N89-19819

SPACE STATION PAYLOAD OPERATIONS SCHEDULING WITH ESP2

Kenneth L. Stacy
and
John P. Jaap
Mission Integration/EL22
Marshall Space Flight Center
Huntsville, Al 35812

ABSTRACT

The Mission Analysis Division of the Systems Analysis and Integration Laboratory at the Marshall Space Flight Center is developing a system of programs to handle all aspects of scheduling payload operations for Space Station. The Expert Scheduling Program (ESP2) is the heart of this system. The task of payload operations scheduling can be simply stated as positioning the payload activities in a mission so that they collect their desired data without interfering with other activities or violating mission constraints. ESP2 is an advanced version of the Experiment Scheduling Program (ESP) which was developed by the Mission Integration Branch beginning in 1979 to schedule Spacelab payload activities. The automatic scheduler in ESP2 is an expert system which embodies the rules that expert planners would use to schedule payload operations by hand. This scheduler uses depth-first searching, backtracking, and forward chaining techniques to place an activity so that constraints (such as crew, resources, and orbit opportunities) are not violated. It has an explanation facility to show why an activity was or was not scheduled at a certain time. The ESP2 user can also place the activities in the schedule manually. The program offers graphical assistance to the user and will advise when constraints are being violated. ESP2 also has an option to identify conflicts introduced into an existing schedule by changes to payload requirements, mission constraints, and orbit opportunities.

INTRODUCTION

In this paper, we shall describe the program's capabilities as seen by the user, the activity and increment constraints the program handles and how the expert system in the program handles these constraints. We have referred to activities and activity timelines, assuming that the reader has an intuitive understanding of these concepts. This discussion of ESP2 can be better understood if we first define several terms.

EXPERIMENT - In general, a collection of procedures and equipment which, when executed, contribute to the body of technological or scientific information. In ESP2, the term experiment refers to the "model" or "models" corresponding to the general definition.

FUNCTIONAL OBJECTIVE - A large section of an experiment's procedures which accomplishes a definite purpose, such as verifying equipment, collecting baseline data, etc.

MODEL (ACTIVITY MODEL) - The database representation of an experiment or part of an experiment. A model is a collection of constraint and execution definitions. Some of the definitions apply to the whole model and some apply to the "steps" of the model.

STEP - The smallest, clearly delineated part of an activity model. Steps are usually executed in sequential order, but are not necessarily contiguous. Resource and crew requirements of a model are shown at the step level.

PERFORMANCE - An execution of an activity model. A model may be performed multiple times to collect additional data.

EXPERIMENT TIMELINE - A time history of experiment performances and related activities planned for a Station increment. These activities are represented by the start and stop times of model steps.

The Expert Scheduling Program is a highly interactive, user-friendly program designed to run on a workstation with high resolution graphics, a mouse and keyboard. It checks all input for reasonableness and provides in-line help text as needed throughout the program for user assistance. The program allows the user to interrupt any of the ESP2 activities without corrupting internal or external data. A journal of all user interactions is maintained to assist in tracking the development of a schedule.

ORIGINAL PAGE IS OF POOR QUALITY

PREPARING THE INPUT DATABASE

The application of ESP2 to Space Station increment planning is a very complex process that requires several weeks to build a final timeline. An outline of the schedule building operations for a Space Station increment is shown below. This outline includes activities done by ESP2 as well as related activities done in preparation for running ESP2.

Preparation:

* Obtain a general understanding of the increment and its objectives.
* Develop a "gross" timeline for the flight increment. This will include such things as STS docking, equipment changeout, crew work cycle, reboost times, etc.
* Determine the resource usage by non-scientific activities. This will include the crew off-duty periods, the sub-systems, etc.
* Determine the names and descriptions of all observation opportunity subjects (opportunity file contents).

Modeling:

* Obtain and enter into the increment model a description of the increment availabilities.
* Convert or generate all subjects required to be on the opportunity file.
* Build models of all activities (both science and non-science) that must be scheduled. These models should be checked for internal consistency.
* Further validate these models by scheduling each one or each related group onto a new (empty) schedule. This will verify that the models can schedule.

There are several goals to keep in mind when developing activity models. The timeline engineer should always attempt to minimize the total number of models required to schedule the increment. Each model should be as simple as possible. Always minimize the number of steps on each model. Steps are required to reflect a change in any one of the resources required. Develop models that will schedule automatically.

ESP2 is the "heart" of the Experiment Scheduling System (ESS). This scheduling system will be used at several levels of Space Station planning. The Discipline Operations Centers (DOC) will use ESP2 to schedule gross models of all their users' requirements. This schedule will define the individual user's operation "windows". The user (scientist) at a User Operations Facility (UOF) will use ESP2 to build a detailed timeline for all his experiment's operation. The detailed timelines will be integrated into a coordinated Discipline activity timeline at the DOC. The Payload Operations Integration Center (POIC) will use ESP2 to integrate all of the DOC timelines. This program must be able to accommodate each of these functions. Figure 1 shows the relationship and data flow within the overall system.

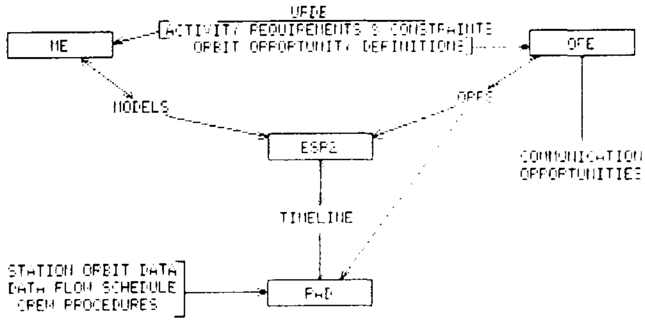

Figure 1. Space Station Timeline Analysis Flow

The scientist whose payload is scheduled to be flown on a Space Station increment will enter his experiment's requirements and constraints in the User Requirements DataBase (URDB). The Model Editor (ME) program and the Opportunity File Editor (OFE) program will use expert systems to extract this information and build files for input to ESP2. The Payload Activity Display (PAD) will use output from ESP2 to produce summary charts for the scientists, crew operations and ground support.

The Model Editor program is a database management tool used to create, modify and copy ESS activity model files used by the ESS programs. The domain of the Expert Scheduling Program includes the activity and increment constraints that are defined in the model file. These files contain the following;

* Increment data
  - Increment start date and duration
  - Constraints and availabilities
* Activity Models
  - Requirements and constraints
  - Grouping data
* ESP2 Control data
  - Opportunity and external retrieval file names
  - External retrieval control data
  - Scheduling control data
  - Grading criteria

The increment data is composed of all the information required to uniquely identify a particular Space Station flight increment. This includes the name or number of the increment, start date, start time and the increment duration. Also, included is a list of the crew members. This input may also be used to specify the crew skills required for activities to be conducted during the flight increment. All resources, constraints and equipment required for the increment operations are identified and availabilities set in this data. The two types of resources are consumables and nondepletables.

Each Activity Model contains constraints and requirements for scheduling the operations that it represents. This data includes the following parameters: maximum performance duration, performance time windows, maximum number of performances, performance separation, startup/shutdown or scenarios, sequencing, concurrence, crew lockin and resource tolerances. The experiment operations are divided into steps to specify the detailed requirements and constraints. Each model may contain up to 50 steps. The data that is specified at the step level is: duration, delays, equipment/constraint requirements, consumable requirements, nondepletable requirements, resource carry-through, fulltime crew or monitoring, and selected, intersected or avoided orbit opportunities. Each of these parameters and requirements will be explained in more detail as we later describe how they are implemented in ESP2.

In most cases, more than one Activity Model is required to represent all of an experiment's operations. The model file contains grouping data that describes which models represent each experiment and discipline.

The ESP2 control data allows the user to tailor the program's operation for specific applications. This data is saved and maintained on the model file.

SCHEDULING

ESP2 provides an array of tools for building, checking and documenting the activity timelines. There are three tools available to build the timeline. The external retriever can retrieve all or part of a previously generated timeline. The automatic scheduler can quickly add multiple performances of multiple models to a timeline. The manual scheduler allows the user to modify or delete what is already scheduled or to add new activities to the timeline. When utilizing any of these development tools, ESP2 will verify that the resources are not overbooked and the opportunity requirements are not violated in the timeline. The program can maintain, in internal storage, up to three timelines (in addition to the current timeline). ESP2 provides a complete output package that allows the user to inspect a timeline and prepare documentation. The relationship of these tools within ESP2 is shown in Figure 2.

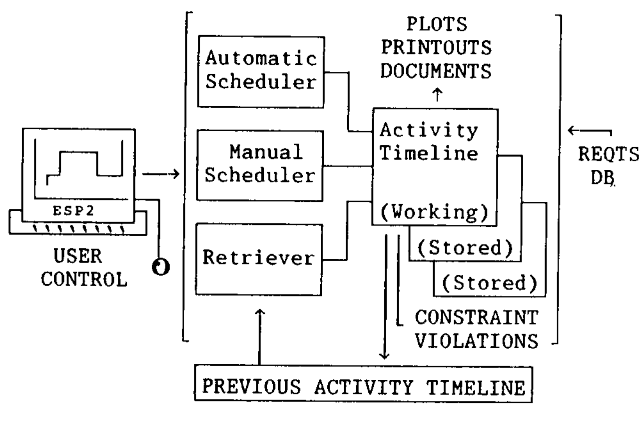

Figure 2. ESP2 Schedule Building Options

There is no hierarchy among these tools. The user may choose them in any logical order to build the final schedule. Each of these will be discussed in this section.

As a result of our experience in planning nine Spacelab missions, we have developed a prototype technique for scheduling Space Station payload operations.

* Start by retrieving all activities that have been defined and scheduled by other sources. These normally include the crew models, sub-system models and other fixed models. A timeline file is usually written at this point so that all future runs of ESP2 can retrieve this partial schedule and add to it.
* Schedule the high priority activities using either the automatic scheduler or the manual scheduler. This is an incremental process usually requiring several days. The activity models may be grouped according to a user defined criteria to provide a set of logical intermediate points during the development of a timeline. At the end of each terminal session, or an intermediate point, a timeline file is written. At the beginning of the next scheduling session an external retrieval from the file is performed to continue the timeline build-up.
* Schedule low priority activities in a manner similar to the high priority activities. During this step, the resource availabilities may become so limited that many activities cannot be scheduled. The timeline developer may either change the requirements of the models or shuffle what is already scheduled to free the needed resources. The explanation facility (trace) is useful to understand why a model will not schedule or why it schedules at an unexpected time.

EXTERNAL RETRIEVAL

External retrieval is used to load into ESP2 a schedule which was previously generated and stored on a file. When using external retrieval, the user specifies the start and stop times of the retrieval and the activity models to exclude. The program does not allow double booking of a crew member or overlapping performances of the same model, but no other validation checks are made. Since the external retrieval accesses the model file, any updates to the resources (consumables, nondepletables and equipment only) will be incorporated into the timeline.

The retrieval function provides several valuable capabilities to the user. Prior scheduled activities may be retrieved, or "loaded" into ESP2, before adding more activities to the timeline. This function is necessary for complex increments that require several weeks to develop. Also, two timelines may be merged. This is done by retrieving from first one timeline file and then another. Output can be produced for a selected subset of the timeline. This is done by excluding all but the desired activity models, retrieving these models and producing the desired output.

AUTOMATIC SCHEDULER

Automatic scheduling is used to add multiple performances of multiple models to the current schedule. This option allows the program to decide where (in the schedule) to place the performances of the models. When using the automatic scheduler, the user specifies the models to be scheduled and, optionally, the order in which to attempt to schedule the models. The user may divide the models into 12 groups and specify the scheduling strategy for each group. The scheduling order does not affect the schedule except that resources are assigned on a first-come, first-serve basis. The user also specifies the weighting parameters reflecting the priority of the models, the schedule start time, the grading criteria for the schedule and the number of scheduling passes to make.

The automatic scheduler is the primary mode for developing timelines. This scheduler minimizes the time required to produce a schedule. It will build a schedule which meets all requirements. It will quickly add activities to a schedule without allowing user input errors. It can automatically generate several schedules and save the best.

A brief top-down explanation of the scheduling process in ESP2 is outlined below.

* Initialize the timeline.

Cycle until all performances are attempted:

* Choose the next model/ performance to try.
* Check the constraints of the chosen model against a timeline of availabilities.
* If successfully checked, insert the performance in the timeline and update the status of increment constraints.

The ESP2 scheduling process is based upon calculating windows which are nested in a hierarchy such that each lower window is totally contained within the window above it. Checking of constraints begins at the topmost window and proceeds downward and across (i.e., "depth-first searching"). Checking is successfully complete whenever an acceptable window is found at the bottom level. Checking fails whenever another window at the top level cannot be defined.

Selection Methods

The choice of which model to schedule next is determined by the selection method. ESP currently provides three selection methods for the user to choose from; Fixed, Random and Grade Maximization (^Grade). Each selection method will choose a scenario for the selected model based on upon a user specified strategy.

The Fixed method requires a user to specify the scheduling order of the models within each group. An optional command, for this method only, will order the models within each group such that the most difficult is listed first. This command computes a difficulty factor for each model based on the time windows on the model, the orbit opportunity requirements of the model, the number and duration of requested performances. These computations consider what is already in the schedule; so that different orders will be obtained by generating the order at different stages of the increment timeline development.

The Random method uses a random number seed and generator to select which model to schedule. Each performance has an equal probability of being chosen. Therefore, requesting an excess number of performances will give a model a higher probability of being selected early. This allows the user to assign a high priority to a model for the purpose of scheduling. By choosing the random selection method and requesting many scheduling passes (trials), the user can effect a Monte-Carlo approach to finding the best solution.

The ^Grade method evaluates which model will provide the greatest improvement in the schedule grade and selects it to be scheduled next. With this method, ESP2 dynamically updates the selection order as each model is scheduled. The grading criteria selected by the user will affect this selection. The grade is based upon the following five factors: the number of performances scheduled, the number of activity models scheduled, the amount of crew time utilized, the amount of activity operation time scheduled and the mean number of performances scheduled.

Loading Algorithms

The placement of the model in the timeline is determined by the loading algorithm and when the requirements of the model are met. This algorithm controls where each performance is scheduled if it is not precisely fixed by performance time windows, performance separation times or other model constraints. ESP2 currently provides two loading algorithms; Front and Back. Front loading will always attempt to schedule the performance at the earliest time which satisfies all constraints. Back loading will always attempt to schedule the performance at the latest time which satisfies all constraints. The loading algorithm can be selected independently for each group of models to be scheduled.

The activity requirements and constraints defined in the model file are not rule based. Each activity model has several different types of requirements and constraints which determine the time at which the model may be scheduled. ESP2 has a fixed rule base which handles all of these requirements. These requirements may be classified as: time constraints, performance options, relational constraint, orbit opportunities and resources.

Time Constraints

An activity model may require that all performances be scheduled within specified time windows. Each model may have up to 10 windows with a start time, a stop time and the number of performances required for each. Multiple performance windows allow the timeline engineer to predictably and variably space performances throughout the schedule.

The size of the scheduling window for a model can be constrained by limiting the maximum performance duration. This gives the user control over the length of an entire performance while allowing ESP2 to stretch out steps as needed. An activity model may also limit the separation between adjacent performances. ESP2 will always attempt to minimize the performance separations.

Each step of an activity model contains limits for the duration and the time to delay before beginning step operations. These limits are defined in the model database with a minimum and maximum value. ESP2 will attempt to maximize step duration (as long as all other requirements are fulfilled) and minimize step delays.

Performance Options

There are two mutually exclusive options for defining the steps to be scheduled on each performance. These options are defined in the models during the ME build process. ESP2 is capable of scheduling either option without any further user interaction.

The first option is Startup/Shutdown. A model may have startup steps which are executed only for the first performance of an activity and shutdown steps which are executed only for the last performance of an activity. However, there must be a core of steps (at least one) which are executed on all performances. With this capability, ESP2 can automatically schedule activity startup/shutdown with a single model.

The second option is Scenarios. A scenario is an alternate ordering of the steps of an activity model. A model may define up to four scenarios. Each scenario has a priority associated with it. Currently, ESP2 has a limited rules set for handling scenarios. This is an area that is still being developed and could provide major improvements in the scheduling capabilities.

Relational Constraints

ESP2 also provides techniques for scheduling experiment activities relative to other Space Station activities. These techniques are sequencing and concurrence. Sequencing is accomplished by requiring a model to be scheduled within a specified time period after the performances of a leading activity model. This time window is restricted by a minimum and maximum value. ESP2 will attempt to minimize the sequence delay. By manipulating these parameters and the performance separation, the user can implement this as either a prerequisite or a one-to-one sequencing requirement. Concurrence is accomplished by requiring that one step of a model be scheduled at the same time as a step of another model. ESP2 provides three levels of concurrence: Mandatory, Necessary and Desired. Each level is handled differently by the scheduler.

Orbit Opportunities

An activity model step may specify a set of orbit opportunities that must exist during its operations. An orbit opportunity may be defined as (1) a celestial object or ground site which is to be observed or (2) an Experiment observation window. Each model step may have up to three categories of requirements for the opportunities. These are Intersected, Selected and Avoided. ESP2 will process any subject on the opportunity file as one of these requirements. Up to three subjects may be intersected for a specific step. The step will be scheduled only if all three subjects exist for the entire duration of the step. A list of up to five subjects may be processed as selected opportunities. The step will be scheduled if at least one of the subjects is available for the entire duration of the step. The list of 5 selected opportunities may also be avoided. In this mode, the step can only be scheduled when none of the opportunities are available for any part of the activity.

Resources

A resource used in discrete integral amounts, such as a TV camera, whose availability is temporarily changed for the duration of its usage is called an Equipment/Constraint. Each activity model step may require up to 15 different types of Equipment/Constraints. In many cases this type of resource is used to prevent two models from scheduling at the same time. It can also be used to make models schedule concurrently and for "bean counting".

The availability of a Nondepletable resource is also temporarily changed only for the duration of its usage. However, this type of resource may be used in fractional amounts (e.g., Power). An activity model step may require up to 10 nondepletables. A negative usage rate will increase the availability for the duration of the step.

A Consumable resource is one whose availability is permanently changed by its usage (e.g., Camera film). Each model step may require up to 10 different types of consumables. A consumable can be defined to be used at a fixed amount, constant rate or based upon a nondepletable rate.

An activity model step may require resource Carry-Through. This function, when used in conjunction with step delays, allows ESP2 to schedule resource usage during a delay between activities.

The Payload Crew is possibly the most valuable resource for any Spacelab mission or Space Station increment. The scheduling of crew members to perform the experiment activities is given special consideration. ESP2 will permit the user to override warnings about overuse of other resources, but it is impossible to overbook a crew member (i.e, schedule him/her to perform more than one activity at once). The program also provides for fulltime crew participation or periodic monitoring. Each model step may specify up to three lists of the eight possible crew members. These lists can be used to identify the crew according to their skill levels. The total number of crew members required for a step can be distributed across the skill levels, but cannot exceed eight. ESP2 allows crew lockin (i.e., require that the same crew members be scheduled for all steps).

MANUAL SCHEDULER

The manual scheduler, also known as the timeline editor, is used to modify existing activities or to enter activities whose requirements are not well modeled or whose placement is predetermined. When using the timeline editor, the user enters the start and stop times and crew usage for each step of an activity. Resource usage is taken from the model steps. The editor presents a screen of data to be edited using form-editing techniques and commands. When the user issues the command to commit the page to the schedule, the entries are checked for constraint violations. If none of the violations will not destroy the integrity of the schedule, the user may ignore the warnings and update the schedule. A chart showing where each required resource of an activity is available and the intersection of these availabilities is provided to assist the user.

The editor also provides easy access to the automatic scheduler for quick scheduling of a model (possibly with some requirements overriden). This override feature will temporarily change a limited set of model requirements within ESP2 only. The model database is not changed.

CHECKPOINTING

Checkpointing/restarting is used to save a schedule internally; and, at some later time, restart from that checkpoint. ESP2 has four slots which may contain schedules. Each checkpoint is timetagged and labeled to identify its contents. One of these slots is used to maintain the current or working schedule. All changes are made and all output is generated from this schedule. The second slot used by the program maintains a copy of the last external retrieval. Two additional checkpoints are available for the user. During the development of a timeline, it is good practice to periodically create a checkpoint to save the existing schedule.

EXPLANATION FACILITY

The engineering trace is an explanation facility which can show the user why ESP2 could not schedule another performance of a model or why ESP2 scheduled the model where it did. ESP2 cannot tell why a model cannot be scheduled. Indeed, there is often not a single reason. The crew may be available when the orbit opportunity is not, or the orbit opportunity may be available when the power is not. The reason for failure was not the lack of crew or the lack of power or the lack of an orbit opportunity. The reason can be stated: "because all requirements were not met at the correct times". ESP2 provides a graphic and textual trace of the checking and loading portions of its scheduling algorithms so that a proficient user can understand how ESP2 arrived at a schedule and therefore why something is not scheduled or is scheduled at a particular time.

OUTPUT

ESP2 documents experiment timelines! The array of output generated upon user request includes tabulations, terminal and laser plots, document outputs, printouts, and experiment timeline and on/off files. A complete list of the types of output available in the current program is shown in Figure 3.

Most of these output options utilize a control form. This form presents a choice of up to six output mediums to accommodate the various hardware capabilities. When ESP2 is run on a non-graphics terminal, the terminal plot options are not available. However, the plots can be generated on a Laser printer. When ESP2 is run on a workstation, the program is capable of displaying up to 9 overlayed terminal plots. These plots can be reduced to icons for later recall.

One of the most useful output formats is the Step Opportunity Plot. This output shows all the requirements of a step and where these requirements are met. The intersection of the time periods where all requirements are met is plotted as the opportunities to schedule the step. This is a valuable tool when building up a timeline.

1-	Composite TL (Timeline file)	TPF
2-	Subset Composite TL	TP
3-	Selected Activity TL's	T F
4-	Crew Timelines	TPF
5-	Selected Equipment TL's	TP
6-	Equipment/Constraint Usage	TPF
7-	Selected Nondepletable TL's	T
8-	Nondepletable resource usage	TP
9-	Schedule Overview	T
10-	Subset Composite Overview	T
11-	Activity Overviews	T
12-	Performance Summary	T
13-	Crew Work Summary	T
14-	Crew Utilization Summary	T
15-	Crew Availability Summary	T
16-	Unscheduled Performances	T
17-	Activity Summary	T
18-	Average Nondepletable Usage	T
19-	Nondepletable Minimum Avail.	T
20-	Step Opportunities	TP
21-	Performance Opportunities	TP
22-	Models	T
23-	Orbit Opportunities	TPF

AVAILABILITIES:
T - Tabulation or Printout
P - Plots
F - File

Figure 3. ESP Output Menu Options

CONCLUSIONS

Although the ESP2 program is still being developed, its predecessor, ESP, has been thoroughly qualified. ESP has successfully supported the activity timeline development for nine Spacelab missions, three of which have flown, and several partial payloads. This program is currently being used not only by MSFC, but also by the Flight Project Engineering Office at JSC to schedule the SLS-1 and SLS-2 Spacelab missions, Martin Marietta Denver Aerospace to schedule the TSS mission and Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (DFVLR), Germany's equivalent to NASA, to schedule the D-2 Spacelab mission. Langley Research Center recently requested a copy of ESP for evaluation of its applicability for their Space Station evolution studies.

ESP is being used at MSFC today to schedule 90-day strawman flight increments for Space Station investigations. This exercise will provide more insight into the modifications required for ESP2 to handle the Space Station scheduling task. ESP2 will utilize AI technology, where it is cost effective, to enhance the program capabilities. Some currently planned improvements include; the ability to edit the activity models within ESP2, the ability to generate observation opportunities within ESP2, and additional loading algorithms. The capacities of the program, as it was designed for Spacelab operations, and the maximum usage experienced for the three successfully completed Spacelab missions are listed in Figure 4.

PROGRAM CAPACITY		HISTORICAL PEAKS	
Mission duration (days)	1000	10	(SL-1)
Activity models	500	361	(SL-1)
Performances per Model	500	135	(SL-1)
Performances per Timeline	50000	1218	(SL-1)
Steps per Model	50	37	(SL-1)
Steps per Timeline	50000	5163	(SL-2)
Crew members	8	7	(SL-3)
Equipment/Constraint types	99	67	(SL-1)
Nondepletable resources	25	5	(SL-3)
Consumable resources	25	7	(SL-1)
Opportunity subjects	200	109	(SL-2)
Acq/Losses per opportunity	500		

Figure 4. ESP Program Capacities

During the Spacelab era, the average overall usage of these capacities was only about 10 percent. Therefore, we believe these limits will be a good starting point for scheduling Space Station payload operations.

In a recent interview for the Marshall Star, John Jaap summed up the status of ESP2 development thus; "Scheduling for Spacelab missions is complex enough, even for short missions. On the other hand, Space Station activity scheduling, with more shared resources and experiment durations counted in weeks rather than hours, requires a more automated computer program. For Space Station we're adding features to ESP that will automate many of the tasks that now must be done by an expert user. New rules and strategies for scheduling are being developed."

REFERENCES

1. Dumoulin, John. "ESP Keeping Spacelab Experiments Conflict-Free," Marshall Star, vol. 28, number 33 (April 27, 1988): page 2.

2. Jaap, John P. and Davis, Elizabeth, "Experiment Scheduling for Spacelab Missions", Conference on Artificial Intelligence for Space Applications, Huntsville, Alabama, November 13-14, 1986.

3. Jaap, John P. and Davis, Elizabeth, "The Scheduling Techniques of ESP2", Second Annual Workshop on Space Automation and Robotics, Dayton, Ohio, July 20-23, 1988.

4. Jaap, John P. and Davis, Elizabeth, Experiment Scheduling Program User's Manual. Huntsville, Alabama: Marshall Space Flight Center, [1988].

5. Maxwell, Theresa G. "Investigation of Experiment Selection Order Determination for Space Mission Scheduling." M.S. thesis, University of Alabama in Huntsville, 1987.

6. Stacy, Kenneth Lamar., Training Course 5200: Mission Activity Model Development, Huntsville, Alabama: Marshall Space Flight Center, [1988].

7. Stacy, Kenneth Lamar., Training Course 5420: Timeline Editing. Huntsville, Alabama: Marshall Space Flight Center, [1988].