The MMP: A Mixed-Initiative Mission Planning System for the Multi-Aircraft Domain

Ruben Strenzke and Axel Schulte
Universität der Bundeswehr München
Department of Aerospace Engineering, Institute of Flight Systems
Werner-Heisenberg-Weg 39, 85577 Neubiberg, GERMANY
{ruben.strenzke, axel.schulte}@unibw.de

Abstract

The Universität der Bundeswehr München is conducting research in the field of single-operator multi-aircraft guidance. This article describes the Mixed-initiative Mission Planner (MMP) as far as requirements, concept, design and implementation are concerned. The MMP is applied to a time-constrained multi-aircraft in-flight mission planning problem. It works in conjunction with a cognitive assistant system for an Uninhabited Aerial Vehicle (UAV) operator, who generates and modifies multi-aircraft mission plans incrementally. The assistant system is able to evaluate, complete, and generate such plans with the aid of the MMP and to communicate with the operator, which results in a mixed-initiative planning approach. Mixed-initiative planning systems need to be able to evaluate the partial or complete human plans as well as the problem itself. For this reason there are two instances of the MMP in our system design. One is configured as a slave to the human input to assume what the human is planning and the other as a free planner that generates reference plans. The assumed human plan includes temporal information about already planned and future tasks, and it can be compared to the reference plan by the assistant system, allowing it to decide whether and when to take initiative. The MMP prototype implementation consists of a Planning Process Manager that dynamically generates problem descriptions, a freely available PDDL 2.2 compatible planner, and multiple domain descriptions.

Introduction

Uninhabited Aerial Vehicles (UAVs) in use today are typically performing preprogrammed missions that can be manually altered in-flight by a crew of at least two human operators. With the advent of multi-UAV scenarios and Manned-Unmanned Teaming (MUM-T), which stands for the joint operation of manned and unmanned assets, the operator-to-vehicle ratio shall be inverted in future applications. For this reason the Institute of Flight Systems at the Universität der Bundeswehr München is 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. 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, Strenzke, and Schulte 2010). On the other hand, the operator shall be supported in mission planning and UAV tasking by a cognitive assistant system (Donath, Rauschert, and Schulte 2010). In this article we describe the mission planning module, which enables interactive online multi-vehicle mission planning, in which both the human (UAV operator) and the machine (assistant system) can take initiative. Hence, we call it Mixed-initiative Mission Planner (MMP).

Of the many mixed-initiative planning systems that we examined, to our knowledge only (Funk et al. 2005) explicitly deals with online planning, i.e. dynamic changes of the situation or the goals during the human problem-solving process and the associated re-planning under time pressure. Their approach allows the user to delegate tasks on different levels of a task hierarchy, which leaves the planning of details unspecified by the user to the automation. This human-automation integration approach is called *supervisory control* (Miller et al. 2005). In contrast to this, we implemented *cooperative control* with real automation-initiative, i.e. the assistant system can initiate dialogs with the user, which is a step into the direction that (Ferguson and Allen 1998) have chosen. But our approach differs by following the Cooperative Automation and assistant system paradigms of (Onken and Schulte 2010), who proclaim a rather passive and invisible assistant, that only interacts with the human operator on its own initiative and does this solely in case he/she shows suboptimal behavior. The assistant system is otherwise invisible and silent, thereby leaving mission responsibility to the human and keeping him/her in the control and decision loop.

First, this article describes the requirements posed by the MUM-T application problem and our assistant system approach. Then, the concept of the MMP is derived from these requirements. After that, we explain its design and give an overview of the implementation. Also, evaluation approaches and first results are presented, and current problems as well as future work are outlined.

Requirements for the MMP

The UAV operator's workplace is located inside a helicopter cockpit, which is part of a large-scale Manned-Unmanned Teaming simulation (Uhrmann, Strenzke, and Schulte 2010). The MMP shall enable the helicopter crew to accomplish experimental MUM-T missions with manageable workload. In this section, we describe the MUM-T application scenario and the assistant system with respect to the requirements they generate for the MMP.

Helicopter and UAV Mission Application Scenario

In our MUM-T scenario, a manned transport helicopter is supposed to carry troops from a pickup zone to an operation area with two possible drop zones nearby (cf. figure 1). In order to get there, the helicopter has to cross the forward line of own troops (FLOT) by the use of defined corridors within specified time windows. In total, there are about 25 mission-relevant locations in the scenario. Three UAVs are taking over the preceding reconnaissance of the helicopter routes and of the drop zone. The more UAVs perform the reconnaissance of a route, the broader is their sensor footprint, thereby increasing safety for the manned high value asset. Prior to the start of the mission, a mission order is provided, which includes certain constraints, e.g. the preferred drop zone, the preferred ingress and egress corridors, the mandatory pickup zone, the permitted times to land at the pickup and drop zones, the corridor opening times, the takeoff clearance time (earliest mission start), as well as the final destinations of all aircraft and troops. Hence, the automated planning of such a mission makes concurrent actions and temporal planning necessary.

During execution of the mission the operator enters a mission plan into the system stepwise by allocating a series of individual tasks to each of the UAVs via a task-based guidance graphical user interface (GUI) (Strenzke et al. 2011). The generated tasks are chronologically ordered, but do not contain any time tags. The following task types can be given to the UAVs: take off, land, transit, cross FLOT, recce route, recce area, and object surveillance. The semi-autonomous UAVs possess restricted planning capabilities that allow them to insert mandatory tasks that the operator left out. In the course of the mission, ground objects are spotted by the UAVs. These events can lead to the necessity of re-planning the mission (e.g. primary landing site or corridor are threatened, cf. figure 1). Also, after the first mission is accomplished, a second troop transport order is given to the helicopter crew, making re-planning of the egress phase necessary.

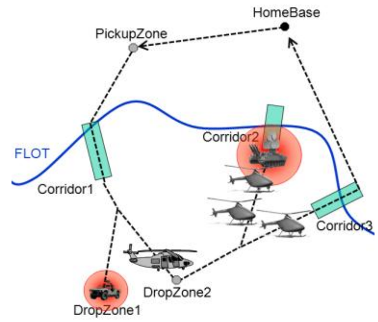

Figure 1: Manned-Unmanned Teaming Mission Scenario

In this simulated scenario we do not regard the movement of any dynamic objects which are not under control of the human (e.g. ground vehicles). The helicopter is also under his/her control (constraint-based guidance, e.g. which corridor and landing site to use) because in our MUM-T scenario the UAV operator and the helicopter commander is the same person. Hence, the mission planning problem is deterministic.

In the helicopter domain, time plays an important role. A mission plan always has to be complete, i.e. including landing in safe territory before fuel runs out. In addition to that, timely coordination is necessary (e.g. site has to be reconnoitered before landing). Therefore, all agents have to be included in the plan. This makes the problem complex. 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, we need optimality criteria, merged together in a cost function. 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).

Manned-Unmanned Teaming Assistant System

The MUM-T assistant system is realized as a knowledge-based system that supports the UAV operator upon detection or anticipation of suboptimal behavior. It mainly holds knowledge about the modes of interaction with the human operator (Donath, Rauschert, and Schulte 2010). In the mentioned cases, the assistant system has three options to aid the operator. It can provide a warning, suggest an action proposal, or initiate an action (e.g. reconfiguration of some system). To communicate with the operator, the assistant system instantiates a dialog or makes an announcement via speech synthesis and the displaying of a message box in the task-based UAV guidance GUI. Whenever appropriate, this message box includes a few buttons that allow the operator to invoke further aid by the assistance system or to either accept or reject its proposals. E.g.

- Assistant takes initiative: *"UAV1 needs follow-up task"*
- Operator presses *"proposal"* button
- Assistant proposes: *"Add task transit HB PZ for UAV1"*
- Operator presses *"accept"* button
- Assistant affirms: *"Added task for UAV1"*

These dialogs can either refer to a single task to be allocated to or executed by a UAV, or to a complete plan to cover the remaining mission goals. Following the Cooperative Automation and assistant system paradigms of (Onken and Schulte 2010) the assistant system initiates these dialogs only if it is found necessary to support the human operator, i.e. he/she made an error (i.e. his/her behavior is below a certain optimality threshold) or an error is anticipated by the assistant system (i.e. his/her plan seems below a certain optimality threshold). To decide whether, when, and how assistance should be provided to the operator

- the assistant system has to be able to anticipate, which tasks the operator has to execute and when he/she is supposed to do this (i.e. *plan is incomplete* and has to be evolved soon due to time constraints), and
- the assistant system has to be able to notice sub-optimality in the past planning performance of the operator (i.e. his/her *plan is too suboptimal*).

Accordingly, during the execution of the mission the assistant system has to check the completeness and optimality of the operator-given UAV tasks upon any operator input that is conflicting with the current plan, any relevant tactical situation change (e.g. new threat enters the scenario), and any mission order change (i.e. new mission objectives received or mission objectives have already been met). The assistant system also needs the ability to propose a new plan to the operator in case his/her plan is infeasible or suboptimal. Taking all this together, the system requires

- the *ability of temporal planning* to complete, generate from scratch, and monitor a task agenda, as well as
- the *ability of plan evaluation* (cost comparison).

Hence, the world model of the MMP has to incorporate the conceptions of time and costs, and it has to be able to perform the necessary calculations in order to provide a basis for assistant system decisions. These decisions have to be made fast, therefore anytime planning is useful.

Concept of the MMP

In order to develop a Mixed-initiative Mission Planner for the MUM-T domain, we first have to analyze the role of the human and of the automation as well as their mixed-initiative interplay. The starting point is that the human shall be able to enter mission plan into the system completely on his own. This is due to our passive assistant system approach and the mentioned responsibility he/she has concerning the mission and the involved systems. This planning task has to be performed by means of a graphical user interface that only allows the incremental generation of a single plan.

Now we suppose that the human has at least one plan in his/her mind and that he/she enters this (the best one or the primary) into the UAV guidance system. Therefore, we can distinguish between the *plan(s) in the human mind* (*human plans – HuP*) and the *plan that is stored in the system* (*system plan – SyP*), i.e. the current active plan for the automatic UAV guidance, that is recognized and modified by the human operator and the assistant system in mixed-initiative fashion. Because it is difficult to get hold of the HuP, the assistant system can only evaluate the SyP.

Plans generated by the assistant system (*assistant system plans – AsP*) constitute a third type of plans in our concept. Figure 2 shows the concept that incorporates all these plan types. Both human and machine shall be able to take initiative in order to manipulate the system plan according to their understanding.

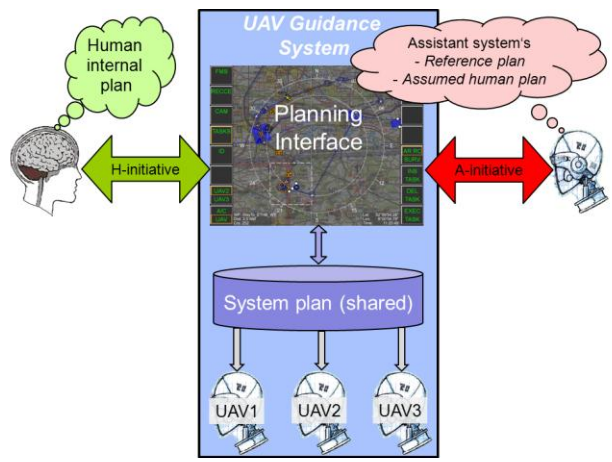

Figure 2: Mixed-initiative planning concept for the MMP

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*). In theory, multiple plans of each subtype can be stored by the assistant system, but this is not regarded in the following. 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 proposals of the assistant system, possibly letting the HuP and SyP converge to either the aHuP or the reference plan of the machine (ReP).

Similar to (Miller et al. 2005) our concept follows a shared task model that allows human and machine to communicate about tasks for the aircraft, goals and plans.

This is explained in more detail in (Strenzke et al. 2011). Furthermore, our MMP concept is based on planning the mission as reaching a defined world state (which is in our case the end state of the mission) and optimizing the way to this state. Hence, partial human plans must be completed in the machine's mind to evaluate them.

Design of the MMP

During the mission, the assistant system receives information about the mission order, the current tactical situation, and the aircraft task agendas out of SyP (cf. figure 3). From this information, the assistant system has to generate the aHuP as well as the ReP. In order to accomplish this, different constraint sets have to be transferred to the MMP. For this purpose the assistant system uses the Simple Temporal Constraint Interface (STCI) as input interface of the MMP. We therefore deploy two instances of the MMP. One is intended to create the aHuP, and the other shall generate the ReP at the same time.

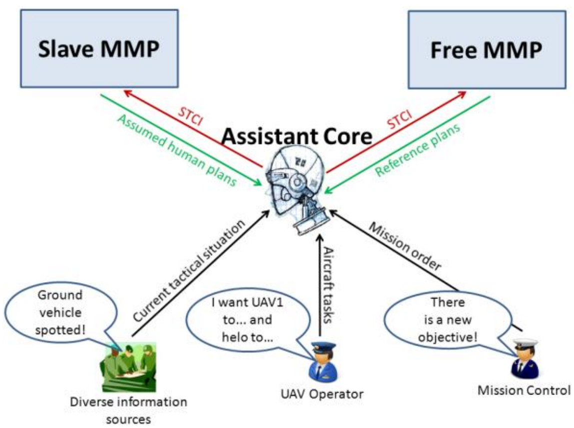

Figure 3: Integration of the assistant core and MMP instances

Slave and Free MMP Instances

Both instances of the MMP receive the information about the current tactical situation, but they differ with respect to the constraints they take into consideration.

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 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 assistant system to monitor if the operator evolves the plan early enough to stay in schedule.

The *Free* instance of the MMP 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 inserted some elements into 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.

Simple Temporal Constraint Interface

The STCI has been developed as an interface for the assistant system core to the MMP to transfer planning constraints. Each constraint refers either to a task to be performed or a state to be reached by an agent (i.e. aircraft or troops). State constraints are "be at ground position", "be at air position" and task constraints include "transit", "unload troops", "load troops", "cross FLOT", "land", "take off", "recce route", "recce area", "object surveillance". To generate multiple (and also open) time windows the temporal specifiers 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 some unspecified point in time or interval. The specification of a closed or half-open interval is possible with the addition of "not before" and/or "not after" constraints. 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). The constraints can be specified as either hard (mandatory) or soft (associated with definable violation costs).

Implementation of the MMP

Although many mission planning systems with symbolic focus used in the aerospace domain are based upon the HTN (Hierarchical Task Network) knowledge-based planning approach, there is a reason for us to prefer a classical operator-based planner. 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 solutions of non-prescribed multi-agent cooperation in a central planning approach seems to be a strong advantage in the complex and dynamic environment of a MUM-T mission, although it poses a heavy burden on solution search performance. This article briefly shows that for the MUM-T scenario described above good solutions can be found sufficiently fast by the MMP in principle. Performance details can be found in (Strenzke and Schulte 2011b).

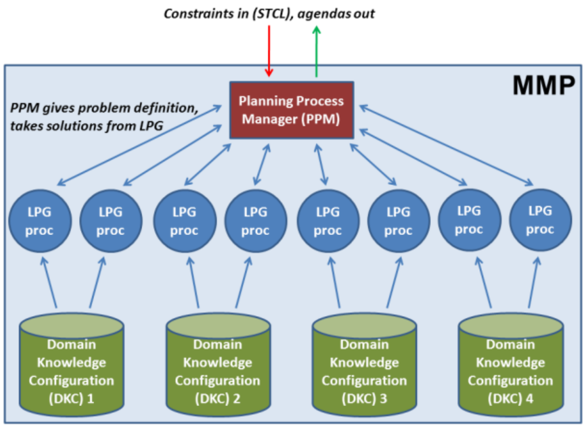

Figure 4: MMP internal structure and functionality

We chose the PDDL (Planning Domain Definition Language) 2.2 (Edelkamp and Hoffmann 2004) representation due to its temporal expressiveness. In our implementation, the 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 PDDL 2.2 problem definition and starts multiple planner processes, which work upon this problem and use different static Domain Knowledge Configurations (DKCs) (see figure 4). The DKCs contains slightly varying MUM-T world models in order to try out different problem-solving heuristics. The generated plans (task agendas) are finally collected by the PPM and provided to the assistant system.

Planning Process Manager

As soon as the PPM receives a planning command via the STCI, it dynamically generates a PDDL problem file containing the complete MUM-T problem with all aircraft. The current tactical situation, which includes all vehicle data (type, position, 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 (i.e. PDDL functions). As mentioned before, the human operator guides the UAVs on a task-based level, i.e. he/she provides tasks to the individual UAVs. Each task can be seen as a declaration of the operator's intent. Hence, these operator-given UAV tasks constitute constraints to the further planning process in addition to the externally given mission order when generating the aHuP. These and the constraints from the mission order are processed as follows.

The conversion of hard temporal constraints into PDDL works with timed initial literals (Edelkamp and Hoffmann 2004) in combination with denying or allowing preconditions for actions defined in the domain. E.g. 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 5). In case of task constraints ("anytime"), a postcondition is defined 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 ("not before"/"not after") constraints with the current implementation.

```
(:goal
 (and
  (landed HELO)
  (ac_pos HELO MOB)          Mission order
  (landed UAV1)               constraints
  (ac_pos UAV1 MOB)
  (landed UAV2)
  (ac_pos UAV2 MOB)
  (landed UAV3)
  (ac_pos UAV3 MOB)
  (troop_pos LEADER SQUAD_A TARGET)
  (location_cleared_by ISAR2 UAV1)
  (section_cleared_by HOA_ENTRY ISAR2 UAV1)
  (flot_crossed_by ZULU_FRIEND ZULU_FOE UAV1)
  …  UAV task constraints / user constraints
 )
)
```

Figure 5: PDDL goals example including constraint sources

Planning Engine

As planning engine we use the PDDL 2.2 compatible LPG-td (*L*ocal Search for *P*lanning *G*raphs – *t*imed initial literals and *d*erived predicates) 1.0 (Gerevini, Saetti, and Serina 2004) due to its full support of temporal planning capabilities defined in PDDL 2.2 and its good performance. Also, due to its local search algorithm, it seems to work well with giving it the SyP as a goal chain that will be followed strictly by the resulting plan (i.e. each next UAV action fulfills one goal in the chain). 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) until there has been provided a new planning request by the assistant system. The different LPG planning processes that are set off by the PPM perform their search each with a distinct initial random seed. Each MMP instance uses 12 LPG processes in the current setup.

Domain Definition

The 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* objects. 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 it is excluded from reconnaissance and surveillance actions. However, it has additional abilities, i.e. the loading and unloading of troops. Some tasks need multiple action models to cover different situations (e.g. “finish departure” as a special case of “departure” when this task is already in progress while starting the planner). This results in 30 different durative actions implemented in total.

The MMP works with multiple PDDL domain configurations in parallel in order to favor different heuristics. E.g. one DKC contains the additional very costly action “land at unrecc’d site”, 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), certain effort is spent on the search for solutions excluding this costly action per se.

All costs can be implemented via functions in PDDL and therefore need not to be part of the domain model but can be generated dynamically in the problem file. The cost model is still being tuned to satisfy test persons’ needs and optimize MMP as well as assistance system performance, therefore we do not present any numbers here.

Evaluation of the MMP

In this chapter we give preliminary evaluation results from the first experimental campaign that includes the MUM-T assistant system and the MMP. Then, some facts about typical problem sizes are provided and an evaluation method for assistant system decisions based on data generated by the MMP is outlined. More detailed data about the subjective and objective MMP evaluation can be found in (Strenzke and Schulte 2011b).

Subjective Evaluation

The test persons were four German Army helicopter pilots acting as UAV operator, each solving two training missions with assistance system aid, then solving two experimental missions without, and then two experimental missions again with assistance system aid. All of the missions were different but following the scheme described above. Subjective questionnaires about the MUM-T assistant system delivered the following results, which show a slightly positive trend, but also improvement potential:

- Slightly better efficiency through automated task insertion by the assistant system (when automated task execution was turned off)
- Proposals to insert tasks lowered workload slightly
- Proposals to insert tasks were considered rather useful
- Proposals to insert tasks seemed rather necessary

Objective Evaluation

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 rounded 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
- 35 timed initial literals
- 50 goal predicates (Slave MMP)
- 75 action steps in the solution
- 20-30 seconds[1] to find a satisfactory plan (viewpoint of developer)

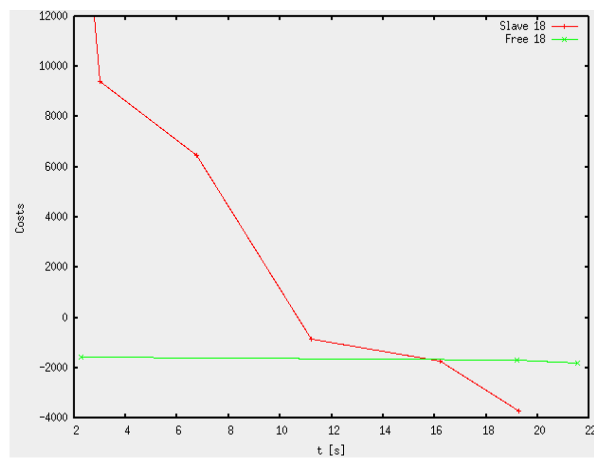

Figure 6: Example of a good aHuP (Slave plan, red)

To analyze situations of false alarms as well as actions missed by the assistant system, it is interesting to compare the plan quality generated over time by the Free and the Slave MMP instances because a future version of the assistant system will perform exactly this comparison in order to decide whether to take initiative to propose not only a

[1] Each MMP instance is running on a high-performance PC with 6 hyper-threading processors (i.e. 12 virtual processors, 1 per LPG process).

single additional UAV task but a completely new mission plan. At the moment this is only done 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 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 in certain mission situations, where re-planning should be proposed by the assistance 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 7). This analysis process is not yet completed.

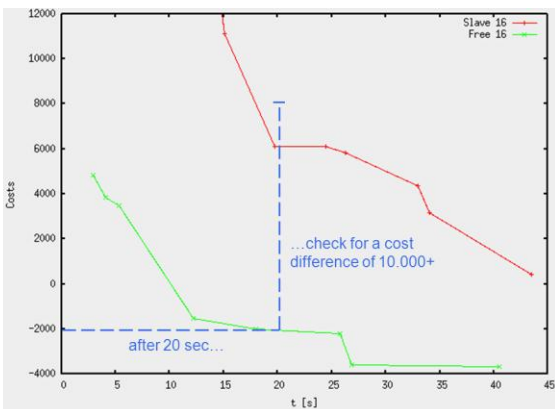

Figure 7: Applying time and cost thresholds to compare quality

A Critical View on the MMP Implementation

Our 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 and Schulte 2011a). However, we see weak points in our 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 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 our current configuration because the Slave planner regards the human input as hard constraints, and therefore all aHuPs 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 v_{max}).

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. Test persons reported that they do not tend so much to plan hierarchically. Instead they liked our forward planning style interface. However, they are indeed used to plan their missions by means of suchlike critical decision points.

Future Work

Figure 8 visualizes two main trends in the development of planners today, which can be seen as the generalization and flexibilization on the one hand (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 performance, which is why Hierarchical Task Network (HTN) planning is used widely in real-world applications (Nau et al. 2005). As depicted in our Sigma-

Delta scheme (figure 8), such domain-configurable planners usually lack the flexibility of classical, domain-independent planning. We explained before, that this was one reason for choosing a domain-independent planner. For planning more complex missions during human-in-the-loop experiments with an acceptable response time, we consider a hybrid approach of classical operator-based and HTN planning, which would be similar to (Estlin, Chien, and Wang 1997; Biundo and Schattenberg 2001; Castillo, Fernández-Olivares, and González 2001), who set a trend towards efficient planning in a dynamic and unforeseeable real world. Further steps that alleviate real-world planning problems, like portfolio-based planning (Gerevini, Saetti, and Vallati 2009) or the situation-dependent assemblage of algorithms and a modular planner (Jameson et al. 2005), will not be taken in the near future.

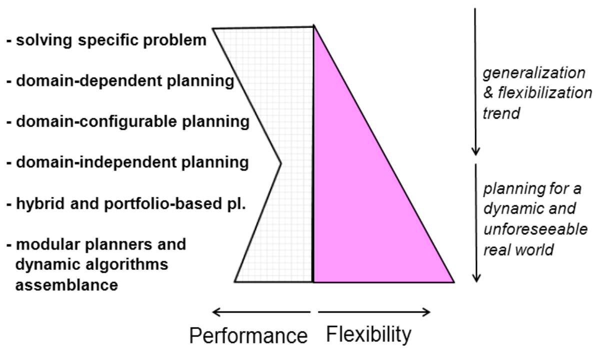

Figure 8: Planning approaches: Trends, performance, flexibility

Our future human-machine interaction research will address brittle (suboptimal) machine solutions (plans and advices) (Strenzke and Schulte 2011b). We will also regard different interaction timing configurations, i.e. the threshold concerning the cost difference between the assumed human plan and the reference plan can be lowered to increase and antedate automation-initiative or raised to make the assistant system intervention less intrusive. Further human-in-the-loop experiments to prove the concept of the MMP and investigate the above mentioned research topics are planned for the near future.

References

Biundo, S., and Schattenberg, B. 2001. From abstract crisis to concrete relief – A preliminary report on combining state abstraction and HTN planning. In *Proceedings of the 6th European Conference on Planning*.

Castillo, L.; Fernández-Olivares, J.; and González, A. 2001. On the adequacy of hierarchical planning characteristics for real-world problem solving. In *Proceedings of the 6th European Conference on Planning*.

Donath, D.; Rauschert, A.; and Schulte, A. 2010. Cognitive assistant system concept for multi-UAV guidance using human operator behaviour models. In *HUMOUS'10*, Toulouse, France.

Edelkamp, S., and Hoffmann, J. 2004. PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition. In *Technical Report 195, Albert-Ludwigs-Universität Freiburg, Institut für Informatik*, Germany.

Estlin, T.A.; Chien, S.A.; and Wang, X. 1997. An argument for a hybrid HTN/operatorbased approach to planning. In *Proceedings of the 4th European Conference on Planning*.

Ferguson, G., and Allen, J. 1998. TRIPS: An Intelligent Integrated Problem-Solving Assistant. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI.

Funk, H.; Goldman, R.; Miller, C.; Meisner, J.; and Wu, P. 2005. A Playbook™ for Real-Time, Closed-Loop Control. In *Proceedings of the First International Conference on Computational Cultural Dynamics*, August 27-28; College Park, MD.

Gerevini, A.; Saetti, A.; and Serina, I. 2004. Planning in PDDL2.2 Domains with LPG-TD. In *Proceedings of ICAPS-04.*

Gerevini, A.; Saetti, A.; and Vallati, M. 2009. An Automatically Configurable Portfolio-based Planner with Macro-actions. In *Proceedings of ICAPS-2009*.

Goldman, R. 2006. Durative planning in HTNs. In *Proceedings of ICAPS-06.*

Jameson, S.; Franke, J.; Szczerba, R.; and Stockdale, S. 2005. Collaborative Autonomy for Manned/Unmanned Teams. In *Proceedings of the American Helicopter Society 61th Annual Forum.*

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's 49th Annual Meeting.* Orlando, FL.

Miller, C.; Goldman, R.; Funk, H.; Wu, P.; and Pate, B. 2004. A Playbook Approach to Variable Autonomy Control: Application for Control of Multiple, Heterogeneous Unmanned Air Vehicles. In *Annual Meeting of the American Helicopter Society*.

Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Wu, D.; Yaman, F.; Munoz-Avila, H.; and Murdock, J.W. 2005. Applications of SHOP and SHOP2. In *Intelligent Systems*, IEEE, Vol. 20 Issue: 2.

Onken, R., and Schulte, A. 2010. *System-ergonomic Design of Cognitive Automation: Dual-Mode Cognitive Design of Vehicle Guidance and Control Work.* Heidelberg, Germany: Springer.

Strenzke, R., and Schulte, A. 2011(a). Mixed-Initiative Multi-UAV Mission Planning by Merging Human and Machine Cognitive Skills. In *8th Conference on Engineering Psychology & Cognitive Ergonomics*, in conjunction with HCI International.

Strenzke, R., and Schulte, A. 2011(b). Design and Evaluation of a Mixed-Initiative Multi-Vehicle Mission Planning System. In *IJCAI-11 Workshop AI in Space: Intelligence beyond Planet Earth*. Barcelona, Spain. 17 July 2011.

Strenzke, R.; Uhrmann, J; Benzler, A.; Maiwald, F.; Rauschert, A.; and Schulte, A. 2011. Managing Cockpit Crew Excess Task Load in Military Manned-Unmanned Teaming Missions by Dual-Mode Cognitive Automation Approaches. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. Portland, Oregon.

Uhrmann, J.; Strenzke, R.; and Schulte, A. 2010. Task-based Guidance of Multiple Detached Unmanned Sensor Platforms in Military Helicopter Operations. In *COGIS*, Crawley, UK.