N88-29402

A Space Station Onboard Scheduling Assistant

A.F. Brindle and B.H. Anderson
Boeing Aerospace Company
P.O. Box 3999 MS 82-58
Seattle, WA 98124

Abstract

One of the goals for the Space Station is to achieve greater autonomy, and have less reliance on ground commanding than previous space missions. This means that the crew will have to take an active role in scheduling and rescheduling their activities onboard, perhaps working from preliminary schedules generated on the ground. Scheduling is a time-intensive task, whether performed manually or automatically, so the best approach to solving onboard scheduling problems may involve crew members working with an interactive software scheduling package. This report describes a project to investigate such a system, which uses knowledge-based techniques for the rescheduling of experiments within the Materials Technology Laboratory of the Space Station. Particular attention is paid to 1) methods for rapid response rescheduling to accommodate unplanned changes in resource availability, 2) the nature of the interface to the crew, 3) the representation of the many types of data within the knowledge base: crew, resources such as power, experiments, schedules, and constraints, and 4) the possibility of applying rule-based and constraint-based reasoning methods to onboard activity scheduling.

1. Introduction

This paper presents the preliminary design of an Onboard Scheduling Assistant (OSA) for the Space Station. A more detailed description of the issues involved and the existing demonstration system may be found in [1].

The Space Station activity scheduling problem has a number of interesting characteristics. A major feature is that scheduling of crew activities and other autonomous onboard tasks will occur both on the ground and onboard. A full schedule including orbit maneuvers, housekeeping and maintenance tasks, and payload experiments will be developed on the ground and periodically transmitted to the Station, where it may undergo some modifications. This implies that a data format for scheduling will be shared between ground and station; in fact, it will be advantageous for the two to have a consistent knowledge representation scheme, as described below. Therefore, the ground based scheduling problem is described briefly, the onboard rescheduling problem is described, and a constraint based representation is proposed as a suitable data organization approach for both ground and Station scheduling. A description is then given of a prototype onboard rescheduling tool which uses a constraint based knowledge base to implement several limited rescheduling algorithms.

Ground based planning and scheduling will have the goal of a highly optimized, detailed schedule. A large variety of constraints will be involved in the scheduling, such as precedence constraints on activities, hard timing constraints, and resource usage constraints.

Multiple, and scarce, resources will be allocated as part of the scheduling. These resources may be logistics elements, such as laboratory equipment, which are allocated in fixed units and not consumed. They may be consumables, such as liquid fuel, or generated consumables which may be stored, such as electrical power. Re-

ORIGINAL PAGE IS
OF POOR QUALITY

sources may be accompanied by a matrix of attributes, for example, crew members with associated skills. Resources may even be created by execution of tasks within the schedule, as is the case with recycled water. The algorithms used to aid in scheduling are often determined by the nature of the resources involved. For example, a bin packing approach [5] to electrical power load management is not feasible, because with a generated, stored resource, the amount of resource available at any time is not independent of the schedule. The amount of resource available at time t depends upon the amount used by tasks scheduled for a period before t.

The nature of task requirements for various resources also influences the applicability of scheduling algorithms. Individual Space Station activities have resource needs that vary over the duration of the activity, and in some cases, such as power, the resource is so tightly constrained and fully utilized that constant approximations for task resource requirements may be undesirable.

Existing scheduling algorithms can manage some portions of the ground based scheduling. These include algorithms arising from project scheduling (e.g. CPM and Pert [8]), job shop scheduling (see Coffman [2]), and especially project scheduling over multiple resources (see the surveys of Davis [3] and Herroelen [6]). However, this optimizing scheduling would benefit from a broad, flexible knowledge base, which would permit the representation of the diverse constraints and resources, and more heuristic data for a wider variety of scheduling algorithms. It would also allow the multiple scheduling efforts now performed on the ground (using tools such as CAPS [10] and the system of Jaap [7]) to be better coordinated.

Onboard Space Station scheduling will consist of rescheduling in response to changes in the operating environment, changes such as unanticipated reductions in resource availability. Such rescheduling will be deliberately limited in scope, on the assumption that the time and computing resources available for rescheduling onboard will be strictly limited, precluding a full scheduling effort. More important, it is assumed that the scheduling data available onboard will be a subset of that employed during ground based optimizing scheduling. This leads to the conclusion that making gross alterations to the existing schedule onboard would probably create more problems than it would solve, as constraints that are not understood onboard would be violated.

The questions which arise, then, in the creation of an onboard rescheduling tool, are 1) what simple alterations to an existing, highly optimized schedule will best respond to the changing onboard environment without violating presumed constraints on timing, resource availability and precedences, and 2) what types of knowledge must be represented onboard for such rescheduling. To these can be added 3) what simple alterations to the schedule could enhance crew job satisfaction by giving them control over their daily activities (again without violating constraints).

It turns out that minimum perturbation rescheduling in some cases involves the manipulation of constraints and allowable alternatives that have not been represented explicitly up until now. For example, if a task requires multiple resources including a crew member with a particular skills matrix, and that person becomes unavailable, then the substitution of another available crew member with similar skills is desirable. The rescheduling of the activity to another time is not a good option, since that might upset the use of the other resources or violate other constraints.

The constraint based knowledge representation of Fox [4,9] for job shop and project scheduling permits the building of a knowledge base for both ground and Station scheduling. It also provides a vocabulary for the description of the scheduling problem. Tasks, resources, and schedules are objects in the representation. Limitations that define a valid schedule are represented explicitly as constraints, including task requirements for resources. Since a major part of most scheduling efforts involve deciding how to make do when all of the constraints cannot be met, each constraint may be associated with relaxations, which describe alternate, possibly less desirable, constraints for consideration. Each relaxation has an associated utility, or desirability metric, and the constraints themselves may be rated as to which ones should be relaxed first in the search for a reasonable schedule.

The suitability of a constraint based representation for ground scheduling is only hypothesized here. However, the development of the Onboard Scheduling Assistant illustrates its utility for onboard processing of schedules, resource allocations, resource attributes, and resource requirements.

2. The Onboard Scheduling Assistant

The OSA is a demonstration system written in Zetalisp and running on Symbolics machines. It employs a menu driven, graphical interface to allow the user (supposedly a crew member) to view scheduling information along many different lines. Displays include all tasks in the timeline of one schedule (see Figure 1), a task's requirement for a resource plotted over time (Figure 2), and the total use of one resource by one schedule plotted over time (Figure 3). As many as four of these displays may be viewed simultaneously (Figure 4). The user may edit resource availability, reschedule individual tasks in a schedule, request a summary of all points at which any resource has been overallocated by a schedule, and request that new schedules be created by any of several simple, fast rescheduling algorithms. The ability of the crew to amend their own availabilities and to make small movements of tasks in time, as well as the "what if" capability resulting from these features, should increase crew acceptance of the schedule and the scheduling process.

The knowledge base is object oriented, with explicit treatment of constraints. Currently constraints are limited to task requirements for resources or resource attributes. Relaxations on constraints, with their utilities, are permitted in the form of alternative resources, alternative attributes, or requirements for any resource within a set. Resources may be either generated consumables or logistics items, and may have discrete attributes.

Other object types included are schedules, resource allocations within schedules, resource utilization summaries, resource overutilization summaries, task types, meta-task types, and tasks. Meta task definitions allow individual activities, such as steps in an experiment, to be joined into large, goal oriented procedures. These definitions can be hierarchical. A task is an instantiation of task or meta-task type, and thus may have a number of sub-tasks. The subtasks are assumed to be independently schedulable, subject to constraints, but decisions on whether to add or delete an activity are made relative to the entire task only.

Each task has a ground assigned priority, a static number indicating its original importance in the health of the station and the achievement of payload goals. Each also has crew assigned priority, which allows the crew to reassess criticality of tasks, if necessary. The knowledge base has the capability to represent time passing, so that the start and end times of tasks can be compared to the "current" time. These three features are employed in the computation of task priorities during scheduling.

Some of the knowledge in a constraint based scheduling representation, such as notions of state, causality, and revision, are required for full optimizing scheduling, but have been postponed in the implementation of the OSA. Nevertheless, the information available in the OSA permits the following rescheduling approaches:

- Resource Substitution. The summaries of resource overallocation are analyzed to determine which tasks are involved in problem areas. Constraints are not ranked in the system, so it is necessary to decide on an order for examining them. Therefore, the problem tasks are ranked by dynamic priority. This is the weighted sum of terms which reflects the importance of 1) the ground assigned priority, 2) the crew assigned priority, 3) whether or not a task has already begun, and 4) how much a task is contributing to the problem areas as a whole. This last is measured as the proportion of use by the task averaged over all problem areas. It reflects the general goal of keeping as many tasks as possible in the schedule; many other goals, priority heuristics, and methods of measuring terms are possible.

Once the tasks are ranked, processing proceeds from lowest ranking to highest. The constraints (here, resource requirements) of the task are considered in random order. For each requirement for a resource within a problem area, an effort is made to replace the allocation of that resource to the task by satisfying a relaxed requirement which utilizes another, available resource. This is done once for all task requirements which are pertinent to the problem areas, resolving as many problems as possible.

- Task Deletion. All tasks involved in problem areas of resource utilization are ranked by dynamic priority, as described under Resource Substitution. Then the problem areas are processed in random order, and the lowest priority tasks involved in each problem are deleted from the schedule until all problems of overutilization are resolved. This is intended only for tasks that are known by the crew to be involved in very few constraints which are not represented in the system, since deletion of a task can easily lead to violations of precedence constraints.
- Task Insertion. One task not currently scheduled is selected by the user. An attempt is made to schedule the task, without creating any problems in resource usage, and without moving any scheduled tasks in time. This is intended only for tasks that are known by the crew to be involved in very few constraints which are not represented in the system, since addition of a task can easily lead to violations of resource use.

These reschedulers could be combined into a full backtracking scheduler, but it would be far too slow in its exhaustive search of a combinatorially large space. More realistically, all three could be used as routines within ground based scheduling which made extensive use of search-limiting constraints and other heuristics. Onboard, it seems preferable to provide an automatic scheduling option which attempts resource substitution initially and then falls back upon task deletion, but which avoids a more comprehensive search for combinations of relaxations or reassignments of start times.

3. Conclusions

The OSA is a running demonstration which illustrates the viability of constraint based representation and limited heuristic based rescheduling for the onboard Space Station scheduling problem. The investigation into the onboard scheduling environment has emphasized the need for consistency between ground and Station scheduling representations. Furthermore, it has revealed that advantages could be gained for crew satisfaction and adaptive response to environment changes if the polished schedule is transmitted to the Station along with a small amount of the knowledge underlying it, such as resource requirements and relaxations. Additionally, it has provided a mechanism for some experimentation with user interfaces for the display of the very complex, multidimensional body of knowledge that is required for scheduling.

Much more work is needed on the full representation of knowledge for ground scheduling, along with the acquisition and analysis of heuristics for optimizing scheduling as it is currently performed for manned space missions.

References

[1] A.F. Brindle and B.H. Anderson. *A Space Station Onboard Scheduling Assistant Project Report.* Technical Report D180-29768-1, Boeing Aerospace Company, 1986.

[2] E.G. Coffman, editor. *Computer and Job-Shop Scheduling Theory.* J. Wiley and Sons, 1976.

[3] E.W. Davis. Project scheduling under resource constraints – historical review and categorization of procedures. *AIIE Transactions*, 5(4):297–313, 1973.

[4] M.S. Fox. *Constraint-Directed Search: A Case Study of Job-Shop Scheduling.* Technical Report CMU-RI-TR-83-22 (Ph.D. Thesis), Carnegie-Mellon University Robotics Institute, 1983.

[5] M.R. Garey, R.L. Graham, and D.S. Johnson. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory (A)*, 21:257–298, 1976.

[6] W.S. Herroelen. Resource-constrained project scheduling – the state of the art. *Operational Research Quarterly*, 23(3):261–275, 1972.

[7] J. Jaap and E. Davis. Experiment scheduling for spacelab missions. In *The Conference on Artificial Intelligence for Space Applications 1986*, co-published in these proceedings.

[8] J.J. Moder and C.R. Phillips. *Project Management with CPM and Pert.* Van Nostrand Reinhold, New York, NY, second edition, 1970.

[9] A. Sathi and M.S. Fox. Representation of activity knowledge for project management. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(5):531–552, 1985.

[10] K.L. Scott. *Crew Activity Planning System II Users Guide.* McDonnell Douglas Astronautics Company, for NASA Johnson Space Center, February 1986.

ORIGINAL PAGE IS
OF POOR QUALITY

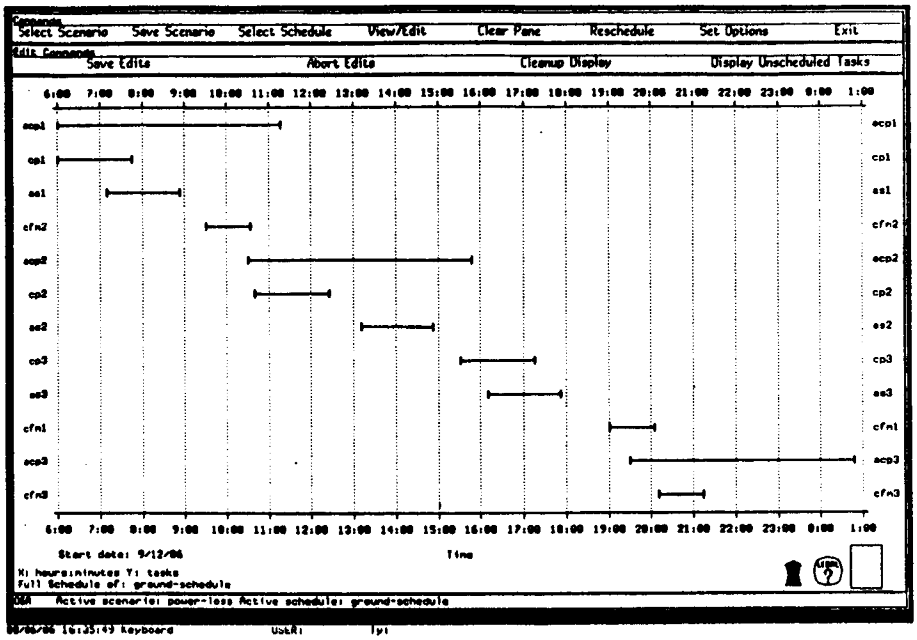

Figure 1: The Tasks in a Materials Technology Laboratory Schedule.

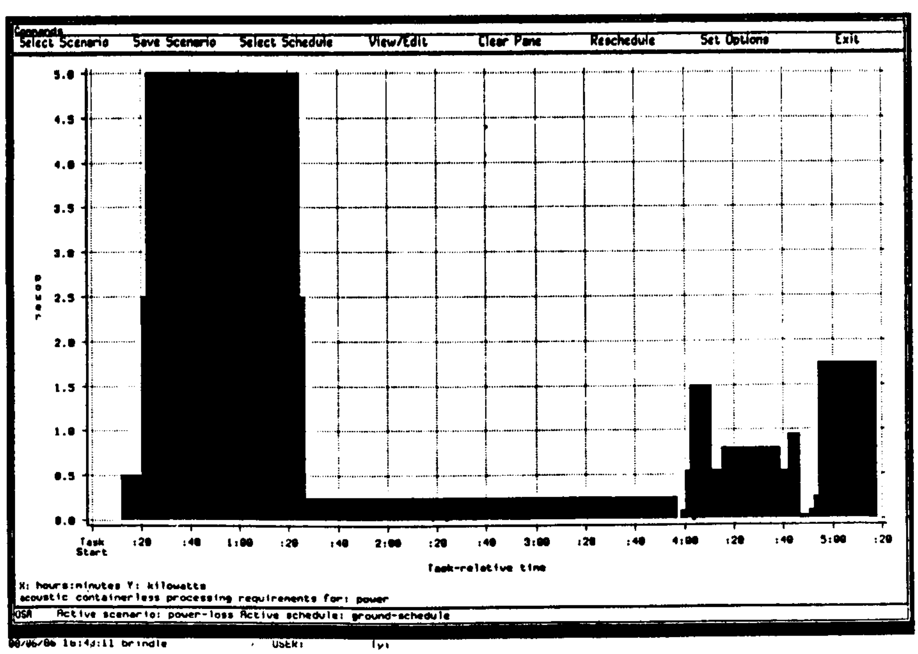

Figure 2: The Requirement of an Acoustic Containerless Processing Experiment for Power.

ORIGINAL PAGE IS
OF POOR QUALITY

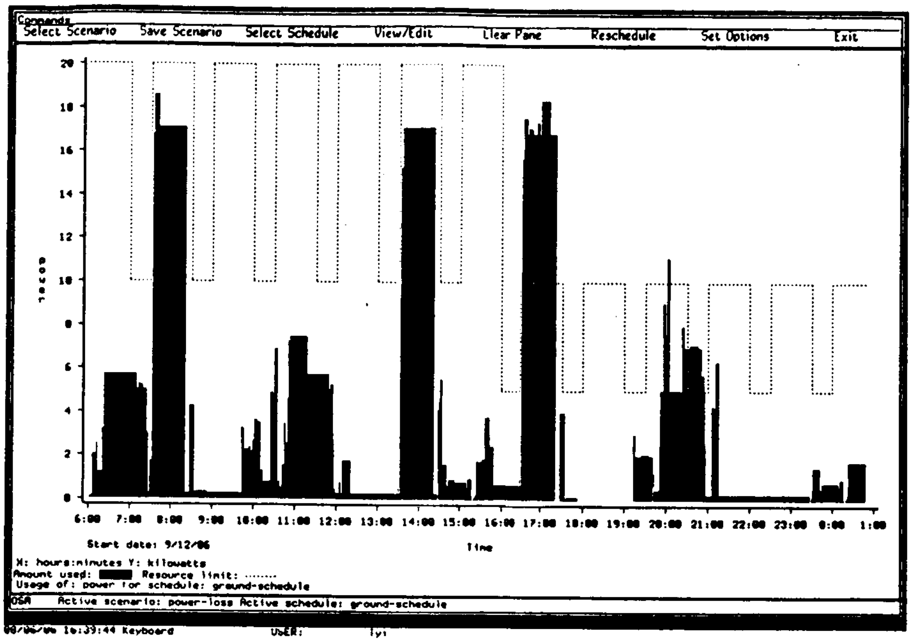

Figure 3: The Summary of Power Usage for the Schedule in Figure 1.

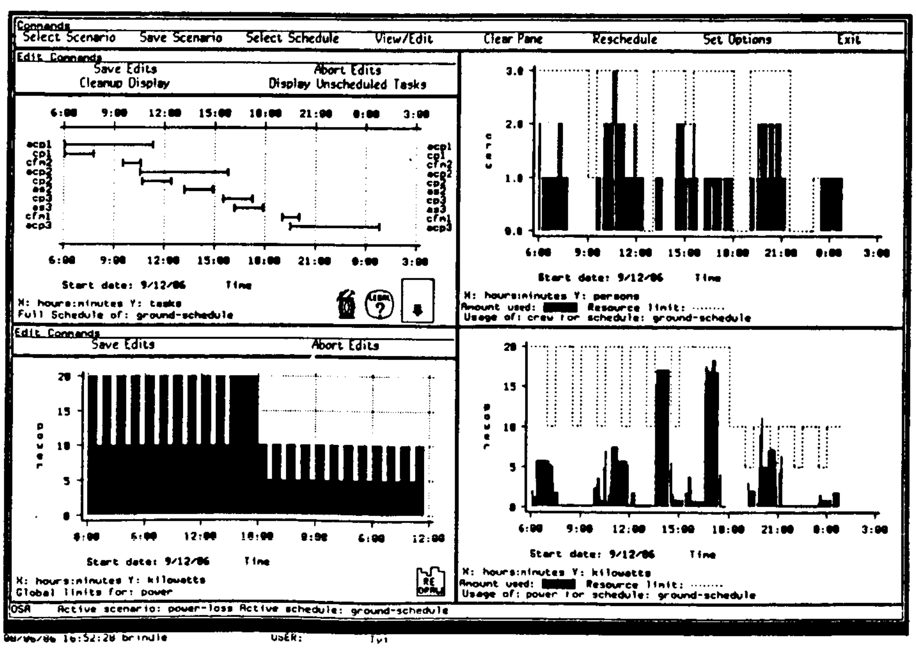

Figure 4: Four Displays as Selected by the User.