1992

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

REQUIREMENTS FOR THE IMPLEMENTATION OF SCHEDULE REPAIR TECHNOLOGY IN THE EXPERIMENT SCHEDULING PROGRAM

Prepared By: Stanley F. Bullington, Ph.D., P.E.

Academic Rank: Assistant Professor

Institution and Department: Mississippi State University
Department of Industrial Engineering

NASA/MSFC:

Office: Mission Operations Laboratory
Division: Mission Analysis
Branch: Planning Systems

MSFC Colleague: John P. Jaap

INTRODUCTION

The following list of requirements specifies the proposed revisions to the Experiment Scheduling Program (ESP2) which deal with schedule repair. These requirements are divided into those which are general in nature, those which relate to measurement and analysis functions of the software, those which relate specifically to conflict resolution, and those relating directly to the user interface. (This list is not a complete list of requirements for the user interface, but only a list of those schedule repair requirements which relate to the interface).

Some of the requirements relate only to uses of the software in real-time operations. Others are clearly for future versions of the software, beyond the upcoming revision. In either case, the fact will be clearly stated.

GENERAL REQUIREMENTS

* The user should be able to control the level of fault tolerance by placing limits on the number of repair iterations and/or the amount of time spent searching for a repair, and by specifying the particular types of repairs to be attempted, the class of conflicts to be repaired, or the repair algorithms to be used.

* A feasible schedule must be kept at all times, in case the schedule repair process is aborted.

* The user should be able to define the horizon for which schedule repairs will be made.

* The user should be able to define the horizon for which activities will be affected by a change in the schedule for a specified activity.

* When supporting real-time operations, schedule repairs must be timely, in the sense that any changes must be implementable at the time the new schedule goes into effect, not at the time the repair process started.

MEASUREMENT AND ANALYSIS REQUIREMENTS

* For a specified resource, the system should be able to determine the time, duration, and severity (e.g., number of activities involved, magnitude of overbooking) of all periods of overbooking.

* For a specified target opportunity, the system should be able to determine the time, duration, and severity of all periods of unavailability of the target.

* For a specified potential schedule change, the system should be able to quantify the effects of the change on the goodness of the schedule (e.g., change in number and severity of resource conflicts, change in schedule grade, change in crew utilization).

* For a specified activity, the system should be able to provide both a composite measure of scheduling difficulty based upon resource usage and observation opportunities, and measures of the usage of individual resources.

* For a specified activity, the system should be able to compute a composite measure of the importance of the activity, relative to other activities, based on a number of different user-input importance measures.

* For a specified activity, the system should be able to provide a measure of the magnitude of the activity's relationships (e.g., concurrency, sequencing, resource generation) to other activities.

* For a specified activity, the system should be able to present other opportunities for the placement of the activity which fall within a user-defined time horizon, and which have no conflicts or fewer conflicts than the specified activity.

* For a specified activity, the system should keep track of the number of performances scheduled relative to the number of performances requested.

* For a specified activity, when supporting real-time operations, the system should be able to report on whether the activity is in progress, and if so, the system should be able to respond to requests to handle stopping, and possibly restarting, the activity using any one of several available preemption modes (e.g., resume from the point where stopped, restart the activity at the beginning, abort the activity and lose the work which was already completed, stop the partially-completed activity, etc.). (This requirement is particularly applicable to possible future on-board scheduling systems).

CONFLICT RESOLUTION REQUIREMENTS

When an activity is moved, that activity (the "transient activity"), along with several others ("conflicting activities"), may combine to form a conflict. Usually, the resolution of such conflicts will consist of attempts to adjust the transient activity first, followed by attempts to adjust one or more of the conflicting activities, if needed. The requirements listed in this section exist in this context.

* The user-specified time horizons (see "General Requirements" above) which limit the search space may be different for the transient activity than for the set of conflicting activities.

* For a specified activity (or class of activities), the system should be able to automatically assign, recommend, or assign in response to a user request, a substitute resource(s), and to update all affected resource profiles accordingly.

* For a specified activity, the system should be able to automatically choose, recommend, or choose in response to a user request, an alternate scenario, and to update all affected resource profiles and timelines accordingly.

* For a specified activity (or class of activities), the system should be able to automatically adjust, recommend adjustment, or adjust in response to a user request, the duration of steps and/or delays between steps, and update all affected resource profiles and timelines accordingly.

* The system should be able to automatically schedule, recommend, or schedule in response to a user request, the performance of an activity which generates a resource which is overbooked, if such resource generation is possible, and to update all affected resource profiles and timelines accordingly.

* The system should be able to automatically delete (only for an autonomous on-board scheduler), recommend deletion, or delete in response to a user request, an activity, and to update all affected resource profiles and timelines accordingly.

* For a specified resource, the system should be able to reduce or increase the capacity of the resource, based upon input from the user. The system should be able to present the effects of such resource changes, and should ask for user confirmation of the changes prior to accepting them as "permanent" changes.

* In the case of an on-board scheduler, for activities which can be preempted while in progress, the system should be able to automatically preempt, recommend preemption, or preempt in response to a user request, and schedule the restart of the activity (in one of several possible modes, to be selected by the model subject to user definition, or defined by the user), and to update all affected resource profiles and timelines accordingly.

USER INTERFACE REQUIREMENTS

* The system should be able, at user request, to shift between a resource-based perspective and an activity-based perspective, in terms of the displays which are presented. The choice of perspective will normally depend on whether the user is attempting to resolve a resource overbooking or to place a specific activity on the timeline.

* The system should report to the user all changes which were actually accomplished in resolving a certain conflict, or group of conflicts.

* For a specified user-requested schedule change, the system should be able to present the effects of making such a change, possibly through a group of graphical "before/after" illustrations. The system should then ask for confirmation before accepting the requested change. (The system could, in future versions, use "filtering heuristics" to recommend acceptance or rejection of any change request, based on the effects of the change).

* The simpler and more-frequently-used interactive schedule repair suggestion capabilities of the system should be made more readily available for the user than more difficult features.

* The system should be able to display specific user-requested timelines, total resource usage profiles, resource requirements for a particular activity, and periods of resource overbooking.

* In a future revision of the system (featuring more intelligent schedule repair capabilities), for a specified user-requested schedule change, the system should query the user regarding the reason for the change (e.g., need to reduce workload on Payload Specialist #1 during the time period in question), and should be able to use this information to make intelligent schedule repairs.

CONCLUSION

A detailed review of literature relating to schedule repair and rescheduling has been performed. Based on this review, the above requirements relating to schedule repair for ESP2 have been identified. A preliminary requirements review has been held with NASA personnel, and the resulting schedule repair requirements will become part of an overall requirements document for a revised scheduling program.