

Automatic Scheduling and Planning (ASAP)
in Future Ground Control Systems

Sam Matlin
GE Aerospace
Valley Forge, PA, 19406

ABSTRACT

This report describes two complementary approaches to the problem of space mission planning and scheduling. The first is an Expert System or Knowledge Based System for automatically resolving most of the activity conflicts in a candidate plan. The second is an Interactive Graphics Decision Aid to assist the operator in manually resolving the residual conflicts which are beyond the scope of the Expert System. The two system designs are consistent with future ground control station activity requirements, support activity timing constraints, resource limits and activity priority guidelines.

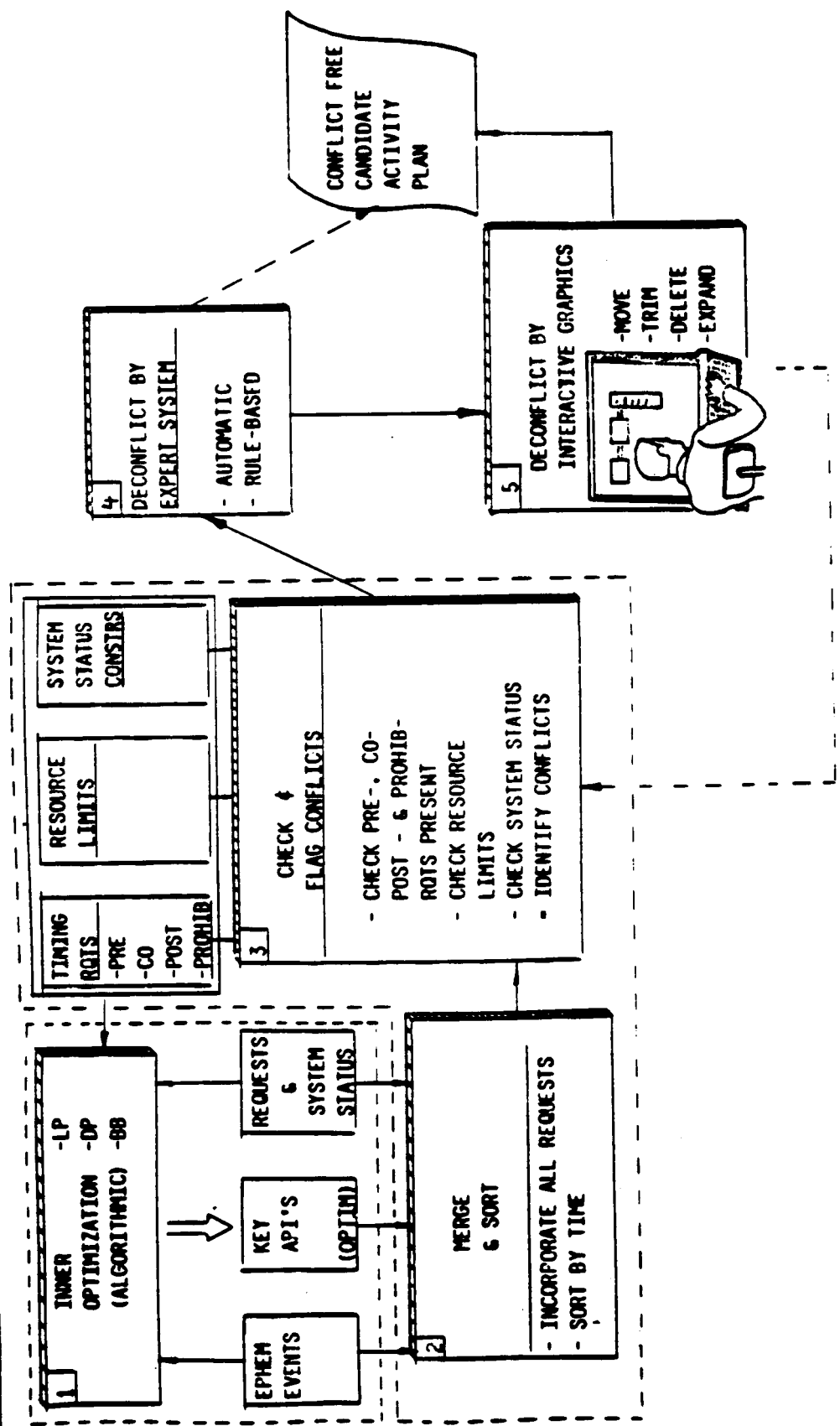
INTRODUCTION

Space mission planning and scheduling is typically performed in a labor-intensive manner, requiring significant numbers of highly skilled personnel, and limited in effectiveness by timeline constraints. By automating these repetitive labor-intensive tasks it will be possible to reduce manpower requirements and provide earlier, more reliable schedules.

Planning and scheduling has been successfully performed at GE by the procedures illustrated in Figure 1. The activities to be scheduled, referred to as Activity Planning Items or APIs, consist of such items as Key Activities (eg, Space Experiments), Special Activities (Calibration, Alignment, Test), Communication Activities (Acquisition), Supporting Activities (Housekeeping, Orbit Adjust), and others. The Key Activities are first scheduled based on a suite of mathematical optimization techniques consisting of Linear Programming, Dynamic Programming, and Branch and Bound. These Key APIs are then merged with other activity requests, and all activities are sorted by start time. Since conflicts may have been introduced by this merging of optimally scheduled activities and ad hoc or late arriving activity requests, conflict criteria (timing requirements, resource limits and system status constraints) are checked and conflicts are flagged. Typically an operator would then manually resolve these conflicts by moving or deleting activities. Instead, it is proposed that the expertise used by the operator be captured in an Expert System (ES), and that most of the conflicts be automatically resolved by the ES. Since not all conflicts are resolvable automatically (too many complex situations would have to be modeled, greatly increasing the cost, size and run time of the ES), it is further proposed to provide a decision aid for the operator in the form of an Interactive Graphics (IG) workstation to assist in resolving the residual hard conflicts.

AUTOMATIC SCHEDULING AND PLANNING

FIGURE 1



APPROACH

The following tasks were undertaken in order to achieve the study objectives of reducing operational costs and timelines:

a) Research Existing Planners. The literature contains dozens of articles on automatic planning and scheduling, including JPL's Devisor, an Artificial Intelligence planner for Voyager missions, and GE/TRW's "Automatic Mission Planning and Scheduling Expert System (AMPASES)". While the literature was not directly applicable to our particular problem, useful elements and techniques were harvested.

b) Determine Application Requirements. Internal documents were reviewed and experts interviewed to ensure that the right problem was being addressed. The task was to generically characterize Key Activities, Supporting Activities, Pre-requisites, Co-requisites, Post-requisites, Prohibited Concurrent Activities, Sequence Constraints, Resource Constraints, and Priority Guidelines.

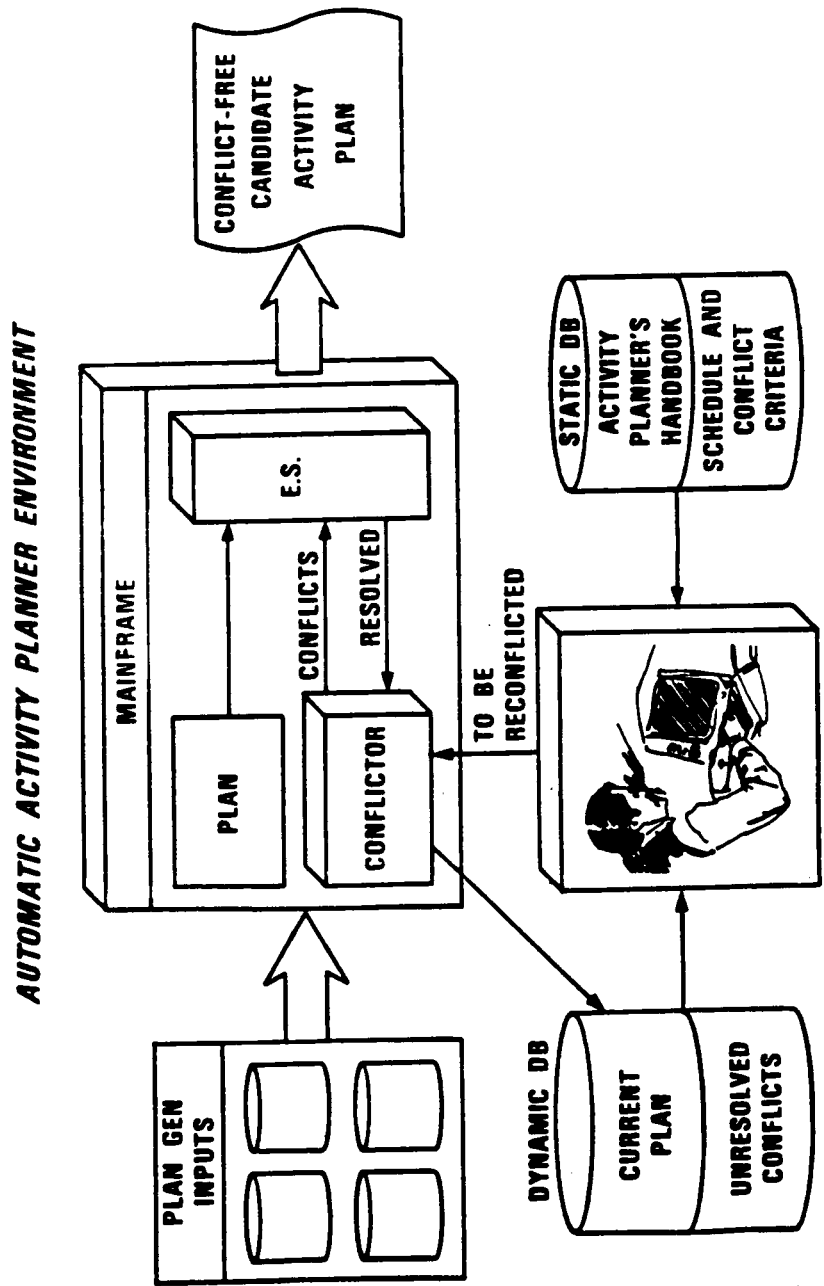
c) Develop Algorithms. Automatic Activity Planning approaches described in the literature include Expert Systems, Tree Searches, 0/1 Programming, Bin Packing, Dynamic Programming, PERT, Network Flow, and others. The effort in this task was to identify the best technique or suite of techniques to use. The conclusion was to develop an Expert System to capture the expertise of current Activity Planners. This ES was then used to remove conflicts generated by the process of accepting all requests, merging them, time-arranging them, and identifying resulting conflicts based on scheduling and conflict criteria.

d) Interactive Graphics. Since it was not feasible to automatically resolve all the conflicts, the approach was to assist the operator with the hard remaining conflicts by providing a computer-based decision aid to facilitate this.

Two prototypes were designed to be configured as shown in Figure 2, based on the above task outputs. Note that after the ES completes its task and the operator completes his, the results of both are reconflicted to ensure that neither the ES nor the operator introduced new conflicts, and the outcome is truly conflict-free.

AUTOMATIC SCHEDULING AND PLANNING

FIGURE 2



RESULTS

The ES prototype was implemented with about 3000 lines of Fortran on an IBM mainframe. The knowledge base consisted of 40 'packed' rules; these are rules containing variables which can be given different values, used in conjunction with the 'Packed Rules Database' which assigns values to these variables. These 40 rules are the equivalent of several hundred ordinary rules but are more compact and more easily maintained. The rules were knowledge-engineered by consulting with several Activity Planners and Operators, based on an initial plan having 117 conflicts. The 40 rules were sufficient to resolve all 117 conflicts; they accomplished this in 1.5 seconds of CPU time, in contrast with an estimated operator time of about 30 minutes. It was anticipated that the ES,, when faced with a new plan it had not seen before, would resolve 50-75% of the conflicts with the same 40 rules. In fact it resolved 93% of the conflicts in a second plan, without introducing any new conflicts.

Figure 3 illustrates a simplified Activity Plan fragment. The conflicts are flagged in the first field, and the corresponding conflict message which the system produces is shown at the bottom. In this case activity SSSS which starts at 07:30:00 is scheduled incorrectly with respect to activity NNNN. The conflict message indicates that the type of conflict is that the required start time of SSSS was scheduled wrong - it was scheduled at 07:30:00, but the schedule criterion was that it had to be scheduled within the window from 07:00:00 to 07:10:00. Figure 4 illustrates a sample rule from the ES knowledge base. The rule states that if a certain pair of Activity Planning Items are in conflict, then move the second relative to the first by a prescribed amount of time. The move is accomplished by deleting and then adding back the offending activity. The rule is generic and applies to many pairs of conflicts. Various instantiations of the rule appear in the packed rules dataset (two are shown in the figure: in the first instance API NNNN and API SSSS play the roles of API1 and API2 in Rule 36; in the second instance these roles are played by APIs V111 and W111). The prescribed amount of time that the second API must be moved to resolve the conflict is also provided in the dataset, corresponding to the particular instantiation involved. In the example the start of API2 (SSSS) must be moved to the start of API1 (NNNN) plus 7 minutes, and the stop 1 second after its start. Note that there are several additional fields available in the Packed Rules Dataset for the inclusion of additional APIs and scheduling constants, to allow for more complicated rules that may involve several APIs in their formulation.

FIGURE 3
AUTOMATIC ACTIVITY PLANNING

INPUT SYSTEM ACTIVITY PLAN

<u>CFLT</u>	<u>TIME</u>	<u>ACTIVITY</u>	<u>ID</u>	<u>START/STOP</u>
	08 FEB 06:50:00	A111	223	START
XX	08 FEB 07:00:00	NNNN	001	START
	08 FEB 07:12:41	R333	020	START
XX	08 FEB 07:30:00	SSSS	001	START
XX	08 FEB 07:30:01	SSSS	001	STOP

CONFLICT MESSAGES

08 FEB 07:00:00 SSSS 001 REQ STRT SCHED WRONG NNNN 07:30:00 07:00:00 07:10:00

Additional features of the Packed Rule Dataset include a priority field (PRI) and a branch code (BR). The priority field allows the user to direct the order in which conflicts are resolved; if a Key Activity is involved in a conflict and several Supporting Activities are in conflict as a result of that, the prioritization allows for the Key Activity conflict to be resolved first, relieving the need to resolve the concomitant supporting activities. The branch code is associated with the type of conflict; this allows modularization of the database so that not all rules need to be searched - only those with the branch code associated with the conflict type. Thus these two fields (PRI and BR) provide a way of efficiently chaining through only that subset of the rules that are of interest for that conflict.

Figure 5 illustrates the output of the ES. The new activity plan shows that API SSSS has been moved to start at 07:07:00, which is within the required window (07:00:00 to 07:10:00) relative to the start of API NNNN. There are no conflicts flagged, and a conflict resolution audit trail message restates the original conflict message and indicates the disposition. Incidentally, no new conflicts were introduced by Rule 36 because it was constructed by experts who knew how to resolve the conflict. However, to be doubly sure, the new activity plan is resubmitted to the conflict identification process used to flag conflicts in the first place. The combination of reconflicting and audit trail gives confidence to the user that the ES has done its job properly.

To summarize the key features of the ES:

design flexibility is achieved by use of packed rules together with the packed rules dataset which makes for an easily maintained and easily extended system;

speed is achieved through the use of branch code and priorities;

confidence is provided by the audit trail and by a final reconflicting.

The Interactive Graphics (IG) decision aid was designed to assist the operator to manually resolve those residual conflicts that were beyond the scope of the ES. It was rapidly prototyped in C on a Sun 3/110 workstation using the Sherrill-Lubinski graphics package. Recommended Human-Machine-Interface procedures were followed throughout. For example, all lines were doubly encoded: first with color, and second with line type (solid, dashed, dotted) to cater to the 10% of American men who are color-blind. Also, all colors are constructed by firing at least 15% of each color gun (Red, Blue, Green) for those operators who may be color-deficient. Early versions of the IG prototype were demonstrated to Activity Planning Operators, and their comments and suggestions were incorporated by fine-tuning the prototype. The IG fulfills the operator needs by providing static data base access (see Figure 2) and ease of use in editing.

AUTOMATIC SCHEDULING AND PLANNING

FIGURE 4

```

SAMPLE RULE
RULE 36

IF API1 IN CONFLICT = API1 IN PACKED RULES AND
API2 IN CONFLICT = API2 IN PACKED RULES
THEN
    DELETE API2 FROM PLAN
    COMPUTE NEW START/STOP TIMES FOR API2 BIASED
    FROM API1
        API2 START = API1 START + DATABASE VALUE 1
        API2 STOP = API2 START + DATABASE VALUE 2
    ADD API2 INTO PLAN AT TIME COMPUTED ABOVE
ENDIF
    
```

PACKED RULES DATASET		API IDENTIFIERS		SCHEDULING CONSTANTS								
PRI	BR	RULE										
1	4	13	Q111	R111	XXXX	XXXX	1800	700	0	0	0	0
2	6	21	T111	U111	XXXX	XXXX	300	1500	90	165	0	0
1	5	36	SSSS	NNNN	XXXX	XXXX	7	1	0	0	0	0
1	5	36	V111	W111	XXXX	XXXX	300	1500	0	0	0	0

FIGURE 5

**AUTOMATIC
SCHEDULING AND PLANNING**

OUTPUT SYSTEM ACTIVITY PLAN

<u>CFLT</u>	<u>TIME</u>	<u>ACTIVITY</u>	<u>ID</u>	<u>START/STOP</u>
08 FEB	06:50:00	A111	223	START
08 FEB	07:00:00	NNNN	001	START
08 FEB	07:07:00	SSSS	001	START
08 FEB	07:07:01	SSSS	001	STOP
08 FEB	07:12:41	R333	020	START

RESOLUTION AUDIT TRAIL

08 FEB 07:00:00 NNNN 001 REQ STRT SCHED WRONG SSSS 07:30:00 07:00:00 07:10:00
 CONFLICT RESOLVED. RULE NUMBER 36 USED. SSSS MOVED TO NNNN START.

CONCLUSIONS

- GE's Activity Planning procedure in which all requests for activities are accepted, merged, sorted by start times and then checked for conflicts using schedule criteria, conflict criteria and resource limits was found to be the most appropriate for the unique problems faced; no other scheme in the literature appeared better.

- It is feasible to expect to resolve most of the conflicts automatically by a simple Expert System

- The Expert System rules require one to four hours each to knowledge engineer, but hundreds rather than thousands of rules are probably adequate.

- The use of 'packed rules' together with a 'packed rule dataset' makes for a highly efficient, easily maintainable implementation.

- Some conflicts require manual intervention; Interactive Graphics can be a valuable aid to the operator.

- Techniques to speed up the Expert System execution time include use of a branch code to segment the rule base and use of rule priorities to eliminate unnecessary resolution of Support Activity conflicts.

ACKNOWLEDGMENTS

A great debt of gratitude is acknowledged for the superlative contributions of Sam Davis, who helped design, knowledge-engineered and programmed the Expert System, and helped design and programmed the Interactive Graphics prototype. Thanks also to Peggy Frederick and Mike Baluta, our planning experts.