Automation in Context: Planning Manned Space Exploration Activities

Debra Keirn-Schreckenghost and Mary Beth Hudson

Metrica/TRACLabs
1012 Hercules
Houston, TX 77058
ghost@hypercon.com

Keywords: mission operations automation, human interaction with automation, space activity planning

Abstract

We report progress made on developing a manned space activity planning application and the intelligent software that assists the crew in using this application. Our approach is to integrate existing automated planning and scheduling software with intelligent assistant software that mediates user interaction with these planner and schedulers. This software assists the user in incrementally building a plan and provides for interim plan evaluation and modification. It assists the user in specifying activity preferences, making plan trade-offs at constraint violation and resource contention, and understanding the implications of these choices on the resulting plan. The final planning toolkit will produce planning products such as crew schedules and control plans for life support systems and robotics, will monitor execution of plans, and will assist the user when re-planning due to unforeseen situations.

1 Operation Concept for Activity Planning

For the long duration missions required for space exploration, the crew should have a high level of autonomy from Earth-based operations when planning their activities [1]. This autonomy is needed both to reduce the workload of Earth-based operations and to give the crew control over their daily schedules, which is important for their psychological health. Thus, in future missions we expect that the crew will be responsible to build and maintain their short-term weekly plan, including any contingency plans deemed necessary (such as a bad weather plan for a Martian planetary site). This plan will include crew activities, robot activities, and policies for automated control of life support system (e.g., raise potable water levels), since these activities can potentially affect crew activities. Execution of tasks in the plan will be monitored daily, using feedback from control instrumentation and electronic forms used during tasks. Plans will be available for remote viewing using handheld computers.

Ground operations will receive daily updates on the status of the plan. The information monitored on the ground includes goal completion and trends in environmental conditions. When needed (not expected to be often), new mission goals (e.g., EVA sites, experiments) and new flight rules (e.g., consumables targets, control set points) will be uplinked to the space site for use during planning. Figure 1 summarizes this concept of planning operations.

Automated planning and scheduling software is an important enabler for this concept of operations [2]. It provides for situated selection of tasks to achieve mission goals, reducing crew workload in building a plan. It supports automatic task execution tracking, which assists both the crew and ground controllers in maintaining awareness of task completion. And many of these systems provide for automatic re-planning at task execution failure, based on this task monitoring.

Most automated planning software today, however, has limitations that constrain its utility for crew planning. These limitations include the following:

- Manage one plan at a time

Planning space activities requires building alternative plans for important anticipated contingency situations (e.g., bad weather contingency plan). The current approach to automated planning requires the user to manage the different variations on input that generate different versions of the plan and provides no support for comparing the resulting plans.
• Provide a software engineering interface to the planner and its products

In most cases the specification of goals and situation information that change with each plan are either text files or simple menu-based systems that show predicate lists. Adding new tasks (i.e., new operators) requires software expertise beyond that of a typical space user. Yet for long duration missions, the crew will need to add new tasks during the course of the mission in response to what is learned from ongoing science operations. They also will need to convert the results of automated planning into domain-specific products, such as crew schedules, robot activity plans, and policies for environmental control in the habitat.

• Do not support modeling domain work practice

It is not expected that planning and scheduling software would model the processes used to build and execute plans in a specific domain of application. Yet such work practice models can be used to assist a user in applying this software to a domain problem.

Our approach to enabling these operations concepts is to develop intelligent software that assists the crew in using automated planners and to integrate this intelligent assistant software with an automated planner to build activity plans. We describe our approach below.

2 Space Planning Application

An activity plan for manned space operations has considerable regularity in the daily activities. Blocks of time are allocated to routine activities like eating meals, exercising, etc. These routine activities are performed at the same time every day. As a result, the time periods where mission tasks (e.g., conduct an experiment) can be performed are limited to a three-hour period in the morning and a four-hour period in the afternoon (see Figure 2). We call these time periods, duty blocks. We use hierarchical task network planning to plan the routine daily blocks (with some variations for weekends). We specify the duration of each routine block in the initial situation, which permits easy modification of the template to accommodate changes (e.g., change time-base from 24 hour Earth day to 25 hour Martian day). To plan duty blocks, we have added a simple scheduling capability described below to select goals that should be achievable in a duty block. This selection logic executes each time the planner selects operations for a duty block. Once a duty goal has been selected, we use hierarchical task planning to identify an operation that will achieve this goal, to assign an agent to perform the operation, and to
manage the temporal ordering for any subtasks in the
operation.

<table>
<thead>
<tr>
<th>Weekday Plan Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>morning block</td>
</tr>
<tr>
<td>morning duty block</td>
</tr>
<tr>
<td>meal block</td>
</tr>
<tr>
<td>afternoon duty block</td>
</tr>
<tr>
<td>exercise block</td>
</tr>
<tr>
<td>evening block</td>
</tr>
<tr>
<td>sleep block</td>
</tr>
</tbody>
</table>

Figure 2. Activity Blocks for the Crew Weekday

The planning software we use is the Adversarial Planner (AP), a non-linear state-based hierarchical task net planner. It provides multi-agent capability that ensures agents (crew, robots) are dedicated to a task. It also includes task execution monitoring and automated re-planning at failure to complete tasks. We have used all these features. Additionally, to plan space-based activities, we found it necessary to enhance the capability of this planner with the following features:

- Planning for agent teams
- Managing temporal and resource constraints
- Handling user preferences

Our approach for each of these topics is discussed in the remainder of this section.

2.1 Planning for Agent Teams

We use the multi-agent capability in the automated planner to model individual crew members and robots as agents, as well as groups of these individuals as agents (called teams). Higher-level tasks are allocated to a team agent, and primitive actions within that task are allocated to specific team-member agents (i.e., crew, robots) based on their skills. Explicit modeling of team agents has a couple of advantages for activity planning in space. First, it permits managing the flight rules and constraints associated with group operations like Extra-Vehicular Activities (EVAs). Crew health considerations require that crew members not do EVAs on consecutive days. Risk mitigation considerations require that crew assigned to an EVA be dedicated to that activity until it is complete. Team assignments can be made considering when a team last did an EVA, and members of a team can be blocked from other task assignments while the team is doing an EVA (approximating the concept of having agents "on call" during the EVA). Second, scheduling daily activities as teams synchronizes the schedules of specific individuals, permitting group training and reinforcing team membership. The requirement for team scheduling resulted from collaboration with JSC operations personnel supporting planning activities for the a mission to Mars [3].

2.2 Managing Temporal and Resource Constraints

We have extended the Adversarial Planner to include simple task scheduling compliant with temporal and resource constraints. As part of its programming language, AP supports specification of temporal ordering constraints (e.g., Allen's temporal logic). AP also guarantees that agents (crew, robots) are assigned to only one task at a time. Although necessary, these features are not sufficient to guarantee that agents are only assigned to perform duty tasks during the specific duty periods described above. Our approach to guaranteeing mission tasks are only assigned during duty periods is to identify duty goals for execution at the time AP is planning a duty block. These duty goals are selected from a set of crew-specified objectives included in the initialization state provided to the planner. These goals are selected using the following heuristics:

- Select the goal with the maximum duration that fits within the duty block (3 hours for morning, 4 hours for afternoon)
- Require the predicted power usage of the goal not to exceed the available power budget
- Prefer team goals to individual goals
- Prefer high priority tasks to low priority tasks
- Satisfy user preferences (if enabled)

These heuristics resemble techniques used by schedulers when selecting tasks that meet temporal and resource constraints. Once a duty goal has been selected, the planner identifies an operation that will achieve this goal, assigns an agent to perform the operation, and manages the temporal ordering for any subtasks in the operation. Figure 3 shows an example of an operation in which the duty is selected during planning.

2.3 Handling User Preferences

For space-based activity scheduling, the user (crew or ground-based controller) should have some ability to advise the planner of his preferences [4]. Such capability accommodates constraints not modeled in the planning operations (e.g., Joe prefers to hold meetings in the morning) and has important implications for crew psychological health for long duration missions by giving the crew some control over their daily activities.
We model user preferences as constraints that are checked when preference handling is enabled. We model three categories of user preferences:

- **Temporal**: specify the preferred time when a task should occur; we plan to support concepts as (1) occur no earlier than, (2) occur no later than (deadline), and (3) occur at this time
- **Agent**: specify the preferred agent to perform a task
- **Priority**: distinguish between mandatory goals (i.e., must be performed during the planning period) and desirable goals (i.e., prefer be performed during this period).

In our approach, the user specifies his preferences as part of the initialization state provided to the planner. The algorithm that selects duty goals (see 2.2) verifies that a candidate goal meets these preferences, if preference handling is enabled. Since user preferences constrain the selection of duty goals, the resulting plan that satisfies these preferences may not be optimal along other dimensions (e.g., amount of free time in plan, number of duty goals achieved). As a result, we will procedurally build two variants of a plan - one with user preferences enabled and one without them enabled.

### 3 Human Interaction for Activity Planning

We are developing intelligent asoftware to assist the crew or flight controllers in planning short-term activities (weekly) of the crew while living in a remote habitat (e.g., International Space Station, lunar or planetary site). Our approach is to integrate existing automated planning and scheduling software with intelligent assistant software that mediates user interaction with these planners and schedulers. This software will assist the user in incrementally building a plan and will provide for interim plan evaluation and modification. It will assist the user in specifying activity preferences, making plan trade-offs at constraint violation and resource contention, and understanding the implications of these choices on the resulting plan. The final planning toolkit will produce planning products such as crew schedules and control plans for life support systems and robotics, will monitor execution of plans, and will assist the user when re-planning due to unforeseen situations. In this section we describe the essential features of the intelligent planning assistant software.

#### 3.1 Modeling Planning Work Practice

The intelligent planning assistant models the work practice of building and revising a plan. This model mediates between the software-specific interaction required to control the planner and database and the domain-specific interaction required to specify and evaluate an activity plan. It assists the user in plan specification (1) by requiring the user to identify, review, and approve the initial goals and states, (2) by guiding the user in specifying new duty tasks, (3) by automatically retrieving required initialization information (e.g., loads archived information from database as needed), (4) by tracking user progress through the planning process and reminding the user of tasks not yet complete, and (5) by grouping and storing the planning initialization information and the associated plans for different plan versions. It also guides the user through iterative plan evaluation and modification [5] (1) by providing alternative views of planning output, (2) by computing metrics for each plan that assist comparing plan versions, and (3) by assisting the user in modifying the user preferences, the goal priorities, and the goals in response to these comparisons. Finally it assists the user in producing flight specific planning products, including crew schedules, robot activity plans, and policies for environmental control in the crew habitat. See Figure 4 for an example of an executable checklist for building a space-based activity plan that is automatically constructed using the model of planning practice.

#### 3.2 Managing Multiple Versions of a Plan

Planning space activities requires building alternative plans for important anticipated contingency
situations (e.g., bad weather contingency plan). Plan variations for a given set of activities can also result when evaluating trade-offs in user preferences and task priorities. Each version of a plan corresponds to a unique set of information used to initialize the planner (i.e., the initial state of the environment and the goals to be achieved by the plan). Most automated planners can only reason about one set initialization information and one resulting plan at a time. This requires the user to manage by hand the variations on planning input that generate different versions of the plan.

The intelligent planning assistant assists the user in managing versions of a plan. It indexes each set of planner initialization information to a case identifier, and handles storing this input in a database when the user selects a case for archival. It also retrieves planner initialization information from the database based on these case identifiers, and can command the planner to regenerate the associated plan using this information. Multiple versions of the planning initialization information can be loaded into the intelligent planning assistant at one time, and the user can easily interact with these different versions using a tabbed interface (see Figures 4 and 5). Planner initialization information also can be retrieved as a starting point for building a weekly plan, permitting the user to carry over routine goals and initial states from week to week.

The intelligent planning assistant also will assist the user in comparing versions of a plan. Plans can be compared from two perspectives:

- Differences in the initialization information used to generate the plan
- Differences in the resulting plan generated from this initialization information

Although not yet implemented, we expect to provide the user with support for comparing plans from both these perspectives. We also expect to automatically build at least two plans from each set of initialization information - a plan that tries to satisfy all user preferences and a plan that ignores these preferences. This comparison capability will be useful in evaluating these plan variations.

Dimensions of comparison for the initialization information that we will support include the following:

- Goal Comparison: how are the goals different between cases?
- State Comparison: what information is missing from one case? what information has been added to one case? what information has been changed between cases?

Dimensions of comparison for the generated plan that we will support include the following:

- Goal Achievement: does the plan include tasks to achieve all user goals?
- Compliance with User Preferences: does the plan comply with all user preferences?
- Goal Priorities: are all high priority goals achieved? how many low priority goals are achieved?
- Plan Efficiency: What is the percentage of free time in plan? Free time refers to a period of duty time where no mission activity is scheduled.
- Resource Utilization: What is the power usage profile for the plan?

![Planning Assistant](image)

Figure 4. Checklist for Building a Plan
3.3 User Specification of New Operations

For long duration missions, it is expected that new mission tasks (i.e., new duties) will arise as a result of what is learned during the mission. For example, new laboratory experiments may be needed in reaction to the results of planned science operations. Or new EVAs may be required in response to unexpected events. Typically, adding new tasks requires modifying the planning application software (i.e., writing new operations). We have developed a concept of duty task templates that describe routine duty tasks (e.g., science operations, EVAs, repairs). We characterize these tasks by parameters that will affect plan building. Specifically, we model tasks with the following parameters:

- duration of the task
- required skills for agent performing the task
- power requirements for task
- equipment needed for the task

The user can add a new duty task by specifying values for these parameters in the initial situation loaded into the planner. When trying to plan the new task, the planner will retrieve these parameters and use them in selecting a time to perform the task and in designating which agent should perform the task.

3.4 Monitoring Plan Execution

When planning space activities for long duration missions, it is important for the crew to monitor progress made on executing a plan. They need to understand which tasks are in progress and which tasks have been completed. We use the plan execution monitoring capability of AP to track the execution of the plan. The model of a planning operation includes the expected effects of that operation. When these effects can be instrumented, the flight data distribution system can provide evidence of task completion to AP. When these effects cannot be instrumented, other means of inferring task completion or querying the user for task status must be used. The intelligent planning assistant can monitor for indirect evidence of task completion. The crew can also specify task completion status as a by-product of other tasks. Figure 6 shows a web-based interface where updating a food inventory database after completing a food processing task can also update the task execution status.

The crew needs assistance in monitoring other aspects of the plan being executed. Specifically, the intelligent planning assistant can optionally monitor the upcoming tasks in a plan and can notify the crew when it is near time to execute a task (task reminder). The intelligent planning assistant also can monitor for environmental conditions that indicate a likely need to switch to an alternative plan (e.g., bad weather, power...
failure). We will be investigating different modalities for remote notification that consider the time urgency and mission criticality of the notice, as well as the crew’s current task and location.

Form for Food Inventory Management

Current Status: process_food_op

<table>
<thead>
<tr>
<th>Component</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>normal, dec 31, 2000</td>
</tr>
<tr>
<td>Task</td>
<td>process_food_op</td>
</tr>
<tr>
<td>Performed</td>
<td>JOHN</td>
</tr>
<tr>
<td>Execution Status</td>
<td></td>
</tr>
</tbody>
</table>

Procedure: process_food_op

<table>
<thead>
<tr>
<th>Component</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wheat grain current levels</td>
<td>42.3 kg</td>
</tr>
<tr>
<td>Wheat grain target levels</td>
<td>40.0 kg</td>
</tr>
<tr>
<td>Wheat grain used or processed</td>
<td>40.0 kg</td>
</tr>
<tr>
<td>Soy current levels</td>
<td>19.4 kg</td>
</tr>
<tr>
<td>Soy target levels</td>
<td>15.0 kg</td>
</tr>
<tr>
<td>Soy used or processed</td>
<td>15.0 kg</td>
</tr>
</tbody>
</table>

Figure 6. Marking Tasks Complete

4 Project Status

We have implemented an initial prototype of the activity planning application using the hierarchical task net planner, the Adversarial Planner (AP) developed by MITRE for military planning [6]. The application developed includes the planning operators needed for a representative usage scenario where the crew at a remote space site builds a weekly plan. We also have implemented a prototype of the intelligent planning assistant software using the Reactive Action Package Software (RAPS) [7]. RAPs models the work practice of plan building and evaluation. It constructs dynamically plan-building suggestions for the user that take into account the actions and choices the user has already made. The focus of the first intelligent planning assistant prototype is on supporting the user in building an activity plan. We developed the initial user interface software to this assistant in Tcl/Tk. We integrated this user interface process with the RAPS intelligent assistant software using the IPC client server software built at Carnegie Mellon University (CMU). We have automatically constructed database tables of planning information using the mySQL database. We have developed a web-based interface that uses the PHP scripting language to retrieve plans from this database, and to mark tasks complete in this database. Figure 7 shows a diagram of our software architecture.

We are reporting on a project that is in progress. Essential features of the software implemented to date include the following:

- An executable model of the crew tasks required to build an activity plan using automated planning software. This model includes tasks that interface to the planner and to a database, as well as to the user.
- A user interface that guides the crew through the plan building process and presents information in the context of the space domain.
- Capability to manage sets of input data and output results from the planner for multiple versions of a plan (supporting the development of contingency plans and what-if analysis).
- Support for crew specification of new scientific operations, such as a traverse to a new EVA site or a new laboratory experiment. The basis of this approach is characterizing these operations in terms of the resources required to perform them (e.g., duration of a task, power and equipment requirements, etc.). Planning operator templates use this information to plan these new activities. The user will specify the resource quantities, capabilities and equipment needed, and task durations for a new task.
- An approach for implementing crew specification of preferred activity times (e.g., specifying a
preferred time for exercise that will be satisfied if possible, but will not cause plan building to fail). We also are investigating the use of web-based forms to enable paperless crew operations that integrate plan monitoring into task performance.

5 References


