresponding location. The UAVs are programmed to take identification photos of detected ground vehicles from different angles. They do this cooperatively, i.e. also UAVs that have not detected the vehicle take photos of it after being informed about the vehicle position. However, the UAVs are not changing their flight path for taking the photos. In case a UAV currently performs a recce task, its sensor switches back orthophoto mode after taking an identification photo. After detection, the objects at the markings in the map have to be identified by the PIC by means of the identification photos made by the UAVs. It is then necessary to specify the general vehicle type and the allegiance of the vehicle.

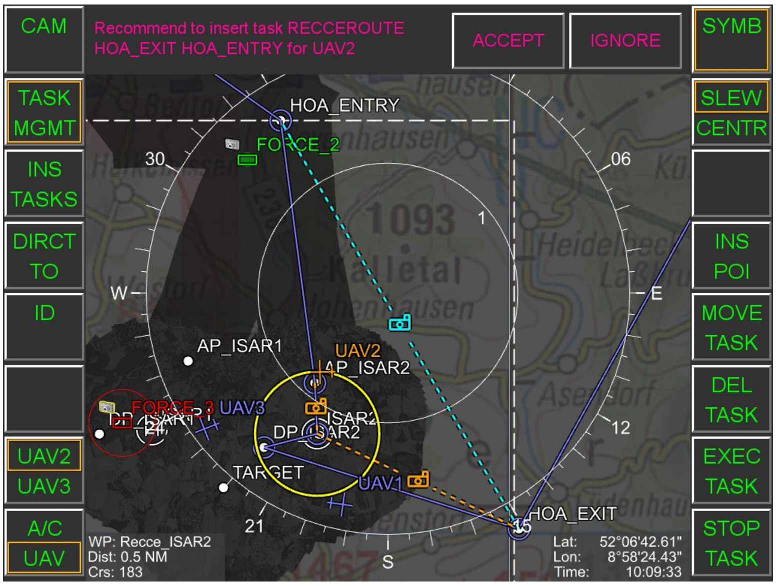

Figure 3-3: Map display covered with UAV-made infrared orthophotos (dark areas)

In order to provide to provide fresh data (vehicles might be moving[18]) on the one hand, and to provide early warning in case of threats, it is the PIC's job to let the UAVs fly in front of the

[18] Although the simulation did not support the movement of ground vehicles, during any experiments this fact was not revealed explicitly.

helicopter and have a distance to it of about five minutes. As depicted in Figure 2-15, the more UAVs recon the same flight leg, the broader is their sensor coverage, and thereby the more thorough is the reconnaissance process concerning the detection of threats. This means, to provide safety for the manned helicopter, which is very important due to the human load in the cargo compartment as well as in the cockpit, it is best to send all three UAVs to recon the same route. However, this is not possible all the time, because, as can be seen in Figure 2-14, there are multiple points of interest located in the operation area (i.e. target of the ground troops, primary and alternate drop zones), which need to be reconnoitered/observed concurrently, and the UAVs therefore disperse geographically during the course of the mission. In addition to that, the reconnaissance of alternative routes can also make sense due to threats that are not known in advance. This gives an impression concerning which decisions the PIC has to make during a MUM-T operation with respect to the tasking of the UAVs.

3.2 Mixed-Initiative Mission Planner Concept

This chapter explains first the development human-automation integration concept for the PIC in the MUM-T setup is explained and then the associated Human-Machine Interface concept.

3.2.1 Human-Automation Integration Concept

In the following, the benefits of giving the human a prominent role in the loop of multi-UAV guidance in the MUM-T setup will be explained.

The author already pointed out in chapter 2.4.4, that the human and the machine have different strengths or skills concerning problem-solving, which should be merged. Also, they have different models of the work domain and the tasks at-hand. In that context, [Sheridan 1992] speaks of “mental models in operator’s head” and “software-based models in the computer”. These models might have different flaws, and by running both models at the same time, these flaws are more likely to be overcome.

Furthermore, even if a fully automated planning engine was able to find an optimal or at least sufficiently cost-efficient MUM-T mission plan in reasonable response time, the human operator would possibly have differing quality criteria in mind as opposed to those formulated in the cost function known to the automated planer. In addition, in working conditions like the MUM-T setup considered here, in which a human operator is working under time pressure, it is not a workable approach to let the human tune a cost function by adding or deleting parameters and changing weights. This can be done in offline planning (e.g. as in [Ryan et al. 2011]), but it is not considered feasible in time-critical conditions as it is the case in the in-flight replanning of a MUM-T mission due to the discovery of new ground-to-air threats.

In addition to that, as long as the human operator is not fully eliminated from the overall system, there is always the need for the operator to check and understand the plan(s) generated by the automation. This is because he/she has the highest authority concerning mission planning and execution due to his/her responsibility for the mission. Therefore, he/she should be involved in certain planning decisions and should always know the resulting mission plan in sufficient detail. In case the machine-generated plan is brittle (e.g. faulty, incomplete, or otherwise insufficient concerning the requirements of the real world) or either the plan or the planning process does not cope with the dynamics of the real world, the operator has at least to be able to intervene. It would be even more beneficial if there was some kind of human-machine cooperation concerning mission planning and execution. In chapter 2.3.6 it has been explained that one important aspect of the Heterarchical Control paradigm is that there is no static function allocation between the human and the machine. In the case of the MUM-T mission planning task, it seems that there is no place for "classical" dynamic function allocation either. This is because dynamic function allocation only works under the precondition that the tasks distributed between human and machine are rather independent from each other. However, even if it would seem at a first glance that planning for example the ingress and the egress routes for a mission have no direct dependencies, there are time constraints and resource constraints (such as fuel[19]), which make a coordination between the planning of each mission phase necessary. The more time is spent for the ingress, the more constrained are the options for planning the egress and the other way around. Another example that supports this argument is referring to the "critical" decisions made during the planning process, such as the selection of a corridor to cross the FLOT or of a landing site. These are critical decisions because they have to be communicated to other allied forces and they can generate a lot of implications (further constraints) concerning the rest of the planning process. Now, if the human-automation integration design let the machine make one of these critical decisions and the human shall decide about the rest, they cannot work independently from each other on the overall problem. And hence, they will always have to wait for the results of each other, which leads to a waste of time, and they may interfere with each other's expectations concerning the outline of the

[19] In the UBM's MUM-T simulation, fuel was not considered, but in real mission, it plays an extremely important role.

plan, thereby causing confusion on either side or even both sides.[20] Furthermore, even if it was possible to determine independent subtasks, their dynamic allocation may cause out-of-the-loop effects for the human operator due to his/her concentration on scattered, disjointed subtasks, which cause losing the overview of the whole operational picture.

As stated in the introductory chapter 1 already, for many (sub)problems it is difficult to foretell, if human or machine will perform better in solving it. As an example, the UAV target clustering task examined by [Malasky et al. 2005] was mentioned. That means, the question, how it can be decided if a (sub)problem shall be allocated to either the human or to the machine, and who shall have the responsibility to decide this, remains open. The only solution is that the human as well as the machine both try to analyze the problem, either on their own or in the best case cooperatively. Problem analysis and problem-solving, which are very much related to each other, should be performed by letting human and automation cooperate. This approach shall be called "*cognitive skill-merging*" [Strenzke & Schulte 2011b]. It will be explained in more detail in the following.

For a closer look at the MMP's Human-Automation Integration concept, a look back at the LOA concept has to be made. First of all, the author has performed small adaptations to the LOAs (see description column in Table 3-1). In addition, a mapping of the LOAs to the corresponding basic principle of human-automation cooperation has been added (see rightmost column in Table 3-1). Whereas LOAs 1, 9, and 10 do not have mentionable cooperation aspects, 2 to 4 can be circumscribed as different forms of decision support through the automation. For this dissertation, LOAs 5 to 8 are the ones that are most interesting to discuss, because they really define different forms of cooperation. The step from LOA 5 (also defined as management-by-consent) to 6 (also defined as management-by-exception) is important, because here the automation receives the right to decide and act automatically. In LOA 7 the autonomy is even higher, but the human is still able to intervene after the automation starts to execute the task. Therefore, the author introduces the term "*management-by-intervention*" for this LOA. Then, in LOA 8, the machine does not automatically inform the human about its decision, which means that the human has to investigate what the machine has done in the more or less recent past. Hence, this can be called "*management-by-investigation*".

[20] One can think of the machine of being confused in case it has to throw away its current plan or its beliefs about how a plan should look like, and it has to replan in a rather lengthy process.

Table 3-1: Levels of Automation matched to human-automation cooperation schemes

LOA	Description [Sheridan 1992]	Human-automation cooperation scheme
1	The computer offers no assistance: human must do it all.	none
2	The computer offers a complete set of action alternatives, and	Decision Support
3	narrows the selection down to a few, or	Decision Support
4	suggests one, and	Decision Support
5	executes that suggestion if the human approves, or	Management-by-Consent
6	allows the human a restricted time to veto before automatic execution, or	Management-by-Exception
7	executes automatically, then necessarily informs humans, or	Mgmt.-by-Intervention
8	informs him only if he asks, or	Mgmt.-by-Investigation
9	informs him after the execution if it, the computer, decides to.	none
10	The computer decides everything and acts autonomously, ignoring the human.	none

The interesting idea that the different problem-solving heuristics of human and machine could be brought together in order to create an overall benefit was already expressed by [Burstein & McDermott 1996]. For example, an important aspect concerning the role of the human in human-machine cooperation is that he/she contributes intuitive and probabilistic reasoning as well as domain expert and background knowledge, all of which are very difficult to either extract or operationalize or implement with sufficient performance into an artificial intelligence system. On the other hand, the machine has certain strengths, like calculating times and costs precisely, reliably and fast. Without doubt, it seems beneficial to add up these strengths.

Modern systems such as the MMP and the MUM-T assistant system shall be conceptually designed to be "human-centered". However, some critique may be applied to the human-centered adaptive automation approach represented by [Hilburn, Byrne & Parasuraman 1997], [Kaber & Riley 1999] and others. In the context of this approach it is postulated that in case of too high operator workload, the LOA should be raised by the adaptive system. However, in the author's view, it is not guaranteed that this will enhance the overall human-machine system's performance, because if the task is transferred to the automation and the automation is not good at solving tasks of this type on its own, the overall result may a worsening of the situation. The setting of policies in advance in order to constrain the system's autonomous behavior (see [Myers & Morley 2003]), may be a key to prevent ad-hoc system reactions that are counter-productive. But the editing of rather complex policies seems not to be a suitable human-machine interface for the PIC in the MUM-T setup, for whom time-critical tasks and small reaction times play a very important role.

This is why the concept of the MMP concentrates on the LOAs that allow for the cooperation of human and machine, allowing them to merge their skills. For this to happen, some form of communication between both partners is required in order to make the human understand what the machine "thinks" it would be a good solution to the planning problem and let the machine get a picture of what plan the human has in mind. The different plan typed will be explained in detail in chapter 4.2. The conceptual approach described here is in line with what [Sheridan 1992] postulates for advanced automation systems: human and machine shall have mutual "mental models" of each other.

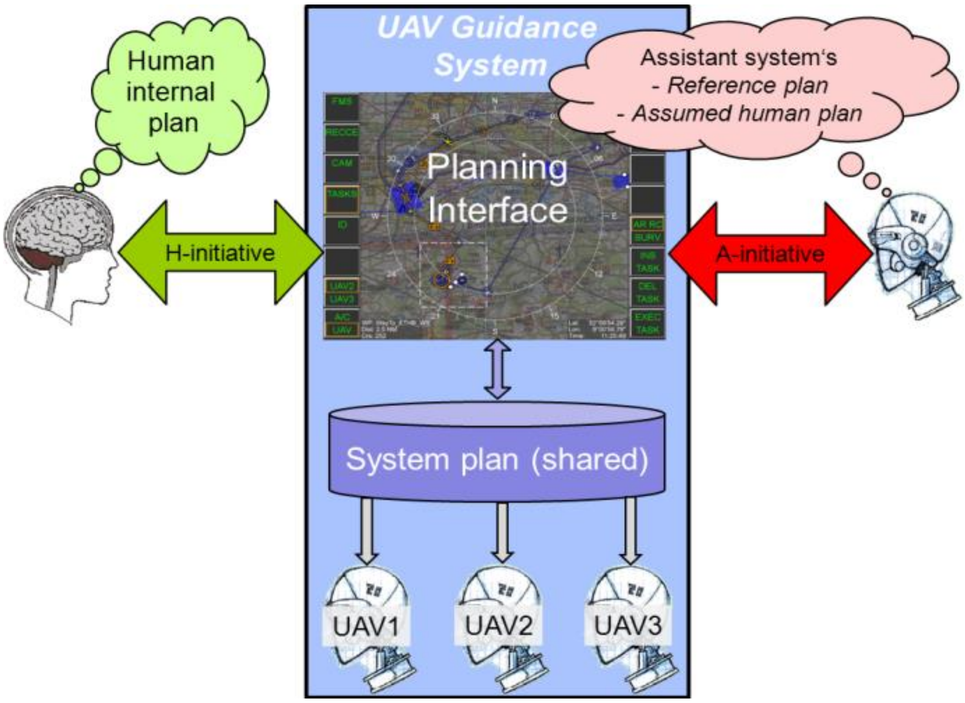

Figure 3-4: Concept for mixed-initiative planning of UAV task agendas

To summarize this, the author's human-automation integration concept for the MUM-T application is to let both the human and the machine reason and work upon the common work objective.[21] This means that the mission planning problem is not divided into subproblems that are then solved by either human or machine. Instead, the two partners reason and work upon the problem individually, but at the same time they communicate about the problem-solving process and thereby influence each other. However, the communication of the human with the machine assistant is rather indirect, as the assistant is not addressed directly, but monitors the

[21] As stated before, the knowledge about the work objective is crucial for true Cooperative Automation.

human inputs into the conventional system. Such kind of machine assistance is proposed by [Onken & Schulte 2010]. It can be circumscribed as a virtual teammate looking over the operator's shoulder.

3.2.2 Human-Machine Interaction Concept

The human-automation integration concept developed in the preceding subchapter lets the human and the machine both analyze and work upon the same task and communicate about it. Therefore, viewing the interaction from a mixed-initiative standpoint seems well suited. The reason for this is that both partners need to be able to take initiative concerning problem-solving and communication. This subchapter explains in detail, which mixed-initiative interaction style (see chapter 2.3.6 has been chosen for the MMP.

As explained before, the MUM-T assistant system is designed according to the basic assistant system requirements [Onken & Schulte 2010] [Rauschert & Schulte 2012], which include that the human is not able to initiate a dialog with the system. Instead he/she works with his tools (i.e. the UAVs), which is tantamount to problem-solving initiative. Again, according to the basic principles of assistant systems, the machine takes initiative to aid the human as soon as erroneous behavior of the human either is expected or has already occurred. This can either take place by the assistant system's initiation of a dialog or by its initiation of a problem-solving related action.[22] Because such an assistant system has the possibility to plan the mission by itself and to aid the human on its own initiative, it can be described as following an assistance-style mixed-initiative planning concept [Strenzke & Schulte 2012a]. This is depicted in Figure 3-6. It appears to be a good choice amongst the alternative interaction styles, because, first, changes in the environment can lead to the necessity of initiating dialogs with the human, which is not in focus of the delegation style. Otherwise, the machine would have to react and replan autonomously. Hence, pure delegation style is not a very cooperative approach. Second, the dialog style seems not well-suited for online (re)planning setups, where the human has to react quickly to a changing environment, because the authority over the goal determination and problem-solving process might first have to be negotiated in a more or less time-consuming process. Hence, the MMP was designed as a cooperative system that does not bind too many of the human operator's resources on dialogs, and therefore implemented the assistance interaction style with simple dialogs, as shown exemplarily in chapter 3.1.2. This example also shows the

[22] The direct plan manipulation by the assistant system is technically possible, but it has been disabled for the experimental evaluation described later in this dissertation.

three different *initiative strengths*,[23] which have been realized for the MUM-T assistant system: advice, proposal and task reallocation. The initiative strength rises during the dialog.

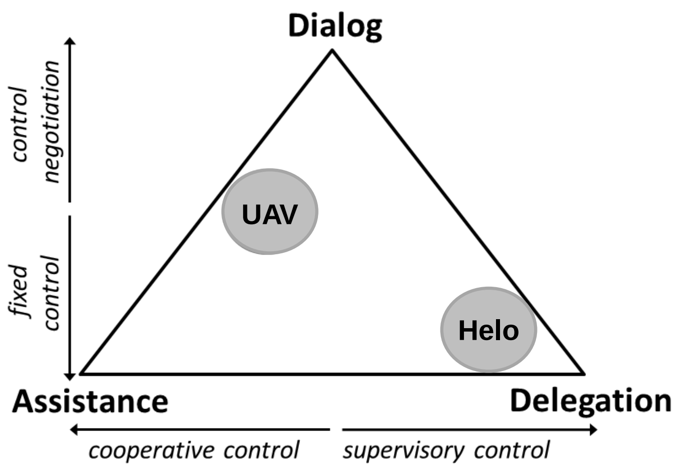

Figure 3-5: MMP interaction styles for the planning of UAV and helicopter tasks

However, the overall concept of the MMP as a mixed-initiative system shall also allow the human to take initiative. Therefore, the MMP shall have two modes of supporting the UAV operator.

The first one is the *passive mode*, which is realized through the MUM-T assistant system: the assistant system supports the UAV operator by interacting with him/her on its own initiative solely in case he/she shows suboptimal behavior or when there is a situational change that has a huge impact on the mission, which is not likely to be manageable by him/her. If the human operator has to add details to the plan or has to trigger the execution of a task by a UAV in order to stay in schedule, a dialog is initiated after a time threshold has passed. This means that if the human performs his/her tasks in time, no initiative is taken by the system. In urgent mission-critical cases however (e.g. necessity for replanning due to threat discovery or reception of a follow-up order), the MUM-T assistant system initiates a dialog with the human immediately. The assistant system behavior is explained in detail by [Rauschert 2013].

This passive behavior is in accordance with the assistant system requirements from [Onken & Schulte 2010]. The functions for which the assistant system uses the MMP are:

- Aiding the operator with respect to inclomplete UAV plans by proposing to add a task

[23] See chapter 1.2.3.6.

- Aiding the operator with respect to overdue UAV tasks by proposing to activate a task
- Aiding the operator with respect to complete replanning of the mission

The second mode is the *active mode*. It is triggered when the cockpit actively crew uses the manned helicopter's mission management system to plan or replan the mission. Because the crew uses the planner as a tool, this mode can be described as Hierarchical Control.

It is important to note that the active mode shall only be used for updating the route and flight log of the manned helicopter; the UAV tasks shall not be modified in this case. The main reason for this is that the change of the route of one aircraft does not confuse the operator's situation awareness as much as changing the routes (or plans) of all four aircraft would. A further reason is that the PF shall also be able to use the active mode for replanning the helicopter route, and the PF should not be able to modify the UAV task agendas, because he/she has neither any awareness nor any responsibility concerning these. In addition, the MMP shall support both the PIC and the PF via delivering data for the manned helicopter's flight log display.

After all, the MMP implements the following functionality:

- UAV plan completion by
- Complete replanning, due to change in mission goals or major change in situation
- Planning / replanning of the helicopter route upon human initiative

3.2.3 Human-Machine Interface Concept

In the following, the human machine interface (HMI) concept developed for the MMP is described. This HMI is the PIC's interface towards the MMP. It contains parts of the assistant system concept developed by [Rauschert 2013].

Figure 3-7 shows the generic helicopter cockpit developed and used for MUM-T and MiRA-T experimental campaigns. It is fitted with two configurable touch screens (the MHDDs) for the PIC, who is sitting on the left side, enabling him/her to guide the UAVs (left map display) and analyze their sensor data (camera display). In addition to that, an MCDU is available to plan the helicopter mission, among other helicopter management functionalities. For the sake of completion, for the PF, there are four touch screens (two on the right side and two in the center center), another MCDU and a flight stick.

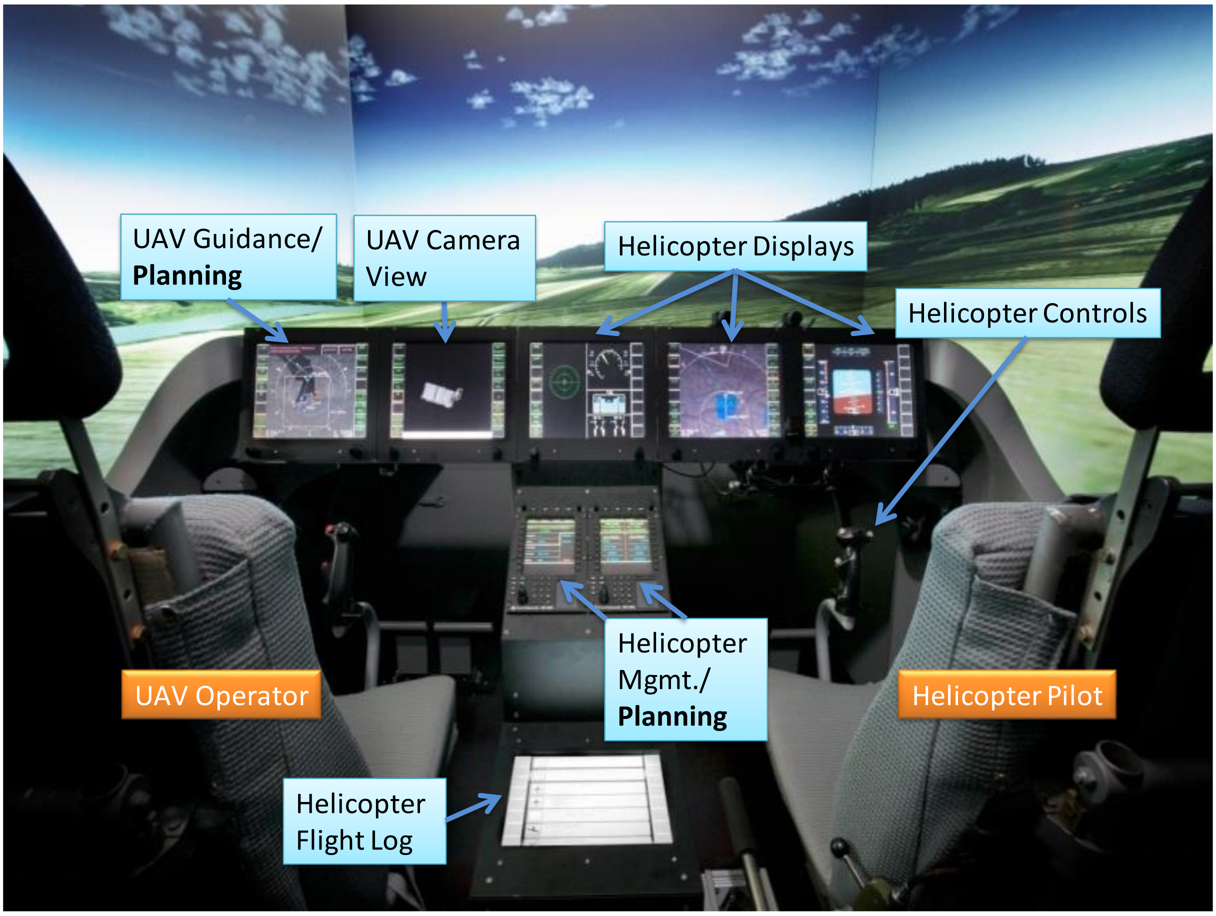

Figure 3-6: Manned-Unmanned Teaming helicopter simulator cockpit

There are mainly three possibilities to initially plan the overall MUM-T mission, as shown in Table 3-2. The helicopter and UAV task planning processes will be described in the following subchapters. Afterwards, the UAV mission execution is explained.

Table 3-2: Possibile MUM-T mission planning workflows

Possibility	Order	Helicopter Task Planning	UAV Task Planning
1	first helicopter, then UAV[24]	constraint-based, human initiative	manual[25] or mixed-initiative
2	first UAV, then helicopter	constraint-based, human initiative	manual or mixed-initiative
3	both at the same time	fully automated by assistant system initiative	

[24] Interleaving of the two processes is possible in principle, but changes to the UAV tasks can lead to a restarting of the helicopter planning process. This is because the MMP tries to optimize the cooperation of the aircraft, and the helicopter route and tasks depend on where the UAVs are going to recon. There is a double-sided dependency, because the helicopter should fly along routes reconned by one or more UAVs, and for the same reason the UAVs should recon the planned helicopter route.

[25] The intelligent functions onboard the UAVs aids the operator in completing the task agenda in case he/she left out necessary tasks in between (either by accident or because he/she intends to transfer the responsibility of planning intermediate tasks to the automation). Therefore, the UAV task planning is never completely manual.

Helicopter Task Planning

In order to plan the tasks for the manned helicopter, the cockpit crew is allowed to load (via the UPLOAD function, see Figure 3-8) the mission goals (e.g. squad A shall be at TARGET) as well as default constraints (e.g. use TANGO as preferred ingress corridor). The crew may modify these constraints (e.g. switch to a different ingress corridor) or delete them (e.g. no specific ingress corridor), and plan/replan on command (via the REPLAN function).

After the first solution is displayed in the map, the crew is able to access more optimized plans as soon as they are available. This is signalized in the MCDU GUI,[26] and the OPTIMIZE function can then be invoked. This process can be regarded as *management-by-intervention* (see chapter 3.2.1), because the system automatically generates a plan and, which will then be followed (by the PF), unless the crew intervenes by pressing either OPTIMIZE or REPLAN.

Figure 3-7: MCDU page for helicopter mission planning

UAV Task Planning

As described above, there are multiple ways to generate tasks for the UAVs. They can be entered *manually* (normally sequentially) by means of a graphical user interface (see Figures 3-9 to 3-12), which only allows the incremental generation of a plans (one per UAV). The manual planning process includes that, when tasks are left out and these are necessary in order to follow common military flight procedures, these tasks are added to the agenda automatically

[26] The availability of the OPTIMIZE function is signalized by the highlighting of the OPTIMIZE button instead of being greyed out. In Figure 3-9 this button is not available. The corresponding slot is labeled "---", instead, because the planning process is still not finished. As can be seen, the state is "Planning…"

be the intelligence onboard the UAVs. This is communicated to the operator by means of the map display, which he/she also uses for UAV task input (cf. Figure 3-3).

Furthermore, the MUM-T assistant system is able to support the PIC in planning the UAV tasks. The PIC is then able to add tasks to the UAV plans by the *acceptance of assistant system proposals concerning the appending of single UAV tasks*. The UAV agenda is not necessarily complete, i.e. it does for example not include the egress path and/or landing at the home base. The assistant system is able to propose the appending of tasks with a certain time buffer for the human to make a decision about this proposal. Figure 3-10 shows such a proposal. This human-automation cooperation method can be categorized as *mixed-initiative planning* and *management-by-consent*.

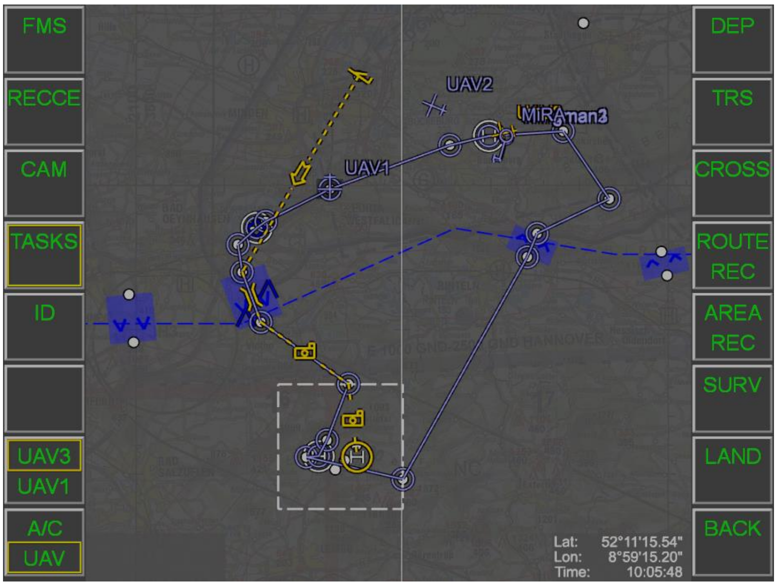

Figure 3-8: MHDD page for mission planning (helicopter route in blue, partial plan for UAV3 in orange)

The third way to generate tasks for the UAVs is accepting the proposed plan made by the assistant system. This case is shown in Figure 3-11. Such a plan includes all aircraft, i.e. the UAVs and the manned helicopter, and it is complete, including landing of all aircraft at the home base . However, this assistance is offered only in special cases, because the assistant system should not take away the PIC’s work, which could kick him/her out of the loop. One such case is the violation of the mission goal, i.e. the current plan does not lead to the fulfillment of one or more of the hard mission goals. This violation occurs directly after loading the mission data, which includes the mission goals, into the helicopter computer, and as soon as the data concerning the follow-up mission is transferred to the computer via datalink. Another case is when the helicopter route becomes threatened by hostile ground forces in such a way, that a mission-level replanning has to take place (i.e. a planned landing site is threatened or a planned corridor). Again, the type of cooperation corresponds to *management-by-consent*.

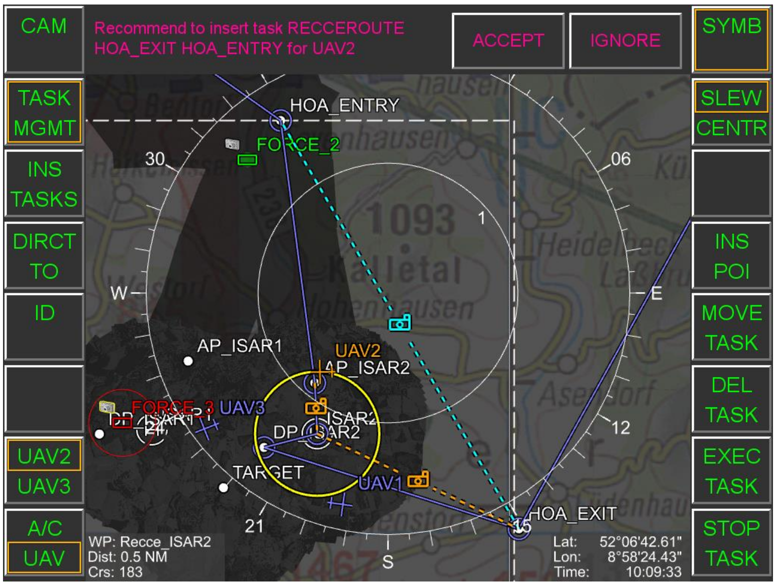

Figure 3-9: System proposal concerning the appending of a single UAV task (marked in turquoise)

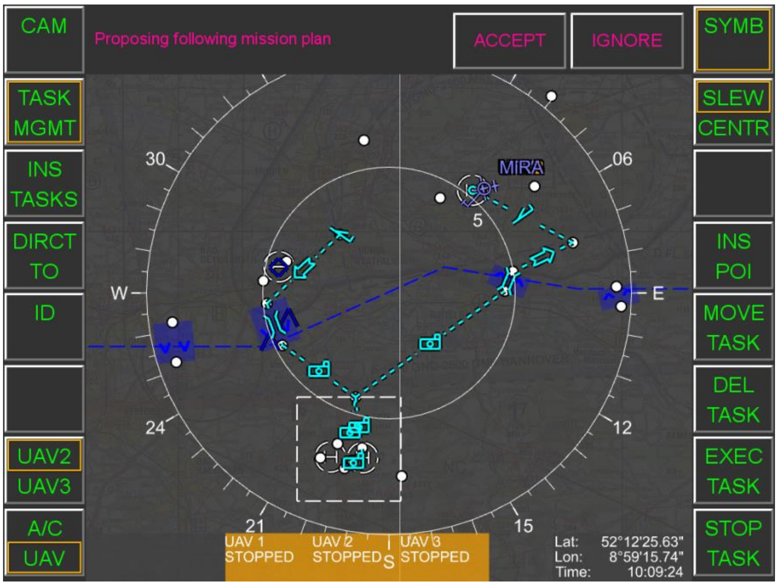

Figure 3-10: System proposal of a complete nission plan (UAV2 plan in turquoise, others to be toggled)

UAV Mission Execution

As described in chapter 3.1.2, the MUM-T assistant system has knowledge about when which task should be executed. This depends on temporal constraints (e.g. takeoff clearance time) as well as on the durations of the preceding tasks. By use of this knowledge, the assistant system is able to monitor the plan execution, and in case the PIC has to activate a task manually, which counts for the takeoff and all tasks directly following an area reconnaissance or object surveillance task, it is able to propose the activation of the corresponding task in *management-by-consent* style. The corresponding mixed-initiative dialog between the assistant system and the PIC is displayed in Figure 3-12.

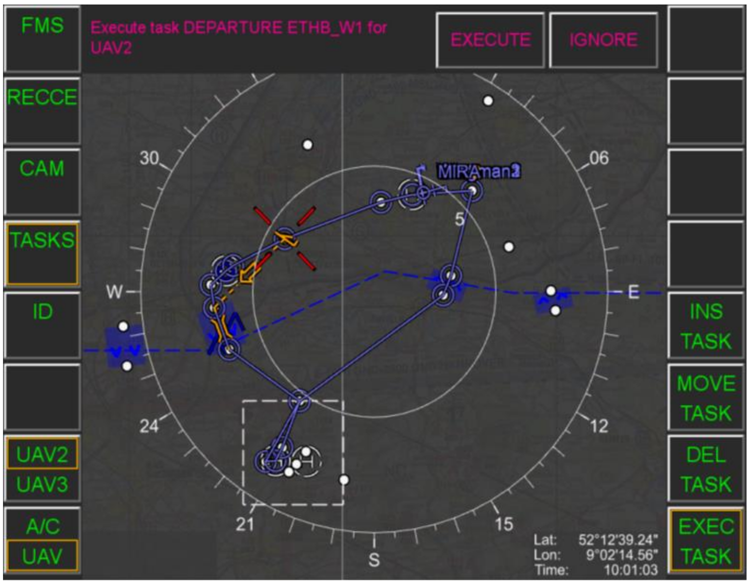

Figure 3-11: MUM-T assistant system proposal concerning mixed-initiative execution

4 Mixed-Initiative Mission Planner Design and Implementation

In the following, first the requirements concerning the MMP are listed, then its design is explained, and finally an overview of the implementation is given.

4.1 Mixed-Initiative Mission Planner Technical Requirements

This subchapter gives an overview of the technical requirements for the MMP. These stem mainly from the MUM-T application domain described in chapter 2.5 and from the MUM-T assistant system concept created by [Rauschert 2013] (see also chapter 3.1.2).

4.1.1 Application-Domain Related Requirements

In order to enable an OCU to derive tasks, subtasks, and/or atomic actions from the work objective, as it is necessary for the MUM-T assistant system described above, the OCU has to be able to perform planning towards reaching the work objective. In the conceptual approach chosen for the MUM-T setup, the MUM-T assistant system can be regarded rather as the dialog and decision driving front-end close to the human operator, whereas the MMP, which is the software component this dissertation focusses upon, is regarded as the problem-solving backend enabling the overall mixed-initiative operation process. Based on the output of the MMP, the MUM-T assistant system can make decisions concerning what and when to communicate to the helicopter commander. More details concerning the MUM-T assistant system itself can be found in [Rauschert 2013] and [Rauschert & Schulte 2012].

- In order to support the MUM-T assistant system, the MMP needs plan reasoning capabilities that include first of all the ability to *generate plans*. This leads to the necessity of having a world model that is sufficient to plan MUM-T missions. **(Req. 1)**
- The aspect of *temporality* is important to fulfill the requirements of the time-constrained MUM-T mission as described in chapter 2.5.2. Hence, the MUM-T world model of the MMP has to include a conception of time. The duration of the actions may not be fixed but depending on parameters like distances and speeds. **(Req. 2)**

- Because of the criticality of every action or decision in a military mission, any proposed plan must be feasible. This means that the landing of the manned helicopter in safe territory must take place before time runs out. This leads to the requirement that a mission plan always has to be *complete*, i.e. it includes reaching the goal state of the planning problem. **(Req. 3)**

- Also, timely coordination is necessary (e.g. site has to be reconnoitered before the landing of the manned helicopter). Therefore, all agents have to be included in a *single overall mission plan*, **(Req. 4)**

- and consideration of *concurrent actions* is necessary. **(Req. 5)**

- Furthermore, the solution process and end state are not well-defined, i.e. there are multiple possible ways of solving the problem and these are not known in advance. Hence, optimality criteria are needed, which can be merged together in a *cost function*, which is to be minimized. This function shall represent the minimization of risk to human life (i.e. the helicopter crew), risk to equipment (i.e. risk to the manned and unmanned aircraft), violation of the mission order, as well as financial costs (i.e. short flight paths). The basis for this is a *metric planning* functionality. **(Req. 6)**

- In the MUM-T scenario, movement of any dynamic objects not under control of the human (e.g. ground vehicles) has been disregarded. The helicopter is also under his/her control (constraint-based guidance, e.g. which corridor and landing site to use) because in the MUM-T scenario the UAV operator and the helicopter commander is the same person. There are no actions with an unpredictable outcome (like firing a weapon and not knowing if it will hit and destroy the target). Hence, the mission planning problem is *deterministic*, and therefore a deterministic planner is required. **(Req. 7)**

- In total, there are about 25 mission-relevant locations in the scenario. These include the ground vehicle objects, which are representable as static locations. This data is necessary for the *performance* requirement of the MMP, as it has to be able to handle the amount of action possibilities generated from the mission-relevant locations **(Req. 8)** and evaluate the distances between locations by temporal (cf. Req. 5) and metric planning (cf. Req. 6).

4.1.2 Assistant-System Related Requirements

In order to decide whether, when and how assistance should be provided to the operator, the MUM-T assistant system has to be able to anticipate, which actions the operator has to execute and when he/she is supposed to do this. The component responsible for this anticipation process is the MMP.

As described in chapter 3.1.2, the assistant system has the sole responsibility in supporting the PIC in performing the UAV guidance, i.e. the planning and execution of the UAV mission. Concerning UAV mission planning, the actions of the human operator are the entering of new tasks for the UAVs. Concerning UAV mission execution, his/her actions are the activation of UAV tasks.

- Anticipating that a task has to be entered by the human operator requires a *plan completeness-checking* ability. **(Req. 9)**
- Deciding which additional UAV tasks have to be entered is necessary in cases where the plan of a UAV is incomplete[27]. Hence, this functionality requires a *plan-completion* feature from the MMP, i.e. the MMP has to be able to plan the mission until the fulfillment of all hard goals under consideration of the partial plans that the human operator has already entered. **(Req. 10)**
- The decision, when an advice concerning a missing UAV task has to be given, depends on the time at which the task is scheduled. Hence, the MMP has to be able to perform *temporal planning*. Which task has to be activated and when a corresponding advice shall be given is an analogous scheduling question. This requirement is covered by Req. 2 already.
- During the execution of the MUM-T mission, the MUM-T assistant system has to check the feasibility of the operator-given UAV tasks upon any relevant tactical situation change (e.g. new threat enters the scenario) any mission order change (i.e. new mission objectives received or mission objectives have already been met) and any operator input that is conflicting with the current plan. During such situations, the assistant system needs to support the human operator in *real-time*. In order to avoid too high delays in dialog instantiation, the MMP has to deliver *suboptimal solutions* as quickly as possible. It therefore needs *anytime planning* (suboptimal planning) capability. **(Req. 11)**

[27] Because in the MUM-T scenario the mission briefing prescribes that all aircraft have to be returned to the home base at the end of the mission, every UAV plan that does not include the return path to the home base is incomplete.

- The assistant system also needs the ability to propose a new complete plan to the operator in case his/her plan is incomplete or faulty. This results in the requirement for a *plan-generation* functionality, which has to consider all agents in the MUM-T scenario, i.e. the helicopter, the UAVs and the ground troops. This feature is also needed for supporting the UAV operator at the very beginning of the mission, before he/she has entered any UAV task. This is covered by Req. 1 and 4 already.

4.1.3 Other Requirements

The rest of the requirements for the MMP come from the MiRA-T assistant system and from the mission management system of the manned helicopter, which can be operated by either the PF or the PIC via their MCDUs.

The PIC must be able to upload mission data into the helicopter's mission management system, i.e. mission-relevant locations and times as well as briefed constraints, as listed below:

- preferred drop zone
- preferred ingress and egress corridors
- mandatory pickup zone
- corridor opening times
- takeoff clearance time (i.e. earliest mission start)
- final destinations of all aircraft and troops

In this context, "preferred" means that the MMP needs to consider the corresponding constraints as *soft constraints*. All these *constraints* are predefined for the mission in accordance with the information from the mission briefing cannot be defined or redefined by the PIC during the mission execution. However, they may change or be extended, when a follow-up mission is commanded by the external mission control.

Furthermore, in order to perform his/her mission planning task, the PIC has to have the possibility to set the following constraints dynamically, which are to be considered as mandatory (i.e. *hard constraints*):

- pickup zone to be used (for each transport mission)
- drop zone to be used (for each transport mission)
- ingress corridor to be used
- egress corridor to be used
- home base to be used as return location

Although costs are already covered by Req. 6, the dealing with soft constraints as input for the MMP is a new requirement. **(Req. 12)**

In addition, the helicopter mission management system shall support the crew in executing the mission by calculating a flight log for the helicopter, which includes also a communication plan for the manned helicopter (see chapter 4.2.6). Important for supporting the crew in mission execution is the display of the order of the helicopter/crew actions and the associated starting time and duration for each action. Again, temporal planning becomes a requirement for the MMP. This is covered by Req. 2 already.

As described in chapter 3.2.3, the PIC shall be able to optimize the flight plan of the manned helicopter, in case he/she finds it to be suboptimal on a subjective level. The plan optimization can be solved by *anytime planning* functionality, where incrementally improving plans are put out over time. These improved plans shall be accessible on demand. This is covered by Req. 11 already.

4.2 Mixed-Initiative Mission Planner Design

Regarding the requirements stated in the preceding subchapter, it is obvious that at least one automated planner is needed in order to be used by the MUM-T assistant system as well as by the helicopter mission management system. This automated planner shall be part of the MMP.

To fulfil the mentioned requirements, the MMP shall process the as-is state (i.e. the current tactical situation) and a set of constraints, which includes the desired goal state as well as further limiting constraints. On the one hand, these constraints stem from the mission order, and on the other hand they stem from the human operator’s input (UAV tasks and helicopter mission constraints entered via UIs). The result of the processing shall be a valid and evaluable temporal mission plan for each MUM-T agent, i.e. a series of operations that are associated with times and costs.

The remainder of this subchapter first describes which plan instances and planner instances are necessary in order to perform this processing. Then, the MMP's input and output interfaces are detailed. After that, the cost function for the evaluation of plans generated by the MMP is developed. Finally, the design of the user interfaces related to the MMP is described.

4.2.1 Plan Instances and Planner Instances

As already described in chapter 3.1.3, the task-based UAV guidance UI allows generating and managing a single mission plan per UAV. If one aggregates all these mission plans, an overall (multi-)UAV mission plan becomes visible, which should correspond to the overall UAV mission plan that the human operator has in his/her mind. Even if he/she has multiple alternative plans in his/her mind, only one of these is the primary plan that he/she actually enters into the UAV guidance system.

However, it is important to distinguish between the *plan(s) in the human mind* (*human plan – HuP*) and the *plan that is stored in the robotic system* (*system plan – SyP*), i.e. the currently active plan for the automatic UAV guidance. This is because the HuP may be a complete plan, whereas the SyP may still be incomplete due to several reasons.[28] Furthermore, the HuP may change, while the SyP is still unmodified, because the modification of the SyP takes time. It is not possible to get hold of the HuP directly, and hence the MUM-T assistant system can only evaluate the SyP.

Plans generated by the MUM-T assistant system (*assistant system plans – AsP*) constitute a third type of plans in the MMP design. Figure 3-5 shows the relations between all these plan types. Both human and machine shall be able to take initiative (problem-solving initiative) in order to manipulate the system plan according to their understanding. The assistant system plans (AsP) can be further divided into *what the machine computes as the best possible plan* (*reference plan – ReP*) and *what the machine supposes that the human is planning* (*assumed human plan – aHuP*).[29] The assumed human plan (aHuP) converges to the true human plan (HuP) with each additional detail the human discloses by tasking the UAVs and thereby expanding the SyP. Furthermore, the operator is driven to detail his plan by the warnings and

[28] Either the human operator may not want to enter the complete plan at this time or he/she did not manage it yet to enter the desired plan completely.

[29] In theory, multiple plans of each subtype can be stored by the assistant system, but this is not regarded in this dissertation.

proposals of the assistant system, possibly letting the HuP and SyP converge to either the aHuP or the ReP.

In order to calculate different plans, there shall be two instances of the automated planner in the MMP design. Both instances shall receive the information about the current tactical situation, but they differ with respect to the constraints they take into consideration.

1. The so-called *Slave* instance of the MMP is slave to the human input, i.e. it uses the constraints expressed by the SyP (aircraft tasks) and the mission order to check the feasibility and completeness of the SyP. If the SyP is feasible, the assistant core receives the start times and durations of the tasks that were calculated by the MMP, which is needed for monitoring the execution of already planned tasks. In case the SyP is incomplete (partial), the missing tasks will be added by the MMP, thereby assembling the aHuP, which allows the MUM-T assistant system to monitor if the operator evolves the plan early enough to stay in schedule.
2. The *Free* instance of the planner is responsible for the generation of the ReP. It receives only the mission order constraints, i.e. it is meant to disregard the SyP completely. Thereby, it checks if the problem is solvable in general, and in this case it generates a complete Free plan (i.e. the ReP), which can then be compared with the best scoring Slave plan (i.e. the aHuP) by the assistant system (see Evaluation chapter). This comparison reveals if the human operator entered some elements to the SyP which might be suboptimal or even counterproductive and can therefore be used by the assistant system as basis for the decision whether to offer the ReP as the new SyP to the human.

As depicted in Figure 4-1, during the execution of the mission, the assistant-system core receives information about the mission order, the current tactical situation and the aircraft task agendas out of SyP. From this information, the aHuP as well as the ReP have to be generated. In order to accomplish this, two different constraint sets have to be transferred to the different planner instances. In both cases, the MUM-T assistant system uses the Simple Temporal Constraint Interface (STCI) of the corresponding mission planner instance.

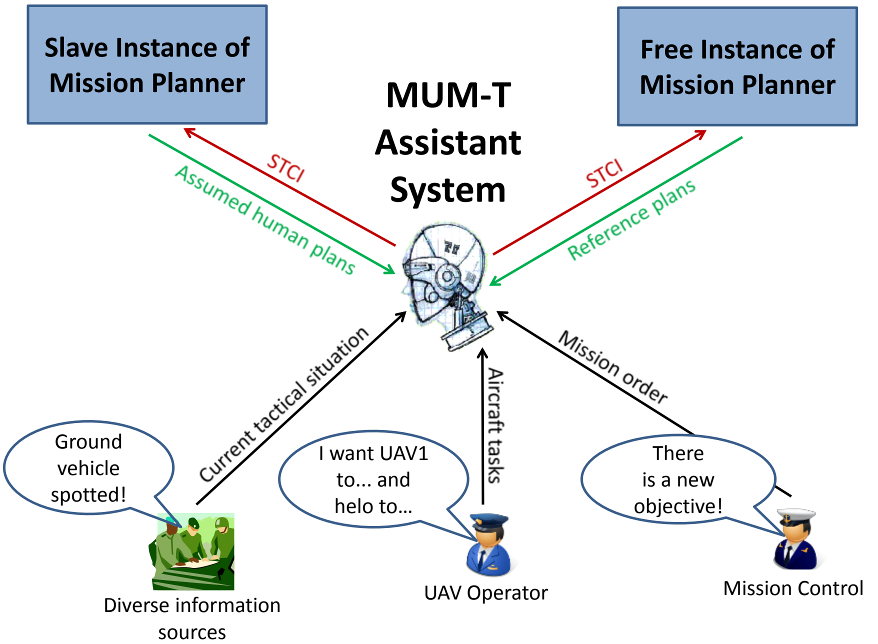

Figure 4-1: Integration of the MUM-T assistant system core and MMP instances

4.2.2 Constraint-Based Interaction Concept

Similar to [Miller et al. 2005], the author's concept to human-machine cooperation follows a *shared task model* that allows human and machine to communicate about tasks for the aircraft, goals and plans. This means that the user interface accepts human input on a convenient cognitive level, and at the same time the machine is able to reason on an identical abstraction level. To be more specific concerning the concrete design, the input of the human's (partial) UAV plans can be regarded as constraints to the planner. With hard constraints, he/she prunes the search space of the machine and with soft constraints he/she guides the search of the machine via its cost-based heuristics. In the other direction, the machine is able to give hints or advices to the human operator, e.g. about actions missing in the plan that are needed to cover mission goals, because it is good at plan checking (e.g. precondition checking, cost and time calculation) as well as maintaining and evaluating multiple plans. Thereby, the MMP in turn influences the problem-solving process of the human. This leads to merging of both partners' strengths in the mixed human-machine team in a way of generating synergy [Strenzke & Schulte 2011b].

In the task-based guidance approach, which is used in the UBM's MUM-T setup for guiding the UAVs in a supervisory control manner, the SCUs onboard of the UAVs have to plan and execute the tasks ordered by the human operator. The details can be planned by the UAV SCUs, e.g. flight path and coordination with the other UAVs, but the mission plan, i.e. the solution to the overall MUM-T mission planning problem, has to be created by the human operator. In contrast to this, the MUM-T assistant system OCU, which knows the work objective (i.e. the goal of the MUM-T mission), is able to perform the mission planning process itself (by means of the MMP). Thereby, the human-machine team, consisting of the PIC and the assistant system are able to plan the mission in mixed-initiative fashion. During this mixed-initiative problem-solving process, the human and the machine may interact on different levels of the planning process. This means that they are able to refer to goals, constraints, or plan elements (i.e. complete or partial plans or single actions). It is important to note that goals and plan elements can be regarded as constraints as well. This allows creating a simple but universal language for different interaction levels.

[Miller 2005] proclaims five delegation types for unmanned military vehicle guidance, which can be seen as mixed-initiative interaction levels. These are:

1. Operator provides goal states to the system
2. Operator provides a partial or complete plan
3. Operator provides negative constraints concerning either states or actions
4. Operator provides positive constraints concerning either states or actions
5. Operator provides a policy (cost function)

The author's approach is to recategorize these mixed-initiative interaction levels from a different perspective which stems from the general approach of Automated Planning (e.g. [Gerevini & Long 2005]), which takes into account, first, the hard constraints that must apply to any valid solution and, secondly, the soft constraints, which end up in a cost function and allow to evaluate the quality of any valid solution. From this viewpoint, every interaction with the system can be seen as providing, modifying or deleting a constraint. Meta-commands, e.g. for plan management,[30] are disregarded here. Such a constraint

- can refer to a certain action or state (content),
- can be either positive or negative (sign),
- can be either hard or soft (hardness) and

[30] Examples for plan management functionality can be found in [Allen & Ferguson 2002].

- can be time-constrained or not or refer to the goal state (temporality).

With this generalized four-dimensional constraint approach, all of the aforementioned delegation types can be realized, as shown in Table 4-1.

Table 4-1: Realizing human-automation delegation types by parameterizable constraints

Delegation type [Miller 2005]	Constraint content	Sign	Hardness	Temporality
1	state	+ or -	hard	*goal*
2	action	+	hard	possibly time-constrained
3	state or action	-	hard	any
4	state or action	+	hard	any
5	state (could also include actions)	+ or -	soft	goal (could also be any)

4.2.3 Simple Temporal Constraint Interface

The STCI has been developed as an interface for the MUM-T assistant system core to the MMP's mission planner instances as a way to transfer sets of planning constraints (as depicted in Figure 4-1). The set is simply a variable-length list, in which the order of the entries plays no role. Each constraint in this set is made up of the following attributes:

- Agent to which it is referring, or named group of agents
- Content type (which *action* or *state* it refers to)
- Objects (one or two objects, e.g. locations, to which the action or state refers)
- Temporality (with the possible values: anytime, at begin, at end, not before, not after)
- Hardness (*soft* or *hard* constraint)
- Costs (only relevant for soft constraints)

Which agents are possible as values depends on the scenario. In the default MUM-T scenario, the agents are the helicopter, the three UAVs and two ground troop objects (one for the primary troop transport mission and one for the secondary follow-up mission). The agents are referred to by their name defined in the tactical situation status. It is also possible to select all UAVs or all aircraft by specific group names.[31]

Each constraint has as content either an action to be performed or a state to be reached by an agent. Furthermore, different parameters need to be specified depending on the constraint content type (see Table 4-2).

Table 4-2: MMP constraint content types with corresponding parameters

Constraint content	State constraint	Action constraint	Parameter "Agent"	Parameter "TargetLoc"	Parameter "StartLoc"	Parameter "TargetAgent"
"be at ground position"	X		X	X		
"be at air position"	X		X	X		
"transit"		X	X	X	X	
"cross FLOT"		X	X	X	X	
"recce route"		X	X	X	X	
"recce area"		X	X	X		
"object surveillance"		X	X	X		
"take-off"		X	X	X		
"land"		X	X		X	
"load troops"		X	X	X		X
"unload troops"		X	X	X		X

To generate multiple (and also open) time windows the temporal specificators for constraints are:

- "at beginning"
- "at end"
- "anytime"
- "not before"
- "not after"

The latter two are associated with a single time value. "Anytime" means the task has to be done or the state has to be reached at any point in time, which is not specified. The specification of a half-open interval is possible with the addition of either a "not before" or "not after" constraint with the "anytime" constraint. Furthermore, a closed interval can be defined by combining a

[31] The group feature is only for convenience. It would make no logical difference, if a list of constraints for every single agent in this group was sent instead.

"not before" with a "not after" constraint. An "at end" constraint specifies a goal state for the planner (non-temporal) and "at-begin" constraints are needed to model tasks that are already in progress at the time of planning and therefore can be finished before the agent starts executing any other task.

Each constraint can be specified either as hard (i.e. mandatory) or soft. In the latter case, the constraint is associated with definable violation costs. This means the constraint may be violated by the MMP's solution, but the plan becomes more expensive.

4.2.4 Plan Output Interface

The plans generated by the MMP have to be passed back to the MUM-T assistant system core. For this purpose, the MMP has a plan output interface. Each message sent over this interface contains a list of valid plans (valid concerning the last set of constraints sent over the STCI), which are ordered by their optimality, i.e. their costs. Each plan stores the associated total costs, total makespan and a list of action items. An action item is composed of

- the starting time,
- the duration,
- the agent,
- the first object and
- the second object.[32]

4.2.5 Cost Function Design

The performance of the overall MUM-T system is evaluated by considering mission briefing violations and risks, such as the helicopter's exposure to (potential) threats. This can be formulated by a cost function as shown in Equation 1, which adds the sum of the reconnaissance-dependent helicopter flight leg costs to the sum of the mission order and threat-safe landing violation costs, where k is a constant that is being reduced by the number of UAVs u_i that scanned the leg i with the length l_i. Cost factors (c) are flight costs in hostile territory and the different violation costs (e.g. landing at a threatened or unreconnoitered site, landing site not briefed as primary choice, or use of designated HOA entry/exit points).

[32] For the MUM-T domain, two objects are sufficient (e.g. starting location and target location of a flight leg).

$$f = \sum_{i=1}^{n} (k\text{-}u_i) l_i c_{flight,foeterr} + \sum_{j=1}^{m} c_{violation,j} \quad (1)$$

Furthermore, in addition to Equation 1, every movement of the agents generates costs (independent from hostile or friendly territory). Except for the ground troops, these costs are rather low compared to the abovementioned ones.

4.2.6 User Interface Design

This subchapter gives more detail concerning the UI design, which is based on the work of [Rauschert 2013] as far as the MUM-T assistant system is concerned and of Benzler (in [Strenzke et al. 2011]) as far as the flight log display is concerned.

The planning process for the helicopter tasks in the MUM-T troop transport mission is performed by using the MCDU depicted in Figure 4-2 as UI. In order to plan the tasks for the manned helicopter, the cockpit crew is allowed to load (via UPLOAD button) the mission goals (e.g. squad A shall be at TARGET) as well as default constraints, e.g. use corridor TANGO for ingress. The crew may modify or delete these constraints (e.g. no specific egress corridor in Figure 4-2) and the mission management system is then able to plan or replan all task details as long as the given constraints are met. This is why this can be seen as constraint-based mission planning.

Figure 4-2: MCDU page for planning the helicopter mission

The helicopter task agenda, which is generated during the planning process, includes the following tasks:

- takeoff
- departure
- flight/transit (with different speeds in friendly and hostile territory respectively)
- approach
- land
- load troops
- cross FLOT
- unload troops

Figure 4-3 shows the flight log display, which can be accessed by both the PIC and the PF. Each agenda item is tagged with a start time and duration. Typically, the initial planning at the beginning of the mission spans about 25 actions for the manned helicopter. Due to the required anytime-planning capability of the MMP, the crew is able to access more optimized plans as soon as they are available (i.e. OPTIMIZE button is highlighted and may then be pressed). In addition to the helicopter tasks mentioned above, the flight log contains crew communication tasks, which are generated by the Communication Planner, which is part of the MiRA-T assistant system.

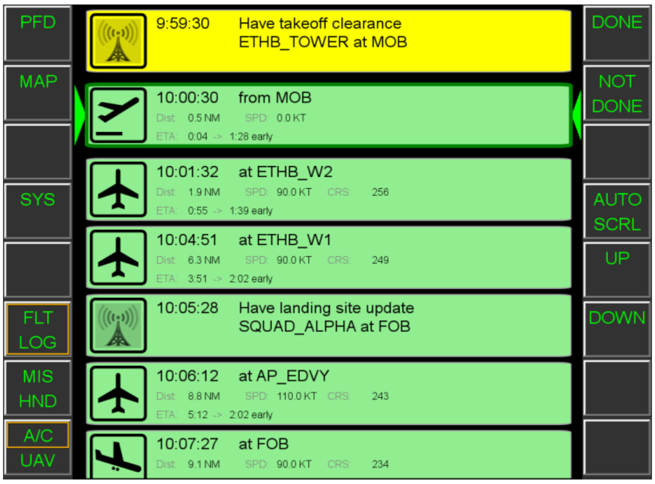

Figure 4-3: MiRA-T flight log display

The UI for the MUM-T assistant system has been designed by [Rauschert 2013] and was already described in sufficient detail throughout the chapters 3.1.2 and 3.2.3.

4.2.7 Overall Technical Design

In the following, the overall technical design for the MUM-T setup is explained. As can be seen in Figure 4-4, a division in five areas has been made. These areas are:

1. Helicopter Systems
2. UAV Systems
3. MUM-T Assistant System Core
4. MMP Core
5. MiRA-T

The helicopter systems area first includes all human-machine interfaces (for UAV guidance and the helicopter flight log as well as towards the MUM-T assistant system and the helicopter mission management system). Then, there is the UAV communication module, which collects and sends messages to/from the multiple UAV mission management systems. Finally, the helicopter management systembelongs to the helicopter systems area.

The UAV systems relevant for the work described in this dissertation are the UAV mission management system. Of course, the UAVs have more onboard systems (in reality as well as in the simulation). Similarly, the only MiRA-T module relevant here is the communication planner, which has as input the currently active mission plan (SyP), and which puts out the complete helicopter flight log including the crew communication tasks, e.g. for communicating with mission command, air traffic control or ground troops.

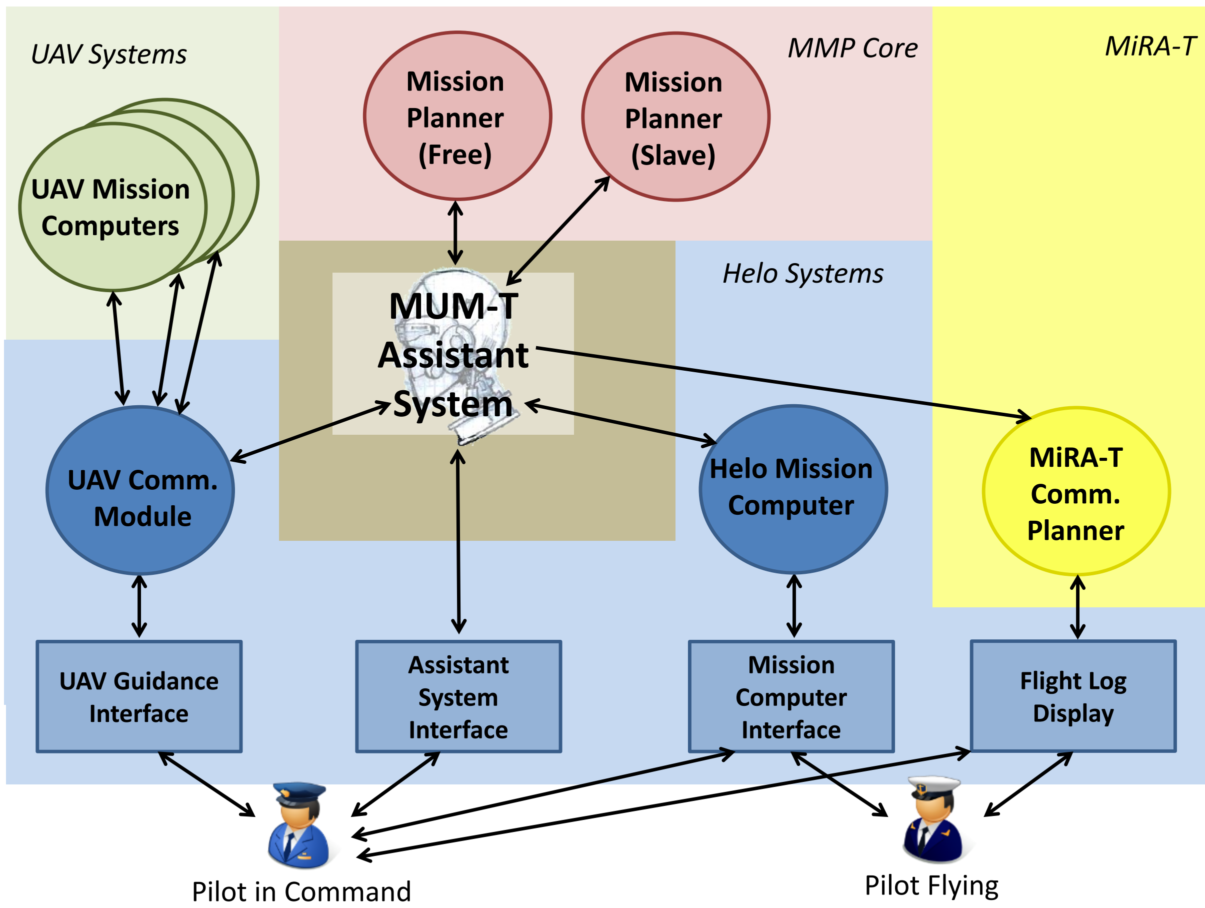

Figure 4-4: The overall technical design for the MUM-T setup

As depicted in Figure 4-4, the *MMP core* area consists of the two mission planner instances. It also shows that all input and output is handled by the MUM-T assistant system core. This is because the assistant system manages the SyP, and the planners themselves are only the tools to work in the background to create, manipulate and check the mission plans. As opposed to this, from the human operators' perspective, the *MMP frontend* includes

- the UAV guidance interface,
- the MUM-T Assistant system interface,
- the mission management system interface and
- the MiRA-T flight log display.

4.3 Mixed-Initiative Mission Planner Implementation

The MMP detailed design is made up of the following components:

- The *Planning Process Manager*, which has a constraint input interface
- The *Planning Engine*, which is a classical operator-based planner
- Multiple *Domain Knowledge Configurations* each describing a variant of the MUM-T mission domain knowledge and cost model

These components work together in the way described in the following. As mentioned before, the MUM-T assistant system sends plan requests and constraints via the STCI to the Planning Process Manager (PPM). The PPM then translates the constraints dynamically into a problem definition compatible with the planning engine and starts multiple processes of the LPG-td planner [Gerevini, Saetti & Serina 2004] in parallel. These work upon the given problem under usage of different Domain Knowledge Configurations (DKCs) (see Figure 4-5). The DKCs contains slightly varying MUM-T world models in order to create domain-specific problem-solving heuristics[33] (cf. Table 4-3). The generated plans are finally collected by the PPM and delivered back to the assistant system.

[33] It is important to note that this is not a heuristic in the sense of heuristic-guided forward planning, i.e. a heuristic that represents an estimation concerning the distance to the goal state. Instead, it is a heuristic that explicitly advertises action alternatives to reach the same goal with different side-effects, which may generate different costs.

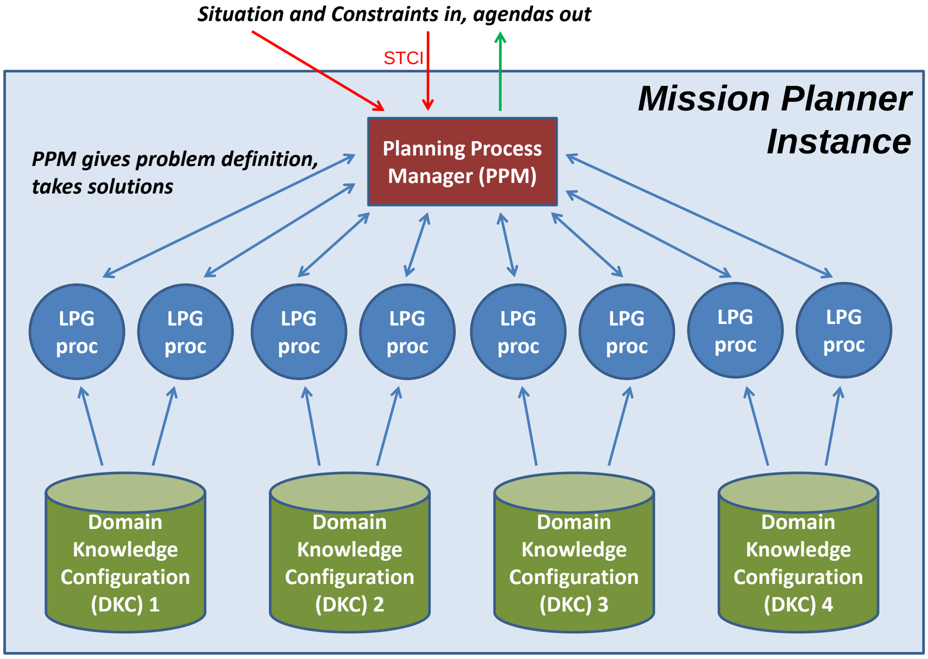

Figure 4-5: MMP internal structure and functionality (schematic)

Table 4-3: PPM planning processes and Domain Knowledge Configuration allocations

Process ID	DKC ID	Forbid helicopter unrecc'd territory	Forbid troops unrecc'd territory	Specialty
1	5	soft	soft	*Helicopter maximum speed instead of military speed*
2	2	soft	**hard**	-
3	3	**hard**	soft	-
4	4	**hard**	**hard**	-
5	1	soft	soft	-
6	2	soft	**hard**	-
7	3	**hard**	soft	-
8	4	**hard**	**hard**	-
9	1	soft	soft	-
10	2	soft	**hard**	-
11	3	**hard**	soft	-
12	4	**hard**	**hard**	-

In the following, the implementation approach used for the MMP is explained in more detail, including the planning engine used, the implementation of the PPM as well as the planning domain and problem definitions. Finally, the cognitive systems to support MUM-T that reside outside of the MMP are briefly described.

4.3.1 Planning Engine

Although many mission planning systems with symbolic focus used in the aerospace domain are based upon the HTN knowledge-based planning approach,[34] there is an advantage of a classical operator-based planner when used for building a cognitive system. An HTN planner is designed to explore different predefined possibilities of task decomposition and perform scheduling. This provides less flexibility compared to an operator-based planner, which is exploring combinations of atomic actions. Also, HTN planners have problems at planning for individual and interleaving actions for multiple agents [Goldman 2006]. For example, in the HTN-based Playbook™ Approach, a play (cooperative action of multiple UAVs) has to be defined before it can be invoked by the operator [Miller et al. 2004], which lowers flexibility. To find fine-granular and creative solutions of non-prescribed multi-agent cooperation (which can be regarded as *emergent behavior*[35] of the technical system) in a central planning approach is to be regarded as a strong advantage in the complex and dynamic environment of a MUM-T mission, although it poses a heavy burden on solution search performance.

For the MMP implementation, a deterministic planner is regarded as sufficient, i.e. all actions are supposed to succeed, and no unforeseen events are regarded in the currently active plan. In the Cognitive Skill Merging approach, it is possible to leave probabilistic reasoning over for the human.[36] In case of an unforeseen event that conflicts with the currently active plan, replanning becomes necessary.

It needs to be emphasized that the helicopter mission planning and the planning of the UAV task agendas are mutually dependent from each other. This is because, in principle, for the helicopter it is best to fly where the UAVs will have reconnoitered the path, and for the UAVs it is best to fly where the helicopter will be passing through later on. The UAV task agendas are also not independent from each other, because they have to maximize their reconnaissance performance. In addition, the ground troops' actions also depend on the helicopter (e.g. where they are dropped) and on the UAVs (i.e. if the troops' path has been reconnoitered by these already). Due to these interdependencies, it is a convenient way to plan the complete MUM-T

[34] See Playbook™ approach described in chapter 2.2.2 and also [Bonasso 1999] or [Gancet et al. 2005].
[35] Emergent behavior has many definitions. The one that the author applies here is that the combination of simple rules (such as action pre- and postconditions in classical planning) can lead to complex results, which are unforeseen during the definition of these rules [Rollings & Adams 2003].
[36] An example for this is the operator's plan to use a certain number of UAVs for the mission. Although a lower number of UAVs would be sufficient to fulfill the mission goals he/she decides to take a reserve with him/her for the case that unforeseen problems arise during mission execution (e.g. loss of a UAV, emergence of enemy forces, follow-up orders from mission command).

mission for all agents with one planner instance. Of course, the planning engine's performance must be sufficient in order to comply with these requirements.

Considering the abovementioned requirements and the ones in chapter 4.1, the selected MMP planning engine was the LPG-td (*L*ocal Search for *P*lanning *G*raphs – *t*imed initial literals and *d*erived predicates) in its version 1.0 [Gerevini, Saetti & Serina 2004] due to its full support of the temporal expressiveness of PDDL 2.2 [Edelkamp & Hoffmann 2004], its anytime planning capability and its good performance [Edelkamp 2004]. The planner is used in *best-quality* mode, i.e. it incrementally puts out the best plan found so far (evaluated by cost minimization constraint).

The LPG-td builds up so-called Temporal Action (TA) Graphs, which are similar to Planning Graphs used in the well-known *GraphPlan* [Blum & Furst 1997] algorithm. These Planning Graphs have alternating layers of *states* with positive propositions and possible *actions*. In TA Graphs, temporal information is added to the nodes in the graph. The LPG-td's algorithm then performs statistical *Local Search* inside the TA Graph with a version of the WalkSAT [Selman, Kautz & Cohen 1993] algorithm.

4.3.2 Planning Process Manager

As soon as the PPM receives a constraint set via the STCI, this is understood as planning command, and the PPM dynamically generates a problem description file containing the complete MUM-T problem including all agents (see chapters 8.2.1 and 8.3.1). The current tactical situation known to the MUM-T assistant system, which includes all agent data (name, type, position and state information, such as landed or not and cargo state) and all mission-relevant locations, is used as the initial state for the problem description. Also, all distances between the locations are calculated and set as numerical values in the problem description. The mission order, which is part of the constraint set, includes the goal state of the planning problem and other constraints, such as the preferred drop zone and so forth (see chapter 4.1.3). In addition to that, in case of the slave instance of the mission planner, the UAV tasks generated by the PIC provide further constraints to the generation of the aHuP. The complete set of constraints is processed as follows.

The conversion of hard temporal constraints into PDDL works with timed initial literals [Edelkamp & Hoffmann 2004 9 in combination with denying or allowing preconditions for actions defined in the domain. For example, if a constraint states that the takeoff of a specific aircraft from a specific location is allowed only after 10:00 ("not before" constraint), then via a timed initial literal at 10:00 a predicate “takeoff_denied” for this aircraft at this airport becomes false, which is a precondition for the “takeoff” action.

The hard “at end”- and “anytime”-constraints are directly translated into goal states for the PDDL planner (see Figure 4-6 and chapter 8.2.1, e.g. code lines 615-623). In case of action constraints (either “anytime” or “at begin”), a post-condition is predefined for the action corresponding to the task, which leads to the fulfillment of the predicate contained in the goal state upon action execution.

Soft constraints are realized via a benefit that is calculated into the total costs of the solution in case the constraint is met (e.g. normally the costs for landing are zero, but if landing is preferred at a specific location then the cost function for this location is set to a negative value). Unfortunately it is not possible to generate soft temporal constraints (“not before”, ”not after”) with the current implementation.[37]

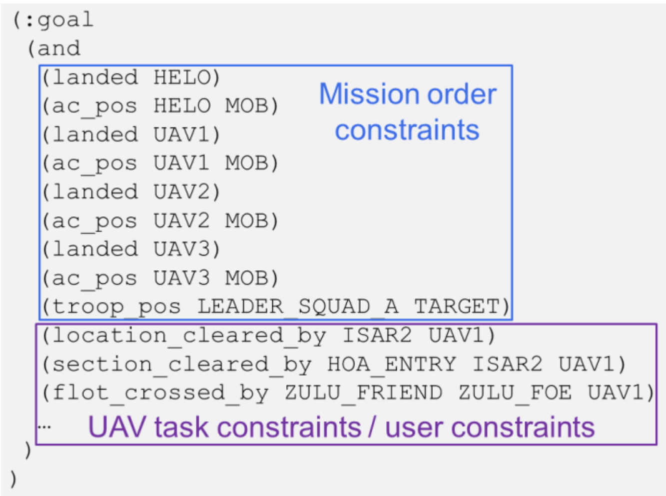

Figure 4-6: PDDL goals generated by the PPM from mission order and user constraints

The twelve different LPG planning processes, which are set off by the PPM, perform their search each with a distinct initial random seed. In case of the receipt of a new constraint set, the planning process (i.e. all LPG-td processes) is started from scratch, which makes the replanning procedure a “brute-force” approach[38]. Although this is not a very sophisticated solution, it is important to mention that the brute-force approach that is often taken in computer games A.I.

[37] PDDL 2.2 is not expressive enough for that purpose, as opposed to PDDL 3.

[38] Although the approach of plan repair would have been more performant and could be easily supported by a local-search based planner, source code modification for the LPG-td would have been necessary for this reason. This was technically not feasible, as the LPG-td was used in a pre-compiled closed-source version.

or for Computer-Generated Forces (CGF) for military simulations [Domshlak, Even-Zur & Golany 2011].

4.3.3 Domain Definition

To be able to cope with the requirements stated in chapter 3.1.1, the MMP holds domain knowledge about possible world states and allowed operations (e.g. UAV tasks), which transform the world state in a specific way. This knowledge is stores in the different DKC files. Each such file is a domain specification for the PDDL language version 2.2. Such a PDDL domain definition includes the description of *object types*, *predicates*, *functions* and *actions*. In the MUM-T world model there are *location*, *aircraft* and *troop* object types. In total, around 60 predicates and functions have been defined for the locations and their interconnections (in order to allow coarse route planning) and the description of the agents (aircraft, troops), e.g. location, speed etc.

All tasks that can be assigned via the task-based UAV guidance interface are represented as *durative actions*. Like the UAVs, the helicopter is of the aircraft object type, but in contrast it is excluded from reconnaissance and surveillance actions. However, it has additional abilities, such as the loading and unloading of troops. It is important to note that some tasks need multiple action models to cover different situations (e.g. “finish departure” as a special case of “departure”, which is valid in case this task is already in progress while starting the planning process). This results in 30 different durative actions implemented in total.

As explained before, the MMP works with multiple PDDL domain configurations in parallel, which are used by different LPG-td processes. This has been implemented in order to force the MMP to explore certain cost-saving or exceptional paths to solving the planning problem. For example, the DKCs with IDs 1 and 2 contain the additional, very costly, exceptional action “land at unreconnoitered site”, thus allowing a solution including this action in principle. Because not all DKCs include this action and the pool of LPG processes is fed with the different DKCs in equal amounts (cf. Figure 4-5 and Table 4-3), certain effort is spent on the search for solutions excluding this costly action per se by turning a soft constraint to a hard one (hard precondition).

All costs have been implemented via functions in PDDL [Fox & Long 2003] and therefore the cost values need not to be part of the domain model but can be generated dynamically in the problem file. The cost model has been described in chapter 4.2.5 already.

4.3.4 Problem Definition

In order to get a better understanding for how the LPG-td is used during runtime, the problem files generated by the PPM during runtime in a MUM-T mission are briefly explained in the following.

The most critical situation for the MMP in terms of performance is the early phase of the mission after the operator has entered the UAV tasks into the system. In this situation the longest action sequence has to be generated in order to accomplish the mission and bring the aircraft back home again. Some approximate benchmark data for this case are given below:

- 35 tasks to be given to the UAVs to fulfill mission
- 600 facts about the 25 locations and their relations
- 45 facts about the 5 agents (aircraft and ground troops)
- 35 timed initial literals
- 50 goal predicates[39]
- 75 action steps in the solution
- 20-30 seconds to find a satisfactory plan[40]

Examples of problem definition files can be found in chapters 8.2.1 and 8.3.1.

4.3.5 Cognitive Systems Outside of the Mixed-Initiative Mission Planner

The Cognitive Agents that support the PIC in the MUM-T setup, i.e. the UAV Cognitive Agents and the MUM-T assistant system Cognitive Agent, are implemented based upon the UBM's Cognitive System Architecture (COSA) [Putzer & Onken 2003]. This allows to program system behavior through mental concepts of the Cognitive Process [Onken & Schulte 2010]. These concepts are environment model (called "belief" when instantiated), desire (called "goal" when instantiated), action alternative (called "plan" when instantiated) and instruction model (called "schedule" when instantiated). Figure 3-2 depicts the underlying processing cycle. Data gathered from the environment is interpreted by the cognitive agent and then stored as a set of beliefs. A change in beliefs may change the agent's goals during the goal determination phase. In order to reach the determined goals, a plan is generated in the planning phase. The state of

[39] This counts for the Slave instance. The Free instance has fewer goal predicates in its constraint set.
[40] Values taken when each mission planner instance was running on a high-performance PC with 6 hyper-threading processor cores (i.e. 12 virtual processors, 1 per LPG process). "Satisfactory" is not a hard criterion here, it was subjectively evaluated by the author.

the plan is continuously compared with the most recent set of beliefs concerning the environment state. Thereby, the action to carry out at the current moment is determined (scheduled) and then carried out in order to influence either the environment or the agent itself towards its goals.

5 Experiments

In May 2011, the final experimental campaign for the UBM's MUM-T project was conducted. The goal of the campaign was to evaluate the complete MUM-T system concept described in the previous chapters in human-in-the-loop simulator experiments. The UBM's generic helicopter simulator (cf. Figure 5-1) and UAV simulators capable of full-scale mission simulation have been used for this purpose.

5.1 Research Questions

The goal of the UBM's MUM-T experiments was the evaluation of the Dual-Mode Cognitive Automation approach to the MUM-T related problems of UAV operator overtaxing, when working as PIC at the same time. To evaluate the MMP as one component of workload reduction for the PIC,[41] the following questions are relevant:

1. Is the simulation realistic enough and suited for getting valid evaluation results?
2. Is the scenario realistic enough and suited for getting valid evaluation results?
3. How is the UI for planning the UAV mission rated by subject matter experts (SMEs)?
4. How is the MMP's performance (processing speed and result quality) rated by SMEs?
5. What is the difference in the MMP's objective performance when planning the UAV missions in mixed-initiative vs. fully automated?
6. What is the difference in the overall human-machine system's objective performance when planning the UAV missions in mixed-initiative vs. manually?
7. What is the difference in the PIC's objective situation awareness when planning the UAV missions in mixed-initiative vs. manually?
8. What is the difference in the PIC's subjective workload when planning the UAV missions in mixed-initiative vs. manually?
9. How is the MUM-T assistant system's performance rated by SMEs as far as its functionality based upon the MMP is concerned? The categories advice quality, workload reduction and efficiency increase are relevant here.

[41] For the evaluation of the other systems see [Uhrmann 2013] (UAV automation) and [Rauschert 2013] or [Rauschert & Schulte 2012] (PIC assistant system).

5.2 Experimental Design and Procedure

In the following, the technical setup, test persons' data, course of action and configurations of the experimental missions will be described.

5.2.1 Technical Setup

For different projects, including the MUM-T project, the UBM has built up and runs a research simulator (see Figure 5-1). It is comprised of twin-seat helicopter glass cockpit with six 10" touch screens, which are usable as MHDDs. Five of these are in the front and one is located in the center console. Every MHDD can be configured to show either of the following pages:

- primary flight display (PFD)
- moving map
- flight log
- mission order page
- systems display (showing RWR and engine parameters)
- UAV camera
- UAV photo page

The default configuration, as shown in Figure 5-1, is that the PIC (left seat) has one or two map pages in front of him/her and can switch one to the target identification page (second screen from left in Figure 5-1). The PF has a map and PFD display, and in the middle between the two crew workstations, the systems page and the radar warning receiver page are displayed.

Underneath each touch screen, two rotary encoders are located. These can be used for quick modification of page-dependent parameters, e.g. the map can be zoomed and its contrast can be changed. In the center console, two MCDUs are located. One of these is depicted in Figure 4-2. The following functions are accessible via the MCDUs:

- Display and modification of the helicopter mission goals and constraints
- Operation of the two onboard radios
- Display and modification of the transponder settings
- Display and modification of the gear state
- Send load/unload troops commands[42]

Figure 5-1: The cockpit of the UBM's generic helicopter simulator

In the simulator cockpit, there is a stereo headset for each the PIC and the PF. The audio system allows the usage of the intercom within the cockpit as well as the communication with the simulator operator, air traffic control, other blue forces and the assistant systems. The out-the-window view is generated by three projectors and three screens with a 60° horizontal field-of-view each.

The UBM developed a scenario simulation, which is has the ability to connect an arbitrary number of aircraft as well as ground forces and manage their interactions. Thereby, the UBM

[42] This could also be regarded as cargo door operation.

is able to simulate a MUM-T mission as outlined in chapter 2.5.2. The MUM-T scenario simulation is able to simulate two different types of interactive enemy ground CGF:

- SAM sites, which have an active radar and can fire guided missiles at the manned helicopter and the UAVs
- Soldiers carrying a rocket-propelled grenade launcher (RPG) can fire unguided rockets at the manned helicopter in case it is flying low[43]

The computer infrastructure used for the MUM-T experiments, which are described in this dissertation, is displayed in Figure 5-2. This logical view is limited to the components directly relevant to the MMP. The physical setup of the simulation laboratory, which was used for conducting the MUM-T experiments, is shown in Figure 5-3. Some computers were also accessible as work stations for software development, debugging and application or simulation control.

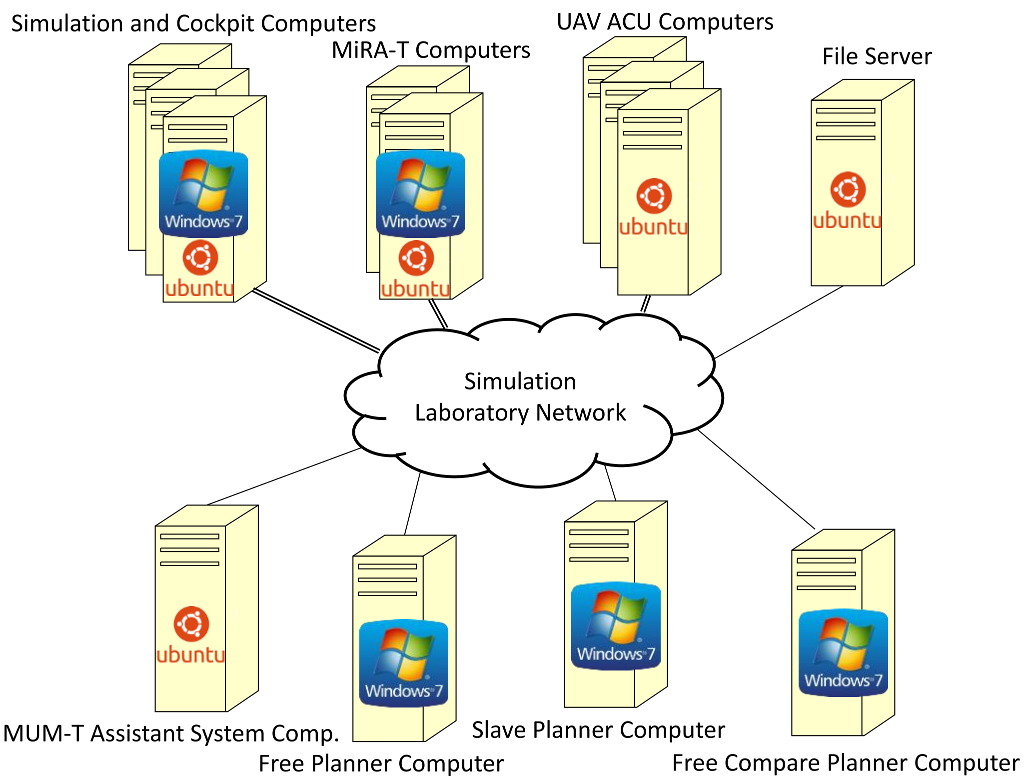

Figure 5-2: MMP-relevant infrastructure of the MUM-T simulation laboratory (logical view)

[43] The default altitude of the UAVs is too high for the use of the RPG weapon against them.

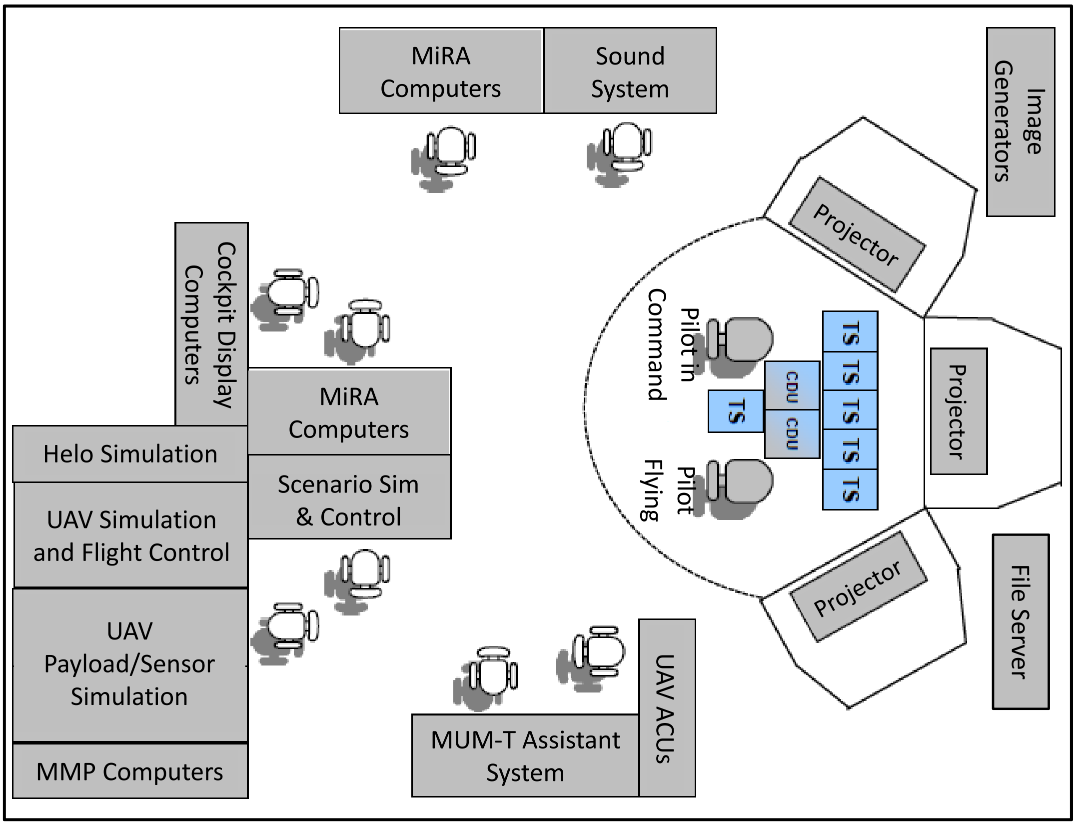

Figure 5-3: Physical setup of the MUM-T simulation laboratory
(adapted from [Rauschert 2013], TS means Touch Screen, dashed line represents light-absorbing curtain)

5.2.2 Test Persons

Eight German Army helicopter pilots with an average age of 37 years (min 28 years, max 51 years) participated as test persons for the experiment. Their flying experience ranged from 830h up to 5100h with an average of 1815h, mainly with the BO-105 anti-tank and liaison helicopter, except one test person with the CH-53 transport helicopter. Additional helicopter types flown by the test persons for a significant duration (more than 20% of the total flight experience, training helicopters excluded) were in two cases the UH-1D transport helicopter and in one case the Mi-24 attack helicopter. The test persons were grouped into fixed crews consisting of two members (alternating in the roles of PIC and PF[44] in the experiments).

5.2.3 Course of Action and Measurements

At the beginning of the experimental campaign, the crews had two days of training for both workstations (PIC and PF). The training began with an instructed phase and then continued

[44] Concerning the description of the experiments, the PF role is not further regarded.

with a free training phase. During both phases, another crew was allowed to watch. This was intended to be additional, passive training.

After the training phase, each test person flew four measurement missions, of which two were in the role as PF and two in the role as PIC. This means that the crew members switched roles between missions. Figure 5-4 shows the experimental setup with one German Army Aviator as PIC (left) and another one as PF (right). The two missions to be flown in each role had *different assistant system configurations* (independent variable in the experiment):

(1) MUM-T assistant *on* (i.e. mixed-initiative UAV mission planning) and MiRA-T assistant in *non-adaptive* mode[45]

(2) MUM-T assistant *off* (i.e. manual UAV mission planning) and MiRA-T assistant in *resource-adaptive* mode

Figure 5-4: German Army Aviators as test persons in the MUM-T simulator

Each mission was a helicopter troop transport mission supported by three reconnaissance UAVs with a second follow-up troop transport task included, as described in chapter 2.1.2, with total mission duration of 30 to 45 minutes. However, because the test persons were not supposed to see any mission twice, four different mission configurations had to be generated, which are

[45] See [Maiwald 2013] for the description of the PF assistant system modes. The PF assistant system is not further regarded here.

described in the next subchapter. Through this approach, the course of action and the locations of mission-relevant objects were not known in advance by the test persons.

Before the start of every measurement mission, the crew received a mission briefing, which was supposed to be realistic enough in order to create a realistic surrounding for the scenario and valid scientific results.[46] Different briefings were necessary, because every mission was unique concerning geography, goals and constraints (i.e. preferred drop zone).

During each mission, the simulation was paused three times in order to measure the crew's subjective workload with NASA Task Load Index (NASA-TLX) [Hart & Staveland 1988] and the objective situation awareness with SAGAT[47] [Endsley 1988]. The pauses were made consistently following this scheme:

1. The first pause was made while the helicopter was still in friendly territory. This measurement served as baseline for flight in friendly territory.
2. The second pause was made when the crew was in a demanding mission phase inside the HOA, after finishing the primary troop transport mission and receiving a follow-up mission order.
3. The third measurement was made while the helicopter was on the egress route, but still in hostile territory (here, only NASA-TLX was measured, as a second baseline for flight in hostile territory).

The experiment was finally stopped as soon as the helicopter returned to friendly territory. For further evaluation and reproducibility, video and audio recordings were taken and all relevant simulation data was logged, which included the system interactions of the PIC and the PF.

5.2.4 Experimental Configurations

As described above, four different mission configurations were needed for the MUM-T experimental campaign. All of these had in common that troops had to be transported by the helicopter from friendly into hostile territory and that there were two replanning situations:

- an ad-hoc follow-up transport mission order, which arrived as soon as the first transport task was completed,

[46] The weather/atmosphere part is normally obligatory, but it was excluded from the briefing. Instead, the briefing focused on the tactical part.
[47] Situation Awareness Global Assesssment Technique

- and the primary landing site was threatened, either during the first or in the follow-up transport task.

The follow-up order contained either a second troop transport (within hostile territory) or the recovery of a crashed pilot (from hostile to friendly territory). In total, all missions took a similar course, but the details differed sufficiently, especially in situations of critical decision-making (e.g. landing site is either threatened or not).

As Table 5-1 shows, the configurations also differed concerning the geographic mission area, whether the MUM-T assistant system was enabled (the independent variable that is relevant here), whether the MiRA-T assistant system communicated with the PF in a resource-adaptive way (or only via text or speech messages respectively). In addition to that, there were different threat configurations (enemy ground forces) leading to mission replanning via the alternate DZ or the alternate corridor as soon as they are detected by a UAV and identified by the PIC. It is important to note that at the beginning of the mission, the enemy forces' quantity, types and whereabouts are not known to the crew. The RPG-carrying soldiers are always grouped with a jeep vehicle, because only objects in the size of a car are recognized by the ATR functionality.

Table 5-1: List of the different experimental configurations

Config	Area	PIC / PF	PIC assistant	PF assistant	Follow-up order	Relevant threats at
1	Northern Germany	A / B	Disabled	Not adaptive Text messages	Pilot recovery	Primary DZ Primary egress corridor
2	Northern Germany	B / A	Disabled	Not adaptive Speech msg.	Troop transport	Primary follow-up DZ
3	Southern Germany	A / B	Enabled	Adaptive	Troop transport	Primary follow-up DZ Primary egress corridor
4	Southern Germany	B / A	Enabled	Adaptive	Pilot recovery	Primary DZ

5.2.5 Questionnaire

The test persons were asked to fill out a questionnaire, which relates to research questions given in chapter 5.1. All the items can be found in the Results section (chapter 5.3). One part of the questionnaire referred to an alternative UAV task planning display, which needs to be described here.

The map-based GUI described before in this dissertation (see Figure 3-3) allows easy geo-referencing of tasks during mission planning, but it lacks input and display features for time values. Because in a MUM-T scenario not only the geographical coordination of the aircraft is of importance, but also the temporal coordination, the author supposed that the test persons would prefer having a Gantt-chart-style [Gantt 1910] GUI for temporal aspects, which could be an addition to the map-based GUI described before. However, only the map-based GUI was made usable for the test persons during the experiments. Hence, after the experiments, the test persons were shown a video presenting an alternative task-base guidance GUI (see Figures 5-5 and 5-6) for evaluation purposes.

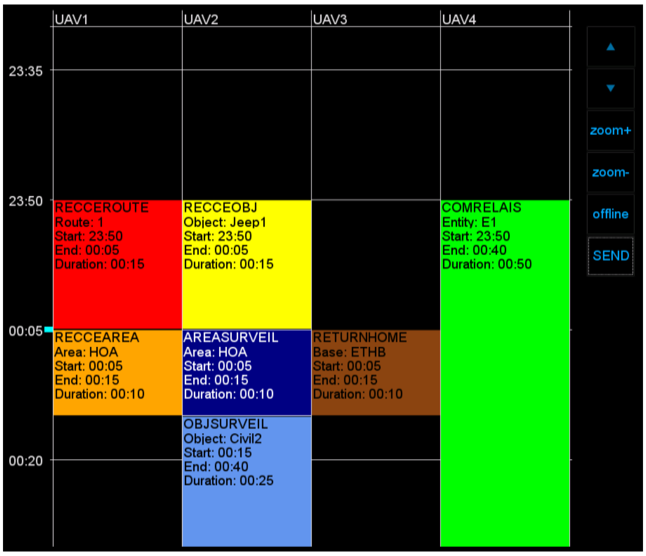

Figure 5-5: Overview screen of the alternative task-based guidance GUI

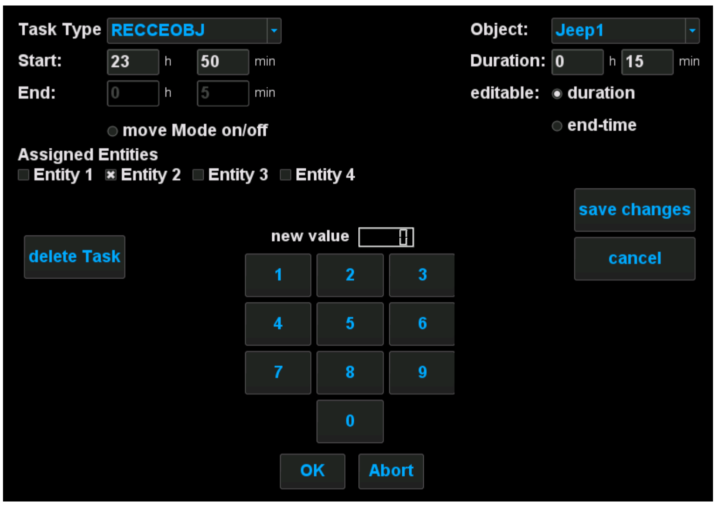

Figure 5-6: Task-entry mask of the alternative task-based guidance GUI

5.3 Experimental Results

In this chapter, the results of the MUM-T experimental campaign are described; first the objective and subjective performance measurements, then the subjective evaluation by the SMEs as well as their statements concerning human planning strategies and requirements for planning systems and finally the general observations made during the experiments. Concerning the objective performance results, a differentiation is made between the overall human-machine system performance (comparison of mixed-initiative vs. manual planning) and the MMP system performance (comparison of mixed-initiative vs. automated planning).

5.3.1 Objective Overall Human-Machine System Performance

As described above, the overall human-machine system performance for the configuration with MUM-T assistant system enabled (i.e. the mixed-initiative UAV mission-planning configuration) shall be compared to the baseline configuration without MUM-T assistant system disabled.

Three different measures for the UAV reconnaissance performance (i.e. UAV operator performance) have been calculated. The exposure of the helicopter to potential threats is an important risk factor, and in the MUM-T configuration, it is the responsibility of the PIC to minimize it by leveraging the reconnaissance abilities of the UAVs. Therefore, first, the *exposure time* was measured, i.e. the time span in which the helicopter had a geographical position (2D) that had not been photographed by a UAV before. Further detailed measures are the *reconnaissance lead time*, i.e. how much in advance the UAV scanning took place (the more seconds the better) and the *reconnaissance actuality* or orthophotos actuality, i.e. how old the photos are in the moment the helicopter passes the corresponding area (the less seconds the better). In these categories, no significant effects were found. [Strenzke et al 2011] contains all measurements of the overall human-machine system performance in detail.

5.3.2 Objective Mixed-Initiative Mission Planner Performance

Although the test persons did not have any direct access to the MMP's functionality of automated planning for the UAV tasks, it is possible to compare the solution quality of the mixed-initiative planning mode with a completely automated planning mode. The necessary data is available after the experiments, because upon every implicit user input (entering of UAV task constraints), the MMP started generating an assumed human plan (transparent to the human operator), which takes the user constraints into account (i.e. *mixed-initiative planning*) and also generating a reference plan[48] (also transparent to the human operator), which disregarded the user input (i.e. *automated planning*).

From the data generated during the experiments (395 valid planning requests leading to at least one solution), a performance comparison can be made of both the mixed-initiative (Slave) and the automated (Free) planner, which had the same software and hardware basis.[49] Table 5-2 shows how many times only one of the two planners found a solution fast enough (before the next planning request), the average time for finding the first solution (including its quality) and the average cost of the best solution for each planning request (including its planning duration). Unfortunately, the costs are not normalized, i.e. they can become negative for a good solution (about -10,000 as lower boundary in most situations).

Table 5-2: MMP technical performance data [Strenzke & Schulte 2012a]

Mixed only	43 samples	
Auto only	117 samples	
Mixed fastest: time [s]	M = 14.2	SD = 19.2
Mixed fastest: cost	M = 14549.9	SD = 12970.9
Auto fastest: time [s]	M = 12.8	SD = 19.9
Auto fastest: cost	M = 12546.7	SD = 70229.4
Mixed best: cost	M = 4541.8	SD = 10010.7
Mixed best: time [s]	M = 92.8	SD = 145.9
Auto best: cost	M = 8.9	SD = 7622.2
Auto best: time [s]	M = 74.5	SD = 146.1

As can be seen from Table 5-2, the automated planning puts out a first plan faster and this plan has in average also lower costs. The automated planning also generates better plans over longer time. This leads to the conclusion that the constraints given by the human did not help the MMP to find a solution faster or more cost-optimized solution.

However, it is important to note that this evaluation has been done with the cost function that is partly pre-programmed into the MMP and partly given by the work objective, both unalterable for the human operator. The plan generated in mixed-initiative can still be better from the viewpoint of the human operator, though. As explained before, the MMP does not

[48] This was done with an otherwise unused third mission planner instance called "Freecmp" (Free – Compare).
[49] The instances were running on different PCs which were equally equipped and had only the MMP software running on it.

perform probabilistic planning, and therefore it generates solutions that concentrate on scanning one main helicopter route with as many UAVs as possible. In contrast to this, most of the time the plans which were generated by the test persons let two UAVs cover the main route and one cover an alternate route in order to minimize risk in case the main route is threatened. For the cost model of the MMP this is a suboptimal solution, which would be one example suited to explain the higher costs for the mixed-initiative planning approach. The MUM-T assistant system allowed the test persons to generate such plans and only warned about missing tasks, not about existing suboptimal tasks. Hence, the acceptance by the users show this is a workable approach to leave the probabilistic planning to the human.

5.3.3 Subjective Workload Measurement

Subjective workload was measured using the NASA-TLX subjective workload assessment tool, which divides workload into the following six categories:

- mental demand
- physical demand
- temporal demand
- performance
- effort
- frustration

The overall NASA-TLX scores for the PIC did not produce significant results because the variation in the SME's individual utilization of the scale is higher than the differences between the system configurations. However*, for the subscale "temporal demand", a significantly lower level (8.3% compared to 13.7%) could be found for the configuration with MUM-T assistant system enabled* (p=.03).

5.3.4 Objective Situation Awareness Measurement

Another interesting measure is the situation awareness of the human operator. The experiments described in this dissertation applied the Situation Awareness Global Assessment Technique (SAGAT) [Endsley 1988]. At two points in time during each mission the test persons had to answer where air and ground vehicles were located, which tasks the UAVs were currently working upon and which the next task was for each UAV.

Although the different experimental missions had a similar course, one of them had a different geometry (unidirectional instead of round-trip). This led to degraded situation awareness, because, although the mission had the same length, the mission-relevant area was larger. This had a negative impact on the section that was visible on the map, which worked as "external storage" for the operator's situation awareness. In addition to that, the number of ground vehicles that were found at the time of measuring was varying largely (between 3 and 8) and also the strategies of the test persons, which vehicles to memorize were different. Therefore, not all SAGAT data is presented here.

Nevertheless, there was a significant effect of improved PIC situation awareness concerning UAV task agendas (p=.03) [Strenzke et al. 2011]. However, as can be seen in Figure 5-7, the SA in the unassisted setup was already very high.

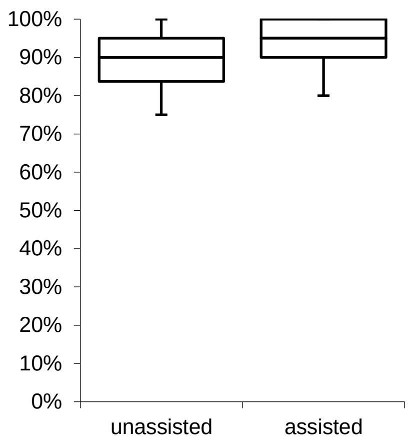

Figure 5-7: PIC situation awareness concerning UAV task agendas (SAGAT) [Strenzke et al. 2011]

5.3.5 Evaluation by Subject Matter Experts

The test persons also evaluated the MMP and the MUM-T assistant system concerning the advice usefulness. These evaluations were realized via Likert scales (ranging from -3 to +3, with 0 being neutral) and have been applied to the following interactive features (see Table 5-3):

1. advice concerning a single UAV task
2. advice concerning complete mission replanning

The single task advices (mixed-initiative planning) were rated positively concerning all aspects (making sense, raising efficiency, slightly relieving). The complete plan proposals (i.e. rather automated planning) were only rated slightly positively, with high variance due to some occurrences of very brittle solutions (detours and the like). However, on average, for this interactive feature, no effect on efficiency was reported. The quality of the complete plan proposals was evaluated slightly positively and the generation duration was evaluated slightly negatively. It was not possible to ask for single task advice generation speed because the underlying planning process was transparent to the test persons.

Table 5-3: Subjective evaluation of MUM-T assistant system planning assistance advices

Item (to be rated on a scale from -3 to +3):	#Valid answers[50]	M[51] =	SD[52] =
Single task advice made sense?	7/8	+1.7	1
Single task advice relieving?	7/8	+0.7	1.3
Single task advice raised efficiency?	5/8	+1.2	0.7
Complete plan advice made sense?	4/8	+0.3	2.4
Complete plan advice relieving?	4/8	+0.5	2.6
Complete plan advice raised efficiency?	4/8	0	2.5

The simulation, the scenario and the mission were rated by the SMEs as appropriate for the context of the experiments and as quite realistic (see Table 5-4). Furthermore, the MUM-T assistant system was considered rather helpful, and the sequential way to generate the UAV plans by entering one UAV task after the other was appreciated by the test persons (see Table 5-5). The quality of the initial plan generated by the MMP was rated as good after optimization (i.e. pressing of the OPTIMIZE button). The planning speed and plan quality for the landing zone switch were also rated as good. The same counts for the qualities of the plans generated on demand for the follow-up mission and of the plans automatically proposed by the MUM-T assistant system. However, in the latter two cases, the planning speed was not appreciated (see Table 5-6). Although the concept of optimization on demand was found helpful, the test persons would have preferred that the first solution be of better quality (see Table 5-7). Surprisingly, the wish for more interaction possibilities to create the helicopter plan was not strong. However, the SMEs would have preferred to be able to view and control the start and end times of the UAV tasks. In the opinion of the SMEs, these times could be calculated automatically and then be modifiable. From this, the general requirement of temporal planning can be derived. According to the SMEs, a project-based GUI for UAV task management should not replace but

[50] Number of test persons asked / number of valid answers
[51] Statistical Mean
[52] Statistical Standard Deviation

rather complement the map-based GUI used in the experiments (see Table 5-8). It has to be noted that some questions referred to in this paragraph were presented to only to the second half of the participating test persons (four instead of eight persons). [53]

Table 5-4: Evaluation of the experiments' validity

Item (to be rated on a scale from -3 to +3):	#	M=	SD=
The scenario was realistic	4/4	+1.8	0.4
The scenario was appropriate for the context of the experiments	4/4	+2.8	0.4
The simulation was realistic	4/4	+1.0	1.2
The simulation was appropriate for the context of the experiments	4/4	+2.5	0.5
The course of action of the mission was realistic	4/4	+0.8	1.6
The course of action of the mission was appropriate for the context of the experiments	4/4	+2.8	0.4

Table 5-5: UAV task planning functionality and interface usefulness evaluation

Item (to be rated on a scale from -3 to +3):	#	M=	SD=
In some situations, the assistant system was able to help with the replanning tasks	3/4	+1.7	1.2
Overall, the assistant system rather helped than hindered during the replanning tasks	3/4	+1.0	1.4
The way of entering/generating the UAV plans was purposeful	4/4	+1.3	0.8

[53] This was due to performance problems and software bugs, which affected the experiments conducted with the first four test persons and which would have influenced the evaluation. It was intended to have the design evaluated and not the implementation maturity. Table 5-8 is an exception, because it refers to evaluation that was independent of the mentioned problems.

Table 5-6: Automated planning performance evaluated by the subject matter experts

Item (to be rated on a scale from -3 to +3):	#	M=	SD=
Speed of initial planning	4/4	+0.5	1.8
Quality of initial planning	4/4	+0.3	1.1
Quality of initial planning after optimization	4/4	+1.8	0.8
Speed of replanning (landing zone switch)	4/4	+1.3	1.1
Quality of replanning (landing zone switch)	4/4	+1.5	1.1
Quality of replanning after optimization (landing zone switch)	3/4	+1.3	0.5
Speed of planning follow-up mission	4/4	-0.5	1.5
Quality of planning follow-up mission	4/4	+1.3	0.8
Quality of planning follow-up mission after optimization	3/4	+2.0	0.8
Speed of replanning when initiated by assistant system	4/4	-0.3	1.8
Quality of replanning when initiated by assistant system	3/4	+1.3	0.5

Table 5-7: Evaluation of the helicopter plan optimization concept

Item (to be rated on a scale from -3 to +3):	#	M=	SD=
I used the “Optimize”-button regularly	4/4	+0.8	0.8
When I wanted to press the “Optimize”-button, it was not available any more ("---")	4/4	-1.8	1.1
When I wanted to press the “Optimize”-button, it was not yet available (grey)	4/4	-1.0	1.2
When usable, the “Optimize”-button helped me to improve the helicopter route	4/4	+1.5	0.5
It often took too long waiting for an optimization possibility	4/4	-1.0	0.0
After pressing “Optimize” once, the helicopter route was good enough	4/4	+1.3	1.1
After pressing “Optimize” multiple times, the helicopter route was good enough	3/4	+2.3	0.5
The concept of quickly offering a suboptimal but optimizable plan pleases me	4/4	-0.5	2.1
I wished for more intervention possibilities concerning the helicopter route	4/4	+1.0	2.3

Table 5-8: Evaluation of the alternative task-based guidance display

Item (to be rated on a scale from -3 to +3):	#	M=	SD=
I would have preferred to see the UAV tasks' start and end times	6/8	+1.2	1.1
The UAV tasks' start and end times should be shown in the map display	6/8	+0.8	1.1
The UAV tasks' start and end times should be shown in another display (as in the video)	6/8	-0.2	1.9
I would have preferred to be able to define start and end times for the UAV tasks (as in the video)	6/8	+1.2	1.1
The UAV tasks' start and end times should be pre-calculated automatically and be modifiable	6/8	+1.2	2.0
The UAV tasks' start and end times should exclusively be calculated automatically	6/8	-1.8	1.5
I prefer the alternative task-based guidance interface to the map-based interface used in the experiments	6/8	-1.5	1.0
It would be good to have the alternative task-based guidance interface as addition to the map display	6/8	+2.2	0.7

5.3.6 Subject Matter Experts' Planning Strategies and Heuristics

To get a better understanding about the implicit cost function that the human operators apply during planning such a mission and to be able to tune the cost model of the MMP for similar future applications, the SMEs were asked for weighting certain factors in a questionnaire. These were:

- reconnaissance of helicopter route
- primary and alternate landing site
- flight path lengths
- different violations concerning the mission briefing

As can be seen in Table 5-9, the SMEs were asked about their strategies and heuristics concerning UAV planning. One of the results is that it seems to be a common strategy to enter incomplete UAV plans, such as flying to the operation area and not planning the way back home. The SMEs also stated that they plan the mission via addressing *critical decision points*, e.g. which corridor to choose, which landing site to choose and so forth. Furthermore, the mission phases (like ingress, operation and egress) seem to be highly interconnected, which supports the author's thesis that the mission planning problem is not dividable (especially during initial planning) [Strenzke & Schulte 2011b]. In addition, according to the SMEs, the replanning of the mission took place under noticeable time pressure. As a further result, the planning assistance was only regarded as useful by the SMEs in the case of in-flight replanning and not in the case of initial planning. This may be the case because during initial planning there was no time pressure. One can derive from this that the MUM-T assistant system was believed to speed up the overall planning process but not to improve the quality of the resulting plan.

Table 5-10 shows the prioritization (or weights) of the goals to be fulfilled during UAV mission planning. Here, one value is surprising: the SMEs did not care about a possibly too low distance between the UAVs and the manned helicopter, which could have been a safety issue due to danger of collision[54] and too short reaction time concerning threats detected by the UAVs. Rather, it was more important to let the distance not become too high.

Table 5-9: Subject matter experts' way of UAV task initial planning and replanning

	Initial planning			Replanning		
Item (to be rated on a scale from -3 to +3):	#	M=	SD=	#	M=	SD=
It is best to enter incomplete UAV plans	8/8	+1.0	2.0	8/8	+1.0	2.2
It is best to enter complete UAV plans	8/8	-1.0	2.2	8/8	-1.3	2.2
I plan such a mission in a hierarchical top-down way	8/8	+0.5	1.9	8/8	+0.4	1.9
I plan such a mission sequentially forward	8/8	-0.1	1.9	8/8	-0.3	1.6
I plan such a mission sequentially backwards	8/8	-1.8	1.5	8/8	-1.9	0.8
I plan such a mission by means of critical decision points	8/8	+1.8	0.8	8/8	+1.8	1.2
I usually plan towards an intermediate state/goal	8/8	+0.3	2.2	8/8	+0.1	2.5
I usually plan towards a final state/goal	8/8	-0.1	2.1	8/8	-0.1	2.3
While planning one mission phase, I do not care about another	8/8	-1.5	1.4	8/8	-0.5	1.9
While performing the planning task, I could use an assistant system	8/8	-0.5	2.0	8/8	+0.4	1.1
While replanning in-flight I felt mostly under time pressure	-	-	-	8/8	+1.4	1.2
My replanning was less systematical than the initial planning	-	-	-	8/8	+0.0	1.7

[54] The danger of collision was normally not so relevant because the manned helicopter and the UAVs had different flight levels.

Table 5-10: Subject matter experts' UAV mission planning goals prioritizations

Goal (freely movable sliders going from 0 to 100):	#	M=	SD=
Reconnaissance of primary landing zone	7/8	82.4	14.5
Reconnaissance of helicopter route	8/8	81.6	23.1
Reconnaissance of alternate landing zone	8/8	59.8	26.7
Reconnaissance of the route and target of the ground troops	8/8	56.8	31.0
Minimization of waiting times / holdings / slowing down of helicopter	8/8	56.1	31.8
Usage of briefed HOA entry and exit points	8/8	51.6	33.1
Distance of UAVs to helicopter not too large (for in-time reconnaissance)	8/8	47.9	41.6
Usage of briefed corridors (including corridors briefed as primary)	8/8	46.5	29.7
Compliance with corridor opening and closing times	8/8	31.8	40.1
Minimization of the UAV route lengths	8/8	18.0	31.4
Distance of UAVs to the helicopter not too small (for flight safety)	8/8	17.5	29.7
Keep one third / one UAV as reserve (free text item)	2/8	97.5	2.5
Reconnaissance of alternate helicopter routes (free text item)	2/8	72.0	21.0
Minimization of threat to UAVs (free text item)	2/8	50.5	1.5
Let UAVs wait / perform holding for further tasks (free text item)	1/8	85.0	0.0

5.3.7 Subject Matter Experts' Requirements for Mission Planning Assistance

The SMEs were also asked questions about the requirements for a UAV planning and plan management assistant. The test persons stated that it would have been beneficial to be able to manage alternative plans (see Table 5-11). However, it was not a problem for them to go without this feature during the experiments. Alternate plans should be on a per-UAV basis instead of including all UAVs at once. A system that generates only plans covering a limited time horizon (e.g. ten minutes in advance) would not have been accepted. Nevertheless, incomplete UAV plans are in principle acceptable. Another interesting finding is that the SMEs would allow the assistant to take about 35 seconds to replan during the most critical mission phases (see Table 5-12).

Table 5-11: Subject matter experts' requirements regarding UAV plan management

Item (to be rated on a scale from -3 to +3):	#	M=	SD=
It would have been beneficial to manage alternative plans also	8/8	+1.3	1.7
It was annoying not to be able to manage alternative plans	7/8	-1.1	2.0
Alternative plans should include all UAVs in a single plan	8/8	-1.1	2.2
Alternative plan should be per UAV	8/8	+1.9	1.6
It would have been beneficial to enter UAV plans hierarchically/top-down	8/8	-1.0	2.3
It would have been beneficial to generate UAV plans by specifying critical decision points	8/8	+0.3	2.1
It would have been beneficial if the system only generated plans for a limited time horizon (e.g. 10 minutes)	8/8	-1.6	2.4
It would have been beneficial if the system only generated plans for a limited time horizon, if planning was faster thereby	8/8	+0.5	2.1
An incomplete helicopter plan that is fulfilling the next goals would be acceptable	8/8	-0.3	2.2
An incomplete UAV plan that is fulfilling the next goals would be acceptable	8/8	+1.5	1.8

Table 5-12: Subject matter experts' Helicopter-mission planning-duration tolerance

Situation (sliders ranging from 0 to 120 seconds):	#	M= [s]	SD=
Initial planning	8/8	112.7	19.4
Replanning, in friendly territory	8/8	72.9	31.7
Replanning in hostile territory due to sensor tracks	8/8	35.4	13.7
Replanning during egress due to follow-up mission	8/8	33.9	25.1

5.3.8 Observations

One general observation made during the experiments and derived from the gathered data was, that the test persons showed a very different amount of *trust* in the MMP or the MUM-T assistant system. Unfortunately, trust in the system has not been directly measured (neither objectively nor subjectively). However, some test persons obviously tended to reject most assistant system proposals, even if they would have been helpful, whereas others tended to accept the proposals, even if they were suboptimal (e.g. appending a transit task for a UAV, where a recce route task would have been more practical).

Although the MMP was designed to work in all phases of the four experimental missions, independent from operator input/behavior, a not very obvious *problem with the symbolic A.I. world model* led to another important observation. The example is interesting, because it not merely about a software bug, but instead it represents a central problem of today's machine intelligence. In the MMP's world model, some representation has to exist concerning the way how the ground troops perform their movement. Early versions of the MMP's world model allowed the troops to move all the way from friendly territory to the target, which does not only take many hours, but also renders the need for the whole air assault mission obsolete.

Now there are multiple ways to force the helicopter to transport the ground troops during the planning process, so that the result will not contain a long march of the troops from their initial position to the target. For example, a temporal constraint could have been included in the mission order, stating that the troops have to arrive at the target before a certain time - and marching would have been too slow. Instead, the marching action was made very expensive, however it is still technically possible to walk all the way, and in case of bad cost optimization this unwanted action can be part of the solution. Therefore, in addition, it was defined that the troops cannot cross the FLOT (it is not so unrealistic that they are denied to cross the FLOT by foot). This worked well for the initial mission order, but in case of the follow-up order, the MMP could fail again, because in two of the four different missions, the second squad had to move solely inside hostile territory, i.e. there was no FLOT blocking them. After the occurrence of this problem, this was fixed for future experiments by limiting the total distance the troops are allowed to march during one mission. But the damage was done already, because the test persons, in their eyes, had a faulty plan for the helicopter on their displays (without pickup and drop of the troops), and to make things worse, they did not know why it had failed. From the viewpoint of the machine, this was a correct plan. It even did not matter that it was a very expensive one, because the automation simply did not find a better one in time.

The underlying problem was that there was no search heuristic to first try out the helicopter as a transporter.[55] This heuristic would have made sense for all four MUM-T missions. However, in another hypothetical but realistic scenario, where there are many ground troops wanting to get to their targets and there is only one helicopter, it may not make sense to try out the helicopter first for every movement. In contrast to machine acting along the lines of a relatively simple algorithm, for an SME, the procedure to solve the problem may be obvious after looking at it for a few seconds.[56]

[55] In the discipline of domain-independent automated planning, the definition of such heuristics is not possible. In contrast, in HTN planning, this is a usual way to define preferences.

[56] The way to solve this problem effectively might be, for example, to generate a priority list of troop objects, for which it makes most sense to transport it via helicopter, then try to move as many of them by helicopter (highest priority first), then sort the transportation tasks to minimize helicopter route length and finally let the rest of the troops march to their targets.

Now this leads back to the beginning, chapter 1, where the author showed that the problem structure decides about an algorithm's performance. However, today's automated problem-solvers normally only follow an inherent problem-solving procedure, but they do not start with generating one after analyzing the problem structure. It takes a lot of meta-reasoning and problem-solving experience to accomplish the latter - something that human experts are very capable of. The ultimate solution would be to let the *human teach the machine on-the-fly concerning how to solve a problem with a given structure*. As long as there is no such powerful machine intelligence, a low-level approach would be to let the human specify a few important constraints, such as "these troops should be transported by this helicopter" and let the automation plan out the rest. In principle, the MMP is capable of this, but in the MUM-T experiments, there was no interface mode for this. The specification of the constraints, as well as the rest of the planning work can be performed in mixed-initiative fashion.

Another example for a missing problem-solving heuristic or procedure is depicted in Figure 5-8. Here, the assistant puts out a valid, but suboptimal plan. One can also speak of *brittle* planning solutions.[57] Although there are costs associated to detours in the MMP's world model, the optimization process sometimes takes too long to eliminate them in time. Of course, an enhancement to the algorithm could filter out detours afterwards, or it should at least try to always go the direct way first. Nevertheless, this does not change anything concerning the principle problem, that from the viewpoint of the machine, a plan may be valid and the best one it has found so far, whereas the human is able to immediately evaluate it as suboptimal by detecting detours and wrong tasks (transit tasks instead of recce route tasks) at first glance. Hence, there is a need for better communication between human and machine and an improved adaptability of the machine to the human operator's preferences. As an example, the human operator could be allowed to inform the machine where he/she detected a detour or a suboptimal task, and the machine could fix the problem, offer alternative solutions and/or even learn to avoid this misbehavior in the future.

[57] The phenomenon of brittleness and its impact on user performance has been addressed in experiments by [Hayes, Larson & Ravinder 2005] as well as [Smith, McCoy & Layton 1997].

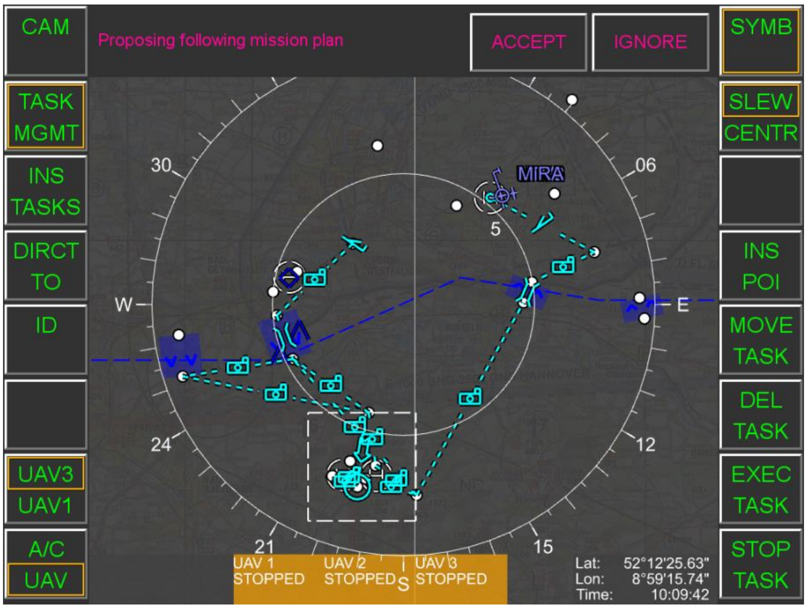

Figure 5-8: Suboptimal UAV action plan including detours

However, it was also possible to observe the heuristics set by the different DKCs working fine during the experiments. In one instance for example, the manned helicopter was on the ground to unload troops. In this situation, the crew received the follow-up mission goals. The plan that was generated with the new goals scheduled a later take-off time for the helicopter, because the new helicopter route had not been reconnoitered by any UAV yet. Hence, the manned helicopter had been scheduled to wait on ground until a UAV arrived at the same location to start making orthophotos of the helicopter route. The MMP was able to generate this intelligent solution because of the cost difference for flying unreconnoitered routes versus reconnoitered ones.[58]

5.3.9 Verification

Whereas the previous chapters were dealing with the validation of the MMP, Table 5-13 is used for the verification of the requirements concerning the MMP, which are listed in chapter 4.1. The design and implementation targeted to fulfill all requirements that had been set up.

The evaluation of the MMP has been described in detail in the preceding subchapters. Testing as method of verification means technical testing during system development.

Concerning the verification by MUM-T assistant system evaluation, [Rauschert 2013] explains that during the MUM-T experimental campaign, there were 79 interventions by the assistant systems during the eight simulation runs. Out of the 74 assistant system advices, 55 were being

[58] One could argue that waiting on ground is also dangerous for the manned helicopter. However, no costs for waiting or loitering have been implemented in the models of the MMP at all. The main reason for this is that this is one of the few commonly required feautures that have not been considered by the specification of PDDL yet.

followed by the test persons. Out of the 17 assistant system proposals, 14 were accepted by the test persons. This indicates a high quality of the proposals, which are the result of the MMP calculations, and of the advices, which are derived from the contents of the proposals.

Table 5-13: Verification of the MMP software

Req. no.	Description	Fulfilled by MMP subcomponents	Verification method
1	Plan generation for MUM-T mission	PPM, LPG	Testing, MMP evaluation
2	Temporal planning	PPM, LPG	Testing, Assistant system evaluation
3	Generated plans need to be complete	LPG	Testing, MMP evaluation
4	Generated plans need to include all agents	PPM	By design
5	Plans need to consider concurrent actions	DKC, LPG	Testing, MMP evaluation
6	Cost function to be minimized	DKC, LPG	Testing, MMP evaluation
7	Deterministic planning	DKC, LPG	By design
8	Performance sufficient for MUM-T scenario complexity	LPG	MMP evaluation, Assistant system evaluation
9	Plan completeness checking functionality	PPM	Assistant system evaluation
10	Plan completion functionality	PPM	Assistant system evaluation
11	Suboptimal / anytime planning	LPG	Testing, MMP evaluation, Assistant system evaluation
12	Soft constraints as input	DKC, PPM	Testing, MMP evaluation

6 Discussion and Further Research Recommendations

This chapter critically discusses the whole previous content of this dissertation. In addition, it shall deliver recommendations for possible further research in MUM-T and mixed-initiative interaction topics. First, the MUM-T and MMP system concepts are treated and then system design. After that, the implementation of the MMP is discussed. Finally, the experimental evaluation and the data analysis are treated.

6.1 Discussion of the System Concept

Although the author claims that the MMP is an implementation of a truly cooperative human-machine system in mixed-initiative fashion, several improvements seem necessary in order to bring such a system into operation.

As [Cortellessa & Cesta 2006] as well as [Linegang et al. 2003] point out, the ability of a mixed-initiative system to explain problem situations, such as conflicts or suboptimalities inside the current plan, is an important functionality. In this contect the author oberseved a weakness of the mixed-initiative planning of UAV tasks: the missing explanation capability of the MUM-T assistant system. When the assistant system proposes the insertion of a UAV task, it does not explain to the human operator, why this task may be beneficial to the mission outcome or which mission goal or intermediate goal makes the task insertion necessary. Such a capability would allow the operator to quickly understand what the assistant system "wants", and it would most probably speed up the operator's decision process (accept versus reject the proposal) and the quality of these decisions. An improvement of the explanation ability would have influence on both the assistant system concept and the MMP concept. This is because the task/action-level cooperation (i.e. atomic tasks as vocabulary) might be too low level in order to accomplish significant improvements in the explanation ability. However, it is important to note that the test persons stated that the action-level cooperation was completely acceptable. It is possible that with an increase of mission complexity, this opinion could change.

Another improvement could be a further increase of autonomy of the UAVs. In the UBM's MUM-T application the human operator is guiding semi-autonomous vehicles. Such systems have dynamic behavior during mission execution, i.e. plan execution. So far, these vehicles are guided in a task-based mode, which means they quite strictly follow the plan generated in mixed-initiative fashion, with the exception of minor route replanning. This can be regarded as giving hard positive action constraints to the UAVs, which is only a limited subset of what is possible with the STCI and the MMP. The hard and soft constraints that are created during the planning phase could in future implementations be transmitted to the executing agents in addition to the current plan to allow local replanning of the agents upon unforeseen events, thereby constituting execution policies, as described above. Such a “constraint-based” vehicle guidance mode was not included in the system concept. For more details on the constraint-based guidance (or constraining autonomy) concept and the related *Mixed-Initiative Operation* concept, in which the constraints defined during initial planning also influence the LOA-switching in the replanning negotiation process, see [Strenzke & Schulte 2012a].

Furthermore, the workload reduction through the MUM-T assistant system could be enhanced by anticipating conflicts between different tasks of human operator. Of course, some tasks, such as vehicle identification, cannot be anticipated, because it is not known, at which point in time an object to identify will be discovered on the flight route. However, in the mission plan, it is known at which time a UAV will have finished its reconnaissance task concerning the primary helicopter landing site. From this, one can conclude, that there is a possibility that the human operator will be occupied with checking the results of the area reconnaissance and with deciding if the landing site should be switched, which goes together with a mission replanning task. Hence, for such cases, a possible workload peak can be anticipated. In other cases, human operator tasks can be definitely anticipated. This approach has been taken by the MiRA-T assistant system already, but not by the MUM-T assistant system. It is important to note that although the MUM-T assistant system already follows the concept of management-by-consent when it gives proposals to the human operator, the anticipation of human operator tasks as reaction to the assistant system dialogs has also not been examined. This could be a very interesting feature, because it would allow the MUM-T assistant system to evaluate which LOA is appropriate in critical timing situations. In addition, this would enable the assistant system to use the operator as a resource when he/she is available for this.[59]

An additional possible enhancement to the assistant system would be extending its *back-up/relieve* approach [Sheridan 1992] to a *relieve and extend* approach [Sheridan 1992], which means that the system would have to interact not only more proactively, but also be able to be queried actively by the human operator in order to extend the overall performance of the human-machine system.

[59] A simple first approach to this would be defining actions in the MMP world model like “grab-operator-attention(vehicle, operator)” and “release”, respectively, which block and free the human operator as a resource.

What could also be an improvement to the human-machine cooperation concept is the implementation of an adaptable automation approach, in which the human operator choses the LOA, by which he/she interacts with the system. In the MUM-T setup, this was only realized for the detailed mission planning, which takes place inside the Cognitive Agents onboard of the UAVs. However, this was not integrated into the overall-mission planning and plan optimization. In addition, for planning the helicopter mission and route, the planning process was associated with a fixed LOA. In contrast, for planning the UAV missions, the MUM-T assistant system takes over the responsibility for selecting the LOA (adaptive automation). It would be interesting to let the PIC access the MMP in a more delegation-focused interaction style, if he/she wishes so, thereby implementing an adaptable automation approach. This would create more flexibility in the human-machine cooperation and would allow the MMP to show its strengths more directly, which are, amongst others, the handling of constraints on different levels of detail at the same time.

Finally, one of the author’s visions for future research is that instead of plans that cover a single thread of actions, *procedures* could be generated by the human operator and/or by the systems assisting him/her. In contrast to a normal plan, such procedures can contain alternate ways of solving subproblems depending on *conditions*, and they can also include *loops*, which are suited to let an agent perform an action or a list of actions until a certain condition is fulfilled. Hence, procedures are suited to deal with uncertainties, which would be relevant for the MUM-T reference scenario, despite its simplification concerning the mobility of hostile vehicles. The author’s high-level concept for mixed-initiative generation and modification of procedures for suchlike applications is explained in more in detail in [Strenzke, Theißing & Schulte 2012].

6.2 Discussion of the System Design

To reason about the system plan [Strenzke & Schulte 2011], which is shared between human and machine, the MMP generates an assumed human plan by making the best possible completion of the system plan that fulfills the shared goals. The assumed human plan therefore allows finding missing actions on the system plan. In addition to that, in future experiments, the assumed human plan could be compared against the MMP’s reference plan that also fulfills the shared goals but disregards the current system plan. Thereby, the quality of the system plan could be evaluated, i.e. violations of soft constraints could be detected. However, the mere comparison of costs makes explanation difficult. In addition to that, human-generated actions that violate the world model of the MMP (e.g. action preconditions not met) prevent the evaluation of such plans, because the problem solver simply fails without explanation.

Furthermore, the system design chosen for the MUM-T setup ignored the following features, which would be important to increase mixed-initiative planning performance:

- feasibility checking of a human plan
- schedule checking of a human plan
- optimality checking of human plan
- re-scheduling of a human plan

What remains an interesting research question is when an assistant system should initiate a monolog or dialog. A machine that warns early and is often wrong will certainly not be accepted as a virtual teammate and will not help increasing overall system performance, because of nuisance alerts. The same counts for a machine that is often right but warns very late. The interaction timing should also consider possible conflicts that may arise e.g. if the human and the machine modify the same plan at the same time or if the human is in high workload situations.

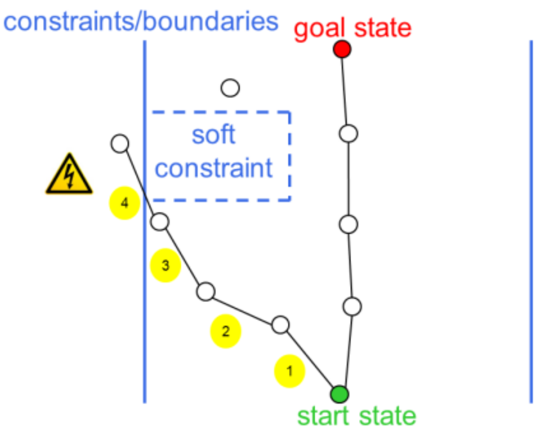

Figure 6-1: Different possibilities for automation to intervene in case of suboptimal human behavior

In order to explain some of the different possibilities, Figure 6-1 displays two planning paths inside a problem space with different world states (small circles) and hard constraints that must not be violated (blue lines) and soft constraints that should not be violated, otherwise resulting in extra costs (dotted blue lines). Although this example looks like a route planning problem it should be seen as a generic problem space visualization that is not available to the operator at the moment of planning (e.g. because of high dimensionality or dynamic problem space construction). One planning path leads quite directly to the goal state (shortest path, therefore optimal solution). Now one can imagine the operator following the other path. After each step he/she is able to leave his path and switch over to the optimal path. When should the machine tell the human that it thinks something is going wrong? After step 4 it is already too late (hard constraint violated). After step 3 the situation can still be solved but it is connected to either high path costs or soft constraint violation costs. Now the machine could also come to the conclusion that something is going wrong at step 2 or even 1, although these are valid operator actions as far as the hard constraints are concerned. But they clearly are not optimal in the view of the machine. It is important to note that the machine not necessarily has the right view. This brittleness is due to its limited world model. A machine that warns early and is often wrong will certainly not be accepted as a virtual teammate and will not help increasing overall system performance, because of nuisance alerts. The same counts for a machine that is often right but warns very late. Therefore, it would make sense to examine different configurations of brittleness and interaction timing in future experiments.

A problem in the MMP design is that there are two planners, which do not cooperate with each other and hence do not create synergies while exploring their private problem-space. This costs performance. Furthermore, one of these has the responsibility to check the plan that the human operator has entered into the system. Instead, to improve performance this task could be taken over by a dedicated plan checker. Such tools for *plan validation and repair* have been developed by [Howey, Long & Fox 2004a] and also been used for the realization of mixed-initiative planning systems [Howey, Long & Fox 2004b]. The abovementioned problem of explanation could be mitigated due to the plan checker’s ability of finding flaws in a plan and clearly expressing the problem.

As a possible improvement, probabilistic instead of deterministic planning could be applied. However, because the probability of a threat at or near a certain position is completely unknown at the beginning of the mission, probabilistic planning is not easily applicable. Nevertheless, as the experiments of the UBM have shown, human experts prefer solutions that cover such uncertainties. This also leads to the ability of generating procedures instead of plans.

In addition, as the test persons stated, a temporal planning capability would be advantageous for UAV task agenda generation, which would have implications on the information associated to the tasks (or constraints) given to the UAVs as well as on the human-machine interface(s) for UAV task planning.

6.3 Discussion of the System Implementation

The Cognitive Skill Merging approach to mixed-initiative has the goal to combine general human strengths and generals machine strengths in order to optimize overall human-machine system performance and compensate each other's weaknesses [Strenzke & Schulte 2011b]. However, there are weak points in the implementation that are in conflict with the concept of this approach. The first is the uninformed search concerning route plan generation. Routes that are longer than necessary can become optimized through the incremental mode of the LPG over time. But these optimizations are associated with relatively small costs (e.g. in comparison to landing at threatened site) and therefore can lead to a time-intensive optimization process, while the sub-optimality of the route is easily visible for the human operator. From the human-automation integration standpoint one would think that the route planning is a machine's strength and not a weakness. The missing of explicit geometrical planning and reasoning leads also to problems concerning reconnaissance coverage optimization, which is an important issue for reconnaissance UAV mission planning.

Another weak point of the MMP is the lack of continuous planning. This means that plan fragments that have already proven to be useful are not re-used. Instead every planning request by the MUM-T assistant system makes the MMP generate a completely new plan (however, certain fragments will reoccur due to the constraints the MMP receives). On the one hand, this leads to the problem that the operator can be confronted with a new machine plan that differs in many aspects from the previous one, which can cause confusion. However, this problem arises rarely in the MUM-T configuration because the Slave planner regards the human input as hard constraints, and therefore all aHuPs (assumed Human Plans) overlap in all tasks that these constraints refer to. On the other hand, the plan optimization process is interrupted and reset very often, even if there are only minor changes to the problem to solve. Hence, it is difficult to maintain or improve plan quality in the long term. This problem could be addressed by remembering constraints that improved the solution and re-applying them. But this leads into the problem of having to try out hard constraint combinations.

A further problem is associated with the plan feasibility checking feature of the MMP. Because the search of the LPG does not terminate in case only temporal constraints deny the solution and there is no other possibility than to define these constraints as hard, the only workaround is to set a timeout concerning the waiting for planner output. To relieve the problem a little, one of the twelve LPG processes is fed with an "emergency plan" problem file with increased helicopter travel speed (near helicopter maximum speed).

One drawback of operator-based planning in comparison to HTN planning also is that critical decision points (e.g. which corridor, which drop zone) are rather implicitly modeled and "lost" in combinatorial space. The test persons reported that although they slightly tend to plan hierarchically, they preferred the implemented forward-planning style interface. However, they are indeed used to plan their missions by means of suchlike critical decision points. In order to plan more complex missions during human-in-the-loop experiments with an acceptable response time, the author suggests taking a hybrid approach of classical operator-based and HTN planning, which would be similar to [Estlin, Chien & Wang 1997] and [Biundo & Schattenberg 2001], who set a trend towards efficient planning in a dynamic and unforeseeable real world. However, also in the domain of classical planning, algorithms that provide a better support for human-machine mixed-initiative planning exist, such as Plan Space Search (Partial-Order Planning algorithm [Kambhampati, Knoblock & Yang 1995]), which explicitly tracks causal relationships of actions towards goals or other actions' preconditions. Independent of applying a hybrid planning approach, the usage domain-specific heuristics could generate extreme performance benefits to an MMP implementation. The heuristics described in chapter 4.3 are only an external workaround to the LPG planning engine. This workaround has the drawback of an exploding number of domain definitions for each heuristic feature added.

Finally, it needs to be mentioned that in principle, the STCI [Strenzke & Schulte 2011], which is used to input constraints into the MMP, can handle all possible combinations of the six constraint dimensions (agent/s, content, object/s, temporality, hardness, costs). However, the MMP itself in the version it was implemented, cannot handle all possible combinations. On the one hand, this is due to pragmatic reasons of implementation complexity and on the other hand, temporal soft constraints can neither be modeled in in PDDL (Planning Domain Definition Language) version 2.2 [Edelkamp & Hoffmann 2004] nor solved by the LPG-td [Gerevini et al. 2004], which is the underlying problem-solver of the MMP. To be precise, this combination would require the trajectory constraint and preferences features of PDDL3 [Gerevini & Long 2005].

6.4 Discussion of the Experimental Evaluation

First of all, it needs to be mentioned that only very few comparable experiments concerning human-automation cooperation are known to the author. There have been implementations of mixed-initiative approaches to multi-UAV guidance and corresponding experiments, but these works regarded a human operator in a ground control station [Billman & Steinberg 2007] [Miller et al. 2011]. Another difference is, that these setups used case-based or hierarchical task network planning respectively, whereas operator-based planning was applied in the MMP - with the corresponding implications for the user interface front-end. There were also analyses of MUM-T missions with the UAV operator sitting inside an attack helicopter, but they did not consider any machine intelligence, assistant systems, or mixed-initiative systems [Kraay, Pouliot & Wallace 1998] [Durbin & Hicks 2009]. In all mentioned studies, measurements comparable to those in the UBM's MUM-T experiments were taken (e.g. operator workload, operator situation awareness).[60] However, in addition, a large amount of data was gathered from SMEs by the use of questionnaires. This data can be used to support further development of automation that supports operators in MUM-T missions.

The MMP can be regarded as a mixed-initiative planning/operation test bed due to its generalized constraint-based interfaces and its modularity, i.e. the planning backend and/or the UI can be modified or exchanged in order to evaluate different approaches to the multi-vehicle guidance and planning problem. As described in this dissertation, in the MUM-T human-machine experiments, so far only two configurations have been compared: manual vs. mixed-initiative planning. To optimize the mixed-initiative system and analyze human operator behavior more thoroughly, different configurations could be constituted by modifying the interaction timing (especially cost threshold used in plan comparison), the mission dynamics (e.g. relaxed mission planning or time-critical replanning) and the assistance solution/advice brittleness (e.g. displaying certain predetermined brittle solutions). In addition to that, the UI could be modified to enable more direct interaction with the automated planner (delegation-focused interaction style). In this configuration, the human operator might be allowed to work on different interaction levels. Such experiments could also be conducted in environments of smaller scale. The complex experimental setup with the full-mission simulation approach has

[60] In [Bergantz et al. 2002], further interesting experiments are described. Unfortunately, their description is too brief for a comparison with the experiments of the UBM. Therefore, the corresponding studies are not considered here.

indeed lowered the level of control as well as the comparativeness between missions and between situations in the missions.

Furthermore, concerning the measurement methods, alternate ways could have been chosen or evaluated. For example, [Linegang et al. 2006] chose the NASA-TLX method to assess the multi-UAV operator's workload but they also state that the Modified Cooper-Harper (MCH) Rating has the advantage of semantic information attached to the workload level, which could be more important than the multi-dimensionality of NASA-TLX. The MCH scale is described in [Casali & Wierwille 1983], a more recent adoption for multi-UAV guidance can be found in [Cummings, Myers & Scott 2006] and the Bedford Workload Scale that is very similar to MCH is explained in [Roscoe 1984]. In addition, scales to assess the human operator's trust in the automation [Cortellessa & Cesta 2005] [Jian et al. 2000] could be applied. As an example, it would be interesting to know, if the differences concerning the trust in the system, as mentioned in chapter 5.3.8, level out after letting the test persons get more training and/or gather more experience with the system.

6.5 Discussion of the Data Analysis

[Sheridan 1992] states that in human-machine systems with decision support it is not possible to differentiate between the human's true decision to accept decision aid (i.e. he/she considers the machine's suggestion as good) and the human's behavior of slavishly following the decision aids (e.g. because he/she has high trust in the machine or has no cognitive resources left to think for him/herself). To evaluate the quality of the automatically generated plans, it is therefore necessary to analyze how well they match with the plans that the human has. In order to achieve this, more data has to be collected and analyzed.

To analyze situations of false alarms as well as actions missed by the MUM-T assistant system, it would be interesting to compare the plan quality generated over time by the Free and the Slave mission planner instances because a future version of the assistant system may have the necessity perform exactly this comparison in order to decide whether to take initiative to propose not only a single additional UAV task but a completely new mission plan. In the implementation described in this dissertation, this is done only in special use cases.

Two thresholds can be set in order to tune the decision process: the time to wait until a decision is made and the cost difference between the Free and the Slave plan. Figure 6-2 shows an example of a good Slave plan (aHuP) beats the Free plan (ReP) after 16 seconds of incremental planning. This means that the human planning heuristics were more effective than those of the machine. Because of the changes in the situation (e.g. aircraft moving, threat blocking primary corridor) and the goals (e.g. follow-up mission), it is not possible to compare the aHuP against any baseline or against the optimal solution because it is not known. Therefore, it is necessary to analyze these graphs (postmortem analysis of the experimental data) in certain mission situations, in which replanning should be proposed by the MUM-T assistant system. The threshold values could be set to start checking for a cost difference of e.g. 10.000 after waiting 20 seconds after starting both planners (see Figure 6-3). This analysis process was not completed.

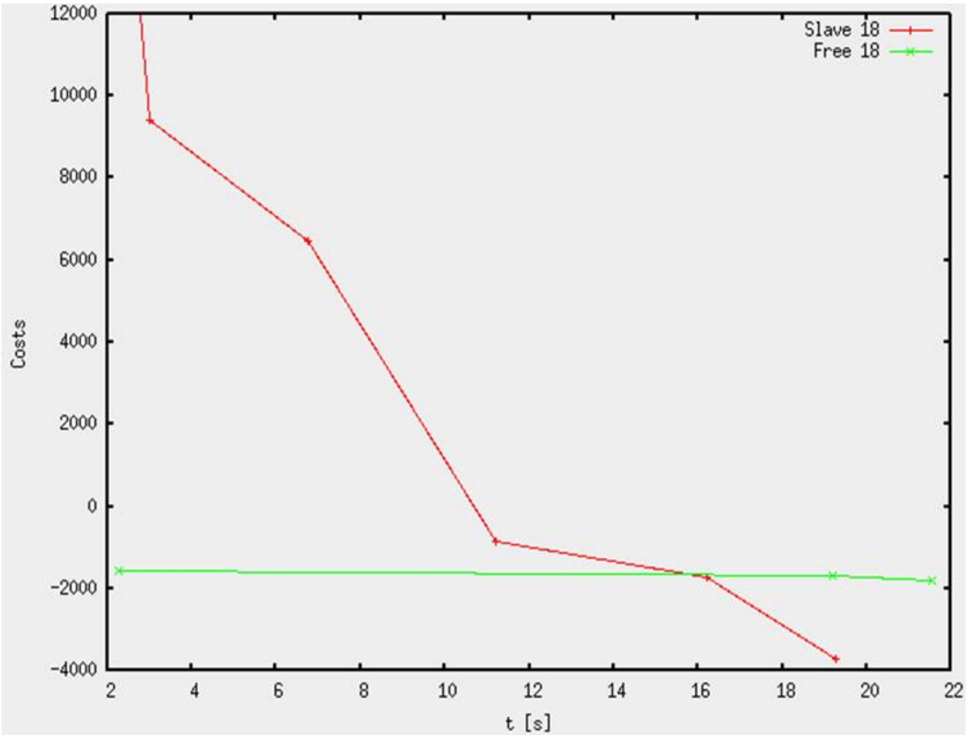

Figure 6-2: Example of a good assumed human plan (Slave plan, in red, compared to Free plan, in green)

Another interesting data analysis that could be performed is to measure the time that the operator spends on the interaction with the MUM-T assistant system. This time can be divided into dealing with useful advices that are accepted, dealing with useful advices that are not properly understood by the operator (and therefore declined) and dealing with brittle advices. This analysis has also not been performed for the MMP yet.

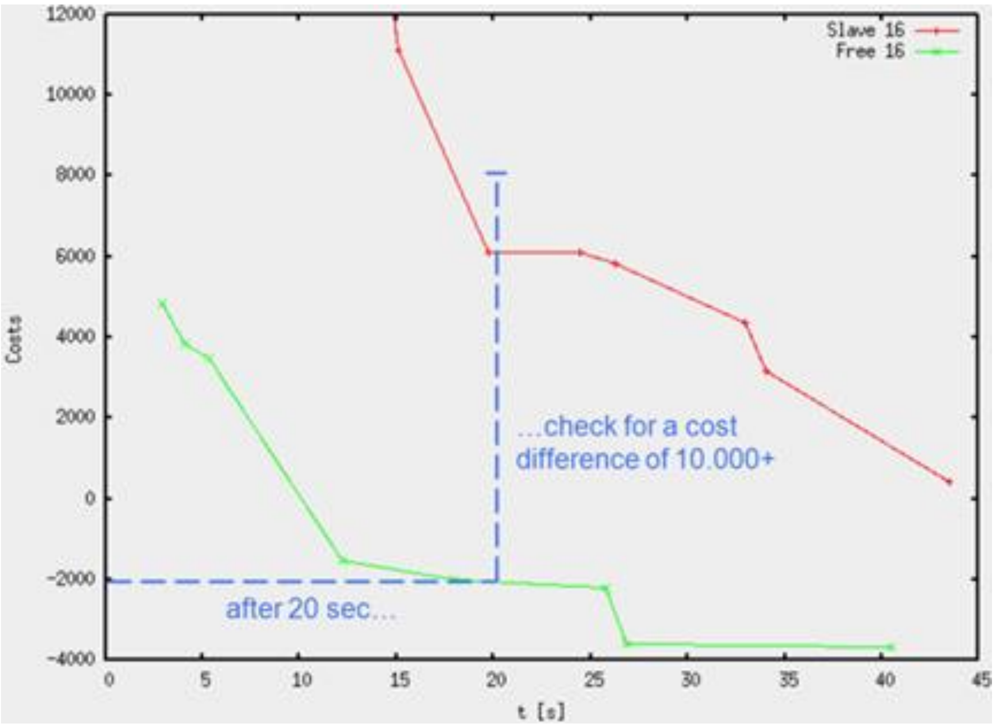

Figure 6-3: Suggestion for applying time and cost thresholds to compare quality

7 Summary

The Manned-Unmanned Teaming (MUM-T) setup of the Universität der Bundeswehr Munich (UBM) consists of *a manned transport helicopter with the pilot in command (PIC), who is guiding multiple Unmanned Aerial Vehicles* (UAVs). These UAVs have the task of supporting the troop transport mission of the manned helicopter by conducting reconnaissance in hostile territory. *In order to relieve the PIC of his/her heavy workload*, which stems from the PIC's tasks as mission commander and as multi-UAV operator, *multiple supporting functions have been designed*, implemented and evaluated by the UBM.

Some of these supporting functions are provided by the *Mixed-Initiative Mission Planner (MMP)*. The MMP supports the PIC in *planning the helicopter mission* on his/her demand, and it also supports the PIC in the *creation, modification and execution of the UAV mission plans*. In the UBM's overall system concept for MUM-T, the latter function is transported through the MUM-T assistant system as dialog initiator and dialog controller. All supporting functions have been prototypically implemented by the UBM and in this form *evaluated by subject matter experts* (SMEs).

The design of the MMP core is based on two identical planners, which are running in parallel and which are used in a different way, i.e. they receive different inputs. One is the "Free" mission planner instance, which receives the current tactical situation and a constraint set solve, which includes only the constraints from the mission order. Hence, the Free mission planner instance generates a solution independent from the plan, which is currently followed by the PIC. This plan can be used by the MUM-T assistant system as a proposal for complete mission replanning in case of major change in the conditions (e.g. designated helicopter landing site is threatened or a follow-up mission is not yet considered in the current mission plan). The other planner is the "Slave" mission planner instance, which receives as additional input the constraints concerning the UAV tasks entered by the PIC into the task-based UAV guidance interface. Therefore, the Slave mission planner creates a solution, which extends the plan currently followed by the PIC. This plan can be used by the MUM-T assistant system to inform the PIC concerning UAV tasks missing in his/her plan and UAV tasks that need to be activated (selected for immediate execution) in order to stay in the schedule of the time-critical mission.

The prototypical implementation of the MMP uses a *centralized planning approach, in which the actions of all agents are planned in a single process*. A *domain-independent operator-based anytime /suboptimal planner* is used due to its flexibility and the limited time available for in-flight mission replanning. It does not deal with uncertainties, i.e. it uses a *deterministic world model*. Probabilistic planning like "primary helicopter landing site could be threatened (although not scanned yet), scanning the secondary also could make sense" as well as domain-specific planning heuristics are left to the human operator. The replanning approach of the MMP upon situation change is to start from scratch again, not re-using previous search space explorations or results. The only exception is that UAV tasks which have been accepted by the PIC in the meantime are now treated as constraints for the "Slave" planner, and thereby the search space can be limited.

The systems developed by the UBM, including the MMP, were integrated into a generic helicopter simulator with a slightly simplified cockpit and *full-mission simulation* capability. With this setup, *human-in-the-loop experiments with eight German Army Aviation pilots* have been conducted. *The independent variable was the configuration of the MUM-T assistant system, which was either enabled or disabled, hence either using the MMP planning results for the UAV missions or not*. Each of the eight test persons had to play the role of the PIC in each of the two mentioned configurations. In order to avoid unwanted training effects, for each test person crew every experimental simulation run had differences in the mission scenario. The measures that were taken included the *objective mission performance*, the *crew's subjective workload* and the *objective situation awareness*. In addition, *via questionnaires the test persons evaluated the supporting systems, made statements concerning their strategies for mission planning and their requirements concerning a mission planning system like the MMP*.

From these experiments, the following results could be gathered. *The PIC's subjective workload was significantly lower with MUM-T assistant system enabled as far as the "temporal demand" was concerned. Furthermore, his/her situation awareness concerning the currently active UAV mission plans was significantly higher. The functionality of advices of the assistant system that were referring to the adding of single tasks into the UAV mission plans was evaluated completely positively* (rather helpful than hindering). Proposals of completely new mission plans were not rated clearly positively or negatively. In the author's opinion this result was related to imperfections in the mission plans proposed by the system. *In total, the MUM-T assistant system was regarded more helpful than hindering*. The quality of the plans generated by the MMP was considered good in three out of four situations in the mission and the MMP's planning speed in two out of four. The MMP's feature of optimizing the helicopter route with a single push of a button was well accepted. However, the test persons would have preferred to have the optimized plan from the beginning on, and they also wished for more modification possibilities concerning the planned helicopter route. The test persons also stated that they would have liked a Gantt-chart based planning GUI in addition to the map-based planning display, which was available during the experiments.

Questioned concerning their planning strategies, the SMEs stated that it is best to enter incomplete UAV plans and that they plan the mission in a hierarchical top-down way and by means of critical decision points (e.g. which corridor to choose, which landing site to choose). However, they also stated that *the planning interface does not have to support planning in a hierarchical top-down way and that sequential entering of UAV tasks, as it was implemented, is preferred*. The SMEs did not plan the mission phases isolated from each other. While replanning the mission in-flight, the SMEs felt mostly under time pressure and preferred having aid from an assistant system for this task. For the initial planning, they stated no need for such assistance. It is important to note that whereas incomplete UAV plans were regarded as acceptable, for the helicopter the mission plan should always be complete.

In the context of this study, the SMEs were also asked to prioritize (weigh) the goals to be fulfilled during UAV mission planning. *Rated as very important was the reconnaissance of the primary landing zone and the helicopter route as well as keeping one third, i.e. one of the three UAVs as reserve for currently unforeseen tasks*. Important was the reconnaissance of alternate helicopter routes. Rather unimportant was the compliance with corridor opening and closing times. Regarded as very unimportant was the minimization of the UAV route lengths and guaranteeing a minimal distance of UAVs to the manned helicopter (for flight safety).

According to the further questionnaire results, it would have been beneficial for the SMEs to manage alternative plans also, but not having this ability was not regarded as problematic. In their view, alternative plans should be on a per-UAV basis instead of managing one overall plan for all UAVs. A system that generates only plans covering a limited time horizon (e.g. ten minutes in advance) would not have been accepted. Nevertheless, an incomplete UAV plan that is fulfilling the next goals would be acceptable. One of the requirements for the planning system performance can be described by the mean value for the maximum acceptable duration of the planning process when helicopter is in hostile territory, which is about 35 seconds. Concerning the validity of the results, it is important to mention that the test persons rated the military scenario, the missions and the simulation used for the experiments as realistic and appropriate.

Finally, the author critically discussed the MMP’s concept, design and implementation as well as the experimental evaluation and the data analysis. The system could be improved concerning its ability to explain itself to the human operator and to anticipate operator workload. Technical approaches to realize both of these features are sketched. Furthermore, the idea of constraint-based UAV guidance, which gives more autonomy to the UAVs concerning their mission execution and multiple levels of detail for the UAV operator to specify the UAV tasks, is described. Because the MMP itself is able to perform in such a way, only the front-end would have to be adapted accordingly. The author also sees an advantage in using procedures (with conditions and loops) instead of fixed plans for future improvements of the system. Probabilistic planning can be applied to generate such procedures. In the view of the author, it would also be interesting to invest more research in the interaction timing (i.e. the point in time when the assistant system initiates a dialog with the human operator) and in the time that the operator spends on the interaction with system as a reaction to high-quality or brittle planning solutions (i.e. plans that have flaws). Furthermore, the author discussed the implementation of the MMP regarding its problems in geographical calculations, missing continuous planning functionality and lack of performance. The performance problem can be relieved for example by domain-specific heuristics and/or hybrid planning approaches (i.e. operator-based planning plus hierarchical task network planning). Furthermore, concerning the evaluation procedure, the test persons' trust in the system should have been measured, because it seemed to play a role in the individual behavior.

8 Appendix

In order to evaluate the complexity of this work as well as the functionality and the cost model of the MMP and to allow the reader to create own planning system prototypes based on the ideas stated in this dissertation, this chapter provides some PDDL modeling examples. This PDDL code can be directly used as input for the LPG-td 1.0 planner and it will put out the same results that the MMP also generates.[61] It is important to note in contrast to the static nature of the domain descriptions used by the MMP, the problem descriptions are completely auto-generated by the MMP's PPM component (*PddlWrapper* class).

Hence, what is given here beside one DKC, i.e. one variant of the domain model, are one example problem from the point in time before the mission starts as well as one example problem from a situation during the mission execution, each with example solutions (plans).

Both problem models have been generated for the Slave planner instance. The first one represents a snapshot from the point in time after the mission has been planned and entered into the system, but before the mission has started. The second problem definition is from a later point in time during the mission. There are more facts now and at the same time less tasks ahead (i.e. less goal constraints).

The solution outputs are not completely ordered by time and might need to be sorted in order to completely understand the mission plan. Common abbreviations in the models are explained in Table 8-1.

[61] To be precise, this is true only if the LPG-td is forced to use the same random seed as provided here in the examples.

Table 8-1: Common abbreviations used in planning domain model

ap	approach point
c	cost
camm	camera mode mapping
camt	camera mode tracking
corr	corridor
d	duration
dep	departure
dir	direction
dist	distance
dp	departure point
dz	drop zone
ent	enemy territory
frt	friendly territory
hoa	helicopter operations area
mob	main operating base
poi	point of interest
pz	pickup zone
terr	territory
unld	unload
v	velocity

8.1 Manned-Unmanned Teaming Planning Domain Model

In the following, DKC number 1 is shown (file "mumt1.DOMAIN"). As can be seen in Table 4-3, this DKC has soft constraints (costs) for forbidding the manned helicopter to transit through unreconnoitered territory as well as forbidding the ground troops to march through unreconnoitered territory.

```
(define (domain MUMT)
(:requirements :adl :typing :fluents :timed-initial-literals :existential-
preconditions
:durative-actions :duration-inequalities :conditional-effects :negative-
preconditions
:disjunctive-preconditions :derived-predicates)
(:types location aircraft troops)

(:predicates
;; location-related predicates
(is_dz ?l - location)
(is_pz ?l - location)
(in_foe_terr ?l - location)
(in_hoa ?l - location)
(is_hoa_entry ?l - location)
(is_hoa_exit ?l - location)
(is_dp_for ?d - location ?w - location)
(is_ap_for ?a - location ?w - location)
(location_cleared ?l - location)
(dist_known ?f - location ?t - location)
(obj_surveiled_by ?l - location ?o - aircraft)
(obj_currently_surv ?l - location)
(cleared_beginning ?f - location ?t - location) ;; not used in this config
(passed_by_heli ?l - location)

;; aircraft-related predicates
(travel_alt ?o - aircraft)
(landed ?o - aircraft)
(empty ?o - aircraft)
(ac_pos ?o - aircraft ?l - location)
(is_heli ?o - aircraft)
(is_uav ?o - aircraft)
(transporting ?o - aircraft ?c - troops)
(mapping_ground ?o - aircraft)
(ready_next_task ?o - aircraft) ;; AC is ready to do any (future) tasks
(ready_continue_task ?o - aircraft) ;; AC is ready to continue with current
 ;; task (at a specific point in time this becomes true)
(approaching ?o - aircraft ?l - location)
(departing ?o - aircraft ?l - location)
(loading ?o - aircraft)
(unloading ?o - aircraft)

;; troop-related predicates
(troop_pos ?c - troops ?l - location)
(troop_can_move ?c - troops)

;; hard constraint related predicates
(corr_open ?f - location ?t - location ?o - aircraft)
(takeoff_denied ?o - aircraft ?l - location)
(landing_denied ?o - aircraft ?l - location)
```

```
(load_constraint ?o - aircraft ?c - troops ?l - location)
(unload_constraint ?o - aircraft ?c - troops ?l - location)
(location_cleared_by ?l - location ?o - aircraft)
(section_cleared_by ?f - location ?t - location ?o - aircraft)
(flot_crossed_by ?f - location ?t - location ?o - aircraft)
(has_departed_via ?o - aircraft ?l - location)
(has_approached_via ?o - aircraft ?l - location)
(has_transited ?o - aircraft ?f - location ?t - location)
;; ATBEGIN constraints
(current_task_crossflot ?o - aircraft ?f - location ?t - location)
(current_task_reccearea ?o - aircraft ?l - location)
(current_task_objsurveil ?o - aircraft ?l - location)
(current_task_departure ?o - aircraft ?f - location ?t - location)
)

(:functions

;; distances & speeds
(v_frt ?o - aircraft)
(v_ent ?o - aircraft)
(dist ?f - location ?t - location)
(troop_has_moved ?c - troops)

;; durations
(d_unloadtrps ?c - troops)
(d_loadtroops ?c - troops)
(d_ac_landing ?o - aircraft)
(d_ac_takeoff ?o - aircraft)
(d_recce_area ?o - aircraft)
(d_obj_surveil) ;; unit in seconds

;; costs
(c_flight ?o - aircraft)
(c_loadtroops ?l - location)
(total-cost)
(troop_speed ?o - troops)
(c_troop_move ?o - troops)
(c_unreccd_route)
(c_obj_surveil)

;; other
(section_cleared_completely ?f - location ?t - location) ;; by how many
 ;; UAVs
(section_cleared_beginning ?f - location ?t - location) ;; by how many UAVs
(obj_surveiled ?l - location) ;; for how long
(has_done_reccetasks ?o - aircraft) ;; how many recce tasks have been done
 ;; by AC

;; soft constraint related costs
(c_corridor ?f - location ?t - location ?o - aircraft)
(c_landing ?l - location ?o - aircraft)
(c_unloadtrps ?l - location ?o - aircraft ?c - troops)

;; ATBEGIN constraint related durations
(d_finish_reccearea ?o - aircraft ?l - location)
(d_finish_objsurv ?o - aircraft ?l - location)
)


;; ACTIONS

(:durative-action approachto
```

```
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
:condition (and (at start(ac_pos ?o ?f))
(over all(is_ap_for ?f ?t))
(at start(travel_alt ?o))
(at start(not(landed ?o)))
(over all(dist_known ?f ?t))
(over all(not (landing_denied ?o ?t)))
(over all(or(not(in_foe_terr ?t)) (location_cleared ?t))) ;; in ENT it has
 ;; to be cleared
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (not (travel_alt ?o)))
(at end (approaching ?o ?t))
(at end (ac_pos ?o ?t))
(at end (increase (total-cost) (*(dist ?f ?t)(c_flight ?o))))
(at end (has_approached_via ?o ?f))
)
)

(:durative-action approach_unreccd_ls
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
:condition (and (at start(ac_pos ?o ?f))
(over all(is_ap_for ?f ?t))
(at start(travel_alt ?o))
(at start(not(landed ?o)))
(over all(dist_known ?f ?t))
(over all(not (landing_denied ?o ?t)))
(over all(and(in_foe_terr ?t) (not(location_cleared ?t))))
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (not (travel_alt ?o)))
(at end (approaching ?o ?t))
(at end (ac_pos ?o ?t))
(at end (increase (total-cost) (+(*(dist ?f ?t)(c_flight ?o)) 7500)))
 ;; alternate: 5.000, threatened: 10.000 => inbetween is 7.500
(at end (has_approached_via ?o ?f))
)
)

(:durative-action departfrom
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
:condition (and (at start(ac_pos ?o ?f))
(at start(departing ?o ?f))
(over all(not (takeoff_denied ?o ?f)))
(over all(is_dp_for ?t ?f))
(over all(dist_known ?f ?t))
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (ac_pos ?o ?t))
(at end (travel_alt ?o))
(at end (not (departing ?o ?f)))
(at end (increase (total-cost) (*(dist ?f ?t)(c_flight ?o))))
(at end (has_departed_via ?o ?t))
)
)
```

```
(:durative-action finish_departure
:parameters (?o - aircraft ?f - location ?t - location ?c - location)
 ;; (c)urrent pos
:duration (= ?duration (*(/(dist ?c ?t)(v_ent ?o))3600))
:condition (and (at start(and(ac_pos ?o ?c)))
(over all(dist_known ?c ?t))
(over all(not (takeoff_denied ?o ?f)))
(at start(current_task_departure ?o ?f ?t))
)
:effect (and
(at start (not (ac_pos ?o ?c)))
(at end (ac_pos ?o ?t))
(at end (travel_alt ?o))
(at end (not(landed ?o)))
(at end (has_departed_via ?o ?t))
(at end (not (current_task_departure ?o ?f ?t)))
(at end (increase (total-cost) (*(dist ?c ?t)(c_flight ?o))))
)
)

(:durative-action deptakeoff
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_ac_takeoff ?o))
:condition (and (over all(ac_pos ?o ?l))
(at start(not (travel_alt ?o)))
(at start(landed ?o))
(over all(not (loading ?o)))
(over all(not (unloading ?o)))
(over all(not (takeoff_denied ?o ?l)))
)
:effect (and
(at start (not (landed ?o)))
(at end (departing ?o ?l))
)
)

(:durative-action applanding
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_ac_landing ?o))
:condition (and (over all(ac_pos ?o ?l))
(over all(or (is_dz ?l) (is_pz ?l)))
(at start(approaching ?o ?l))
(at start(not (landed ?o)))
(over all(not (landing_denied ?o ?l)))
)
:effect (and
(at end (landed ?o))
(at end (not(approaching ?o ?l)))
)
)

(:durative-action dirtakeoff
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_ac_takeoff ?o))
:condition (and (over all(ac_pos ?o ?l))
(at start(not (travel_alt ?o)))
(at start(landed ?o))
(over all(not (loading ?o)))
(over all(not (unloading ?o)))
(over all(not (takeoff_denied ?o ?l)))
(over all(not(exists (?d - location) (is_dp_for ?d ?l))))
)
```

```
:effect (and
(at start (not (landed ?o)))
(at end (travel_alt ?o))
)
)

(:durative-action dirlanding
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_ac_landing ?o))
:condition (and (over all(ac_pos ?o ?l))
(over all(or (is_dz ?l) (is_pz ?l)))
(at start(travel_alt ?o))
(at start(not (landed ?o)))
(over all(not (landing_denied ?o ?l)))
(over all(not(exists (?a - location) (is_ap_for ?a ?l))))
)
:effect (and
(at end (landed ?o))
(at end (not (travel_alt ?o)))
)
)

(:durative-action ent_flight
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
:condition (and (over all(dist_known ?f ?t))
(over all(in_foe_terr ?f))
(over all(in_foe_terr ?t))
(at start(and(ac_pos ?o ?f)))
(over all(travel_alt ?o))
(over all(or (and (not(in_hoa ?t)) (not(in_hoa ?f)))
      (and (in_hoa ?t) (in_hoa ?f))
      (and (not(in_hoa ?t)) (is_hoa_exit ?t))
      (and (not(in_hoa ?f)) (is_hoa_entry ?f))))
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (ac_pos ?o ?t))
(at start (increase (total-cost) (* (*(dist ?f ?t)(c_flight ?o)) (-
c_unreccd_route (section_cleared_beginning ?f ?t)) ) ))
(at end (has_transited ?o ?f ?t))
)
)

(:durative-action ent_flight_wrong_hoa_dir
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
:condition (and (over all(dist_known ?f ?t))
(over all(in_foe_terr ?f))
(over all(in_foe_terr ?t))
(at start(and(ac_pos ?o ?f)))
(over all(travel_alt ?o))
(over all(or
      (and (not(in_hoa ?f)) (in_hoa ?t) (is_hoa_exit ?f))
      (and (in_hoa ?f) (not(in_hoa ?t)) (is_hoa_entry ?t))))
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (ac_pos ?o ?t))
(at start (increase (total-cost) (+ (* (*(dist ?f ?t)(c_flight ?o)) (-
c_unreccd_route (section_cleared_beginning ?f ?t)) ) 2000 )))
(at end (has_transited ?o ?f ?t))
```

```
)
)

(:durative-action frt_flight
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_frt ?o))3600))
:condition (and (at start(and(ac_pos ?o ?f)))
(over all(not(in_foe_terr ?f)))
(over all(not(in_foe_terr ?t)))
(over all(travel_alt ?o))
(over all(dist_known ?f ?t))
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (ac_pos ?o ?t))
(at end (increase (total-cost) (*(dist ?f ?t)(c_flight ?o))))
(at end (has_transited ?o ?f ?t))
)
)

(:durative-action cross_flot
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
:condition (and (at start(and(ac_pos ?o ?f)))
(over all(or (and (in_foe_terr ?f) (not(in_foe_terr ?t))) (and
(not(in_foe_terr ?f)) (in_foe_terr ?t))))
(over all(travel_alt ?o))
(over all(dist_known ?f ?t))
(over all(corr_open ?f ?t ?o))
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (ac_pos ?o ?t))
(at end (flot_crossed_by ?f ?t ?o))
(at end (increase (total-cost) (+ (*(dist ?f ?t)(c_flight ?o)) (c_corridor
?f ?t ?o))))
)
)

(:durative-action finish_cross_flot
:parameters (?o - aircraft ?f - location ?t - location ?c - location)
 ;; (c)urrent pos
:duration (= ?duration (*(/(dist ?c ?t)(v_ent ?o))3600))
:condition (and (at start(and(ac_pos ?o ?c)))
(over all(travel_alt ?o))
(over all(dist_known ?c ?t))
(over all(corr_open ?f ?t ?o))
(at start(current_task_crossflot ?o ?f ?t))
)
:effect (and
(at start (not (ac_pos ?o ?c)))
(at end (ac_pos ?o ?t))
(at end (flot_crossed_by ?f ?t ?o))
(at end (not (current_task_crossflot ?o ?f ?t)))
(at end (increase (total-cost) (+ (*(dist ?c ?t)(c_flight ?o)) (c_corridor
?f ?t ?o))))
)
)

(:durative-action recceroute
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
```

```
:condition (and (over all(is_uav ?o)) ;; only UAVs used for recce purposes
(over all(dist_known ?f ?t))
(at start(ac_pos ?o ?f))
(over all(and (in_foe_terr ?f) (in_foe_terr ?t)))
(over all(travel_alt ?o))
(over all(mapping_ground ?o))
(over all(or (and (not(in_hoa ?t)) (not(in_hoa ?f)))
      (and (in_hoa ?t) (in_hoa ?f))
      (and (not(in_hoa ?t)) (is_hoa_exit ?t))
      (and (not(in_hoa ?f)) (is_hoa_entry ?f))))
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (ac_pos ?o ?t))
(at start (increase (section_cleared_beginning ?f ?t) 1))
(at start (cleared_beginning ?f ?t))
(at end (cleared_beginning ?t ?f))
(at end (increase (section_cleared_beginning ?t ?f) 1))
(at end (section_cleared_by ?f ?t ?o))
(at end (increase (has_done_reccetasks ?o) 1))
(at end (increase (total-cost) 1))
)
)

(:durative-action recceroute_frt
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
:condition (and (over all(is_uav ?o)) ;; only UAVs used for recce purposes
(over all(dist_known ?f ?t))
(at start(ac_pos ?o ?f))
(over all(and (not(in_foe_terr ?f)) (not(in_foe_terr ?t))))
(over all(travel_alt ?o))
(over all(mapping_ground ?o))
(over all(or (and (not(in_hoa ?t)) (not(in_hoa ?f)))
      (and (in_hoa ?t) (in_hoa ?f))
      (and (not(in_hoa ?t)) (is_hoa_exit ?t))
      (and (not(in_hoa ?f)) (is_hoa_entry ?f))))
)
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (ac_pos ?o ?t))
(at start (increase (section_cleared_beginning ?f ?t) 1))
(at start (cleared_beginning ?f ?t))
(at end (section_cleared_by ?f ?t ?o))
(at end (increase (total-cost) (*(dist ?f ?t)(c_flight ?o))))
)
)

(:durative-action recceroute_wrong_hoa_dir
:parameters (?o - aircraft ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(v_ent ?o))3600))
:condition (and (over all(is_uav ?o)) ;; only UAVs used for recce purposes
(over all(dist_known ?f ?t))
(at start(ac_pos ?o ?f))
(over all(in_foe_terr ?f))
(over all(in_foe_terr ?t))
(over all(travel_alt ?o))
(over all(mapping_ground ?o))
(over all(or
      (and (not(in_hoa ?f)) (in_hoa ?t) (is_hoa_exit ?f))
      (and (in_hoa ?f) (not(in_hoa ?t)) (is_hoa_entry ?t))))
)
```

```
:effect (and
(at start (not (ac_pos ?o ?f)))
(at end (ac_pos ?o ?t))
(at start (increase (section_cleared_beginning ?f ?t) 1))
(at end (section_cleared_by ?f ?t ?o))
(at end (increase (has_done_reccetasks ?o) 1))
(at end (increase (total-cost) (+ (*(dist ?f ?t)(c_flight ?o)) 2000)))
)
)

(:durative-action recce_area
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_recce_area ?o))
:condition (and (at start (travel_alt ?o))
(over all (ac_pos ?o ?l))
(over all(is_uav ?o))
(over all(in_foe_terr ?l))
(over all(mapping_ground ?o))
)
:effect (and
(at end (location_cleared ?l))
(at end (location_cleared_by ?l ?o))
)
)

(:durative-action recce_area_frt
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_recce_area ?o))
:condition (and (at start (travel_alt ?o))
(over all (ac_pos ?o ?l))
(over all(is_uav ?o))
(over all(not (in_foe_terr ?l)))
(over all(mapping_ground ?o))
)
:effect (and
(at end (location_cleared ?l))
(at end (location_cleared_by ?l ?o))
)
)

(:durative-action finish_recce_area
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_finish_reccearea ?o ?l))
:condition (and (at start (travel_alt ?o))
(at start (current_task_reccearea ?o ?l))
(over all (ac_pos ?o ?l))
(over all(is_uav ?o))
(over all(in_foe_terr ?l))
(over all(mapping_ground ?o))
)
:effect (and
(at end (location_cleared ?l))
(at end (location_cleared_by ?l ?o))
(at end (not (current_task_reccearea ?o ?l)))
)
)

(:durative-action finish_recce_area_frt
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_finish_reccearea ?o ?l))
:condition (and (at start (travel_alt ?o))
(at start (current_task_reccearea ?o ?l))
```

```
(over all (ac_pos ?o ?l))
(over all(is_uav ?o))
(over all(not (in_foe_terr ?l)))
(over all(mapping_ground ?o))
)
:effect (and
(at end (location_cleared ?l))
(at end (location_cleared_by ?l ?o))
(at end (not (current_task_reccearea ?o ?l)))
)
)

(:durative-action objsurveil
:parameters (?o - aircraft ?l - location)
:duration (= ?duration d_obj_surveil)
:condition (and (over all(is_uav ?o)) ;; only UAVs used for surveil
 ;; purposes
(over all(ac_pos ?o ?l))
(over all(travel_alt ?o))
(over all(not (mapping_ground ?o)))
(at start(not (obj_currently_surv ?l))) ;; only one UAV shall surveil at
 ;; the same time
)
:effect (and
(at start (obj_currently_surv ?l))
(at end (not(obj_currently_surv ?l)))
(at end (increase (obj_surveiled ?l) 1))
(at end (obj_surveiled_by ?l ?o))
(at end (increase total-cost c_obj_surveil))
)
)

(:durative-action finish_objsurveil
:parameters (?o - aircraft ?l - location)
:duration (= ?duration (d_finish_objsurv ?o ?l))
:condition (and (over all(is_uav ?o)) ;; only UAVs used for surveil
 ;; purposes
(at start (current_task_objsurveil ?o ?l))
(over all(ac_pos ?o ?l))
(over all(travel_alt ?o))
(over all(not (mapping_ground ?o)))
(at start(not (obj_currently_surv ?l))) ;; only one UAV shall surveil at
 ;; the same time
)
:effect (and
(at start (obj_currently_surv ?l))
(at end (not(obj_currently_surv ?l)))
(at end (increase (obj_surveiled ?l) 1))
(at end (obj_surveiled_by ?l ?o))
(at end (increase total-cost c_obj_surveil))
(at end (not (current_task_objsurveil ?o ?l)))
)
)



(:durative-action loadtroops
:parameters (?o - aircraft ?l - location ?c - troops)
:duration (= ?duration (d_loadtroops ?c))
:condition (and
(at start(landed ?o))
(over all(landed ?o))
```

```
(at end(landed ?o))
(over all(loading ?o))
(over all (is_pz ?l))
(over all (ac_pos ?o ?l))
(at start (troop_pos ?c ?l))
(at start (empty ?o))
(over all(not(exists (?e - location) (load_constraint ?o ?c ?e)))))
:effect (and
(at end (not (empty ?o)))
(at end (not (troop_pos ?c ?l)))
(at end (transporting ?o ?c))
(at end (load_constraint ?o ?c ?l))
(at end (increase(total-cost) (c_loadtroops ?l)))
)
)

(:durative-action unloadtrps
:parameters (?o - aircraft ?l - location ?c - troops)
:duration (= ?duration (d_unloadtrps ?c))
:condition (and (over all (ac_pos ?o ?l))
(at start(transporting ?o ?c))
(at start(landed ?o))
(over all(landed ?o))
(at end(landed ?o))
(over all(unloading ?o))
(over all(is_dz ?l))
(over all(not(exists (?e - location) (unload_constraint ?o ?c ?e)))))
:effect (and
(at end (empty ?o))
(at end (not (transporting ?o ?c)))
(at end (troop_pos ?c ?l))
(at end (unload_constraint ?o ?c ?l))
(at end (increase(total-cost) (c_unloadtrps ?l ?o ?c)))
)
)

(:durative-action troopsmove
:parameters (?o - troops ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(troop_speed ?o))3600))
:condition (and (at start(troop_pos ?o ?f))
(over all (troop_can_move ?o))
(over all (or (and (in_foe_terr ?f) (in_foe_terr ?t))
(and (not(in_foe_terr ?f)) (not(in_foe_terr ?t)))))
(over all(dist_known ?f ?t))
(over all (forall (?a - aircraft) (not(transporting ?a ?o))))
(at end (location_cleared ?t))
)
:effect (and
(at start (not (troop_pos ?o ?f)))
(at end (troop_pos ?o ?t))
(at end (increase (total-cost) (* (*(dist ?f ?t)(c_troop_move ?o)) (-
c_unreccd_route (section_cleared_beginning ?f ?t))) ) )
(at end (increase (troop_has_moved ?o) (dist ?f ?t)))
)
)

(:durative-action troopsmove_tounreccdtarget
:parameters (?o - troops ?f - location ?t - location)
:duration (= ?duration (*(/(dist ?f ?t)(troop_speed ?o))3600))
:condition (and (at start(troop_pos ?o ?f))
(over all (troop_can_move ?o))
(over all (or (and (in_foe_terr ?f) (in_foe_terr ?t))
```

```
(and (not(in_foe_terr ?f)) (not(in_foe_terr ?t)))))
(over all(dist_known ?f ?t))
(over all (forall (?a - aircraft) (not(transporting ?a ?o))))
(at end (not(location_cleared ?t)))
)
:effect (and
(at start (not (troop_pos ?o ?f)))
(at end (troop_pos ?o ?t))
(at end (increase (total-cost) (+ (* (*(dist ?f ?t)(c_troop_move ?o)) (-
c_unreccd_route (section_cleared_beginning ?f ?t))) 7500) ) )
(at end (increase (troop_has_moved ?o) (dist ?f ?t)))
)
)

(:durative-action switchcamm
:parameters (?o - aircraft)
:duration (= ?duration 0.1)
:condition (at start(not(mapping_ground ?o)))
:effect (at end(mapping_ground ?o))
)

(:durative-action switchcamt
:parameters (?o - aircraft)
:duration (= ?duration 0.1)
:condition (at start(mapping_ground ?o))
:effect (at end(not(mapping_ground ?o)))
)

(:durative-action start_load
:parameters (?o - aircraft)
:duration (= ?duration 0.1)
:condition (and(over all (landed ?o)) (at start(not(loading ?o))))
:effect (at end(loading ?o))
)

(:durative-action stop_load
:parameters (?o - aircraft)
:duration (= ?duration 0.1)
:condition (and(over all (landed ?o)) (at start(loading ?o)))
:effect (at end(not(loading ?o)))
)

(:durative-action start_unld
:parameters (?o - aircraft)
:duration (= ?duration 0.1)
:condition (and(over all (landed ?o)) (at start(not(unloading ?o))))
:effect (at end(unloading ?o))
)

(:durative-action stop_unld
:parameters (?o - aircraft)
:duration (= ?duration 0.1)
:condition (and(over all (landed ?o)) (at start(unloading ?o)))
:effect (at end(not(unloading ?o)))
)
)
```

8.2 Planning Problem Example before Mission Start

In the following, an example problem for the Slave planner instance, generated before mission start, is shown together with an example solution for exactly that problem.

8.2.1 Planning Problem Description before Mission Start

This PDDL problem description file has been generated during the experimental campaign described in this dissertation. See Table 8-2 for details.

Table 8-2: Planning problem example before mission start - metadata

File name	Slave22.5.PROBLEM
Experiment start date	19.05.2011
Experiment start time	14:36
Mission[62]	4
Run / test person crew	number II
Problem instance	22
LPG process[63]	5

```
;; AUTO-GENERATED BY PDDL_WRAPPER

(define (problem autogen)

(:domain MUMT)

(:objects
 MiRA UAV1 UAV2 UAV3 - aircraft
 LEADER_SQUAD_A - troops
 TANGO_FRIEND TANGO_FOE ZULU_FRIEND ZULU_FOE LIMA_FRIEND LIMA_FOE
MIKE_FRIEND MIKE_FOE TARGET POI_0 EDNY_SE1 EDNY_W2 MOB AP_FOB DP_FOB FOB
AP_ISAR1 DP_ISAR1 ISAR1 AP_ISAR2 DP_ISAR2 ISAR2 HOA_ENTRY HOA_EXIT -
location
)

(:init
 (in_foe_terr TANGO_FOE)
 (in_foe_terr ZULU_FOE)
 (in_foe_terr LIMA_FOE)
 (in_foe_terr MIKE_FOE)
 (in_foe_terr TARGET)
 (in_hoa TARGET)
 (in_foe_terr POI_0)
```

[62] See Table 5-1.
[63] See Table 4-1.

```
(in_hoa POI_0)
(in_foe_terr AP_ISAR1)
(in_hoa AP_ISAR1)
(in_foe_terr DP_ISAR1)
(in_hoa DP_ISAR1)
(in_foe_terr ISAR1)
(in_hoa ISAR1)
(in_foe_terr AP_ISAR2)
(in_hoa AP_ISAR2)
(in_foe_terr DP_ISAR2)
(in_hoa DP_ISAR2)
(in_foe_terr ISAR2)
(in_hoa ISAR2)
(in_foe_terr HOA_ENTRY)
(in_foe_terr HOA_EXIT)
(dist_known TANGO_FRIEND TANGO_FOE)
(= (dist TANGO_FRIEND TANGO_FOE) 1.47)
(dist_known ZULU_FOE ZULU_FRIEND)
(= (dist ZULU_FOE ZULU_FRIEND) 1.51)
(dist_known LIMA_FOE LIMA_FRIEND)
(= (dist LIMA_FOE LIMA_FRIEND) 1.54)
(dist_known MIKE_FRIEND MIKE_FOE)
(= (dist MIKE_FRIEND MIKE_FOE) 1.44)
(dist_known TANGO_FRIEND ZULU_FRIEND)
(= (dist TANGO_FRIEND ZULU_FRIEND) 8.81)
(dist_known TANGO_FRIEND LIMA_FRIEND)
(= (dist TANGO_FRIEND LIMA_FRIEND) 12.50)
(dist_known TANGO_FRIEND MIKE_FRIEND)
(= (dist TANGO_FRIEND MIKE_FRIEND) 3.18)
(dist_known TANGO_FRIEND EDNY_SE1)
(= (dist TANGO_FRIEND EDNY_SE1) 8.86)
(dist_known TANGO_FRIEND EDNY_W2)
(= (dist TANGO_FRIEND EDNY_W2) 6.44)
(dist_known TANGO_FRIEND MOB)
(= (dist TANGO_FRIEND MOB) 7.46)
(dist_known TANGO_FRIEND AP_FOB)
(= (dist TANGO_FRIEND AP_FOB) 6.47)
(dist_known TANGO_FRIEND DP_FOB)
(= (dist TANGO_FRIEND DP_FOB) 5.41)
(dist_known TANGO_FRIEND FOB)
(= (dist TANGO_FRIEND FOB) 6.37)
(dist_known ZULU_FRIEND TANGO_FRIEND)
(= (dist ZULU_FRIEND TANGO_FRIEND) 8.81)
(dist_known ZULU_FRIEND LIMA_FRIEND)
(= (dist ZULU_FRIEND LIMA_FRIEND) 3.69)
(dist_known ZULU_FRIEND MIKE_FRIEND)
(= (dist ZULU_FRIEND MIKE_FRIEND) 11.86)
(dist_known ZULU_FRIEND EDNY_SE1)
(= (dist ZULU_FRIEND EDNY_SE1) 10.92)
(dist_known ZULU_FRIEND EDNY_W2)
(= (dist ZULU_FRIEND EDNY_W2) 12.41)
(dist_known ZULU_FRIEND MOB)
(= (dist ZULU_FRIEND MOB) 11.63)
(dist_known ZULU_FRIEND AP_FOB)
(= (dist ZULU_FRIEND AP_FOB) 13.04)
(dist_known ZULU_FRIEND DP_FOB)
(= (dist ZULU_FRIEND DP_FOB) 12.87)
(dist_known ZULU_FRIEND FOB)
(= (dist ZULU_FRIEND FOB) 13.67)
(dist_known LIMA_FRIEND TANGO_FRIEND)
(= (dist LIMA_FRIEND TANGO_FRIEND) 12.50)
(dist_known LIMA_FRIEND ZULU_FRIEND)
```

```
(= (dist LIMA_FRIEND ZULU_FRIEND) 3.69)
(dist_known LIMA_FRIEND MIKE_FRIEND)
(= (dist LIMA_FRIEND MIKE_FRIEND) 15.52)
(dist_known LIMA_FRIEND EDNY_SE1)
(= (dist LIMA_FRIEND EDNY_SE1) 13.67)
(dist_known LIMA_FRIEND EDNY_W2)
(= (dist LIMA_FRIEND EDNY_W2) 15.82)
(dist_known LIMA_FRIEND MOB)
(= (dist LIMA_FRIEND MOB) 14.78)
(dist_known LIMA_FRIEND AP_FOB)
(= (dist LIMA_FRIEND AP_FOB) 16.51)
(dist_known LIMA_FRIEND DP_FOB)
(= (dist LIMA_FRIEND DP_FOB) 16.46)
(dist_known LIMA_FRIEND FOB)
(= (dist LIMA_FRIEND FOB) 17.22)
(dist_known MIKE_FRIEND TANGO_FRIEND)
(= (dist MIKE_FRIEND TANGO_FRIEND) 3.18)
(dist_known MIKE_FRIEND ZULU_FRIEND)
(= (dist MIKE_FRIEND ZULU_FRIEND) 11.86)
(dist_known MIKE_FRIEND LIMA_FRIEND)
(= (dist MIKE_FRIEND LIMA_FRIEND) 15.52)
(dist_known MIKE_FRIEND EDNY_SE1)
(= (dist MIKE_FRIEND EDNY_SE1) 10.92)
(dist_known MIKE_FRIEND EDNY_W2)
(= (dist MIKE_FRIEND EDNY_W2) 7.25)
(dist_known MIKE_FRIEND MOB)
(= (dist MIKE_FRIEND MOB) 8.99)
(dist_known MIKE_FRIEND AP_FOB)
(= (dist MIKE_FRIEND AP_FOB) 6.85)
(dist_known MIKE_FRIEND DP_FOB)
(= (dist MIKE_FRIEND DP_FOB) 5.33)
(dist_known MIKE_FRIEND FOB)
(= (dist MIKE_FRIEND FOB) 6.16)
(dist_known EDNY_SE1 TANGO_FRIEND)
(= (dist EDNY_SE1 TANGO_FRIEND) 8.86)
(dist_known EDNY_SE1 ZULU_FRIEND)
(= (dist EDNY_SE1 ZULU_FRIEND) 10.92)
(dist_known EDNY_SE1 LIMA_FRIEND)
(= (dist EDNY_SE1 LIMA_FRIEND) 13.67)
(dist_known EDNY_SE1 MIKE_FRIEND)
(= (dist EDNY_SE1 MIKE_FRIEND) 10.92)
(dist_known EDNY_SE1 EDNY_W2)
(= (dist EDNY_SE1 EDNY_W2) 4.77)
(dist_known EDNY_SE1 MOB)
(= (dist EDNY_SE1 MOB) 2.49)
(dist_known EDNY_SE1 AP_FOB)
(= (dist EDNY_SE1 AP_FOB) 5.70)
(dist_known EDNY_SE1 DP_FOB)
(= (dist EDNY_SE1 DP_FOB) 6.95)
(dist_known EDNY_SE1 FOB)
(= (dist EDNY_SE1 FOB) 6.97)
(dist_known EDNY_W2 TANGO_FRIEND)
(= (dist EDNY_W2 TANGO_FRIEND) 6.44)
(dist_known EDNY_W2 ZULU_FRIEND)
(= (dist EDNY_W2 ZULU_FRIEND) 12.41)
(dist_known EDNY_W2 LIMA_FRIEND)
(= (dist EDNY_W2 LIMA_FRIEND) 15.82)
(dist_known EDNY_W2 MIKE_FRIEND)
(= (dist EDNY_W2 MIKE_FRIEND) 7.25)
(dist_known EDNY_W2 EDNY_SE1)
(= (dist EDNY_W2 EDNY_SE1) 4.77)
(dist_known EDNY_W2 MOB)
```

```
(= (dist EDNY_W2 MOB) 2.29)
(dist_known EDNY_W2 AP_FOB)
(= (dist EDNY_W2 AP_FOB) 0.94)
(dist_known EDNY_W2 DP_FOB)
(= (dist EDNY_W2 DP_FOB) 2.30)
(dist_known EDNY_W2 FOB)
(= (dist EDNY_W2 FOB) 2.20)
(dist_known MOB TANGO_FRIEND)
(= (dist MOB TANGO_FRIEND) 7.46)
(dist_known MOB ZULU_FRIEND)
(= (dist MOB ZULU_FRIEND) 11.63)
(dist_known MOB LIMA_FRIEND)
(= (dist MOB LIMA_FRIEND) 14.78)
(dist_known MOB MIKE_FRIEND)
(= (dist MOB MIKE_FRIEND) 8.99)
(dist_known MOB EDNY_SE1)
(= (dist MOB EDNY_SE1) 2.49)
(dist_known MOB EDNY_W2)
(= (dist MOB EDNY_W2) 2.29)
(dist_known MOB AP_FOB)
(= (dist MOB AP_FOB) 3.21)
(dist_known MOB DP_FOB)
(= (dist MOB DP_FOB) 4.53)
(dist_known MOB FOB)
(= (dist MOB FOB) 4.49)
(dist_known AP_FOB TANGO_FRIEND)
(= (dist AP_FOB TANGO_FRIEND) 6.47)
(dist_known AP_FOB ZULU_FRIEND)
(= (dist AP_FOB ZULU_FRIEND) 13.04)
(dist_known AP_FOB LIMA_FRIEND)
(= (dist AP_FOB LIMA_FRIEND) 16.51)
(dist_known AP_FOB MIKE_FRIEND)
(= (dist AP_FOB MIKE_FRIEND) 6.85)
(dist_known AP_FOB EDNY_SE1)
(= (dist AP_FOB EDNY_SE1) 5.70)
(dist_known AP_FOB EDNY_W2)
(= (dist AP_FOB EDNY_W2) 0.94)
(dist_known AP_FOB MOB)
(= (dist AP_FOB MOB) 3.21)
(dist_known AP_FOB DP_FOB)
(= (dist AP_FOB DP_FOB) 1.60)
(dist_known AP_FOB FOB)
(= (dist AP_FOB FOB) 1.28)
(dist_known DP_FOB TANGO_FRIEND)
(= (dist DP_FOB TANGO_FRIEND) 5.41)
(dist_known DP_FOB ZULU_FRIEND)
(= (dist DP_FOB ZULU_FRIEND) 12.87)
(dist_known DP_FOB LIMA_FRIEND)
(= (dist DP_FOB LIMA_FRIEND) 16.46)
(dist_known DP_FOB MIKE_FRIEND)
(= (dist DP_FOB MIKE_FRIEND) 5.33)
(dist_known DP_FOB EDNY_SE1)
(= (dist DP_FOB EDNY_SE1) 6.95)
(dist_known DP_FOB EDNY_W2)
(= (dist DP_FOB EDNY_W2) 2.30)
(dist_known DP_FOB MOB)
(= (dist DP_FOB MOB) 4.53)
(dist_known DP_FOB AP_FOB)
(= (dist DP_FOB AP_FOB) 1.60)
(dist_known DP_FOB FOB)
(= (dist DP_FOB FOB) 0.96)
(dist_known FOB TANGO_FRIEND)
```

```
(= (dist FOB TANGO_FRIEND) 6.37)
(dist_known FOB ZULU_FRIEND)
(= (dist FOB ZULU_FRIEND) 13.67)
(dist_known FOB LIMA_FRIEND)
(= (dist FOB LIMA_FRIEND) 17.22)
(dist_known FOB MIKE_FRIEND)
(= (dist FOB MIKE_FRIEND) 6.16)
(dist_known FOB EDNY_SE1)
(= (dist FOB EDNY_SE1) 6.97)
(dist_known FOB EDNY_W2)
(= (dist FOB EDNY_W2) 2.20)
(dist_known FOB MOB)
(= (dist FOB MOB) 4.49)
(dist_known FOB AP_FOB)
(= (dist FOB AP_FOB) 1.28)
(dist_known FOB DP_FOB)
(= (dist FOB DP_FOB) 0.96)
(dist_known TANGO_FOE ZULU_FOE)
(= (dist TANGO_FOE ZULU_FOE) 8.46)
(dist_known TANGO_FOE LIMA_FOE)
(= (dist TANGO_FOE LIMA_FOE) 11.81)
(dist_known TANGO_FOE MIKE_FOE)
(= (dist TANGO_FOE MIKE_FOE) 2.86)
(dist_known TANGO_FOE HOA_ENTRY)
(= (dist TANGO_FOE HOA_ENTRY) 5.48)
(dist_known TANGO_FOE HOA_EXIT)
(= (dist TANGO_FOE HOA_EXIT) 6.77)
(dist_known ZULU_FOE TANGO_FOE)
(= (dist ZULU_FOE TANGO_FOE) 8.46)
(dist_known ZULU_FOE LIMA_FOE)
(= (dist ZULU_FOE LIMA_FOE) 3.36)
(dist_known ZULU_FOE MIKE_FOE)
(= (dist ZULU_FOE MIKE_FOE) 11.21)
(dist_known ZULU_FOE HOA_ENTRY)
(= (dist ZULU_FOE HOA_ENTRY) 8.77)
(dist_known ZULU_FOE HOA_EXIT)
(= (dist ZULU_FOE HOA_EXIT) 6.16)
(dist_known LIMA_FOE TANGO_FOE)
(= (dist LIMA_FOE TANGO_FOE) 11.81)
(dist_known LIMA_FOE ZULU_FOE)
(= (dist LIMA_FOE ZULU_FOE) 3.36)
(dist_known LIMA_FOE MIKE_FOE)
(= (dist LIMA_FOE MIKE_FOE) 14.54)
(dist_known LIMA_FOE HOA_ENTRY)
(= (dist LIMA_FOE HOA_ENTRY) 11.52)
(dist_known LIMA_FOE HOA_EXIT)
(= (dist LIMA_FOE HOA_EXIT) 8.50)
(dist_known MIKE_FOE TANGO_FOE)
(= (dist MIKE_FOE TANGO_FOE) 2.86)
(dist_known MIKE_FOE ZULU_FOE)
(= (dist MIKE_FOE ZULU_FOE) 11.21)
(dist_known MIKE_FOE LIMA_FOE)
(= (dist MIKE_FOE LIMA_FOE) 14.54)
(dist_known MIKE_FOE HOA_ENTRY)
(= (dist MIKE_FOE HOA_ENTRY) 6.03)
(dist_known MIKE_FOE HOA_EXIT)
(= (dist MIKE_FOE HOA_EXIT) 8.39)
(dist_known HOA_ENTRY TANGO_FOE)
(= (dist HOA_ENTRY TANGO_FOE) 5.48)
(dist_known HOA_ENTRY ZULU_FOE)
(= (dist HOA_ENTRY ZULU_FOE) 8.77)
(dist_known HOA_ENTRY LIMA_FOE)
```

```
(= (dist HOA_ENTRY LIMA_FOE) 11.52)
(dist_known HOA_ENTRY MIKE_FOE)
(= (dist HOA_ENTRY MIKE_FOE) 6.03)
(dist_known HOA_ENTRY HOA_EXIT)
(= (dist HOA_ENTRY HOA_EXIT) 3.26)
(dist_known HOA_EXIT TANGO_FOE)
(= (dist HOA_EXIT TANGO_FOE) 6.77)
(dist_known HOA_EXIT ZULU_FOE)
(= (dist HOA_EXIT ZULU_FOE) 6.16)
(dist_known HOA_EXIT LIMA_FOE)
(= (dist HOA_EXIT LIMA_FOE) 8.50)
(dist_known HOA_EXIT MIKE_FOE)
(= (dist HOA_EXIT MIKE_FOE) 8.39)
(dist_known HOA_EXIT HOA_ENTRY)
(= (dist HOA_EXIT HOA_ENTRY) 3.26)
(dist_known TARGET POI_0)
(= (dist TARGET POI_0) 0.32)
(dist_known TARGET AP_ISAR1)
(= (dist TARGET AP_ISAR1) 1.35)
(dist_known TARGET DP_ISAR1)
(= (dist TARGET DP_ISAR1) 0.69)
(dist_known TARGET ISAR1)
(= (dist TARGET ISAR1) 1.00)
(dist_known TARGET AP_ISAR2)
(= (dist TARGET AP_ISAR2) 0.53)
(dist_known TARGET DP_ISAR2)
(= (dist TARGET DP_ISAR2) 1.31)
(dist_known TARGET ISAR2)
(= (dist TARGET ISAR2) 1.18)
(dist_known TARGET HOA_ENTRY)
(= (dist TARGET HOA_ENTRY) 1.67)
(dist_known TARGET HOA_EXIT)
(= (dist TARGET HOA_EXIT) 1.69)
(dist_known POI_0 TARGET)
(= (dist POI_0 TARGET) 0.32)
(dist_known POI_0 AP_ISAR1)
(= (dist POI_0 AP_ISAR1) 1.36)
(dist_known POI_0 DP_ISAR1)
(= (dist POI_0 DP_ISAR1) 0.58)
(dist_known POI_0 ISAR1)
(= (dist POI_0 ISAR1) 0.97)
(dist_known POI_0 AP_ISAR2)
(= (dist POI_0 AP_ISAR2) 0.63)
(dist_known POI_0 DP_ISAR2)
(= (dist POI_0 DP_ISAR2) 1.30)
(dist_known POI_0 ISAR2)
(= (dist POI_0 ISAR2) 1.35)
(dist_known POI_0 HOA_ENTRY)
(= (dist POI_0 HOA_ENTRY) 1.76)
(dist_known POI_0 HOA_EXIT)
(= (dist POI_0 HOA_EXIT) 1.51)
(dist_known AP_ISAR1 TARGET)
(= (dist AP_ISAR1 TARGET) 1.35)
(dist_known AP_ISAR1 POI_0)
(= (dist AP_ISAR1 POI_0) 1.36)
(dist_known AP_ISAR1 DP_ISAR1)
(= (dist AP_ISAR1 DP_ISAR1) 0.80)
(dist_known AP_ISAR1 ISAR1)
(= (dist AP_ISAR1 ISAR1) 0.40)
(dist_known AP_ISAR1 AP_ISAR2)
(= (dist AP_ISAR1 AP_ISAR2) 1.88)
(dist_known AP_ISAR1 DP_ISAR2)
```

```
(= (dist AP_ISAR1 DP_ISAR2) 2.64)
(dist_known AP_ISAR1 ISAR2)
(= (dist AP_ISAR1 ISAR2) 2.51)
(dist_known AP_ISAR1 HOA_ENTRY)
(= (dist AP_ISAR1 HOA_ENTRY) 0.52)
(dist_known AP_ISAR1 HOA_EXIT)
(= (dist AP_ISAR1 HOA_EXIT) 2.82)
(dist_known DP_ISAR1 TARGET)
(= (dist DP_ISAR1 TARGET) 0.69)
(dist_known DP_ISAR1 POI_0)
(= (dist DP_ISAR1 POI_0) 0.58)
(dist_known DP_ISAR1 AP_ISAR1)
(= (dist DP_ISAR1 AP_ISAR1) 0.80)
(dist_known DP_ISAR1 ISAR1)
(= (dist DP_ISAR1 ISAR1) 0.40)
(dist_known DP_ISAR1 AP_ISAR2)
(= (dist DP_ISAR1 AP_ISAR2) 1.17)
(dist_known DP_ISAR1 DP_ISAR2)
(= (dist DP_ISAR1 DP_ISAR2) 1.88)
(dist_known DP_ISAR1 ISAR2)
(= (dist DP_ISAR1 ISAR2) 1.87)
(dist_known DP_ISAR1 HOA_ENTRY)
(= (dist DP_ISAR1 HOA_ENTRY) 1.26)
(dist_known DP_ISAR1 HOA_EXIT)
(= (dist DP_ISAR1 HOA_EXIT) 2.02)
(dist_known ISAR1 TARGET)
(= (dist ISAR1 TARGET) 1.00)
(dist_known ISAR1 POI_0)
(= (dist ISAR1 POI_0) 0.97)
(dist_known ISAR1 AP_ISAR1)
(= (dist ISAR1 AP_ISAR1) 0.40)
(dist_known ISAR1 DP_ISAR1)
(= (dist ISAR1 DP_ISAR1) 0.40)
(dist_known ISAR1 AP_ISAR2)
(= (dist ISAR1 AP_ISAR2) 1.51)
(dist_known ISAR1 DP_ISAR2)
(= (dist ISAR1 DP_ISAR2) 2.26)
(dist_known ISAR1 ISAR2)
(= (dist ISAR1 ISAR2) 2.17)
(dist_known ISAR1 HOA_ENTRY)
(= (dist ISAR1 HOA_ENTRY) 0.88)
(dist_known ISAR1 HOA_EXIT)
(= (dist ISAR1 HOA_EXIT) 2.42)
(dist_known AP_ISAR2 TARGET)
(= (dist AP_ISAR2 TARGET) 0.53)
(dist_known AP_ISAR2 POI_0)
(= (dist AP_ISAR2 POI_0) 0.63)
(dist_known AP_ISAR2 AP_ISAR1)
(= (dist AP_ISAR2 AP_ISAR1) 1.88)
(dist_known AP_ISAR2 DP_ISAR1)
(= (dist AP_ISAR2 DP_ISAR1) 1.17)
(dist_known AP_ISAR2 ISAR1)
(= (dist AP_ISAR2 ISAR1) 1.51)
(dist_known AP_ISAR2 DP_ISAR2)
(= (dist AP_ISAR2 DP_ISAR2) 0.80)
(dist_known AP_ISAR2 ISAR2)
(= (dist AP_ISAR2 ISAR2) 0.72)
(dist_known AP_ISAR2 HOA_ENTRY)
(= (dist AP_ISAR2 HOA_ENTRY) 2.19)
(dist_known AP_ISAR2 HOA_EXIT)
(= (dist AP_ISAR2 HOA_EXIT) 1.34)
(dist_known DP_ISAR2 TARGET)
```

```
(= (dist DP_ISAR2 TARGET) 1.31)
(dist_known DP_ISAR2 POI_0)
(= (dist DP_ISAR2 POI_0) 1.30)
(dist_known DP_ISAR2 AP_ISAR1)
(= (dist DP_ISAR2 AP_ISAR1) 2.64)
(dist_known DP_ISAR2 DP_ISAR1)
(= (dist DP_ISAR2 DP_ISAR1) 1.88)
(dist_known DP_ISAR2 ISAR1)
(= (dist DP_ISAR2 ISAR1) 2.26)
(dist_known DP_ISAR2 AP_ISAR2)
(= (dist DP_ISAR2 AP_ISAR2) 0.80)
(dist_known DP_ISAR2 ISAR2)
(= (dist DP_ISAR2 ISAR2) 0.72)
(dist_known DP_ISAR2 HOA_ENTRY)
(= (dist DP_ISAR2 HOA_ENTRY) 2.98)
(dist_known DP_ISAR2 HOA_EXIT)
(= (dist DP_ISAR2 HOA_EXIT) 0.84)
(dist_known ISAR2 TARGET)
(= (dist ISAR2 TARGET) 1.18)
(dist_known ISAR2 POI_0)
(= (dist ISAR2 POI_0) 1.35)
(dist_known ISAR2 AP_ISAR1)
(= (dist ISAR2 AP_ISAR1) 2.51)
(dist_known ISAR2 DP_ISAR1)
(= (dist ISAR2 DP_ISAR1) 1.87)
(dist_known ISAR2 ISAR1)
(= (dist ISAR2 ISAR1) 2.17)
(dist_known ISAR2 AP_ISAR2)
(= (dist ISAR2 AP_ISAR2) 0.72)
(dist_known ISAR2 DP_ISAR2)
(= (dist ISAR2 DP_ISAR2) 0.72)
(dist_known ISAR2 HOA_ENTRY)
(= (dist ISAR2 HOA_ENTRY) 2.73)
(dist_known ISAR2 HOA_EXIT)
(= (dist ISAR2 HOA_EXIT) 1.55)
(dist_known HOA_ENTRY TARGET)
(= (dist HOA_ENTRY TARGET) 1.67)
(dist_known HOA_ENTRY POI_0)
(= (dist HOA_ENTRY POI_0) 1.76)
(dist_known HOA_ENTRY AP_ISAR1)
(= (dist HOA_ENTRY AP_ISAR1) 0.52)
(dist_known HOA_ENTRY DP_ISAR1)
(= (dist HOA_ENTRY DP_ISAR1) 1.26)
(dist_known HOA_ENTRY ISAR1)
(= (dist HOA_ENTRY ISAR1) 0.88)
(dist_known HOA_ENTRY AP_ISAR2)
(= (dist HOA_ENTRY AP_ISAR2) 2.19)
(dist_known HOA_ENTRY DP_ISAR2)
(= (dist HOA_ENTRY DP_ISAR2) 2.98)
(dist_known HOA_ENTRY ISAR2)
(= (dist HOA_ENTRY ISAR2) 2.73)
(dist_known HOA_ENTRY HOA_EXIT)
(= (dist HOA_ENTRY HOA_EXIT) 3.26)
(dist_known HOA_EXIT TARGET)
(= (dist HOA_EXIT TARGET) 1.69)
(dist_known HOA_EXIT POI_0)
(= (dist HOA_EXIT POI_0) 1.51)
(dist_known HOA_EXIT AP_ISAR1)
(= (dist HOA_EXIT AP_ISAR1) 2.82)
(dist_known HOA_EXIT DP_ISAR1)
(= (dist HOA_EXIT DP_ISAR1) 2.02)
(dist_known HOA_EXIT ISAR1)
```

```
(= (dist HOA_EXIT ISAR1) 2.42)
(dist_known HOA_EXIT AP_ISAR2)
(= (dist HOA_EXIT AP_ISAR2) 1.34)
(dist_known HOA_EXIT DP_ISAR2)
(= (dist HOA_EXIT DP_ISAR2) 0.84)
(dist_known HOA_EXIT ISAR2)
(= (dist HOA_EXIT ISAR2) 1.55)
(dist_known HOA_EXIT HOA_ENTRY)
(= (dist HOA_EXIT HOA_ENTRY) 3.26)
(is_pz MOB)
(is_dz MOB)
(is_ap_for EDNY_SE1 MOB)
(is_dp_for EDNY_W2 MOB)
(is_pz FOB)
(is_dz FOB)
(is_ap_for AP_FOB FOB)
(is_dp_for DP_FOB FOB)
(is_pz ISAR1)
(is_dz ISAR1)
(is_ap_for AP_ISAR1 ISAR1)
(is_dp_for DP_ISAR1 ISAR1)
(is_pz ISAR2)
(is_dz ISAR2)
(is_ap_for AP_ISAR2 ISAR2)
(is_dp_for DP_ISAR2 ISAR2)
(is_hoa_entry HOA_ENTRY)
(is_hoa_exit HOA_EXIT)
(is_heli MiRA)
(empty MiRA)
(= (v_frt MiRA) 110)
(= (v_ent MiRA) 90)
(= (c_flight MiRA) 100)
(= (d_ac_landing MiRA) 60)
(= (d_ac_takeoff MiRA) 30)
(ac_pos MiRA MOB)
(landed MiRA)
(= (has_done_reccetasks MiRA) 0)
(is_uav UAV1)
(= (v_frt UAV1) 119)
(= (v_ent UAV1) 119)
(= (c_flight UAV1) 5)
(= (d_ac_landing UAV1) 60)
(= (d_ac_takeoff UAV1) 30)
(= (d_recce_area UAV1) 75)
(ac_pos UAV1 MOB)
(landed UAV1)
(= (has_done_reccetasks UAV1) 0)
(is_uav UAV2)
(= (v_frt UAV2) 119)
(= (v_ent UAV2) 119)
(= (c_flight UAV2) 5)
(= (d_ac_landing UAV2) 60)
(= (d_ac_takeoff UAV2) 30)
(= (d_recce_area UAV2) 75)
(ac_pos UAV2 MOB)
(landed UAV2)
(= (has_done_reccetasks UAV2) 0)
(is_uav UAV3)
(= (v_frt UAV3) 119)
(= (v_ent UAV3) 119)
(= (c_flight UAV3) 5)
(= (d_ac_landing UAV3) 60)
```

```
(= (d_ac_takeoff UAV3) 30)
(= (d_recce_area UAV3) 75)
(ac_pos UAV3 MOB)
(landed UAV3)
(= (has_done_reccetasks UAV3) 0)
(= (troop_speed LEADER_SQUAD_A) 9.72)
(= (d_unloadtrps LEADER_SQUAD_A) 60)
(= (d_loadtroops LEADER_SQUAD_A) 60)
(= (c_troop_move LEADER_SQUAD_A) 500)
(troop_pos LEADER_SQUAD_A FOB)
(= (troop_has_moved LEADER_SQUAD_A) 0)
(troop_can_move LEADER_SQUAD_A)
(= c_obj_surveil 0)
(= d_obj_surveil 300)
(= c_unreccd_route 3)
(= total-cost 0)
(takeoff_denied MiRA MOB)
(at 419 (not (takeoff_denied MiRA MOB)))
(takeoff_denied UAV3 MOB)
(at 389 (not (takeoff_denied UAV3 MOB)))
(takeoff_denied UAV2 MOB)
(at 389 (not (takeoff_denied UAV2 MOB)))
(takeoff_denied UAV1 MOB)
(at 389 (not (takeoff_denied UAV1 MOB)))
(at 389 (corr_open MIKE_FRIEND MIKE_FOE MiRA))
(at 389 (corr_open MIKE_FRIEND MIKE_FOE UAV1))
(at 389 (corr_open MIKE_FRIEND MIKE_FOE UAV2))
(at 389 (corr_open MIKE_FRIEND MIKE_FOE UAV3))
(at 1289 (not (corr_open MIKE_FRIEND MIKE_FOE MiRA)))
(at 1289 (not (corr_open MIKE_FRIEND MIKE_FOE UAV1)))
(at 1289 (not (corr_open MIKE_FRIEND MIKE_FOE UAV2)))
(at 1289 (not (corr_open MIKE_FRIEND MIKE_FOE UAV3)))
(at 1589 (corr_open LIMA_FOE LIMA_FRIEND MiRA))
(at 1589 (corr_open LIMA_FOE LIMA_FRIEND UAV1))
(at 1589 (corr_open LIMA_FOE LIMA_FRIEND UAV2))
(at 1589 (corr_open LIMA_FOE LIMA_FRIEND UAV3))
(at 2789 (not (corr_open LIMA_FOE LIMA_FRIEND MiRA)))
(at 2789 (not (corr_open LIMA_FOE LIMA_FRIEND UAV1)))
(at 2789 (not (corr_open LIMA_FOE LIMA_FRIEND UAV2)))
(at 2789 (not (corr_open LIMA_FOE LIMA_FRIEND UAV3)))
(at 1589 (corr_open ZULU_FOE ZULU_FRIEND MiRA))
(at 1589 (corr_open ZULU_FOE ZULU_FRIEND UAV1))
(at 1589 (corr_open ZULU_FOE ZULU_FRIEND UAV2))
(at 1589 (corr_open ZULU_FOE ZULU_FRIEND UAV3))
(at 2789 (not (corr_open ZULU_FOE ZULU_FRIEND MiRA)))
(at 2789 (not (corr_open ZULU_FOE ZULU_FRIEND UAV1)))
(at 2789 (not (corr_open ZULU_FOE ZULU_FRIEND UAV2)))
(at 2789 (not (corr_open ZULU_FOE ZULU_FRIEND UAV3)))
(at 389 (corr_open TANGO_FRIEND TANGO_FOE MiRA))
(at 389 (corr_open TANGO_FRIEND TANGO_FOE UAV1))
(at 389 (corr_open TANGO_FRIEND TANGO_FOE UAV2))
(at 389 (corr_open TANGO_FRIEND TANGO_FOE UAV3))
(at 1289 (not (corr_open TANGO_FRIEND TANGO_FOE MiRA)))
(at 1289 (not (corr_open TANGO_FRIEND TANGO_FOE UAV1)))
(at 1289 (not (corr_open TANGO_FRIEND TANGO_FOE UAV2)))
(at 1289 (not (corr_open TANGO_FRIEND TANGO_FOE UAV3)))
(= (c_corridor ZULU_FOE ZULU_FRIEND MiRA) -1000)
(= (c_corridor ZULU_FOE ZULU_FRIEND UAV1) -1000)
(= (c_corridor ZULU_FOE ZULU_FRIEND UAV2) -1000)
(= (c_corridor ZULU_FOE ZULU_FRIEND UAV3) -1000)
(= (c_corridor TANGO_FRIEND TANGO_FOE MiRA) -1000)
(= (c_corridor TANGO_FRIEND TANGO_FOE UAV1) -1000)
```

```
  (= (c_corridor TANGO_FRIEND TANGO_FOE UAV2) -1000)
  (= (c_corridor TANGO_FRIEND TANGO_FOE UAV3) -1000)
  (= (c_unloadtrps ISAR1 MiRA LEADER_SQUAD_A) -5000)
 )

 (:goal
  (and
  (load_constraint MiRA LEADER_SQUAD_A FOB)
  (flot_crossed_by ZULU_FOE ZULU_FRIEND MiRA)
  (flot_crossed_by TANGO_FRIEND TANGO_FOE MiRA)
  (unload_constraint MiRA LEADER_SQUAD_A ISAR1)
  (location_cleared_by ISAR2 UAV3)
  (section_cleared_by HOA_ENTRY ISAR2 UAV3)
  (section_cleared_by TANGO_FOE HOA_ENTRY UAV3)
  (flot_crossed_by TANGO_FRIEND TANGO_FOE UAV3)
  (has_transited UAV3 EDNY_W2 TANGO_FRIEND)
  (has_departed_via UAV3 EDNY_W2)
  (location_cleared_by ISAR1 UAV2)
  (section_cleared_by HOA_ENTRY ISAR1 UAV2)
  (section_cleared_by TANGO_FOE HOA_ENTRY UAV2)
  (flot_crossed_by TANGO_FRIEND TANGO_FOE UAV2)
  (has_transited UAV2 EDNY_W2 TANGO_FRIEND)
  (has_departed_via UAV2 EDNY_W2)
  (section_cleared_by ISAR2 POI_0 UAV1)
  (section_cleared_by AP_ISAR2 ISAR2 UAV1)
  (section_cleared_by TARGET AP_ISAR2 UAV1)
  (section_cleared_by ISAR1 TARGET UAV1)
  (section_cleared_by AP_ISAR1 ISAR1 UAV1)
  (section_cleared_by HOA_ENTRY AP_ISAR1 UAV1)
  (section_cleared_by TANGO_FOE HOA_ENTRY UAV1)
  (flot_crossed_by TANGO_FRIEND TANGO_FOE UAV1)
  (has_transited UAV1 EDNY_W2 TANGO_FRIEND)
  (has_departed_via UAV1 EDNY_W2)
  (landed MiRA)
  (ac_pos MiRA MOB)
  (landed UAV1)
  (ac_pos UAV1 MOB)
  (landed UAV2)
  (ac_pos UAV2 MOB)
  (landed UAV3)
  (ac_pos UAV3 MOB)
  (troop_pos LEADER_SQUAD_A TARGET)
  (has_departed_via MiRA EDNY_W2)
  (has_transited MiRA EDNY_W2 AP_FOB)
  (has_approached_via MiRA AP_FOB)
  (load_constraint MiRA LEADER_SQUAD_A FOB)
  (has_departed_via MiRA DP_FOB)
  (has_transited MiRA DP_FOB TANGO_FRIEND)
  (flot_crossed_by TANGO_FRIEND TANGO_FOE MiRA)
  (has_transited MiRA TANGO_FOE HOA_ENTRY)
  (has_transited MiRA HOA_ENTRY AP_ISAR1)
  (has_approached_via MiRA AP_ISAR1)
  (unload_constraint MiRA LEADER_SQUAD_A ISAR1)
  (has_departed_via MiRA DP_ISAR1)
  (has_transited MiRA DP_ISAR1 HOA_EXIT)
  (has_transited MiRA HOA_EXIT ZULU_FOE)
  (flot_crossed_by ZULU_FOE ZULU_FRIEND MiRA)
  (has_transited MiRA ZULU_FRIEND EDNY_SE1)
  (has_approached_via MiRA EDNY_SE1)
  (> (has_done_reccetasks UAV1) 2)
  (> (has_done_reccetasks UAV2) 2)
  (> (has_done_reccetasks UAV3) 2)
```

```
  )
)

(:metric minimize (total-cost))

)
```

8.2.2 Planning Solution before Mission Start

Due to the anytime planning feature of the LPG-td planner, multiple solutions are generated sequentially. The solution shown here has the sequence number six. See Table 8-3 for further details.

Table 8-3: Example planning solution before mission start - metadata

File name	plan_Slave22.5.PROBLEM_6.SOL
Input domain file	mumt1.DOMAIN
Input problem file	Slave22.5.PROBLEM
Solution sequence number	6

```
; Version LPG-td-1.0
; Seed 18473062
; Command line: /cygdrive/c/REPOS/lpg/LPG-td-1.0/LPG-td-1.0 -o
C:\REPOS\aquasim\software\projects\mum-t\missionPlanner\domain\mumt1.DOMAIN
-f P:\Slave\Slave22.5.PROBLEM -n 9999 -v off -cputime 2400
; Problem Slave22.5.PROBLEM
; Time 48.20
; Search time 47.50
; Parsing time 0.16
; Mutex time 0.53
; Quality 3250.40


Time 48.20

0.0003:   (SWITCHCAMM UAV1) [0.1000]
0.0005:   (SWITCHCAMM UAV2) [0.1000]
0.0008:   (SWITCHCAMM UAV3) [0.1000]
389.0010:   (DEPTAKEOFF UAV1 MOB) [30.0000]
389.0013:   (DEPTAKEOFF UAV2 MOB) [30.0000]
389.0015:   (DEPTAKEOFF UAV3 MOB) [30.0000]
419.0017:   (DEPARTFROM UAV2 MOB EDNY_W2) [69.2773]
419.0020:   (DEPTAKEOFF MIRA MOB) [30.0000]
419.0023:   (DEPARTFROM UAV3 MOB EDNY_W2) [69.2773]
419.0025:   (DEPARTFROM UAV1 MOB EDNY_W2) [69.2773]
488.2801:   (FRT_FLIGHT UAV1 EDNY_W2 TANGO_FRIEND) [194.8235]
449.0030:   (DEPARTFROM MIRA MOB EDNY_W2) [91.6000]
488.2805:   (FRT_FLIGHT UAV2 EDNY_W2 TANGO_FRIEND) [194.8235]
488.2808:   (FRT_FLIGHT UAV3 EDNY_W2 TANGO_FRIEND) [194.8235]
540.6037:   (FRT_FLIGHT MIRA EDNY_W2 AP_FOB) [30.7636]
571.3676:   (APPROACHTO MIRA AP_FOB FOB) [51.2000]
622.5679:   (APPLANDING MIRA FOB) [60.0000]
682.5681:   (START_LOAD MIRA) [0.1000]
682.6683:   (LOADTROOPS MIRA FOB LEADER_SQUAD_A) [60.0000]
683.1057:   (CROSS_FLOT UAV2 TANGO_FRIEND TANGO_FOE) [44.4706]
683.1060:   (CROSS_FLOT UAV1 TANGO_FRIEND TANGO_FOE) [44.4706]
683.1062:   (CROSS_FLOT UAV3 TANGO_FRIEND TANGO_FOE) [44.4706]
```

```
727.5771:    (RECCEROUTE UAV2 TANGO_FOE HOA_ENTRY) [165.7815]
727.5774:    (RECCEROUTE UAV3 TANGO_FOE HOA_ENTRY) [165.7815]
742.5698:    (STOP_LOAD MIRA) [0.1000]
742.6700:    (DEPTAKEOFF MIRA FOB) [30.0000]
772.6703:    (DEPARTFROM MIRA FOB DP_FOB) [38.4000]
811.0706:    (FRT_FLIGHT MIRA DP_FOB TANGO_FRIEND) [177.0545]
727.5787:    (RECCEROUTE UAV1 TANGO_FOE HOA_ENTRY) [165.7815]
893.3603:    (RECCEROUTE UAV1 HOA_ENTRY AP_ISAR1) [15.7311]
893.3607:    (RECCEROUTE UAV2 HOA_ENTRY ISAR1) [26.6218]
893.3608:    (RECCEROUTE UAV3 HOA_ENTRY ISAR2) [82.5882]
909.0922:    (RECCEROUTE UAV1 AP_ISAR1 ISAR2) [75.9328]
919.9832:    (RECCE_AREA UAV2 ISAR1) [75.0000]
975.9497:    (RECCE_AREA UAV3 ISAR2) [75.0000]
985.0258:    (RECCEROUTE UAV1 ISAR2 POI_0) [40.8403]
988.1274:    (CROSS_FLOT MIRA TANGO_FRIEND TANGO_FOE) [58.8000]
994.9842:    (ENT_FLIGHT_WRONG_HOA_DIR UAV2 ISAR1 HOA_ENTRY) [26.6218]
1021.6064:    (RECCEROUTE UAV2 HOA_ENTRY ZULU_FOE) [265.3109]
1025.8671:    (RECCEROUTE UAV1 POI_0 AP_ISAR2) [19.0588]
1044.9261:    (RECCEROUTE UAV1 AP_ISAR2 ISAR2) [21.7815]
1046.9285:    (ENT_FLIGHT MIRA TANGO_FOE HOA_ENTRY) [219.2000]
1050.9518:    (RECCEROUTE UAV3 ISAR2 HOA_EXIT) [46.8908]
1066.7084:    (RECCEROUTE UAV1 ISAR2 TARGET) [35.6975]
1097.8430:    (RECCEROUTE UAV3 HOA_EXIT ZULU_FOE) [186.3529]
1102.4064:    (RECCEROUTE UAV1 TARGET AP_ISAR2) [16.0336]
1118.4401:    (RECCEROUTE UAV1 AP_ISAR2 ISAR1) [45.6807]
1164.1211:    (RECCEROUTE UAV1 ISAR1 TARGET) [30.2521]
1194.3734:    (RECCEROUTE UAV1 TARGET DP_ISAR1) [20.8739]
1215.2476:    (RECCEROUTE UAV1 DP_ISAR1 AP_ISAR1) [24.2017]
1239.4495:    (RECCEROUTE UAV1 AP_ISAR1 ISAR1) [12.1008]
1251.5505:    (RECCEROUTE UAV1 ISAR1 POI_0) [29.3445]
1266.1313:    (ENT_FLIGHT MIRA HOA_ENTRY AP_ISAR1) [20.8000]
1280.8955:    (RECCEROUTE UAV1 POI_0 HOA_EXIT) [45.6807]
1284.1985:    (RECCEROUTE UAV3 ZULU_FOE LIMA_FOE) [101.6471]
1286.9321:    (APPROACHTO MIRA AP_ISAR1 ISAR1) [16.0000]
1302.9324:    (APPLANDING MIRA ISAR1) [60.0000]
1326.5773:    (RECCEROUTE UAV1 HOA_EXIT TANGO_FOE) [204.8067]
1362.9329:    (START_UNLD MIRA) [0.1000]
1385.8468:    (RECCEROUTE UAV3 LIMA_FOE ZULU_FOE) [101.6471]
1363.0333:    (UNLOADTRPS MIRA ISAR1 LEADER_SQUAD_A) [60.0000]
1423.0334:    (TROOPSMOVE_TOUNRECCDTARGET LEADER_SQUAD_A ISAR1 TARGET)
[370.3704]
1422.9338:    (STOP_UNLD MIRA) [0.1000]
1423.0341:    (DEPTAKEOFF MIRA ISAR1) [30.0000]
1531.3856:    (RECCEROUTE UAV1 TANGO_FOE ZULU_FOE) [255.9328]
1589.0165:    (CROSS_FLOT UAV2 ZULU_FOE ZULU_FRIEND) [45.6807]
1589.0167:    (CROSS_FLOT UAV3 ZULU_FOE ZULU_FRIEND) [45.6807]
1634.6976:    (RECCEROUTE_FRT UAV2 ZULU_FRIEND EDNY_SE1) [330.3529]
1634.6978:    (FRT_FLIGHT UAV3 ZULU_FRIEND LIMA_FRIEND) [111.6303]
1746.3285:    (FRT_FLIGHT UAV3 LIMA_FRIEND EDNY_SE1) [413.5462]
1787.3199:    (RECCEROUTE UAV1 ZULU_FOE TANGO_FOE) [255.9328]
1453.0360:    (DEPARTFROM MIRA ISAR1 DP_ISAR1) [16.0000]
1469.0364:    (ENT_FLIGHT MIRA DP_ISAR1 HOA_EXIT) [80.8000]
1549.8367:    (ENT_FLIGHT MIRA HOA_EXIT ZULU_FOE) [246.4000]
1796.2369:    (CROSS_FLOT MIRA ZULU_FOE ZULU_FRIEND) [60.4000]
1856.6372:    (FRT_FLIGHT MIRA ZULU_FRIEND EDNY_SE1) [357.3818]
1965.0529:    (APPROACHTO UAV2 EDNY_SE1 MOB) [75.3277]
2040.3809:    (APPLANDING UAV2 MOB) [60.0000]
2043.2549:    (RECCEROUTE UAV1 TANGO_FOE HOA_ENTRY) [165.7815]
2159.8772:    (APPROACHTO UAV3 EDNY_SE1 MOB) [75.3277]
2209.0369:    (RECCEROUTE UAV1 HOA_ENTRY ZULU_FOE) [265.3109]
2214.0205:    (APPROACHTO MIRA EDNY_SE1 MOB) [99.6000]
2235.2056:    (APPLANDING UAV3 MOB) [60.0000]
```

```
2313.6211:   (APPLANDING MIRA MOB) [60.0000]
2474.3489:   (CROSS_FLOT UAV1 ZULU_FOE ZULU_FRIEND) [45.6807]
2520.0298:   (FRT_FLIGHT UAV1 ZULU_FRIEND EDNY_SE1) [330.3529]
2850.3831:   (APPROACHTO UAV1 EDNY_SE1 MOB) [75.3277]
2925.7109:   (APPLANDING UAV1 MOB) [60.0000]
```

8.3 Planning Problem Example during Mission Execution

In the following, an example problem for the Slave planner instance, generated during mission execution, is shown together with an example solution for exactly that problem.

8.3.1 Planning Problem Description during Mission Execution

This PDDL problem description file has been generated during the experimental campaign described in this dissertation. See Table 8-4 for details.

Table 8-4: Planning problem example during mission execution - metadata

File name	Slave93.5.PROBLEM
Experiment start date	19.05.2011
Experiment start time	14:36
Mission[64]	4
Run / test person crew	number II
Problem instance	93
LPG process[65]	5

```
;; AUTO-GENERATED BY PDDL_WRAPPER

(define (problem autogen)

(:domain MUMT)

(:objects
 MiRA UAV1 UAV2 UAV3 - aircraft
 LEADER_SQUAD_A LEADER_SQUAD_B - troops
 TANGO_FRIEND TANGO_FOE ZULU_FRIEND ZULU_FOE LIMA_FRIEND LIMA_FOE
MIKE_FRIEND MIKE_FOE TARGET POI_0 UAV1_MMP_POS_32 CRASH_SITE X_UAV2_3100
EDNY_SE1 EDNY_W2 MOB AP_FOB DP_FOB FOB AP_ISAR1 DP_ISAR1 ISAR1 AP_ISAR2
DP_ISAR2 ISAR2 AP_PICKUP DP_PICKUP PICKUP_POINT HOA_ENTRY HOA_EXIT
MiRA_MMP_POS_93 UAV1_MMP_POS_93 FORCE_0 FORCE_1 FORCE_2 FORCE_3 FORCE_4
FORCE_5 FORCE_6 FORCE_7 FORCE_8 - location
)

(:init
 (in_foe_terr TANGO_FOE)
 (in_foe_terr ZULU_FOE)
 (in_foe_terr LIMA_FOE)
 (in_foe_terr MIKE_FOE)
 (in_foe_terr TARGET)
 (in_hoa TARGET)
 (in_foe_terr POI_0)
```

[64] See Table 5-1.
[65] See Table 4-1.

```
(in_hoa POI_0)
(in_foe_terr UAV1_MMP_POS_32)
(in_foe_terr CRASH_SITE)
(in_foe_terr X_UAV2_3100)
(in_foe_terr AP_ISAR1)
(in_hoa AP_ISAR1)
(in_foe_terr DP_ISAR1)
(in_hoa DP_ISAR1)
(in_foe_terr ISAR1)
(in_hoa ISAR1)
(in_foe_terr AP_ISAR2)
(in_hoa AP_ISAR2)
(in_foe_terr DP_ISAR2)
(in_hoa DP_ISAR2)
(in_foe_terr ISAR2)
(in_hoa ISAR2)
(in_foe_terr AP_PICKUP)
(in_foe_terr DP_PICKUP)
(in_foe_terr PICKUP_POINT)
(in_foe_terr HOA_ENTRY)
(in_foe_terr HOA_EXIT)
(in_foe_terr MiRA_MMP_POS_93)
(in_foe_terr FORCE_0)
(in_foe_terr FORCE_1)
(in_foe_terr FORCE_2)
(in_foe_terr FORCE_3)
(in_foe_terr FORCE_4)
(in_hoa FORCE_4)
(in_foe_terr FORCE_5)
(in_foe_terr FORCE_6)
(in_foe_terr FORCE_7)
(in_foe_terr FORCE_8)
(dist_known TANGO_FRIEND TANGO_FOE)
(= (dist TANGO_FRIEND TANGO_FOE) 1.47)
(dist_known ZULU_FOE ZULU_FRIEND)
(= (dist ZULU_FOE ZULU_FRIEND) 1.51)
(dist_known LIMA_FOE LIMA_FRIEND)
(= (dist LIMA_FOE LIMA_FRIEND) 1.54)
(dist_known MIKE_FRIEND MIKE_FOE)
(= (dist MIKE_FRIEND MIKE_FOE) 1.44)
(dist_known TANGO_FRIEND ZULU_FRIEND)
(= (dist TANGO_FRIEND ZULU_FRIEND) 8.81)
(dist_known TANGO_FRIEND LIMA_FRIEND)
(= (dist TANGO_FRIEND LIMA_FRIEND) 12.50)
(dist_known TANGO_FRIEND MIKE_FRIEND)
(= (dist TANGO_FRIEND MIKE_FRIEND) 3.18)
(dist_known TANGO_FRIEND EDNY_SE1)
(= (dist TANGO_FRIEND EDNY_SE1) 8.86)
(dist_known TANGO_FRIEND EDNY_W2)
(= (dist TANGO_FRIEND EDNY_W2) 6.44)
(dist_known TANGO_FRIEND MOB)
(= (dist TANGO_FRIEND MOB) 7.46)
(dist_known TANGO_FRIEND AP_FOB)
(= (dist TANGO_FRIEND AP_FOB) 6.47)
(dist_known TANGO_FRIEND DP_FOB)
(= (dist TANGO_FRIEND DP_FOB) 5.41)
(dist_known TANGO_FRIEND FOB)
(= (dist TANGO_FRIEND FOB) 6.37)
(dist_known TANGO_FRIEND UAV1_MMP_POS_93)
(= (dist TANGO_FRIEND UAV1_MMP_POS_93) 8.02)
(dist_known ZULU_FRIEND TANGO_FRIEND)
(= (dist ZULU_FRIEND TANGO_FRIEND) 8.81)
```

```
(dist_known ZULU_FRIEND LIMA_FRIEND)
(= (dist ZULU_FRIEND LIMA_FRIEND) 3.69)
(dist_known ZULU_FRIEND MIKE_FRIEND)
(= (dist ZULU_FRIEND MIKE_FRIEND) 11.86)
(dist_known ZULU_FRIEND EDNY_SE1)
(= (dist ZULU_FRIEND EDNY_SE1) 10.92)
(dist_known ZULU_FRIEND EDNY_W2)
(= (dist ZULU_FRIEND EDNY_W2) 12.41)
(dist_known ZULU_FRIEND MOB)
(= (dist ZULU_FRIEND MOB) 11.63)
(dist_known ZULU_FRIEND AP_FOB)
(= (dist ZULU_FRIEND AP_FOB) 13.04)
(dist_known ZULU_FRIEND DP_FOB)
(= (dist ZULU_FRIEND DP_FOB) 12.87)
(dist_known ZULU_FRIEND FOB)
(= (dist ZULU_FRIEND FOB) 13.67)
(dist_known ZULU_FRIEND UAV1_MMP_POS_93)
(= (dist ZULU_FRIEND UAV1_MMP_POS_93) 5.15)
(dist_known LIMA_FRIEND TANGO_FRIEND)
(= (dist LIMA_FRIEND TANGO_FRIEND) 12.50)
(dist_known LIMA_FRIEND ZULU_FRIEND)
(= (dist LIMA_FRIEND ZULU_FRIEND) 3.69)
(dist_known LIMA_FRIEND MIKE_FRIEND)
(= (dist LIMA_FRIEND MIKE_FRIEND) 15.52)
(dist_known LIMA_FRIEND EDNY_SE1)
(= (dist LIMA_FRIEND EDNY_SE1) 13.67)
(dist_known LIMA_FRIEND EDNY_W2)
(= (dist LIMA_FRIEND EDNY_W2) 15.82)
(dist_known LIMA_FRIEND MOB)
(= (dist LIMA_FRIEND MOB) 14.78)
(dist_known LIMA_FRIEND AP_FOB)
(= (dist LIMA_FRIEND AP_FOB) 16.51)
(dist_known LIMA_FRIEND DP_FOB)
(= (dist LIMA_FRIEND DP_FOB) 16.46)
(dist_known LIMA_FRIEND FOB)
(= (dist LIMA_FRIEND FOB) 17.22)
(dist_known LIMA_FRIEND UAV1_MMP_POS_93)
(= (dist LIMA_FRIEND UAV1_MMP_POS_93) 7.70)
(dist_known MIKE_FRIEND TANGO_FRIEND)
(= (dist MIKE_FRIEND TANGO_FRIEND) 3.18)
(dist_known MIKE_FRIEND ZULU_FRIEND)
(= (dist MIKE_FRIEND ZULU_FRIEND) 11.86)
(dist_known MIKE_FRIEND LIMA_FRIEND)
(= (dist MIKE_FRIEND LIMA_FRIEND) 15.52)
(dist_known MIKE_FRIEND EDNY_SE1)
(= (dist MIKE_FRIEND EDNY_SE1) 10.92)
(dist_known MIKE_FRIEND EDNY_W2)
(= (dist MIKE_FRIEND EDNY_W2) 7.25)
(dist_known MIKE_FRIEND MOB)
(= (dist MIKE_FRIEND MOB) 8.99)
(dist_known MIKE_FRIEND AP_FOB)
(= (dist MIKE_FRIEND AP_FOB) 6.85)
(dist_known MIKE_FRIEND DP_FOB)
(= (dist MIKE_FRIEND DP_FOB) 5.33)
(dist_known MIKE_FRIEND FOB)
(= (dist MIKE_FRIEND FOB) 6.16)
(dist_known MIKE_FRIEND UAV1_MMP_POS_93)
(= (dist MIKE_FRIEND UAV1_MMP_POS_93) 11.11)
(dist_known EDNY_SE1 TANGO_FRIEND)
(= (dist EDNY_SE1 TANGO_FRIEND) 8.86)
(dist_known EDNY_SE1 ZULU_FRIEND)
(= (dist EDNY_SE1 ZULU_FRIEND) 10.92)
```

```
(dist_known EDNY_SE1 LIMA_FRIEND)
(= (dist EDNY_SE1 LIMA_FRIEND) 13.67)
(dist_known EDNY_SE1 MIKE_FRIEND)
(= (dist EDNY_SE1 MIKE_FRIEND) 10.92)
(dist_known EDNY_SE1 EDNY_W2)
(= (dist EDNY_SE1 EDNY_W2) 4.77)
(dist_known EDNY_SE1 MOB)
(= (dist EDNY_SE1 MOB) 2.49)
(dist_known EDNY_SE1 AP_FOB)
(= (dist EDNY_SE1 AP_FOB) 5.70)
(dist_known EDNY_SE1 DP_FOB)
(= (dist EDNY_SE1 DP_FOB) 6.95)
(dist_known EDNY_SE1 FOB)
(= (dist EDNY_SE1 FOB) 6.97)
(dist_known EDNY_SE1 UAV1_MMP_POS_93)
(= (dist EDNY_SE1 UAV1_MMP_POS_93) 5.97)
(dist_known EDNY_W2 TANGO_FRIEND)
(= (dist EDNY_W2 TANGO_FRIEND) 6.44)
(dist_known EDNY_W2 ZULU_FRIEND)
(= (dist EDNY_W2 ZULU_FRIEND) 12.41)
(dist_known EDNY_W2 LIMA_FRIEND)
(= (dist EDNY_W2 LIMA_FRIEND) 15.82)
(dist_known EDNY_W2 MIKE_FRIEND)
(= (dist EDNY_W2 MIKE_FRIEND) 7.25)
(dist_known EDNY_W2 EDNY_SE1)
(= (dist EDNY_W2 EDNY_SE1) 4.77)
(dist_known EDNY_W2 MOB)
(= (dist EDNY_W2 MOB) 2.29)
(dist_known EDNY_W2 AP_FOB)
(= (dist EDNY_W2 AP_FOB) 0.94)
(dist_known EDNY_W2 DP_FOB)
(= (dist EDNY_W2 DP_FOB) 2.30)
(dist_known EDNY_W2 FOB)
(= (dist EDNY_W2 FOB) 2.20)
(dist_known EDNY_W2 UAV1_MMP_POS_93)
(= (dist EDNY_W2 UAV1_MMP_POS_93) 8.65)
(dist_known MOB TANGO_FRIEND)
(= (dist MOB TANGO_FRIEND) 7.46)
(dist_known MOB ZULU_FRIEND)
(= (dist MOB ZULU_FRIEND) 11.63)
(dist_known MOB LIMA_FRIEND)
(= (dist MOB LIMA_FRIEND) 14.78)
(dist_known MOB MIKE_FRIEND)
(= (dist MOB MIKE_FRIEND) 8.99)
(dist_known MOB EDNY_SE1)
(= (dist MOB EDNY_SE1) 2.49)
(dist_known MOB EDNY_W2)
(= (dist MOB EDNY_W2) 2.29)
(dist_known MOB AP_FOB)
(= (dist MOB AP_FOB) 3.21)
(dist_known MOB DP_FOB)
(= (dist MOB DP_FOB) 4.53)
(dist_known MOB FOB)
(= (dist MOB FOB) 4.49)
(dist_known MOB UAV1_MMP_POS_93)
(= (dist MOB UAV1_MMP_POS_93) 7.24)
(dist_known AP_FOB TANGO_FRIEND)
(= (dist AP_FOB TANGO_FRIEND) 6.47)
(dist_known AP_FOB ZULU_FRIEND)
(= (dist AP_FOB ZULU_FRIEND) 13.04)
(dist_known AP_FOB LIMA_FRIEND)
(= (dist AP_FOB LIMA_FRIEND) 16.51)
```

```
(dist_known AP_FOB MIKE_FRIEND)
(= (dist AP_FOB MIKE_FRIEND) 6.85)
(dist_known AP_FOB EDNY_SE1)
(= (dist AP_FOB EDNY_SE1) 5.70)
(dist_known AP_FOB EDNY_W2)
(= (dist AP_FOB EDNY_W2) 0.94)
(dist_known AP_FOB MOB)
(= (dist AP_FOB MOB) 3.21)
(dist_known AP_FOB DP_FOB)
(= (dist AP_FOB DP_FOB) 1.60)
(dist_known AP_FOB FOB)
(= (dist AP_FOB FOB) 1.28)
(dist_known AP_FOB UAV1_MMP_POS_93)
(= (dist AP_FOB UAV1_MMP_POS_93) 9.48)
(dist_known DP_FOB TANGO_FRIEND)
(= (dist DP_FOB TANGO_FRIEND) 5.41)
(dist_known DP_FOB ZULU_FRIEND)
(= (dist DP_FOB ZULU_FRIEND) 12.87)
(dist_known DP_FOB LIMA_FRIEND)
(= (dist DP_FOB LIMA_FRIEND) 16.46)
(dist_known DP_FOB MIKE_FRIEND)
(= (dist DP_FOB MIKE_FRIEND) 5.33)
(dist_known DP_FOB EDNY_SE1)
(= (dist DP_FOB EDNY_SE1) 6.95)
(dist_known DP_FOB EDNY_W2)
(= (dist DP_FOB EDNY_W2) 2.30)
(dist_known DP_FOB MOB)
(= (dist DP_FOB MOB) 4.53)
(dist_known DP_FOB AP_FOB)
(= (dist DP_FOB AP_FOB) 1.60)
(dist_known DP_FOB FOB)
(= (dist DP_FOB FOB) 0.96)
(dist_known DP_FOB UAV1_MMP_POS_93)
(= (dist DP_FOB UAV1_MMP_POS_93) 9.88)
(dist_known FOB TANGO_FRIEND)
(= (dist FOB TANGO_FRIEND) 6.37)
(dist_known FOB ZULU_FRIEND)
(= (dist FOB ZULU_FRIEND) 13.67)
(dist_known FOB LIMA_FRIEND)
(= (dist FOB LIMA_FRIEND) 17.22)
(dist_known FOB MIKE_FRIEND)
(= (dist FOB MIKE_FRIEND) 6.16)
(dist_known FOB EDNY_SE1)
(= (dist FOB EDNY_SE1) 6.97)
(dist_known FOB EDNY_W2)
(= (dist FOB EDNY_W2) 2.20)
(dist_known FOB MOB)
(= (dist FOB MOB) 4.49)
(dist_known FOB AP_FOB)
(= (dist FOB AP_FOB) 1.28)
(dist_known FOB DP_FOB)
(= (dist FOB DP_FOB) 0.96)
(dist_known FOB UAV1_MMP_POS_93)
(= (dist FOB UAV1_MMP_POS_93) 10.44)
(dist_known UAV1_MMP_POS_93 TANGO_FRIEND)
(= (dist UAV1_MMP_POS_93 TANGO_FRIEND) 8.02)
(dist_known UAV1_MMP_POS_93 ZULU_FRIEND)
(= (dist UAV1_MMP_POS_93 ZULU_FRIEND) 5.15)
(dist_known UAV1_MMP_POS_93 LIMA_FRIEND)
(= (dist UAV1_MMP_POS_93 LIMA_FRIEND) 7.70)
(dist_known UAV1_MMP_POS_93 MIKE_FRIEND)
(= (dist UAV1_MMP_POS_93 MIKE_FRIEND) 11.11)
```

```
(dist_known UAV1_MMP_POS_93 EDNY_SE1)
(= (dist UAV1_MMP_POS_93 EDNY_SE1) 5.97)
(dist_known UAV1_MMP_POS_93 EDNY_W2)
(= (dist UAV1_MMP_POS_93 EDNY_W2) 8.65)
(dist_known UAV1_MMP_POS_93 MOB)
(= (dist UAV1_MMP_POS_93 MOB) 7.24)
(dist_known UAV1_MMP_POS_93 AP_FOB)
(= (dist UAV1_MMP_POS_93 AP_FOB) 9.48)
(dist_known UAV1_MMP_POS_93 DP_FOB)
(= (dist UAV1_MMP_POS_93 DP_FOB) 9.88)
(dist_known UAV1_MMP_POS_93 FOB)
(= (dist UAV1_MMP_POS_93 FOB) 10.44)
(dist_known TANGO_FOE ZULU_FOE)
(= (dist TANGO_FOE ZULU_FOE) 8.46)
(dist_known TANGO_FOE LIMA_FOE)
(= (dist TANGO_FOE LIMA_FOE) 11.81)
(dist_known TANGO_FOE MIKE_FOE)
(= (dist TANGO_FOE MIKE_FOE) 2.86)
(dist_known TANGO_FOE UAV1_MMP_POS_32)
(= (dist TANGO_FOE UAV1_MMP_POS_32) 7.05)
(dist_known TANGO_FOE CRASH_SITE)
(= (dist TANGO_FOE CRASH_SITE) 10.44)
(dist_known TANGO_FOE X_UAV2_3100)
(= (dist TANGO_FOE X_UAV2_3100) 11.41)
(dist_known TANGO_FOE AP_PICKUP)
(= (dist TANGO_FOE AP_PICKUP) 10.83)
(dist_known TANGO_FOE DP_PICKUP)
(= (dist TANGO_FOE DP_PICKUP) 11.31)
(dist_known TANGO_FOE PICKUP_POINT)
(= (dist TANGO_FOE PICKUP_POINT) 11.12)
(dist_known TANGO_FOE HOA_ENTRY)
(= (dist TANGO_FOE HOA_ENTRY) 5.48)
(= (section_cleared_beginning TANGO_FOE HOA_ENTRY) 3)
(= (section_cleared_completely TANGO_FOE HOA_ENTRY) 3)
(dist_known TANGO_FOE HOA_EXIT)
(= (dist TANGO_FOE HOA_EXIT) 6.77)
(dist_known TANGO_FOE MiRA_MMP_POS_93)
(= (dist TANGO_FOE MiRA_MMP_POS_93) 11.49)
(dist_known TANGO_FOE FORCE_0)
(= (dist TANGO_FOE FORCE_0) 0.69)
(dist_known TANGO_FOE FORCE_1)
(= (dist TANGO_FOE FORCE_1) 1.88)
(dist_known TANGO_FOE FORCE_2)
(= (dist TANGO_FOE FORCE_2) 2.11)
(dist_known TANGO_FOE FORCE_3)
(= (dist TANGO_FOE FORCE_3) 3.72)
(dist_known TANGO_FOE FORCE_5)
(= (dist TANGO_FOE FORCE_5) 8.06)
(dist_known TANGO_FOE FORCE_6)
(= (dist TANGO_FOE FORCE_6) 7.22)
(dist_known TANGO_FOE FORCE_7)
(= (dist TANGO_FOE FORCE_7) 10.95)
(dist_known TANGO_FOE FORCE_8)
(= (dist TANGO_FOE FORCE_8) 10.61)
(dist_known ZULU_FOE TANGO_FOE)
(= (dist ZULU_FOE TANGO_FOE) 8.46)
(dist_known ZULU_FOE LIMA_FOE)
(= (dist ZULU_FOE LIMA_FOE) 3.36)
(dist_known ZULU_FOE MIKE_FOE)
(= (dist ZULU_FOE MIKE_FOE) 11.21)
(dist_known ZULU_FOE UAV1_MMP_POS_32)
(= (dist ZULU_FOE UAV1_MMP_POS_32) 3.01)
```

```
(dist_known ZULU_FOE CRASH_SITE)
(= (dist ZULU_FOE CRASH_SITE) 4.71)
(dist_known ZULU_FOE X_UAV2_3100)
(= (dist ZULU_FOE X_UAV2_3100) 3.27)
(dist_known ZULU_FOE AP_PICKUP)
(= (dist ZULU_FOE AP_PICKUP) 5.41)
(dist_known ZULU_FOE DP_PICKUP)
(= (dist ZULU_FOE DP_PICKUP) 5.11)
(dist_known ZULU_FOE PICKUP_POINT)
(= (dist ZULU_FOE PICKUP_POINT) 5.29)
(dist_known ZULU_FOE HOA_ENTRY)
(= (dist ZULU_FOE HOA_ENTRY) 8.77)
(dist_known ZULU_FOE HOA_EXIT)
(= (dist ZULU_FOE HOA_EXIT) 6.16)
(dist_known ZULU_FOE MiRA_MMP_POS_93)
(= (dist ZULU_FOE MiRA_MMP_POS_93) 4.11)
(dist_known ZULU_FOE FORCE_0)
(= (dist ZULU_FOE FORCE_0) 8.48)
(dist_known ZULU_FOE FORCE_1)
(= (dist ZULU_FOE FORCE_1) 7.92)
(dist_known ZULU_FOE FORCE_2)
(= (dist ZULU_FOE FORCE_2) 7.97)
(dist_known ZULU_FOE FORCE_3)
(= (dist ZULU_FOE FORCE_3) 8.02)
(dist_known ZULU_FOE FORCE_5)
(= (dist ZULU_FOE FORCE_5) 0.45)
(dist_known ZULU_FOE FORCE_6)
(= (dist ZULU_FOE FORCE_6) 4.62)
(dist_known ZULU_FOE FORCE_7)
(= (dist ZULU_FOE FORCE_7) 2.74)
(dist_known ZULU_FOE FORCE_8)
(= (dist ZULU_FOE FORCE_8) 3.89)
(dist_known LIMA_FOE TANGO_FOE)
(= (dist LIMA_FOE TANGO_FOE) 11.81)
(dist_known LIMA_FOE ZULU_FOE)
(= (dist LIMA_FOE ZULU_FOE) 3.36)
(dist_known LIMA_FOE MIKE_FOE)
(= (dist LIMA_FOE MIKE_FOE) 14.54)
(dist_known LIMA_FOE UAV1_MMP_POS_32)
(= (dist LIMA_FOE UAV1_MMP_POS_32) 5.62)
(dist_known LIMA_FOE CRASH_SITE)
(= (dist LIMA_FOE CRASH_SITE) 5.04)
(dist_known LIMA_FOE X_UAV2_3100)
(= (dist LIMA_FOE X_UAV2_3100) 1.50)
(dist_known LIMA_FOE AP_PICKUP)
(= (dist LIMA_FOE AP_PICKUP) 5.62)
(dist_known LIMA_FOE DP_PICKUP)
(= (dist LIMA_FOE DP_PICKUP) 4.83)
(dist_known LIMA_FOE PICKUP_POINT)
(= (dist LIMA_FOE PICKUP_POINT) 5.23)
(dist_known LIMA_FOE HOA_ENTRY)
(= (dist LIMA_FOE HOA_ENTRY) 11.52)
(dist_known LIMA_FOE HOA_EXIT)
(= (dist LIMA_FOE HOA_EXIT) 8.50)
(dist_known LIMA_FOE MiRA_MMP_POS_93)
(= (dist LIMA_FOE MiRA_MMP_POS_93) 3.07)
(dist_known LIMA_FOE FORCE_0)
(= (dist LIMA_FOE FORCE_0) 11.81)
(dist_known LIMA_FOE FORCE_1)
(= (dist LIMA_FOE FORCE_1) 11.18)
(dist_known LIMA_FOE FORCE_2)
(= (dist LIMA_FOE FORCE_2) 11.21)
```

```
(dist_known LIMA_FOE FORCE_3)
(= (dist LIMA_FOE FORCE_3) 11.05)
(dist_known LIMA_FOE FORCE_5)
(= (dist LIMA_FOE FORCE_5) 3.75)
(dist_known LIMA_FOE FORCE_6)
(= (dist LIMA_FOE FORCE_6) 6.86)
(dist_known LIMA_FOE FORCE_7)
(= (dist LIMA_FOE FORCE_7) 1.43)
(dist_known LIMA_FOE FORCE_8)
(= (dist LIMA_FOE FORCE_8) 3.81)
(dist_known MIKE_FOE TANGO_FOE)
(= (dist MIKE_FOE TANGO_FOE) 2.86)
(dist_known MIKE_FOE ZULU_FOE)
(= (dist MIKE_FOE ZULU_FOE) 11.21)
(dist_known MIKE_FOE LIMA_FOE)
(= (dist MIKE_FOE LIMA_FOE) 14.54)
(dist_known MIKE_FOE UAV1_MMP_POS_32)
(= (dist MIKE_FOE UAV1_MMP_POS_32) 9.45)
(dist_known MIKE_FOE CRASH_SITE)
(= (dist MIKE_FOE CRASH_SITE) 12.62)
(dist_known MIKE_FOE X_UAV2_3100)
(= (dist MIKE_FOE X_UAV2_3100) 14.02)
(dist_known MIKE_FOE AP_PICKUP)
(= (dist MIKE_FOE AP_PICKUP) 12.91)
(dist_known MIKE_FOE DP_PICKUP)
(= (dist MIKE_FOE DP_PICKUP) 13.51)
(dist_known MIKE_FOE PICKUP_POINT)
(= (dist MIKE_FOE PICKUP_POINT) 13.27)
(dist_known MIKE_FOE HOA_ENTRY)
(= (dist MIKE_FOE HOA_ENTRY) 6.03)
(dist_known MIKE_FOE HOA_EXIT)
(= (dist MIKE_FOE HOA_EXIT) 8.39)
(dist_known MIKE_FOE MiRA_MMP_POS_93)
(= (dist MIKE_FOE MiRA_MMP_POS_93) 13.93)
(dist_known MIKE_FOE FORCE_0)
(= (dist MIKE_FOE FORCE_0) 2.73)
(dist_known MIKE_FOE FORCE_1)
(= (dist MIKE_FOE FORCE_1) 3.54)
(dist_known MIKE_FOE FORCE_2)
(= (dist MIKE_FOE FORCE_2) 3.60)
(dist_known MIKE_FOE FORCE_3)
(= (dist MIKE_FOE FORCE_3) 4.72)
(dist_known MIKE_FOE FORCE_5)
(= (dist MIKE_FOE FORCE_5) 10.79)
(dist_known MIKE_FOE FORCE_6)
(= (dist MIKE_FOE FORCE_6) 9.26)
(dist_known MIKE_FOE FORCE_7)
(= (dist MIKE_FOE FORCE_7) 13.59)
(dist_known MIKE_FOE FORCE_8)
(= (dist MIKE_FOE FORCE_8) 12.97)
(dist_known UAV1_MMP_POS_32 TANGO_FOE)
(= (dist UAV1_MMP_POS_32 TANGO_FOE) 7.05)
(dist_known UAV1_MMP_POS_32 ZULU_FOE)
(= (dist UAV1_MMP_POS_32 ZULU_FOE) 3.01)
(dist_known UAV1_MMP_POS_32 LIMA_FOE)
(= (dist UAV1_MMP_POS_32 LIMA_FOE) 5.62)
(dist_known UAV1_MMP_POS_32 MIKE_FOE)
(= (dist UAV1_MMP_POS_32 MIKE_FOE) 9.45)
(dist_known UAV1_MMP_POS_32 CRASH_SITE)
(= (dist UAV1_MMP_POS_32 CRASH_SITE) 3.53)
(dist_known UAV1_MMP_POS_32 X_UAV2_3100)
(= (dist UAV1_MMP_POS_32 X_UAV2_3100) 4.72)
```

```
(dist_known UAV1_MMP_POS_32 AP_PICKUP)
(= (dist UAV1_MMP_POS_32 AP_PICKUP) 4.06)
(dist_known UAV1_MMP_POS_32 DP_PICKUP)
(= (dist UAV1_MMP_POS_32 DP_PICKUP) 4.34)
(dist_known UAV1_MMP_POS_32 PICKUP_POINT)
(= (dist UAV1_MMP_POS_32 PICKUP_POINT) 4.23)
(dist_known UAV1_MMP_POS_32 HOA_ENTRY)
(= (dist UAV1_MMP_POS_32 HOA_ENTRY) 5.92)
(dist_known UAV1_MMP_POS_32 HOA_EXIT)
(= (dist UAV1_MMP_POS_32 HOA_EXIT) 3.15)
(= (section_cleared_beginning UAV1_MMP_POS_32 HOA_EXIT) 1)
(= (section_cleared_completely UAV1_MMP_POS_32 HOA_EXIT) 1)
(dist_known UAV1_MMP_POS_32 MiRA_MMP_POS_93)
(= (dist UAV1_MMP_POS_32 MiRA_MMP_POS_93) 4.48)
(dist_known UAV1_MMP_POS_32 FORCE_0)
(= (dist UAV1_MMP_POS_32 FORCE_0) 6.83)
(dist_known UAV1_MMP_POS_32 FORCE_1)
(= (dist UAV1_MMP_POS_32 FORCE_1) 5.93)
(dist_known UAV1_MMP_POS_32 FORCE_2)
(= (dist UAV1_MMP_POS_32 FORCE_2) 5.90)
(dist_known UAV1_MMP_POS_32 FORCE_3)
(= (dist UAV1_MMP_POS_32 FORCE_3) 5.49)
(dist_known UAV1_MMP_POS_32 FORCE_5)
(= (dist UAV1_MMP_POS_32 FORCE_5) 2.62)
(dist_known UAV1_MMP_POS_32 FORCE_6)
(= (dist UAV1_MMP_POS_32 FORCE_6) 1.66)
(dist_known UAV1_MMP_POS_32 FORCE_7)
(= (dist UAV1_MMP_POS_32 FORCE_7) 4.38)
(dist_known UAV1_MMP_POS_32 FORCE_8)
(= (dist UAV1_MMP_POS_32 FORCE_8) 3.57)
(dist_known CRASH_SITE TANGO_FOE)
(= (dist CRASH_SITE TANGO_FOE) 10.44)
(dist_known CRASH_SITE ZULU_FOE)
(= (dist CRASH_SITE ZULU_FOE) 4.71)
(dist_known CRASH_SITE LIMA_FOE)
(= (dist CRASH_SITE LIMA_FOE) 5.04)
(dist_known CRASH_SITE MIKE_FOE)
(= (dist CRASH_SITE MIKE_FOE) 12.62)
(dist_known CRASH_SITE UAV1_MMP_POS_32)
(= (dist CRASH_SITE UAV1_MMP_POS_32) 3.53)
(dist_known CRASH_SITE X_UAV2_3100)
(= (dist CRASH_SITE X_UAV2_3100) 3.56)
(dist_known CRASH_SITE AP_PICKUP)
(= (dist CRASH_SITE AP_PICKUP) 0.70)
(dist_known CRASH_SITE DP_PICKUP)
(= (dist CRASH_SITE DP_PICKUP) 0.90)
(dist_known CRASH_SITE PICKUP_POINT)
(= (dist CRASH_SITE PICKUP_POINT) 0.71)
(dist_known CRASH_SITE HOA_ENTRY)
(= (dist CRASH_SITE HOA_ENTRY) 7.92)
(dist_known CRASH_SITE HOA_EXIT)
(= (dist CRASH_SITE HOA_EXIT) 4.67)
(dist_known CRASH_SITE MiRA_MMP_POS_93)
(= (dist CRASH_SITE MiRA_MMP_POS_93) 2.21)
(dist_known CRASH_SITE FORCE_0)
(= (dist CRASH_SITE FORCE_0) 10.14)
(dist_known CRASH_SITE FORCE_1)
(= (dist CRASH_SITE FORCE_1) 9.09)
(dist_known CRASH_SITE FORCE_2)
(= (dist CRASH_SITE FORCE_2) 9.01)
(dist_known CRASH_SITE FORCE_3)
(= (dist CRASH_SITE FORCE_3) 8.18)
```

```
(dist_known CRASH_SITE FORCE_5)
(= (dist CRASH_SITE FORCE_5) 4.61)
(dist_known CRASH_SITE FORCE_6)
(= (dist CRASH_SITE FORCE_6) 3.38)
(dist_known CRASH_SITE FORCE_7)
(= (dist CRASH_SITE FORCE_7) 3.65)
(dist_known CRASH_SITE FORCE_8)
(= (dist CRASH_SITE FORCE_8) 1.23)
(dist_known X_UAV2_3100 TANGO_FOE)
(= (dist X_UAV2_3100 TANGO_FOE) 11.41)
(dist_known X_UAV2_3100 ZULU_FOE)
(= (dist X_UAV2_3100 ZULU_FOE) 3.27)
(dist_known X_UAV2_3100 LIMA_FOE)
(= (dist X_UAV2_3100 LIMA_FOE) 1.50)
(dist_known X_UAV2_3100 MIKE_FOE)
(= (dist X_UAV2_3100 MIKE_FOE) 14.02)
(dist_known X_UAV2_3100 UAV1_MMP_POS_32)
(= (dist X_UAV2_3100 UAV1_MMP_POS_32) 4.72)
(dist_known X_UAV2_3100 CRASH_SITE)
(= (dist X_UAV2_3100 CRASH_SITE) 3.56)
(dist_known X_UAV2_3100 AP_PICKUP)
(= (dist X_UAV2_3100 AP_PICKUP) 4.13)
(dist_known X_UAV2_3100 DP_PICKUP)
(= (dist X_UAV2_3100 DP_PICKUP) 3.32)
(dist_known X_UAV2_3100 PICKUP_POINT)
(= (dist X_UAV2_3100 PICKUP_POINT) 3.72)
(dist_known X_UAV2_3100 HOA_ENTRY)
(= (dist X_UAV2_3100 HOA_ENTRY) 10.49)
(dist_known X_UAV2_3100 HOA_EXIT)
(= (dist X_UAV2_3100 HOA_EXIT) 7.35)
(dist_known X_UAV2_3100 MiRA_MMP_POS_93)
(= (dist X_UAV2_3100 MiRA_MMP_POS_93) 1.59)
(dist_known X_UAV2_3100 FORCE_0)
(= (dist X_UAV2_3100 FORCE_0) 11.32)
(dist_known X_UAV2_3100 FORCE_1)
(= (dist X_UAV2_3100 FORCE_1) 10.56)
(dist_known X_UAV2_3100 FORCE_2)
(= (dist X_UAV2_3100 FORCE_2) 10.56)
(dist_known X_UAV2_3100 FORCE_3)
(= (dist X_UAV2_3100 FORCE_3) 10.21)
(dist_known X_UAV2_3100 FORCE_5)
(= (dist X_UAV2_3100 FORCE_5) 3.54)
(dist_known X_UAV2_3100 FORCE_6)
(= (dist X_UAV2_3100 FORCE_6) 5.73)
(dist_known X_UAV2_3100 FORCE_7)
(= (dist X_UAV2_3100 FORCE_7) 0.53)
(dist_known X_UAV2_3100 FORCE_8)
(= (dist X_UAV2_3100 FORCE_8) 2.33)
(dist_known AP_PICKUP TANGO_FOE)
(= (dist AP_PICKUP TANGO_FOE) 10.83)
(dist_known AP_PICKUP ZULU_FOE)
(= (dist AP_PICKUP ZULU_FOE) 5.41)
(dist_known AP_PICKUP LIMA_FOE)
(= (dist AP_PICKUP LIMA_FOE) 5.62)
(dist_known AP_PICKUP MIKE_FOE)
(= (dist AP_PICKUP MIKE_FOE) 12.91)
(dist_known AP_PICKUP UAV1_MMP_POS_32)
(= (dist AP_PICKUP UAV1_MMP_POS_32) 4.06)
(dist_known AP_PICKUP CRASH_SITE)
(= (dist AP_PICKUP CRASH_SITE) 0.70)
(dist_known AP_PICKUP X_UAV2_3100)
(= (dist AP_PICKUP X_UAV2_3100) 4.13)
```

```
(dist_known AP_PICKUP DP_PICKUP)
(= (dist AP_PICKUP DP_PICKUP) 0.97)
(dist_known AP_PICKUP PICKUP_POINT)
(= (dist AP_PICKUP PICKUP_POINT) 0.51)
(dist_known AP_PICKUP HOA_ENTRY)
(= (dist AP_PICKUP HOA_ENTRY) 7.98)
(dist_known AP_PICKUP HOA_EXIT)
(= (dist AP_PICKUP HOA_EXIT) 4.76)
(dist_known AP_PICKUP MiRA_MMP_POS_93)
(= (dist AP_PICKUP MiRA_MMP_POS_93) 2.66)
(dist_known AP_PICKUP FORCE_0)
(= (dist AP_PICKUP FORCE_0) 10.50)
(dist_known AP_PICKUP FORCE_1)
(= (dist AP_PICKUP FORCE_1) 9.41)
(dist_known AP_PICKUP FORCE_2)
(= (dist AP_PICKUP FORCE_2) 9.32)
(dist_known AP_PICKUP FORCE_3)
(= (dist AP_PICKUP FORCE_3) 8.38)
(dist_known AP_PICKUP FORCE_5)
(= (dist AP_PICKUP FORCE_5) 5.31)
(dist_known AP_PICKUP FORCE_6)
(= (dist AP_PICKUP FORCE_6) 3.66)
(dist_known AP_PICKUP FORCE_7)
(= (dist AP_PICKUP FORCE_7) 4.27)
(dist_known AP_PICKUP FORCE_8)
(= (dist AP_PICKUP FORCE_8) 1.84)
(dist_known DP_PICKUP TANGO_FOE)
(= (dist DP_PICKUP TANGO_FOE) 11.31)
(dist_known DP_PICKUP ZULU_FOE)
(= (dist DP_PICKUP ZULU_FOE) 5.11)
(dist_known DP_PICKUP LIMA_FOE)
(= (dist DP_PICKUP LIMA_FOE) 4.83)
(dist_known DP_PICKUP MIKE_FOE)
(= (dist DP_PICKUP MIKE_FOE) 13.51)
(dist_known DP_PICKUP UAV1_MMP_POS_32)
(= (dist DP_PICKUP UAV1_MMP_POS_32) 4.34)
(dist_known DP_PICKUP CRASH_SITE)
(= (dist DP_PICKUP CRASH_SITE) 0.90)
(dist_known DP_PICKUP X_UAV2_3100)
(= (dist DP_PICKUP X_UAV2_3100) 3.32)
(dist_known DP_PICKUP AP_PICKUP)
(= (dist DP_PICKUP AP_PICKUP) 0.97)
(dist_known DP_PICKUP PICKUP_POINT)
(= (dist DP_PICKUP PICKUP_POINT) 0.46)
(dist_known DP_PICKUP HOA_ENTRY)
(= (dist DP_PICKUP HOA_ENTRY) 8.80)
(dist_known DP_PICKUP HOA_EXIT)
(= (dist DP_PICKUP HOA_EXIT) 5.55)
(dist_known DP_PICKUP MiRA_MMP_POS_93)
(= (dist DP_PICKUP MiRA_MMP_POS_93) 1.78)
(dist_known DP_PICKUP FORCE_0)
(= (dist DP_PICKUP FORCE_0) 11.02)
(dist_known DP_PICKUP FORCE_1)
(= (dist DP_PICKUP FORCE_1) 9.99)
(dist_known DP_PICKUP FORCE_2)
(= (dist DP_PICKUP FORCE_2) 9.91)
(dist_known DP_PICKUP FORCE_3)
(= (dist DP_PICKUP FORCE_3) 9.08)
(dist_known DP_PICKUP FORCE_5)
(= (dist DP_PICKUP FORCE_5) 5.08)
(dist_known DP_PICKUP FORCE_6)
(= (dist DP_PICKUP FORCE_6) 4.28)
```

```
(dist_known DP_PICKUP FORCE_7)
(= (dist DP_PICKUP FORCE_7) 3.54)
(dist_known DP_PICKUP FORCE_8)
(= (dist DP_PICKUP FORCE_8) 1.24)
(dist_known PICKUP_POINT TANGO_FOE)
(= (dist PICKUP_POINT TANGO_FOE) 11.12)
(dist_known PICKUP_POINT ZULU_FOE)
(= (dist PICKUP_POINT ZULU_FOE) 5.29)
(dist_known PICKUP_POINT LIMA_FOE)
(= (dist PICKUP_POINT LIMA_FOE) 5.23)
(dist_known PICKUP_POINT MIKE_FOE)
(= (dist PICKUP_POINT MIKE_FOE) 13.27)
(dist_known PICKUP_POINT UAV1_MMP_POS_32)
(= (dist PICKUP_POINT UAV1_MMP_POS_32) 4.23)
(dist_known PICKUP_POINT CRASH_SITE)
(= (dist PICKUP_POINT CRASH_SITE) 0.71)
(dist_known PICKUP_POINT X_UAV2_3100)
(= (dist PICKUP_POINT X_UAV2_3100) 3.72)
(dist_known PICKUP_POINT AP_PICKUP)
(= (dist PICKUP_POINT AP_PICKUP) 0.51)
(dist_known PICKUP_POINT DP_PICKUP)
(= (dist PICKUP_POINT DP_PICKUP) 0.46)
(dist_known PICKUP_POINT HOA_ENTRY)
(= (dist PICKUP_POINT HOA_ENTRY) 8.43)
(dist_known PICKUP_POINT HOA_EXIT)
(= (dist PICKUP_POINT HOA_EXIT) 5.20)
(dist_known PICKUP_POINT MiRA_MMP_POS_93)
(= (dist PICKUP_POINT MiRA_MMP_POS_93) 2.21)
(dist_known PICKUP_POINT FORCE_0)
(= (dist PICKUP_POINT FORCE_0) 10.81)
(dist_known PICKUP_POINT FORCE_1)
(= (dist PICKUP_POINT FORCE_1) 9.75)
(dist_known PICKUP_POINT FORCE_2)
(= (dist PICKUP_POINT FORCE_2) 9.67)
(dist_known PICKUP_POINT FORCE_3)
(= (dist PICKUP_POINT FORCE_3) 8.78)
(dist_known PICKUP_POINT FORCE_5)
(= (dist PICKUP_POINT FORCE_5) 5.22)
(dist_known PICKUP_POINT FORCE_6)
(= (dist PICKUP_POINT FORCE_6) 4.01)
(dist_known PICKUP_POINT FORCE_7)
(= (dist PICKUP_POINT FORCE_7) 3.91)
(dist_known PICKUP_POINT FORCE_8)
(= (dist PICKUP_POINT FORCE_8) 1.52)
(dist_known HOA_ENTRY TANGO_FOE)
(= (dist HOA_ENTRY TANGO_FOE) 5.48)
(= (section_cleared_beginning HOA_ENTRY TANGO_FOE) 3)
(= (section_cleared_completely HOA_ENTRY TANGO_FOE) 3)
(dist_known HOA_ENTRY ZULU_FOE)
(= (dist HOA_ENTRY ZULU_FOE) 8.77)
(dist_known HOA_ENTRY LIMA_FOE)
(= (dist HOA_ENTRY LIMA_FOE) 11.52)
(dist_known HOA_ENTRY MIKE_FOE)
(= (dist HOA_ENTRY MIKE_FOE) 6.03)
(dist_known HOA_ENTRY UAV1_MMP_POS_32)
(= (dist HOA_ENTRY UAV1_MMP_POS_32) 5.92)
(dist_known HOA_ENTRY CRASH_SITE)
(= (dist HOA_ENTRY CRASH_SITE) 7.92)
(dist_known HOA_ENTRY X_UAV2_3100)
(= (dist HOA_ENTRY X_UAV2_3100) 10.49)
(dist_known HOA_ENTRY AP_PICKUP)
(= (dist HOA_ENTRY AP_PICKUP) 7.98)
```

```
(dist_known HOA_ENTRY DP_PICKUP)
(= (dist HOA_ENTRY DP_PICKUP) 8.80)
(dist_known HOA_ENTRY PICKUP_POINT)
(= (dist HOA_ENTRY PICKUP_POINT) 8.43)
(dist_known HOA_ENTRY HOA_EXIT)
(= (dist HOA_ENTRY HOA_EXIT) 3.26)
(dist_known HOA_ENTRY MiRA_MMP_POS_93)
(= (dist HOA_ENTRY MiRA_MMP_POS_93) 9.80)
(dist_known HOA_ENTRY FORCE_0)
(= (dist HOA_ENTRY FORCE_0) 4.81)
(dist_known HOA_ENTRY FORCE_1)
(= (dist HOA_ENTRY FORCE_1) 3.62)
(dist_known HOA_ENTRY FORCE_2)
(= (dist HOA_ENTRY FORCE_2) 3.39)
(dist_known HOA_ENTRY FORCE_3)
(= (dist HOA_ENTRY FORCE_3) 1.79)
(dist_known HOA_ENTRY FORCE_5)
(= (dist HOA_ENTRY FORCE_5) 8.33)
(dist_known HOA_ENTRY FORCE_6)
(= (dist HOA_ENTRY FORCE_6) 4.78)
(dist_known HOA_ENTRY FORCE_7)
(= (dist HOA_ENTRY FORCE_7) 10.22)
(dist_known HOA_ENTRY FORCE_8)
(= (dist HOA_ENTRY FORCE_8) 8.70)
(dist_known HOA_EXIT TANGO_FOE)
(= (dist HOA_EXIT TANGO_FOE) 6.77)
(dist_known HOA_EXIT ZULU_FOE)
(= (dist HOA_EXIT ZULU_FOE) 6.16)
(dist_known HOA_EXIT LIMA_FOE)
(= (dist HOA_EXIT LIMA_FOE) 8.50)
(dist_known HOA_EXIT MIKE_FOE)
(= (dist HOA_EXIT MIKE_FOE) 8.39)
(dist_known HOA_EXIT UAV1_MMP_POS_32)
(= (dist HOA_EXIT UAV1_MMP_POS_32) 3.15)
(= (section_cleared_beginning HOA_EXIT UAV1_MMP_POS_32) 1)
(= (section_cleared_completely HOA_EXIT UAV1_MMP_POS_32) 1)
(dist_known HOA_EXIT CRASH_SITE)
(= (dist HOA_EXIT CRASH_SITE) 4.67)
(dist_known HOA_EXIT X_UAV2_3100)
(= (dist HOA_EXIT X_UAV2_3100) 7.35)
(dist_known HOA_EXIT AP_PICKUP)
(= (dist HOA_EXIT AP_PICKUP) 4.76)
(dist_known HOA_EXIT DP_PICKUP)
(= (dist HOA_EXIT DP_PICKUP) 5.55)
(dist_known HOA_EXIT PICKUP_POINT)
(= (dist HOA_EXIT PICKUP_POINT) 5.20)
(dist_known HOA_EXIT HOA_ENTRY)
(= (dist HOA_EXIT HOA_ENTRY) 3.26)
(dist_known HOA_EXIT MiRA_MMP_POS_93)
(= (dist HOA_EXIT MiRA_MMP_POS_93) 6.56)
(dist_known HOA_EXIT FORCE_0)
(= (dist HOA_EXIT FORCE_0) 6.28)
(dist_known HOA_EXIT FORCE_1)
(= (dist HOA_EXIT FORCE_1) 5.05)
(dist_known HOA_EXIT FORCE_2)
(= (dist HOA_EXIT FORCE_2) 4.90)
(dist_known HOA_EXIT FORCE_3)
(= (dist HOA_EXIT FORCE_3) 3.70)
(dist_known HOA_EXIT FORCE_5)
(= (dist HOA_EXIT FORCE_5) 5.76)
(dist_known HOA_EXIT FORCE_6)
(= (dist HOA_EXIT FORCE_6) 1.64)
```

```
(dist_known HOA_EXIT FORCE_7)
(= (dist HOA_EXIT FORCE_7) 7.13)
(dist_known HOA_EXIT FORCE_8)
(= (dist HOA_EXIT FORCE_8) 5.46)
(dist_known MiRA_MMP_POS_93 TANGO_FOE)
(= (dist MiRA_MMP_POS_93 TANGO_FOE) 11.49)
(dist_known MiRA_MMP_POS_93 ZULU_FOE)
(= (dist MiRA_MMP_POS_93 ZULU_FOE) 4.11)
(dist_known MiRA_MMP_POS_93 LIMA_FOE)
(= (dist MiRA_MMP_POS_93 LIMA_FOE) 3.07)
(dist_known MiRA_MMP_POS_93 MIKE_FOE)
(= (dist MiRA_MMP_POS_93 MIKE_FOE) 13.93)
(dist_known MiRA_MMP_POS_93 UAV1_MMP_POS_32)
(= (dist MiRA_MMP_POS_93 UAV1_MMP_POS_32) 4.48)
(dist_known MiRA_MMP_POS_93 CRASH_SITE)
(= (dist MiRA_MMP_POS_93 CRASH_SITE) 2.21)
(dist_known MiRA_MMP_POS_93 X_UAV2_3100)
(= (dist MiRA_MMP_POS_93 X_UAV2_3100) 1.59)
(dist_known MiRA_MMP_POS_93 AP_PICKUP)
(= (dist MiRA_MMP_POS_93 AP_PICKUP) 2.66)
(dist_known MiRA_MMP_POS_93 DP_PICKUP)
(= (dist MiRA_MMP_POS_93 DP_PICKUP) 1.78)
(dist_known MiRA_MMP_POS_93 PICKUP_POINT)
(= (dist MiRA_MMP_POS_93 PICKUP_POINT) 2.21)
(dist_known MiRA_MMP_POS_93 HOA_ENTRY)
(= (dist MiRA_MMP_POS_93 HOA_ENTRY) 9.80)
(dist_known MiRA_MMP_POS_93 HOA_EXIT)
(= (dist MiRA_MMP_POS_93 HOA_EXIT) 6.56)
(dist_known MiRA_MMP_POS_93 FORCE_0)
(= (dist MiRA_MMP_POS_93 FORCE_0) 11.31)
(dist_known MiRA_MMP_POS_93 FORCE_1)
(= (dist MiRA_MMP_POS_93 FORCE_1) 10.41)
(dist_known MiRA_MMP_POS_93 FORCE_2)
(= (dist MiRA_MMP_POS_93 FORCE_2) 10.37)
(dist_known MiRA_MMP_POS_93 FORCE_3)
(= (dist MiRA_MMP_POS_93 FORCE_3) 9.79)
(dist_known MiRA_MMP_POS_93 FORCE_5)
(= (dist MiRA_MMP_POS_93 FORCE_5) 4.23)
(dist_known MiRA_MMP_POS_93 FORCE_6)
(= (dist MiRA_MMP_POS_93 FORCE_6) 5.05)
(dist_known MiRA_MMP_POS_93 FORCE_7)
(= (dist MiRA_MMP_POS_93 FORCE_7) 1.90)
(dist_known MiRA_MMP_POS_93 FORCE_8)
(= (dist MiRA_MMP_POS_93 FORCE_8) 1.11)
(dist_known FORCE_0 TANGO_FOE)
(= (dist FORCE_0 TANGO_FOE) 0.69)
(dist_known FORCE_0 ZULU_FOE)
(= (dist FORCE_0 ZULU_FOE) 8.48)
(dist_known FORCE_0 LIMA_FOE)
(= (dist FORCE_0 LIMA_FOE) 11.81)
(dist_known FORCE_0 MIKE_FOE)
(= (dist FORCE_0 MIKE_FOE) 2.73)
(dist_known FORCE_0 UAV1_MMP_POS_32)
(= (dist FORCE_0 UAV1_MMP_POS_32) 6.83)
(dist_known FORCE_0 CRASH_SITE)
(= (dist FORCE_0 CRASH_SITE) 10.14)
(dist_known FORCE_0 X_UAV2_3100)
(= (dist FORCE_0 X_UAV2_3100) 11.32)
(dist_known FORCE_0 AP_PICKUP)
(= (dist FORCE_0 AP_PICKUP) 10.50)
(dist_known FORCE_0 DP_PICKUP)
(= (dist FORCE_0 DP_PICKUP) 11.02)
```

```
(dist_known FORCE_0 PICKUP_POINT)
(= (dist FORCE_0 PICKUP_POINT) 10.81)
(dist_known FORCE_0 HOA_ENTRY)
(= (dist FORCE_0 HOA_ENTRY) 4.81)
(dist_known FORCE_0 HOA_EXIT)
(= (dist FORCE_0 HOA_EXIT) 6.28)
(dist_known FORCE_0 MiRA_MMP_POS_93)
(= (dist FORCE_0 MiRA_MMP_POS_93) 11.31)
(dist_known FORCE_0 FORCE_1)
(= (dist FORCE_0 FORCE_1) 1.27)
(dist_known FORCE_0 FORCE_2)
(= (dist FORCE_0 FORCE_2) 1.47)
(dist_known FORCE_0 FORCE_3)
(= (dist FORCE_0 FORCE_3) 3.08)
(dist_known FORCE_0 FORCE_5)
(= (dist FORCE_0 FORCE_5) 8.07)
(dist_known FORCE_0 FORCE_6)
(= (dist FORCE_0 FORCE_6) 6.85)
(dist_known FORCE_0 FORCE_7)
(= (dist FORCE_0 FORCE_7) 10.88)
(dist_known FORCE_0 FORCE_8)
(= (dist FORCE_0 FORCE_8) 10.39)
(dist_known FORCE_1 TANGO_FOE)
(= (dist FORCE_1 TANGO_FOE) 1.88)
(dist_known FORCE_1 ZULU_FOE)
(= (dist FORCE_1 ZULU_FOE) 7.92)
(dist_known FORCE_1 LIMA_FOE)
(= (dist FORCE_1 LIMA_FOE) 11.18)
(dist_known FORCE_1 MIKE_FOE)
(= (dist FORCE_1 MIKE_FOE) 3.54)
(dist_known FORCE_1 UAV1_MMP_POS_32)
(= (dist FORCE_1 UAV1_MMP_POS_32) 5.93)
(dist_known FORCE_1 CRASH_SITE)
(= (dist FORCE_1 CRASH_SITE) 9.09)
(dist_known FORCE_1 X_UAV2_3100)
(= (dist FORCE_1 X_UAV2_3100) 10.56)
(dist_known FORCE_1 AP_PICKUP)
(= (dist FORCE_1 AP_PICKUP) 9.41)
(dist_known FORCE_1 DP_PICKUP)
(= (dist FORCE_1 DP_PICKUP) 9.99)
(dist_known FORCE_1 PICKUP_POINT)
(= (dist FORCE_1 PICKUP_POINT) 9.75)
(dist_known FORCE_1 HOA_ENTRY)
(= (dist FORCE_1 HOA_ENTRY) 3.62)
(dist_known FORCE_1 HOA_EXIT)
(= (dist FORCE_1 HOA_EXIT) 5.05)
(dist_known FORCE_1 MiRA_MMP_POS_93)
(= (dist FORCE_1 MiRA_MMP_POS_93) 10.41)
(dist_known FORCE_1 FORCE_0)
(= (dist FORCE_1 FORCE_0) 1.27)
(dist_known FORCE_1 FORCE_2)
(= (dist FORCE_1 FORCE_2) 0.24)
(dist_known FORCE_1 FORCE_3)
(= (dist FORCE_1 FORCE_3) 1.85)
(dist_known FORCE_1 FORCE_5)
(= (dist FORCE_1 FORCE_5) 7.48)
(dist_known FORCE_1 FORCE_6)
(= (dist FORCE_1 FORCE_6) 5.75)
(dist_known FORCE_1 FORCE_7)
(= (dist FORCE_1 FORCE_7) 10.15)
(dist_known FORCE_1 FORCE_8)
(= (dist FORCE_1 FORCE_8) 9.44)
```

```
(dist_known FORCE_2 TANGO_FOE)
(= (dist FORCE_2 TANGO_FOE) 2.11)
(dist_known FORCE_2 ZULU_FOE)
(= (dist FORCE_2 ZULU_FOE) 7.97)
(dist_known FORCE_2 LIMA_FOE)
(= (dist FORCE_2 LIMA_FOE) 11.21)
(dist_known FORCE_2 MIKE_FOE)
(= (dist FORCE_2 MIKE_FOE) 3.60)
(dist_known FORCE_2 UAV1_MMP_POS_32)
(= (dist FORCE_2 UAV1_MMP_POS_32) 5.90)
(dist_known FORCE_2 CRASH_SITE)
(= (dist FORCE_2 CRASH_SITE) 9.01)
(dist_known FORCE_2 X_UAV2_3100)
(= (dist FORCE_2 X_UAV2_3100) 10.56)
(dist_known FORCE_2 AP_PICKUP)
(= (dist FORCE_2 AP_PICKUP) 9.32)
(dist_known FORCE_2 DP_PICKUP)
(= (dist FORCE_2 DP_PICKUP) 9.91)
(dist_known FORCE_2 PICKUP_POINT)
(= (dist FORCE_2 PICKUP_POINT) 9.67)
(dist_known FORCE_2 HOA_ENTRY)
(= (dist FORCE_2 HOA_ENTRY) 3.39)
(dist_known FORCE_2 HOA_EXIT)
(= (dist FORCE_2 HOA_EXIT) 4.90)
(dist_known FORCE_2 MiRA_MMP_POS_93)
(= (dist FORCE_2 MiRA_MMP_POS_93) 10.37)
(dist_known FORCE_2 FORCE_0)
(= (dist FORCE_2 FORCE_0) 1.47)
(dist_known FORCE_2 FORCE_1)
(= (dist FORCE_2 FORCE_1) 0.24)
(dist_known FORCE_2 FORCE_3)
(= (dist FORCE_2 FORCE_3) 1.62)
(dist_known FORCE_2 FORCE_5)
(= (dist FORCE_2 FORCE_5) 7.53)
(dist_known FORCE_2 FORCE_6)
(= (dist FORCE_2 FORCE_6) 5.66)
(dist_known FORCE_2 FORCE_7)
(= (dist FORCE_2 FORCE_7) 10.15)
(dist_known FORCE_2 FORCE_8)
(= (dist FORCE_2 FORCE_8) 9.39)
(dist_known FORCE_3 TANGO_FOE)
(= (dist FORCE_3 TANGO_FOE) 3.72)
(dist_known FORCE_3 ZULU_FOE)
(= (dist FORCE_3 ZULU_FOE) 8.02)
(dist_known FORCE_3 LIMA_FOE)
(= (dist FORCE_3 LIMA_FOE) 11.05)
(dist_known FORCE_3 MIKE_FOE)
(= (dist FORCE_3 MIKE_FOE) 4.72)
(dist_known FORCE_3 UAV1_MMP_POS_32)
(= (dist FORCE_3 UAV1_MMP_POS_32) 5.49)
(dist_known FORCE_3 CRASH_SITE)
(= (dist FORCE_3 CRASH_SITE) 8.18)
(dist_known FORCE_3 X_UAV2_3100)
(= (dist FORCE_3 X_UAV2_3100) 10.21)
(dist_known FORCE_3 AP_PICKUP)
(= (dist FORCE_3 AP_PICKUP) 8.38)
(dist_known FORCE_3 DP_PICKUP)
(= (dist FORCE_3 DP_PICKUP) 9.08)
(dist_known FORCE_3 PICKUP_POINT)
(= (dist FORCE_3 PICKUP_POINT) 8.78)
(dist_known FORCE_3 HOA_ENTRY)
(= (dist FORCE_3 HOA_ENTRY) 1.79)
```

```
(dist_known FORCE_3 HOA_EXIT)
(= (dist FORCE_3 HOA_EXIT) 3.70)
(dist_known FORCE_3 MiRA_MMP_POS_93)
(= (dist FORCE_3 MiRA_MMP_POS_93) 9.79)
(dist_known FORCE_3 FORCE_0)
(= (dist FORCE_3 FORCE_0) 3.08)
(dist_known FORCE_3 FORCE_1)
(= (dist FORCE_3 FORCE_1) 1.85)
(dist_known FORCE_3 FORCE_2)
(= (dist FORCE_3 FORCE_2) 1.62)
(dist_known FORCE_3 FORCE_5)
(= (dist FORCE_3 FORCE_5) 7.57)
(dist_known FORCE_3 FORCE_6)
(= (dist FORCE_3 FORCE_6) 4.81)
(dist_known FORCE_3 FORCE_7)
(= (dist FORCE_3 FORCE_7) 9.87)
(dist_known FORCE_3 FORCE_8)
(= (dist FORCE_3 FORCE_8) 8.74)
(dist_known FORCE_5 TANGO_FOE)
(= (dist FORCE_5 TANGO_FOE) 8.06)
(dist_known FORCE_5 ZULU_FOE)
(= (dist FORCE_5 ZULU_FOE) 0.45)
(dist_known FORCE_5 LIMA_FOE)
(= (dist FORCE_5 LIMA_FOE) 3.75)
(dist_known FORCE_5 MIKE_FOE)
(= (dist FORCE_5 MIKE_FOE) 10.79)
(dist_known FORCE_5 UAV1_MMP_POS_32)
(= (dist FORCE_5 UAV1_MMP_POS_32) 2.62)
(dist_known FORCE_5 CRASH_SITE)
(= (dist FORCE_5 CRASH_SITE) 4.61)
(dist_known FORCE_5 X_UAV2_3100)
(= (dist FORCE_5 X_UAV2_3100) 3.54)
(dist_known FORCE_5 AP_PICKUP)
(= (dist FORCE_5 AP_PICKUP) 5.31)
(dist_known FORCE_5 DP_PICKUP)
(= (dist FORCE_5 DP_PICKUP) 5.08)
(dist_known FORCE_5 PICKUP_POINT)
(= (dist FORCE_5 PICKUP_POINT) 5.22)
(dist_known FORCE_5 HOA_ENTRY)
(= (dist FORCE_5 HOA_ENTRY) 8.33)
(dist_known FORCE_5 HOA_EXIT)
(= (dist FORCE_5 HOA_EXIT) 5.76)
(dist_known FORCE_5 MiRA_MMP_POS_93)
(= (dist FORCE_5 MiRA_MMP_POS_93) 4.23)
(dist_known FORCE_5 FORCE_0)
(= (dist FORCE_5 FORCE_0) 8.07)
(dist_known FORCE_5 FORCE_1)
(= (dist FORCE_5 FORCE_1) 7.48)
(dist_known FORCE_5 FORCE_2)
(= (dist FORCE_5 FORCE_2) 7.53)
(dist_known FORCE_5 FORCE_3)
(= (dist FORCE_5 FORCE_3) 7.57)
(dist_known FORCE_5 FORCE_6)
(= (dist FORCE_5 FORCE_6) 4.25)
(dist_known FORCE_5 FORCE_7)
(= (dist FORCE_5 FORCE_7) 3.02)
(dist_known FORCE_5 FORCE_8)
(= (dist FORCE_5 FORCE_8) 3.89)
(dist_known FORCE_6 TANGO_FOE)
(= (dist FORCE_6 TANGO_FOE) 7.22)
(dist_known FORCE_6 ZULU_FOE)
(= (dist FORCE_6 ZULU_FOE) 4.62)
```

```
(dist_known FORCE_6 LIMA_FOE)
(= (dist FORCE_6 LIMA_FOE) 6.86)
(dist_known FORCE_6 MIKE_FOE)
(= (dist FORCE_6 MIKE_FOE) 9.26)
(dist_known FORCE_6 UAV1_MMP_POS_32)
(= (dist FORCE_6 UAV1_MMP_POS_32) 1.66)
(dist_known FORCE_6 CRASH_SITE)
(= (dist FORCE_6 CRASH_SITE) 3.38)
(dist_known FORCE_6 X_UAV2_3100)
(= (dist FORCE_6 X_UAV2_3100) 5.73)
(dist_known FORCE_6 AP_PICKUP)
(= (dist FORCE_6 AP_PICKUP) 3.66)
(dist_known FORCE_6 DP_PICKUP)
(= (dist FORCE_6 DP_PICKUP) 4.28)
(dist_known FORCE_6 PICKUP_POINT)
(= (dist FORCE_6 PICKUP_POINT) 4.01)
(dist_known FORCE_6 HOA_ENTRY)
(= (dist FORCE_6 HOA_ENTRY) 4.78)
(dist_known FORCE_6 HOA_EXIT)
(= (dist FORCE_6 HOA_EXIT) 1.64)
(dist_known FORCE_6 MiRA_MMP_POS_93)
(= (dist FORCE_6 MiRA_MMP_POS_93) 5.05)
(dist_known FORCE_6 FORCE_0)
(= (dist FORCE_6 FORCE_0) 6.85)
(dist_known FORCE_6 FORCE_1)
(= (dist FORCE_6 FORCE_1) 5.75)
(dist_known FORCE_6 FORCE_2)
(= (dist FORCE_6 FORCE_2) 5.66)
(dist_known FORCE_6 FORCE_3)
(= (dist FORCE_6 FORCE_3) 4.81)
(dist_known FORCE_6 FORCE_5)
(= (dist FORCE_6 FORCE_5) 4.25)
(dist_known FORCE_6 FORCE_7)
(= (dist FORCE_6 FORCE_7) 5.50)
(dist_known FORCE_6 FORCE_8)
(= (dist FORCE_6 FORCE_8) 3.96)
(dist_known FORCE_7 TANGO_FOE)
(= (dist FORCE_7 TANGO_FOE) 10.95)
(dist_known FORCE_7 ZULU_FOE)
(= (dist FORCE_7 ZULU_FOE) 2.74)
(dist_known FORCE_7 LIMA_FOE)
(= (dist FORCE_7 LIMA_FOE) 1.43)
(dist_known FORCE_7 MIKE_FOE)
(= (dist FORCE_7 MIKE_FOE) 13.59)
(dist_known FORCE_7 UAV1_MMP_POS_32)
(= (dist FORCE_7 UAV1_MMP_POS_32) 4.38)
(dist_known FORCE_7 CRASH_SITE)
(= (dist FORCE_7 CRASH_SITE) 3.65)
(dist_known FORCE_7 X_UAV2_3100)
(= (dist FORCE_7 X_UAV2_3100) 0.53)
(dist_known FORCE_7 AP_PICKUP)
(= (dist FORCE_7 AP_PICKUP) 4.27)
(dist_known FORCE_7 DP_PICKUP)
(= (dist FORCE_7 DP_PICKUP) 3.54)
(dist_known FORCE_7 PICKUP_POINT)
(= (dist FORCE_7 PICKUP_POINT) 3.91)
(dist_known FORCE_7 HOA_ENTRY)
(= (dist FORCE_7 HOA_ENTRY) 10.22)
(dist_known FORCE_7 HOA_EXIT)
(= (dist FORCE_7 HOA_EXIT) 7.13)
(dist_known FORCE_7 MiRA_MMP_POS_93)
(= (dist FORCE_7 MiRA_MMP_POS_93) 1.90)
```

```
(dist_known FORCE_7 FORCE_0)
(= (dist FORCE_7 FORCE_0) 10.88)
(dist_known FORCE_7 FORCE_1)
(= (dist FORCE_7 FORCE_1) 10.15)
(dist_known FORCE_7 FORCE_2)
(= (dist FORCE_7 FORCE_2) 10.15)
(dist_known FORCE_7 FORCE_3)
(= (dist FORCE_7 FORCE_3) 9.87)
(dist_known FORCE_7 FORCE_5)
(= (dist FORCE_7 FORCE_5) 3.02)
(dist_known FORCE_7 FORCE_6)
(= (dist FORCE_7 FORCE_6) 5.50)
(dist_known FORCE_7 FORCE_8)
(= (dist FORCE_7 FORCE_8) 2.43)
(dist_known FORCE_8 TANGO_FOE)
(= (dist FORCE_8 TANGO_FOE) 10.61)
(dist_known FORCE_8 ZULU_FOE)
(= (dist FORCE_8 ZULU_FOE) 3.89)
(dist_known FORCE_8 LIMA_FOE)
(= (dist FORCE_8 LIMA_FOE) 3.81)
(dist_known FORCE_8 MIKE_FOE)
(= (dist FORCE_8 MIKE_FOE) 12.97)
(dist_known FORCE_8 UAV1_MMP_POS_32)
(= (dist FORCE_8 UAV1_MMP_POS_32) 3.57)
(dist_known FORCE_8 CRASH_SITE)
(= (dist FORCE_8 CRASH_SITE) 1.23)
(dist_known FORCE_8 X_UAV2_3100)
(= (dist FORCE_8 X_UAV2_3100) 2.33)
(dist_known FORCE_8 AP_PICKUP)
(= (dist FORCE_8 AP_PICKUP) 1.84)
(dist_known FORCE_8 DP_PICKUP)
(= (dist FORCE_8 DP_PICKUP) 1.24)
(dist_known FORCE_8 PICKUP_POINT)
(= (dist FORCE_8 PICKUP_POINT) 1.52)
(dist_known FORCE_8 HOA_ENTRY)
(= (dist FORCE_8 HOA_ENTRY) 8.70)
(dist_known FORCE_8 HOA_EXIT)
(= (dist FORCE_8 HOA_EXIT) 5.46)
(dist_known FORCE_8 MiRA_MMP_POS_93)
(= (dist FORCE_8 MiRA_MMP_POS_93) 1.11)
(dist_known FORCE_8 FORCE_0)
(= (dist FORCE_8 FORCE_0) 10.39)
(dist_known FORCE_8 FORCE_1)
(= (dist FORCE_8 FORCE_1) 9.44)
(dist_known FORCE_8 FORCE_2)
(= (dist FORCE_8 FORCE_2) 9.39)
(dist_known FORCE_8 FORCE_3)
(= (dist FORCE_8 FORCE_3) 8.74)
(dist_known FORCE_8 FORCE_5)
(= (dist FORCE_8 FORCE_5) 3.89)
(dist_known FORCE_8 FORCE_6)
(= (dist FORCE_8 FORCE_6) 3.96)
(dist_known FORCE_8 FORCE_7)
(= (dist FORCE_8 FORCE_7) 2.43)
(dist_known TARGET POI_0)
(= (dist TARGET POI_0) 0.32)
(dist_known TARGET AP_ISAR1)
(= (dist TARGET AP_ISAR1) 1.35)
(dist_known TARGET DP_ISAR1)
(= (dist TARGET DP_ISAR1) 0.69)
(dist_known TARGET ISAR1)
(= (dist TARGET ISAR1) 1.00)
```

```
(= (section_cleared_beginning TARGET ISAR1) 2)
(= (section_cleared_completely TARGET ISAR1) 2)
(dist_known TARGET AP_ISAR2)
(= (dist TARGET AP_ISAR2) 0.53)
(= (section_cleared_beginning TARGET AP_ISAR2) 1)
(= (section_cleared_completely TARGET AP_ISAR2) 1)
(dist_known TARGET DP_ISAR2)
(= (dist TARGET DP_ISAR2) 1.31)
(dist_known TARGET ISAR2)
(= (dist TARGET ISAR2) 1.18)
(dist_known TARGET HOA_ENTRY)
(= (dist TARGET HOA_ENTRY) 1.67)
(dist_known TARGET HOA_EXIT)
(= (dist TARGET HOA_EXIT) 1.69)
(dist_known TARGET FORCE_4)
(= (dist TARGET FORCE_4) 1.00)
(dist_known POI_0 TARGET)
(= (dist POI_0 TARGET) 0.32)
(dist_known POI_0 AP_ISAR1)
(= (dist POI_0 AP_ISAR1) 1.36)
(dist_known POI_0 DP_ISAR1)
(= (dist POI_0 DP_ISAR1) 0.58)
(dist_known POI_0 ISAR1)
(= (dist POI_0 ISAR1) 0.97)
(dist_known POI_0 AP_ISAR2)
(= (dist POI_0 AP_ISAR2) 0.63)
(dist_known POI_0 DP_ISAR2)
(= (dist POI_0 DP_ISAR2) 1.30)
(= (section_cleared_beginning POI_0 DP_ISAR2) 1)
(= (section_cleared_completely POI_0 DP_ISAR2) 1)
(dist_known POI_0 ISAR2)
(= (dist POI_0 ISAR2) 1.35)
(= (section_cleared_beginning POI_0 ISAR2) 1)
(= (section_cleared_completely POI_0 ISAR2) 1)
(dist_known POI_0 HOA_ENTRY)
(= (dist POI_0 HOA_ENTRY) 1.76)
(dist_known POI_0 HOA_EXIT)
(= (dist POI_0 HOA_EXIT) 1.51)
(dist_known POI_0 FORCE_4)
(= (dist POI_0 FORCE_4) 0.97)
(dist_known AP_ISAR1 TARGET)
(= (dist AP_ISAR1 TARGET) 1.35)
(dist_known AP_ISAR1 POI_0)
(= (dist AP_ISAR1 POI_0) 1.36)
(dist_known AP_ISAR1 DP_ISAR1)
(= (dist AP_ISAR1 DP_ISAR1) 0.80)
(dist_known AP_ISAR1 ISAR1)
(= (dist AP_ISAR1 ISAR1) 0.40)
(= (section_cleared_beginning AP_ISAR1 ISAR1) 1)
(= (section_cleared_completely AP_ISAR1 ISAR1) 1)
(dist_known AP_ISAR1 AP_ISAR2)
(= (dist AP_ISAR1 AP_ISAR2) 1.88)
(dist_known AP_ISAR1 DP_ISAR2)
(= (dist AP_ISAR1 DP_ISAR2) 2.64)
(dist_known AP_ISAR1 ISAR2)
(= (dist AP_ISAR1 ISAR2) 2.51)
(dist_known AP_ISAR1 HOA_ENTRY)
(= (dist AP_ISAR1 HOA_ENTRY) 0.52)
(= (section_cleared_beginning AP_ISAR1 HOA_ENTRY) 1)
(= (section_cleared_completely AP_ISAR1 HOA_ENTRY) 1)
(dist_known AP_ISAR1 HOA_EXIT)
(= (dist AP_ISAR1 HOA_EXIT) 2.82)
```

```
(dist_known AP_ISAR1 FORCE_4)
(= (dist AP_ISAR1 FORCE_4) 0.40)
(dist_known DP_ISAR1 TARGET)
(= (dist DP_ISAR1 TARGET) 0.69)
(dist_known DP_ISAR1 POI_0)
(= (dist DP_ISAR1 POI_0) 0.58)
(dist_known DP_ISAR1 AP_ISAR1)
(= (dist DP_ISAR1 AP_ISAR1) 0.80)
(dist_known DP_ISAR1 ISAR1)
(= (dist DP_ISAR1 ISAR1) 0.40)
(dist_known DP_ISAR1 AP_ISAR2)
(= (dist DP_ISAR1 AP_ISAR2) 1.17)
(dist_known DP_ISAR1 DP_ISAR2)
(= (dist DP_ISAR1 DP_ISAR2) 1.88)
(dist_known DP_ISAR1 ISAR2)
(= (dist DP_ISAR1 ISAR2) 1.87)
(dist_known DP_ISAR1 HOA_ENTRY)
(= (dist DP_ISAR1 HOA_ENTRY) 1.26)
(dist_known DP_ISAR1 HOA_EXIT)
(= (dist DP_ISAR1 HOA_EXIT) 2.02)
(dist_known DP_ISAR1 FORCE_4)
(= (dist DP_ISAR1 FORCE_4) 0.41)
(dist_known ISAR1 TARGET)
(= (dist ISAR1 TARGET) 1.00)
(= (section_cleared_beginning ISAR1 TARGET) 2)
(= (section_cleared_completely ISAR1 TARGET) 2)
(dist_known ISAR1 POI_0)
(= (dist ISAR1 POI_0) 0.97)
(dist_known ISAR1 AP_ISAR1)
(= (dist ISAR1 AP_ISAR1) 0.40)
(= (section_cleared_beginning ISAR1 AP_ISAR1) 1)
(= (section_cleared_completely ISAR1 AP_ISAR1) 1)
(dist_known ISAR1 DP_ISAR1)
(= (dist ISAR1 DP_ISAR1) 0.40)
(dist_known ISAR1 AP_ISAR2)
(= (dist ISAR1 AP_ISAR2) 1.51)
(dist_known ISAR1 DP_ISAR2)
(= (dist ISAR1 DP_ISAR2) 2.26)
(dist_known ISAR1 ISAR2)
(= (dist ISAR1 ISAR2) 2.17)
(dist_known ISAR1 HOA_ENTRY)
(= (dist ISAR1 HOA_ENTRY) 0.88)
(= (section_cleared_beginning ISAR1 HOA_ENTRY) 1)
(= (section_cleared_completely ISAR1 HOA_ENTRY) 1)
(dist_known ISAR1 HOA_EXIT)
(= (dist ISAR1 HOA_EXIT) 2.42)
(dist_known ISAR1 FORCE_4)
(= (dist ISAR1 FORCE_4) 0.01)
(dist_known AP_ISAR2 TARGET)
(= (dist AP_ISAR2 TARGET) 0.53)
(= (section_cleared_beginning AP_ISAR2 TARGET) 1)
(= (section_cleared_completely AP_ISAR2 TARGET) 1)
(dist_known AP_ISAR2 POI_0)
(= (dist AP_ISAR2 POI_0) 0.63)
(dist_known AP_ISAR2 AP_ISAR1)
(= (dist AP_ISAR2 AP_ISAR1) 1.88)
(dist_known AP_ISAR2 DP_ISAR1)
(= (dist AP_ISAR2 DP_ISAR1) 1.17)
(dist_known AP_ISAR2 ISAR1)
(= (dist AP_ISAR2 ISAR1) 1.51)
(dist_known AP_ISAR2 DP_ISAR2)
(= (dist AP_ISAR2 DP_ISAR2) 0.80)
```

```
(dist_known AP_ISAR2 ISAR2)
(= (dist AP_ISAR2 ISAR2) 0.72)
(= (section_cleared_beginning AP_ISAR2 ISAR2) 1)
(= (section_cleared_completely AP_ISAR2 ISAR2) 1)
(dist_known AP_ISAR2 HOA_ENTRY)
(= (dist AP_ISAR2 HOA_ENTRY) 2.19)
(dist_known AP_ISAR2 HOA_EXIT)
(= (dist AP_ISAR2 HOA_EXIT) 1.34)
(dist_known AP_ISAR2 FORCE_4)
(= (dist AP_ISAR2 FORCE_4) 1.51)
(dist_known DP_ISAR2 TARGET)
(= (dist DP_ISAR2 TARGET) 1.31)
(dist_known DP_ISAR2 POI_0)
(= (dist DP_ISAR2 POI_0) 1.30)
(= (section_cleared_beginning DP_ISAR2 POI_0) 1)
(= (section_cleared_completely DP_ISAR2 POI_0) 1)
(dist_known DP_ISAR2 AP_ISAR1)
(= (dist DP_ISAR2 AP_ISAR1) 2.64)
(dist_known DP_ISAR2 DP_ISAR1)
(= (dist DP_ISAR2 DP_ISAR1) 1.88)
(dist_known DP_ISAR2 ISAR1)
(= (dist DP_ISAR2 ISAR1) 2.26)
(dist_known DP_ISAR2 AP_ISAR2)
(= (dist DP_ISAR2 AP_ISAR2) 0.80)
(dist_known DP_ISAR2 ISAR2)
(= (dist DP_ISAR2 ISAR2) 0.72)
(dist_known DP_ISAR2 HOA_ENTRY)
(= (dist DP_ISAR2 HOA_ENTRY) 2.98)
(dist_known DP_ISAR2 HOA_EXIT)
(= (dist DP_ISAR2 HOA_EXIT) 0.84)
(= (section_cleared_beginning DP_ISAR2 HOA_EXIT) 1)
(= (section_cleared_completely DP_ISAR2 HOA_EXIT) 1)
(dist_known DP_ISAR2 FORCE_4)
(= (dist DP_ISAR2 FORCE_4) 2.26)
(dist_known ISAR2 TARGET)
(= (dist ISAR2 TARGET) 1.18)
(dist_known ISAR2 POI_0)
(= (dist ISAR2 POI_0) 1.35)
(= (section_cleared_beginning ISAR2 POI_0) 1)
(= (section_cleared_completely ISAR2 POI_0) 1)
(dist_known ISAR2 AP_ISAR1)
(= (dist ISAR2 AP_ISAR1) 2.51)
(dist_known ISAR2 DP_ISAR1)
(= (dist ISAR2 DP_ISAR1) 1.87)
(dist_known ISAR2 ISAR1)
(= (dist ISAR2 ISAR1) 2.17)
(dist_known ISAR2 AP_ISAR2)
(= (dist ISAR2 AP_ISAR2) 0.72)
(= (section_cleared_beginning ISAR2 AP_ISAR2) 1)
(= (section_cleared_completely ISAR2 AP_ISAR2) 1)
(dist_known ISAR2 DP_ISAR2)
(= (dist ISAR2 DP_ISAR2) 0.72)
(dist_known ISAR2 HOA_ENTRY)
(= (dist ISAR2 HOA_ENTRY) 2.73)
(= (section_cleared_beginning ISAR2 HOA_ENTRY) 1)
(= (section_cleared_completely ISAR2 HOA_ENTRY) 1)
(dist_known ISAR2 HOA_EXIT)
(= (dist ISAR2 HOA_EXIT) 1.55)
(dist_known ISAR2 FORCE_4)
(= (dist ISAR2 FORCE_4) 2.17)
(dist_known HOA_ENTRY TARGET)
(= (dist HOA_ENTRY TARGET) 1.67)
```

```
(dist_known HOA_ENTRY POI_0)
(= (dist HOA_ENTRY POI_0) 1.76)
(dist_known HOA_ENTRY AP_ISAR1)
(= (dist HOA_ENTRY AP_ISAR1) 0.52)
(= (section_cleared_beginning HOA_ENTRY AP_ISAR1) 1)
(= (section_cleared_completely HOA_ENTRY AP_ISAR1) 1)
(dist_known HOA_ENTRY DP_ISAR1)
(= (dist HOA_ENTRY DP_ISAR1) 1.26)
(dist_known HOA_ENTRY ISAR1)
(= (dist HOA_ENTRY ISAR1) 0.88)
(= (section_cleared_beginning HOA_ENTRY ISAR1) 1)
(= (section_cleared_completely HOA_ENTRY ISAR1) 1)
(dist_known HOA_ENTRY AP_ISAR2)
(= (dist HOA_ENTRY AP_ISAR2) 2.19)
(dist_known HOA_ENTRY DP_ISAR2)
(= (dist HOA_ENTRY DP_ISAR2) 2.98)
(dist_known HOA_ENTRY ISAR2)
(= (dist HOA_ENTRY ISAR2) 2.73)
(= (section_cleared_beginning HOA_ENTRY ISAR2) 1)
(= (section_cleared_completely HOA_ENTRY ISAR2) 1)
(dist_known HOA_ENTRY HOA_EXIT)
(= (dist HOA_ENTRY HOA_EXIT) 3.26)
(dist_known HOA_ENTRY FORCE_4)
(= (dist HOA_ENTRY FORCE_4) 0.87)
(dist_known HOA_EXIT TARGET)
(= (dist HOA_EXIT TARGET) 1.69)
(dist_known HOA_EXIT POI_0)
(= (dist HOA_EXIT POI_0) 1.51)
(dist_known HOA_EXIT AP_ISAR1)
(= (dist HOA_EXIT AP_ISAR1) 2.82)
(dist_known HOA_EXIT DP_ISAR1)
(= (dist HOA_EXIT DP_ISAR1) 2.02)
(dist_known HOA_EXIT ISAR1)
(= (dist HOA_EXIT ISAR1) 2.42)
(dist_known HOA_EXIT AP_ISAR2)
(= (dist HOA_EXIT AP_ISAR2) 1.34)
(dist_known HOA_EXIT DP_ISAR2)
(= (dist HOA_EXIT DP_ISAR2) 0.84)
(= (section_cleared_beginning HOA_EXIT DP_ISAR2) 1)
(= (section_cleared_completely HOA_EXIT DP_ISAR2) 1)
(dist_known HOA_EXIT ISAR2)
(= (dist HOA_EXIT ISAR2) 1.55)
(dist_known HOA_EXIT HOA_ENTRY)
(= (dist HOA_EXIT HOA_ENTRY) 3.26)
(dist_known HOA_EXIT FORCE_4)
(= (dist HOA_EXIT FORCE_4) 2.43)
(dist_known FORCE_4 TARGET)
(= (dist FORCE_4 TARGET) 1.00)
(dist_known FORCE_4 POI_0)
(= (dist FORCE_4 POI_0) 0.97)
(dist_known FORCE_4 AP_ISAR1)
(= (dist FORCE_4 AP_ISAR1) 0.40)
(dist_known FORCE_4 DP_ISAR1)
(= (dist FORCE_4 DP_ISAR1) 0.41)
(dist_known FORCE_4 ISAR1)
(= (dist FORCE_4 ISAR1) 0.01)
(dist_known FORCE_4 AP_ISAR2)
(= (dist FORCE_4 AP_ISAR2) 1.51)
(dist_known FORCE_4 DP_ISAR2)
(= (dist FORCE_4 DP_ISAR2) 2.26)
(dist_known FORCE_4 ISAR2)
(= (dist FORCE_4 ISAR2) 2.17)
```

```
(dist_known FORCE_4 HOA_ENTRY)
(= (dist FORCE_4 HOA_ENTRY) 0.87)
(dist_known FORCE_4 HOA_EXIT)
(= (dist FORCE_4 HOA_EXIT) 2.43)
(location_cleared TARGET)
(is_pz MOB)
(is_dz MOB)
(is_ap_for EDNY_SE1 MOB)
(is_dp_for EDNY_W2 MOB)
(is_pz FOB)
(is_dz FOB)
(is_ap_for AP_FOB FOB)
(is_dp_for DP_FOB FOB)
(is_pz ISAR1)
(is_dz ISAR1)
(is_ap_for AP_ISAR1 ISAR1)
(is_dp_for DP_ISAR1 ISAR1)
(location_cleared ISAR1)
(is_pz ISAR2)
(is_dz ISAR2)
(is_ap_for AP_ISAR2 ISAR2)
(is_dp_for DP_ISAR2 ISAR2)
(location_cleared ISAR2)
(is_pz PICKUP_POINT)
(is_dz PICKUP_POINT)
(is_ap_for AP_PICKUP PICKUP_POINT)
(is_dp_for DP_PICKUP PICKUP_POINT)
(is_hoa_entry HOA_ENTRY)
(is_hoa_exit HOA_EXIT)
(location_cleared HOA_EXIT)
(is_heli MiRA)
(= (v_frt MiRA) 110)
(= (v_ent MiRA) 90)
(= (c_flight MiRA) 100)
(= (d_ac_landing MiRA) 60)
(= (d_ac_takeoff MiRA) 30)
(ac_pos MiRA MiRA_MMP_POS_93)
(travel_alt MiRA)
(transporting MiRA LEADER_SQUAD_B)
(= (has_done_reccetasks MiRA) 0)
(is_uav UAV1)
(= (v_frt UAV1) 119)
(= (v_ent UAV1) 119)
(= (c_flight UAV1) 5)
(= (d_ac_landing UAV1) 60)
(= (d_ac_takeoff UAV1) 30)
(= (d_recce_area UAV1) 75)
(ac_pos UAV1 UAV1_MMP_POS_93)
(travel_alt UAV1)
(= (has_done_reccetasks UAV1) 9)
(is_uav UAV2)
(= (v_frt UAV2) 119)
(= (v_ent UAV2) 119)
(= (c_flight UAV2) 5)
(= (d_ac_landing UAV2) 60)
(= (d_ac_takeoff UAV2) 30)
(= (d_recce_area UAV2) 75)
(ac_pos UAV2 LIMA_FOE)
(travel_alt UAV2)
(= (has_done_reccetasks UAV2) 3)
(is_uav UAV3)
(= (v_frt UAV3) 119)
```

```
(= (v_ent UAV3) 119)
(= (c_flight UAV3) 5)
(= (d_ac_landing UAV3) 60)
(= (d_ac_takeoff UAV3) 30)
(= (d_recce_area UAV3) 75)
(ac_pos UAV3 ISAR2)
(travel_alt UAV3)
(= (has_done_reccetasks UAV3) 2)
(= (troop_speed LEADER_SQUAD_A) 9.72)
(= (d_unloadtrps LEADER_SQUAD_A) 60)
(= (d_loadtroops LEADER_SQUAD_A) 60)
(= (c_troop_move LEADER_SQUAD_A) 500)
(troop_pos LEADER_SQUAD_A AP_ISAR2)
(= (troop_has_moved LEADER_SQUAD_A) 0)
(troop_can_move LEADER_SQUAD_A)
(= (troop_speed LEADER_SQUAD_B) 9.72)
(= (d_unloadtrps LEADER_SQUAD_B) 60)
(= (d_loadtroops LEADER_SQUAD_B) 60)
(= (c_troop_move LEADER_SQUAD_B) 500)
(= (troop_has_moved LEADER_SQUAD_B) 0)
(troop_can_move LEADER_SQUAD_B)
(= c_obj_surveil 0)
(= d_obj_surveil 300)
(= c_unreccd_route 3)
(= total-cost 0)
(current_task_crossflot UAV2 LIMA_FOE LIMA_FRIEND)
(corr_open MIKE_FRIEND MIKE_FOE MiRA)
(corr_open MIKE_FRIEND MIKE_FOE UAV1)
(corr_open MIKE_FRIEND MIKE_FOE UAV2)
(corr_open MIKE_FRIEND MIKE_FOE UAV3)
(at 1 (not (corr_open MIKE_FRIEND MIKE_FOE MiRA)))
(at 1 (not (corr_open MIKE_FRIEND MIKE_FOE UAV1)))
(at 1 (not (corr_open MIKE_FRIEND MIKE_FOE UAV2)))
(at 1 (not (corr_open MIKE_FRIEND MIKE_FOE UAV3)))
(corr_open LIMA_FOE LIMA_FRIEND MiRA)
(corr_open LIMA_FOE LIMA_FRIEND UAV1)
(corr_open LIMA_FOE LIMA_FRIEND UAV2)
(corr_open LIMA_FOE LIMA_FRIEND UAV3)
(at 798 (not (corr_open LIMA_FOE LIMA_FRIEND MiRA)))
(at 798 (not (corr_open LIMA_FOE LIMA_FRIEND UAV1)))
(at 798 (not (corr_open LIMA_FOE LIMA_FRIEND UAV2)))
(at 798 (not (corr_open LIMA_FOE LIMA_FRIEND UAV3)))
(corr_open ZULU_FOE ZULU_FRIEND MiRA)
(corr_open ZULU_FOE ZULU_FRIEND UAV1)
(corr_open ZULU_FOE ZULU_FRIEND UAV2)
(corr_open ZULU_FOE ZULU_FRIEND UAV3)
(at 798 (not (corr_open ZULU_FOE ZULU_FRIEND MiRA)))
(at 798 (not (corr_open ZULU_FOE ZULU_FRIEND UAV1)))
(at 798 (not (corr_open ZULU_FOE ZULU_FRIEND UAV2)))
(at 798 (not (corr_open ZULU_FOE ZULU_FRIEND UAV3)))
(corr_open TANGO_FRIEND TANGO_FOE MiRA)
(corr_open TANGO_FRIEND TANGO_FOE UAV1)
(corr_open TANGO_FRIEND TANGO_FOE UAV2)
(corr_open TANGO_FRIEND TANGO_FOE UAV3)
(at 1 (not (corr_open TANGO_FRIEND TANGO_FOE MiRA)))
(at 1 (not (corr_open TANGO_FRIEND TANGO_FOE UAV1)))
(at 1 (not (corr_open TANGO_FRIEND TANGO_FOE UAV2)))
(at 1 (not (corr_open TANGO_FRIEND TANGO_FOE UAV3)))
(= (c_unloadtrps ISAR1 MiRA LEADER_SQUAD_A) 15000)
(= (c_unloadtrps ISAR1 UAV1 LEADER_SQUAD_A) 20000)
(= (c_unloadtrps ISAR1 UAV2 LEADER_SQUAD_A) 20000)
(= (c_unloadtrps ISAR1 UAV3 LEADER_SQUAD_A) 20000)
```

```
 (= (c_corridor ZULU_FOE ZULU_FRIEND MiRA) 4000)
 (= (c_corridor ZULU_FOE ZULU_FRIEND UAV1) 4000)
 (= (c_corridor ZULU_FOE ZULU_FRIEND UAV2) 4000)
 (= (c_corridor ZULU_FOE ZULU_FRIEND UAV3) 4000)
 (= (c_unloadtrps MOB MiRA LEADER_SQUAD_B) -5000)
 (= (c_corridor TANGO_FRIEND TANGO_FOE MiRA) -1000)
 (= (c_corridor TANGO_FRIEND TANGO_FOE UAV1) -1000)
 (= (c_corridor TANGO_FRIEND TANGO_FOE UAV2) -1000)
 (= (c_corridor TANGO_FRIEND TANGO_FOE UAV3) -1000)
)

(:goal
 (and
 (unload_constraint MiRA LEADER_SQUAD_B MOB)
 (flot_crossed_by LIMA_FOE LIMA_FRIEND MiRA)
 (not (current_task_crossflot UAV2 LIMA_FOE LIMA_FRIEND))
 (has_approached_via UAV2 EDNY_SE1)
 (has_transited UAV2 LIMA_FRIEND EDNY_SE1)
 (has_approached_via UAV1 EDNY_SE1)
 (troop_pos LEADER_SQUAD_B MOB)
 (landed MiRA)
 (ac_pos MiRA MOB)
 (landed UAV1)
 (ac_pos UAV1 MOB)
 (landed UAV2)
 (ac_pos UAV2 MOB)
 (landed UAV3)
 (ac_pos UAV3 MOB)
 (troop_pos LEADER_SQUAD_A TARGET)
 (> (has_done_reccetasks UAV3) 2)
 (< (troop_has_moved LEADER_SQUAD_B) 2.5)
 )
)

(:metric minimize (total-cost))

)
```

8.3.2 Planning Solution during Mission Execution

Due to the anytime planning feature of the LPG-td planner, multiple solutions are generated sequentially. The solution shown here has the sequence number six. See Table 8-5 for further details.

Table 8-5: Example planning solution before mission start - metadata

File name	plan_Slave93.5.PROBLEM_5.SOL
Input domain file	mumt1.DOMAIN
Input problem file	Slave93.5.PROBLEM
Solution sequence number	5

```
; Version LPG-td-1.0
; Seed 50861681
; Command line: /cygdrive/c/REPOS/lpg/LPG-td-1.0/LPG-td-1.0 -o
C:\REPOS\aquasim\software\projects\mum-t\missionPlanner\domain\mumt1.DOMAIN
-f P:\Slave\Slave93.5.PROBLEM -n 9999 -v off -cputime 2400
; Problem Slave93.5.PROBLEM
; Time 51.23
; Search time 35.13
; Parsing time 0.61
; Mutex time 15.47
; Quality -1556.10


Time 51.23

0.0003:   (FRT_FLIGHT UAV1 UAV1_MMP_POS_93 EDNY_SE1) [180.6050]
0.0005:   (FINISH_CROSS_FLOT UAV2 LIMA_FOE LIMA_FRIEND LIMA_FOE) [46.5882]
46.5890:   (FRT_FLIGHT UAV2 LIMA_FRIEND EDNY_SE1) [413.5462]
180.6060:   (APPROACHTO UAV1 EDNY_SE1 MOB) [75.3277]
255.9339:   (APPLANDING UAV1 MOB) [60.0000]
0.0015:   (ENT_FLIGHT MIRA MIRA_MMP_POS_93 LIMA_FOE) [122.8000]
460.1361:   (APPROACHTO UAV2 EDNY_SE1 MOB) [75.3277]
535.4641:   (APPLANDING UAV2 MOB) [60.0000]
122.8023:   (CROSS_FLOT MIRA LIMA_FOE LIMA_FRIEND) [61.6000]
184.4025:   (FRT_FLIGHT MIRA LIMA_FRIEND EDNY_SE1) [447.3818]
631.7845:   (APPROACHTO MIRA EDNY_SE1 MOB) [99.6000]
731.3848:   (APPLANDING MIRA MOB) [60.0000]
791.3850:   (START_UNLD MIRA) [0.1000]
791.4853:   (UNLOADTRPS MIRA MOB LEADER_SQUAD_B) [60.0000]
0.0037:   (SWITCHCAMM UAV3) [0.1000]
0.1040:   (RECCEROUTE UAV3 ISAR2 HOA_EXIT) [46.8908]
46.9950:   (RECCEROUTE UAV3 HOA_EXIT TANGO_FOE) [204.8067]
251.8020:   (RECCEROUTE UAV3 TANGO_FOE LIMA_FOE) [357.2773]
609.0796:   (CROSS_FLOT UAV3 LIMA_FOE LIMA_FRIEND) [46.5882]
655.6680:   (FRT_FLIGHT UAV3 LIMA_FRIEND EDNY_SE1) [413.5462]
1069.2145:   (APPROACHTO UAV3 EDNY_SE1 MOB) [75.3277]
1144.5424:   (APPLANDING UAV3 MOB) [60.0000]
0.0058:   (TROOPSMOVE LEADER_SQUAD_A AP_ISAR2 TARGET) [196.2963]
```

9 References

Bainbridge, L. (1983). Ironies of automation. *Automatica*, *19*(6), 775-779.

Beavers, G., & Hexmoor, H. (2003). Types and limits of agent autonomy. In *International Workshop on Computational Autonomy* (pp. 95-102). Springer Berlin/Heidelberg.

Bergantz, G. J., Delashaw, J., MacWillie, S., & Woodbury, D. (2002). Manned and unmanned experimentation: Enabling effective objective force operations. *Aircraft Survivability*.

Betsch, T., Funke, J., & Plessner, H. (2011). *Denken - Urteilen, Entscheiden, Problemlösen.* Springer Berlin/Heidelberg.

Billman, L., & Steinberg, M. (2007). Human system performance metrics for evaluation of mixed-initiative heterogeneous autonomous systems. In *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems* (pp. 120-126). ACM.

Billings, C. E. (1996). Human-centered aviation automation: Principles and guidelines (NASA TM 110381). *Moffett Field, CA: NASA Ames Research Center*.

Biundo, S. & Schattenberg, B. (2001). From abstract crisis to concrete relief – a preliminary report on combining state abstraction and HTN planning. *In Proceedings of the 6^th^ European Conference on Planning (ECP-01)* (p. 157-168).

Blum, A. L., & Furst, M. L. (1997). Fast planning through planning graph analysis. *Artificial intelligence*, *90*(1), 281-300.

Bonasso, R. P. (1992). Using parallel program specifications for reactive control of underwater vehicles. *Applied Intelligence*, *2*(3), 201-224.

Bonasso, R. P. (1999). Issues in providing adjustable autonomy in the 3T architecture. In *Proceedings of the AAAI Spring Symposium on Agents with Adjustable Autonomy*.

Bradshaw, J. M., Feltovich, P. J., Jung, H., Kulkarni, S., Taysom, W., & Uszok, A. (2003). Dimensions of adjustable autonomy and mixed-initiative interaction. In *International Workshop on Computational Autonomy* (pp. 17-39). Springer Berlin/Heidelberg.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, *2*(1), 14-23.

De Brun, M. L., Moffitt, V. Z., Franke, J. L., Yiantsios, D., Housten, T., Hughes, A., Fouse, S. & Housten, D. (2008). Mixed-initiative adjustable autonomy for human/unmanned system teaming. In *AUVSI Unmanned Systems North America Conference*.

Burstein, M. H., & McDermott, D. V. (1996). Issues in the development of human-computer mixed-initiative planning. *Advances in Psychology*, *113*, 285-303.

Cantrell, M. (2005). The Pilot Who Wasn't There. *Military Officer*, *2*.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, *4*(1), 55-81

Cohen, R., Allaby, C., Cumbaa, C., Fitzgerald, M., Ho, K., Hui, B., Latulipe, C., Lu, F., Moussa, N., Pooley, D., Qian, A. & Siddiqi, S. (1998). What is initiative? *User Modeling and User-Adapted Interaction*, *8*(3-4), 171-214.

Cummings, M. L., & Mitchell, P. J. (2007). Operator scheduling strategies in supervisory control of multiple UAVs. *Aerospace Science and Technology*, *11*(4), 339-348.

Cummings, M. L., & Morales, D. (2005). UAVs as tactical wingmen: Control methods and pilots' perceptions. *Unmanned Systems*, *23*(1), 25-27.

DARPA. Joint Unmanned Combat Air Systems (J-UCAS) Project Homepage. Retrieved from http://www.darpa.mil/j-ucas/ on 2009-06-15

Despouys, O., & Ingrand, F. F. (2000). Propice-plan: Toward a unified framework for planning and execution. In *Recent Advances in AI Planning* (pp. 278-293). Springer Berlin/Heidelberg.

Domshlak, C., Even-Zur, Z., Golany, Y., Karpas, E., & Nus, Y. (2011). Command and control training centers: Computer generated forces meet classical planning. In *ICAPS Workshop on Scheduling and Planning Applications (SPARK)*.

Donath, D. (2012). *Verhaltensanalyse der Beanspruchung des Operateurs in der Multi-UAV-Führung*. Doctoral dissertation, Universität der Bundeswehr. Munich.

Duncker, K. (1945). On problem-solving. *Psychological monographs*, *58*(5).

Durbin, D. B., & Hicks, J. S. (2009). *AH-64D Apache Longbow aircrew workload assessment for unmanned aerial system (UAS) employment*. U.S. Army Research Lab Report ARL-TR-4707, Aberdeen Proving Ground MD.

Edelkamp, S. (2004). IPC-4 Results. Retrieved from http://www.tzi.de/~edelkamp/ipc-4/

Edelkamp, S., & Hoffmann, J. (2004). PDDL2. 2: The language for the classical part of the 4th international planning competition. *4th International Planning Competition (IPC'04), at ICAPS'04*.

Eisenführ, F., Weber, M., & Langer, T. (2010). *Rational decision making*. Springer Berlin/Heidelberg.

Endsley, M. R. (1988). Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 32, No. 2, pp. 97-101). SAGE Publications.

Endsley, M. R. (1996). Automation and situation awareness. In R. Parasuraman and M. Mouloua (eds.), *Automation and human performance: Theory and applications* (pp. 163-181). Mahwah, NJ: Lawrence Erlbaum Associates

Estlin, T. A., Chien, S. A., & Wang, X. (1997, September). An argument for a hybrid HTN/operator-based approach to planning. In *European Conference on Planning* (pp. 182-194). Springer Berlin/Heidelberg.

Ferguson, G., Hayes, C. & Sullivan, G. (Eds.) (2005). *Proceedings of ICAPS Workshop on Mixed-Initiative Planning and Scheduling*. Monterey, CA.

Ferguson, G., & Allen, J. F. (1998). TRIPS: An integrated intelligent problem-solving assistant. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)* (pp. 567-572). AAAI Press.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, *2*(3-4), 189-208.

Firby, R. J. (1987). An investigation into reactive planning in complex domains. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)* (pp. 202-206). AAAI Press.

Fitter, M. J., & Sime, M. E. (1980). Responsibility and shared decision making. In H. T. Smith, T. R. G. Green (Eds.) *Human interaction with computers*, (pp. 32-60). London: Academic Press.

Fitts, P. M., Viteles, M. S., Barr, N. L., Brimhall, D. R., Finch, G., Gardner, E., Grether, W. F., Kellum, W. E & Stevens, S. S. (1951). *Human engineering for an effective air-navigation and traffic-control system*. Columbus, OH: Ohio State University Research Foundation.

Fox, M., & Long, D. (2003). PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research (JAIR)*, *20*, 61-124.

Franke, J. L., Zaychik, V., Spura, T. M., & Alves, E. E. (2005). Inverting the operator/vehicle ratio: Approaches to next generation UAV command and control. *Proceedings of AUVSI Unmanned Systems North America 2005*.

Franke, J. L., Hughes, A., Jameson, S. M., Clark, J. G., & Szczerba, R. J. (2006). Holistic contingency management for autonomous unmanned systems. *Proceedings of the AUVSI's Unmanned Systems North America 2006*.

Gancet, J., Hattenberger, G., Alami, R., & Lacroix, S. (2005). Task planning and control for a multi-UAV system: architecture and algorithms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)* (pp. 1017-1022). IEEE.

Gantt, H. L. (1910). *Work, wages, and profits: their influence on the cost of living*. Engineering magazine.

Gat, E. (1992). Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)* (pp. 809-815). AAAI Press.

Gat, E. (1998). On three-layer architectures. In *Artificial intelligence and mobile robots* (pp. 195-210). AAAI Press.

Georgeff, M. P., & Lansky, A. L. (1987). Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)* (pp. 677-682). AAAI Press.

Gerevini, A., & Long, D. (2005). Plan constraints and preferences in PDDL3. *The Language of the Fifth International Planning Competition.* Technical Report RT-2005-08-47, Department of Electronics for Automation. Italy: University of Brescia.

Gerevini, A., Saetti, A., Serina, I., & Toninelli, P. (2004). LPG-TD: a fully automated planner for PDDL2. 2 domains. In *In Proc. of the 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04) International Planning Competition abstracts*.

Gerevini, A., Saetti, A., & Vallati, M. (2009). An Automatically Configurable Portfolio-based Planner with Macro-actions: PbP. In *Proceedings of ICAPS-2009.*

Gilhooly, K. J., (1989a). Human and Machine Problem Solving: Toward a Comparative Cognitive Science. In K. J. Gilhooly (Ed.) *Human and Machine Problem Solving* (pp. 1-13). New York: Plenum Press.

Gilhooly, K. J. (1989b). Human and Machine Problem Solving: A Comparative Overview. In K. J. Gilhooly (Ed.) *Human and Machine Problem Solving* (pp. 363-371). New York: Plenum Press.

Goldman, R. P. (2006). Durative Planning in HTNs. In *Proceedings of ICAPS-2006* (pp. 382-385).

Goldman, R. P., Miller, C. A., Wu, P., Funk, H. B., & Meisner, J. (2005). Optimizing to satisfice: Using optimization to guide users. In *In Proceedings of the American Helicopter Society's International Specialists Meeting on Unmanned Aerial Vehicles*.

Gunderson, J. P. & Gunderson L.F. (2004), Intelligence ≠ Autonomy ≠ Capability. In *Performance Metrics for Intelligent Systems (PERMIS)*. Gaithersburg, MD.

Hancock, P. A., & Chignell, M. H. (1987). Adaptive control in human-machine systems. In *Human factors psychology* (pp. 305-345). Amsterdam: North-Holland Publishing Co.

Hancock, P. A., & Scallen, S. F. (1996). The future of function allocation. *Ergonomics in Design: The Quarterly of Human Factors Applications, 4*(4), 24-29.

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology, 52*, 139-183.

Hilburn, B., Jorna, P. G., Byrne, E. A., & Parasuraman, R. (1997). The effect of adaptive air traffic control (ATC) decision aiding on controller mental workload. *Human-automation interaction: Research and practice*, 84-91.

Hilburn, B., Molloy, R., Wong, D. and Parasuraman, R. (1993). Operator versus computer control of adaptive automation. In R. S. Jensen and D. Neumeister (Eds.) *Proceedings of the 7th International Symposium on Aviation Psychology* (pp. 161-166). Columbus, OH: Ohio State University.

Hoc, J. M. (1988). *Cognitive psychology of planning*. Academic Press Professional Inc.

Hoffman, D. (1999). I had a funny feeling in my gut. *Washington Post*, *February 10*, A19.

Hollnagel, E., & Bye, A. (2000). Principles for modelling function allocation. *International Journal of Human-Computer Studies*, *52*(2), 253-265.

Hollnagel, E., & Woods, D. D. (1983). Cognitive systems engineering: New wine in new bottles. *International Journal of Man-machine Studies*, *18*(6), 583-600.

Hollnagel, E., & Woods, D. D. (2005). *Joint cognitive systems: Foundations of cognitive systems engineering*. CRC Press.

Hou, M., & Kobierski, R. D. (2006). *Intelligent Adaptive Interfaces: Summary Report on Design, Development, and Evaluation of Intelligent Adaptive Interfaces for the Control of Multiple UAVs from an Airborne Platform* (DRDC-T-TR-2006-292). Toronto: Defence Research and Development.

Hoyos, C. (1990). Menschliches Handeln in technischen Systemen. In Hoyos, C., Zimolong, B. (Eds.) *Enzyklopädie der Psychologie, D3* (Vol.2, pp. 396-425). Göttingen, Germany: Hogrefe.

Inagaki, T. (2003). Automation and the cost of authority. *International Journal of Industrial Ergonomics*, *31*(3), 169-174.

Jameson, S., Franke, J., Szczerba, R., & Stockdale, S. (2005). Collaborative autonomy for manned/unmanned teams. In *Proceedings of the American Helicopter Society 61th Annual Forum* (Vol. 2, pp. 1673-1682). American Helicopter Society.

Kaelbling, L. P. (1987). An architecture for intelligent reactive systems. In M. Georgeff, A. Lansky (Eds.) *Proceedings of the 1986 Workshop: Reasoning about Actions and Plans* (pp. 395-410). Morgan Kaufmann.

Kaber, D. B. (1997). *The Effect of Level of Automation and Adaptive Automation on Performance in Dynamic Control Environments* (AN RCP-NGITWD-97-01). Amarillo, TX: Amarillo National Resource Center for Plutonium.

Kaber, D. B., & Endsley, M. R. (2004). The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science*, *5*(2), 113-153.

Kaber, D. B., & Riley, J. M. (1999). Adaptive automation of a dynamic control task based on secondary task workload measurement. *International journal of cognitive ergonomics*, *3*(3), 169-187.

Kaber, D. B., Riley, J. M., Tan, K. W., & Endsley, M. R. (2001). On the design of adaptive automation for complex systems. *International Journal of Cognitive Ergonomics*, *5*(1), 37-57.

Kambhampati, S., Knoblock, C. A., & Yang, Q. (1995). Planning as refinement search: A unified framework for evaluating design tradeoffs in partial-order planning. *Artificial Intelligence*, *76*(1), 167-238.

Sperling, B., & Kewley Jr., R. H. (2008). *Cooperative Engagements and Levels of Interoperability (CELI) between Unmanned Aircraft Systems and the AH-64D Longbow* (DSE-TR-0806). West Point, NY: Military Academy West Point Operations Research Center.

Klein, G. (2008). Naturalistic decision making. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 50*(3), 456-460.

Kraay, A. G., Pouliot, M. L., & Wallace, W. J. (1998). Test and Evaluation of the Man-Machine Interface between the Apache Longbow and an Unmanned Aerial Vehicle. In *Proceedings of the American Helicopter Society 54th Annual Forum*. Washington, DC.
Kurzweil, R. (2005). *The Singularity is Near. When Humans Transcend Biology*. London: Penguin Science.

Lewis, M., Wang, J., & Scerri, P. (2006). Teamwork coordination for realistically complex multi robot systems. In *Proceedings of the NATO Symposium on Human Factors of Uninhabited Military Vehicles as Force Multipliers* (pp. 1-12).

Linegang, M. P., Stoner, H. A., Patterson, M. J., Seppelt, B. D., Hoffman, J. D., Crittendon, Z. B., & Lee, J. D. (2006). Human-automation collaboration in dynamic mission planning: A challenge requiring an ecological approach. In *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting* (Vol. 23, pp. 2482-2486). SAGE Publications.

Maddock, I. (2004). The battle for UCAR. Two defense giants are poised to build the unmanned combat armed rotorcraft for Darpa and the US Army. *Vertiflite Fall 2004*, 40-48.

Maiwald, F. (2013). *Maschinelle Beanspruchungsprädiktion zur ressourcengerechten Adaption eines Pilotenassistenzsystems*. Doctoral dissertation, Universität der Bundeswehr. Verlag Dr. Kovac.

Malasky, J., Forest, L. M., Kahn, A. C., & Key, J. R. (2005). Experimental evaluation of human-machine collaborative algorithms in planning for multiple UAVs. In *2005 IEEE International Conference on Systems, Man and Cybernetics* (Vol. 3, pp. 2469-2475). IEEE.

Miller, C. (2005). Delegation Architectures: Playbooks and Policy for Keeping Operators in Charge. In *Proceedings of ICAPS Workshop on Mixed-Initiative Planning and Scheduling*. Monterey, CA.

Miller, C., Funk, H., Wu, P., Goldman, R., Meisner, J., & Chapman, M. (2005). The Playbook™ Approach to Adaptive Automation. In *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting* (Vol. 1, pp. 15-19). SAGE Publications.

Miller, C., Goldman, R., Funk, H., Wu, P., & Pate, B. (2004,). A playbook approach to variable autonomy control: Application for control of multiple, heterogeneous unmanned air vehicles. In *Proceedings of the American Helicopter Society 60th Annual Forum* (pp. 7-10). American Helicopter Society.

Miller, C. A., & Hannen, M. D. (1999). The Rotorcraft Pilot's Associate: design and evaluation of an intelligent user interface for cockpit information management. *Knowledge-Based Systems, 12*(8), 443-456.

Miller, C. A., & Parasuraman, R. (2007). Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *49*(1), 57-75.

Miller, C. A., Shaw, T. H., Hamell, J. D., Emfield, A., Musliner, D. J., De Visser, E., & Parasurman, R. (2011). Delegation to automation: performance and implications in non-optimal situations. In *International Conference on Engineering Psychology and Cognitive Ergonomics* (pp. 322-331). Springer Berlin/Heidelberg.

Moray, N., Inagaki, T., & Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, *6*(1), 44-58.

Myers, K. L., & Morley, D. N. (2003). Policy-based agent directability. In *Agent Autonomy* (pp. 185-209). Springer US.

N.A.T.O. (2010). *Standardisation Aggreement (STANAG) 4586: Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability*, Ed. 2. NATO Standardisation Agency.

Naval Studies Board (2005). *Autonomous Vehicles in Support of Naval Operations*. National Academic Press.

Nau, D., Au, T. C., Ilghami, O., Kuter, U., Wu, D., Yaman, F., ... & Murdock, J. W. (2005). Applications of SHOP and SHOP2. *IEEE Intelligent Systems*, *20*(2), 34-41.

Newell, A., Shaw, J. C., & Simon, H. A. (1959). Report on a general problem solving program. In *Proceedings of the IFIP Congress* (pp. 256-264).

Oates, T., & Cohen, P. (1994). Mixed-Initiative Schedule Maintenance: a First Step Toward Plan Steering. In *ARPA/Rome Laboratory knowledge-based planning and scheduling initiative: workshop proceedings* (pp. 133-143).

Onken, R. & Schulte, A. (2010). *System-ergonomic Design of Cognitive Automation: Dual-Mode Cognitive Design of Vehicle Guidance and Control Work*. Springer Berlin/Heidelberg.

Platts, J. (2006). Autonomous systems design–a human centric paradox. In *Presentation to Humans and Technology Symposium* (Vol. 24). MIT.

Prinzel III, L. J. (2003). *Team-centered perspective for adaptive automation design*. Hampton, VA: Langley Research Center.

Putzer, H., & Onken, R. (2003). COSA–A generic cognitive system architecture based on a cognitive model of human behavior. *Cognition, Technology & Work*, *5*(2), 140-151.

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE transactions on systems, man, and cybernetics*, (13), 257-266.

Rauschert, A. (2013). *Kognitives Assistenzsystem zur Führung unbemannter Luftfahrzeuge in bemannt-unbemannten Flugmissionen*. Doctoral dissertation, Universität der Bundeswehr. Munich.

Rauschert, A., & Schulte, A. (2012). Cognitive and cooperative assistant system for aerial manned-unmanned teaming missions. *NATO Research and Technology Agency, Human Factors and Medicine Panel, Task Group HFM-170 on Supervisory Control of Multiple Uninhabited Systems: Methodology and Enabling Operator Interface Technologies. RTO-TR-HFM*, *170*, 1-16.

Reason, J. (1990). *Human error*. Cambridge University Press.

Reimann, P., & Chi, M. T. (1989). Human expertise. In K. Gilhooly (Ed.) *Human and machine problem solving* (pp. 161-191). New York: Plenum Press.

Reitman, W. R. (1965). Cognition and thought: an information processing approach. New York: Wiley.

Rollings, A., & Adams, E. (2003). On game design. Indianapolis, IN: New Riders Publishing.

Rouse, W. B. (1976). Adaptive allocation of decision making responsibility between supervisor and computer. In T.B. Sheridan & G. Johannsen (Eds.) *Monitoring behavior and supervisory control* (pp. 295-306). New York: Plenum Press.

Rouse, W. B. (1977). Human-computer interaction in multitask situations. *IEEE Transactions on Systems, Man, and Cybernetics*, *7*(5), 384-392.

Rousseau, D. M., Sitkin, S. B., Burt, R. S., & Camerer, C. (1998). Not so different after all: A cross-discipline view of trust. *Academy of management review*, *23*(3), 393-404.

Ruff, H. A., Calhoun, G. L., Draper, M. H., Fontejon, J. V., & Guilfoos, B. J. (2004).

Exploring automation issues in supervisory control of multiple UAVs. In *Proceedings of the 2nd Human Performance, Situation Awareness, and Automation Conference (HPSAA II).*

Ruff, H. A., Narayanan, S., & Draper, M. H. (2002). Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. *Presence: Teleoperators and virtual environments*, *11*(4), 335-351.

Russell, S. J. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall.

Ryan, J. C., Cummings, M. L., Roy, N., Banerjee, A., & Schulte, A. S. (2011). Designing an interactive local and global decision support system for aircraft carrier deck scheduling. In *Proceedings of AIAA Infotech*.

Sanner, S. (2011). *Relational Dynamic Influence Diagram Language (RDDL): Language Description*. Australian National University.

Scerbo, M.W. (1996). Theoretical perspectives on adaptive automation. In R. Parasuraman and M. Mouloua (Eds.) *Automation and human performance: Theory and applications* (pp. 37-63). Mahwah, NJ: Erlbaum.

Schulte, A. (2012). Kognitive und kooperative Automation zur Führung unbemannter Luftfahrzeuge. In *2. Interdisziplinärer Workshop Kognitive Systeme: Mensch, Teams, Systeme und Automaten – Verstehen, Beschreiben und Gestalten Kognitiver (Technischer) Systeme.*

Schulte, A., Donath, D., & Lange, D. S. (2016). Design patterns for human-cognitive agent teaming. In *International Conference on Engineering Psychology and Cognitive Ergonomics* (pp. 231-243). Springer, Cham.

Selman, B., Kautz, H., & Cohen, B. (1993). Local search strategies for satisfiability testing. *Cliques, coloring, and satisfiability: Second DIMACS implementation challenge*, *26*, 521-532.

Shelton, M. (2011). Manned Unmanned Systems Integration: Mission accomplished. Retrieved from https://www.army.mil/article/67838/

Sheridan, T. B. (1992). *Telerobotics, Automation and Human Supervisory Control*. Cambridge, MA: MIT Press.

Sheridan, T. B., & Verplank, W. L. (1978). *Human and computer control of undersea teleoperators*. MIT.

Simon, H. A. (1955). A behavioral model of rational choice. *The quarterly journal of economics* *69*(1), 99-118.

Simon, H. (1957). *Models of Man, Social and Rational: Mathematical Essays on Rational Human Behavior in a Social Setting*, New York: The Free Press.

Simon, H. (1964). On the Concept of Organizational Goal. *Administrative Science Quarterly, 9*(1), 1-22.

Solem, A. R. (1992). Some Applications of Problem-Solving Versus Decision-Making to Management. *Journal of Business and Psychology, 6*(3), 401-411.

Steinhauser, N. B., Pavlas, D., & Hancock, P. A. (2009). Design principles for adaptive automation and aiding. *Ergonomics in Design: The Quarterly of Human Factors Applications*, *17*(2), 6-10.

Strenzke, R., & Schulte, A. (2011a). The MMP: A Mixed-Initiative Mission Planning System for the Multi-Aircraft Domain. In *Proceedings of the Scheduling and Planning Application workshop (SPARK'11)* (pp. 74-81).

Strenzke, R., & Schulte, A. (2011b). Mixed-initiative multi-UAV mission planning by merging human and machine cognitive skills. In *Proceedings of the International Conference on Engineering Psychology and Cognitive Ergonomics* (pp. 608-617). Springer Berlin/Heidelberg.

Strenzke, R. & Schulte, A., (2012a). Design and evaluation of a system for mixed-initiative operation. *Acta Futura, 5,* 83-97.

Strenzke, R. & Schulte, A. (2012b). Human-Automation Cooperation Issues in Manned-Unmanned Teaming. In *Proceedings of the AUVSI's Unmanned Systems North America 2012*, (pp. 851-870).

Strenzke, R., Theißing, N. & Schulte, A. (2012). Mixed-Initiative Procedure Generation and Modification - the Future of Unmanned Military Systems Guidance? In F. Py, D. Musliner (Eds.) *PlanEx 2012 - Proceedings of the Workshop on Planning and Plan Execution for Real-World Systems* (pp. 47-50).

Strenzke, R., Uhrmann, J., Benzler, A., Maiwald, F., Rauschert, A. & Schulte, A. (2011). Managing Cockpit Crew Excess Task Load in Military Manned-Unmanned Teaming Missions by Dual-Mode Cognitive Automation Approaches. In *Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference* (Vol.1, pp. 647-670).

Sussman, G. J. (1975). *A Computer Model of Skill Acquisition*. New York: Elsevier Science.

Tecuci, G., Aha, D., Boicu, M., Cox M.T., Ferguson G. & Tate, A. (Eds.) (2003). *Proceedings of the IJCAI Workshop on Mixed-Initiative Intelligent Systems*. AAAI Press.

Tecuci, G., Boicu, M. & Cox, M.T. (Eds.) (2007). AI. Magazine, Special Issue on Mixed-initiative Assistants. Volume 28(2), AAAI Press.

Tecuci, G., Boicu, M., & Cox, M. T. (2007). Seven aspects of mixed-initiative reasoning: An introduction to this special issue on mixed-initiative assistants. *AI Magazine*, *28*(2), 11-18.

Theißing, N., & Schulte, A. (2016). Designing a support system to mitigate pilot error while minimizing out-of-the-loop-effects. In *International Conference on Engineering Psychology and Cognitive Ergonomics* (pp. 439-451). Springer, Cham.

Uhrmann, J., & Schulte, A. (2012). Concept, Design and Evaluation of Cognitive Task-based UAV Guidance. *International Journal On Advances in Intelligent Systems, 5*(1), 145-158.

Uhrmann, J., Strenzke, R., Rauschert, A., Meitinger, C., & Schulte, A. (2009). Manned-unmanned teaming: Artificial cognition applied to multiple UAV guidance. In *NATO RTO SCI-202 Symposium on Intelligent Uninhabited Vehicle Guidance Systems*.

Uhrmann, J. (2013). *Auftragsbasierte Multi-UAV-Führung aus dem Helikoptercockpit durch kognitive Automation*. Doctoral dissertation, Universität der Bundeswehr. Munich.

Vallati, M. (2013). Efficient planning through automatic configuration and machine learning. *AI Communications*, *26*(3), 319-321.

Veloso, M. M., Mulvehill, A. M., & Cox, M. T. (1997). Rationale-supported mixed-initiative case-based planning. In *AAAI/IAAI* (pp. 1072-1077).

Wallis, P., Ronnquist, R., Jarvis, D., & Lucas, A. (2002). The automated wingman-using JACK intelligent agents for unmanned autonomous vehicles. In *Aerospace Conference Proceedings, 2002. IEEE* (Vol. 5, pp. 5-2615). IEEE.

Walliser, J. C. (2011). *Trust in automated systems the effect of automation level on trust calibration*. Doctoral dissertation, Naval Postgraduate School. Monterey, CA.

Ward, G. & Morris, R. (2005). Introduction to the Psychology of Planning. . In R. Morris & G. Ward (Eds.) *The Cognitive Psychology of Planning* (pp. 1-34). Hove, UK: Psychology Press.

Wiener, E.L. (1989). *Human factors of advanced technology ('glass cockpit') transport aircraft*. Technical report 117528. Moffett Field, CA: NASA Ames Research Center.

Wikipedia. (n.d.). Retrieved from http://en.wikipedia.org/wiki/General_Atomics_MQ-9_Reaper on 2014-08-03

Wohlers, K. & Blohm, C. (2007). Flugführungskonzept der KZO Drohne – Autonomie und Verantwortlichkeiten, In *Workshop der DGLR Fachausschüsse*. Neubiberg, Germany.

Younes, H. L. S. & M. L. Littman, M. L. (2004). *PPDDL 1.0: an extension to PDDL for expressing planning domains with probabilistic effects*. Technical Report CMU-CS-04-167. Pittsburgh, PA: Carnegie Mellon University.