Maximally Expressive Modeling of Operations Tasks

John Jaap (John.P.Jaap@nasa.gov)
Lea Richardson (Lea.M.Richardson@nasa.gov)
Elizabeth Davis (Elizabeth.K.Davis@nasa.gov)
Mission Support Systems Group
Ground Systems Department
Flight Projects Directorate
Marshall Space Flight Center
National Aeronautics and Space Administration

Abstract — Planning and scheduling systems organize "tasks" into a timeline or schedule. The tasks are defined within the scheduling system in logical containers called models. The dictionary might define a model of this type as "a system of things and relations satisfying a set of rules that, when applied to the things and relations, produce certainty about the tasks that are being modeled." One challenging domain for a planning and scheduling system is the operation of on-board experiments for the International Space Station. In these experiments, the equipment used is among the most complex hardware ever developed, the information sought is at the cutting edge of scientific endeavor, and the procedures are intricate and exacting. Scheduling is made more difficult by a scarcity of station resources. The models to be fed into the scheduler must describe both the complexity of the experiments and procedures (to ensure a valid schedule) and the flexibilities of the procedures and the equipment (to effectively utilize available resources). Clearly, scheduling International Space Station experiment operations calls for a "maximally expressive" modeling schema.

TABLE OF CONTENTS

.....
1. INTRODUCTION 1
2. THE SCHEMA 2
3. EXTENDED EXAMPLE 4
4. POTENTIAL BENEFITS..... 9
5. CONCLUSION 10

1. INTRODUCTION

Modeling even the simplest of activities cannot be automated, no sensor can be attached to a piece of equipment that can discern how to use that piece of equipment, and no camera can quantify how to operate a piece of equipment. Modeling is a human enterprise – both an art and a science. The modeling schema should allow the models to flow from the keyboard of the user as easily as works of literature flowed from the pen of Shakespeare.

U.S. Government work not protected by U.S. copyright.

The Ground Systems Department at the Marshall Space Flight Center has embarked on an effort to develop a new scheduling engine that is highlighted by a maximally expressive modeling schema. The schema, presented in this paper, is a synergy of technological advances and domain-specific innovations. Some of the key features are given below:

- Decomposition of the problem into salient components — Operations are decomposed into activities that define resource requirements and sequences that define relationships between activities. Sequences can also contain other sequences, repeated activities and sequences, and optional activities and sequences.
- Graphical paradigms — Simple graphical paradigms such as outlines and networks are used to build and depict the models. Modeling itself is done using techniques such as drag-and-drop.
- Modeling equipment modes — Implicit resource requirements are defined by equipment mode models, thereby more closely representing the real world.
- Intuitive and rich expression of the relationships between components — The schema employs common-sense representations of temporal relationships using everyday concepts like sequential, during, and overlap. Innovative enhancements to represent the continuance of resource usage between tasks, the interruption of tasks, minimal percent coverage, and temporal relationships to outside tasks have been added to the modeling schema.
- Public services — The schema also introduces the concept of public services, models that are scheduled at the request of another model.

Poor modeling is the downfall of automatic scheduling. If all the requirements are not included in the model, then the scheduler has little chance of producing a satisfactory schedule. The modeling schema must have an available representation for all the constraints and be friendly enough to allow the user to enter them all without excessive labor. The scheduling systems currently used in NASA's manned space flight program cannot capture many of the constraints which describe the operation sequences required to operate the Shuttle or the International Space Station, especially

those required by the science payloads. This failure of the modeling schema has begotten the "scheduling cadre," which digests all the requirements, builds the best models allowed by the current schema, makes notes containing the remainder of the requirements, and then generates the timeline using a mixed-initiative approach to scheduling.

The objective of the maximally expressive modeling schema is to capture easily *all* the requirements and constraints so that an automatic scheduler can produce a satisfactory schedule.

2. THE SCHEMA

The inspiration for the modeling schema is the real world that we interact with and observe each day. The schema is based on the scientist who goes to a lab to perform an experiment, the instructor who explains a complex procedure to students, the housewife who prepares dinner for guests while helping with homework, and the various experiments that are performed on the International Space Station. The modeling schema is an evolutionary improvement of the modeling schema currently used by the Marshall Space Flight Center for International Space Station payloads [1].

Decomposition into Salient Components

This schema models scheduling requirements by defining "activities" and "sequences of activities." Activities generally equate to the simplest or lowest-level tasks. A sequence of activities is usually required to represent scheduling entities. Consider the following example: to do the laundry a housewife must wash, dry and put away the clothes. Doing the activities out of sequence or standalone does not accomplish the objective.

Activities define the resource requirements (with alternatives) and other quantitative constraints and requirements of the tasks to be performed. Activity requirements may be grouped into "all-of" groups or "one-of" groups. Groups may be hierarchical. For example: the housewife can use the oven or the stovetop to cook a roast; however, the duration would be different, and a different pan would be used. Activity requirements include the specification of the minimum, maximum and preferred duration of the activity.

Sequences define the temporal relationships between activities. In our laundry example, we discussed three activities that would be done one after the other (i.e., in a sequence). Sequences may also define relationships with other sequences, as well as with events. For example: laundry is done after taking the children to school, and dinner is served between the evening news and primetime TV. Temporal relationships include during, sequential, separated, overlap, standby, fragmentable, percent coverage, and (for repeated items) cyclic. Resource lock-in and one-to-one relationships are also included. Other temporal constraints are modeled for the International Space Station but are outside the scope of this paper.

Graphical Paradigms

The schema is implemented using graphical paradigms to interact with the user – both for presenting the data and for entering the data. An outline paradigm is used for activities and a network paradigm is used for sequences.

Hierarchies of groups of requirements best describe the constraints of most non-trivial activities. The outline paradigm is well suited to modeling hierarchies of groups, because it can be manipulated by a drag-and-drop interface

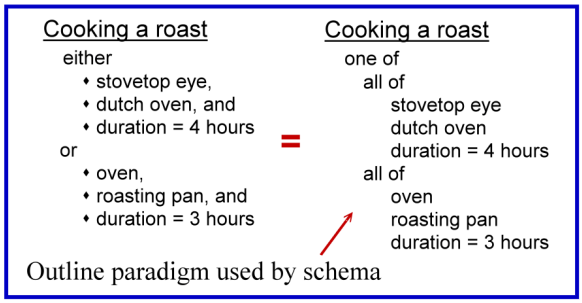

Figure 1 – Constraint Hierarchies

and nested to any depth without ambiguity. Figure 1 compares two representations of the "cooking a roast" example; notice the similarity between the two. The implementation provides a list of predefined constraints; clicking on one of the constraints adds it to the activity model at the current cursor location. "One of" and "all of" headings are also added by clicking on their respective icons. Constraints are arranged into groups and hierarchies via drag-and-drop. Values are added by double-clicking on a constraint item and entering data into a dialog box.

Sequences use a "network" paradigm to define the relationships between tasks (activities and included sequences). The method consists of selecting tasks from a list and placing them on a drawing canvas. They are moved with the primary mouse button and connected to each other by the secondary mouse button. When a connection is made, a dialog box appears to allow the user to specify the type of relationship and the time delays between tasks. Figure 2 shows a sequence for a Thursday evening in a hypothetical household. The "followed by"

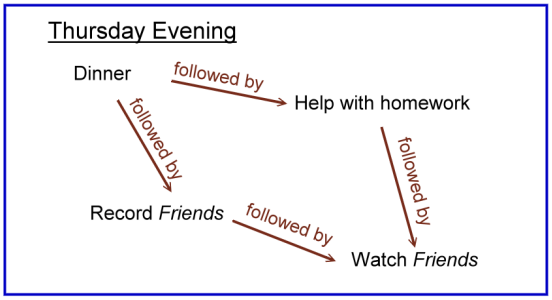

Figure 2 – Network Paradigm

relationships in this example include slack time so that homework and the recording are not artificially forced to be the same duration.

Modeling Equipment Modes

Most tasks are accomplished using equipment of some sort. Most equipment have various operating modes: e.g., a microwave has modes such as defrost, reheat, and cook. The power requirements of each mode are predefined. On the International Space Station, the characteristics of each piece of equipment are well known to those building and integrating the equipment into the International Space Station systems. The equipment and their modes may be modeled independently of the experiments that will use the equipment. Occasionally an experiment will need to use a piece of equipment in a new or novel manner; consequently, a new mode must be defined. Equipment mode models use an outline paradigm like that used by activity models.

Public Services

A public service is a task (usually a sequence) that can be scheduled in conjunction with a user's sequence. When a user includes a public service in a sequence, the process of scheduling the sequence will also cause the public service to be scheduled. Note that the details of the public service (such as tasks of the public service sequence, resource usage, conditions required, etc.) are not visible to the requesting sequence, but will be booked when scheduling. Public services are usually modeled in advance. For example: a housewife might ask her husband to bring home a loaf of bread for dinner. She does not need to define where to get the bread or how to get there. She needs only to request the bread.

Intuitive and Rich Expression of the Relationships

The sequence model may include one or more of the relationships listed below. As stated earlier, sequences may contain activities, other sequences, public services, and external events (such as launch and docking).

Sequential — Items follow each other. Minimum and maximum separations may be specified.

Separated — Items may not overlap, but the order of execution is not defined. Minimum and maximum separations may be specified.

During — Items occur simultaneously; when items are of different durations, one contains the other. Which item is during the other may be specified. Minimum and maximum separations of both the start and end times may be specified.

Overlap — Items overlap; which item starts first may be defined. Minimum and maximum durations of the overlap may be specified.

Percent Coverage — One item must be scheduled during another item so that it covers a certain percentage of the duration of the other item. For example: a parent needs to provide assistance to a certain young child playing on the computer about 60% of the time. This time may be broken into reasonably short segments. The minimum coverage, the maximum number of segments, the minimum duration of a segment, and the maximum separation between segments may be specified.

Standby — During a delay between sequential or separated items, a standby item is scheduled to book (consume) the resources that are used during the delay. For example: if there is delay between washing the clothes and drying the clothes, an item would be scheduled to show that the washer is in use. If drying follows immediately after washing, then the standby item is not scheduled.

Fragmentable — When an activity may be fragmented into parts, an activity or sequence is scheduled to book the resources that are used during the interruption. For example, when a stamp collection is being organized, it could be laid out on the kitchen table. Sorting the collection could be fragmented into multiple short sessions, but between the sessions the table is in use and cannot be used for anything else. The maximum number of fragments, the minimum duration of a fragment, and the maximum duration of an interruption may be specified.

Cyclic — An item in a sequence may be repeated; minimum and maximum repetition counts may be specified. The frequency in hours, days or weeks may be specified. For the daily and weekly options, the time of day (with variation) may be specified. For the weekly option, days of the week may be specified. Additionally, the temporal relationship of the repetitions can also be separated or overlapped with time constraints. When the minimum repetition count is 0, the item is considered optional.

Lock-In — If two activities in a sequence contain identical "one-of" selection groups, then the same constraints must be chosen when scheduling the sequence. For example: assume there is a choice of which car to drive to the grocery and which car to drive from the grocery. When scheduling the grocery shopping sequence, the same car must be chosen for both the trip to and the trip from the market.

One-to-One — When an item is to be done multiple times, and each repetition of this item is related to a pre-existing item in the timeline, but only one instance of the item is to be scheduled for each instance of the pre-existing item, then a one-to-one relationship is required. For example: many pictures of the crew having breakfast are to be taken, but only one picture is to be taken per meal.

Modeling Flexibility and Nuances of the Tasks

Several of the features that have been defined are especially useful for modeling flexibility. They are alternate choice of constraints ("one-of" groups) in the activity, variable durations of an activity, variable separations in a sequence, sequence scenarios, and optional items in a sequence. The

subtle nuances of tasks can be modeled with features like lock-in, standby, fragmentation, and percent-coverage relationships.

The temporal relationships defined by the schema are common sense relationships, not the classical (and sometimes esoteric) temporal relationships [2]. Sequential, separated, during, and overlap can be mapped directly to the classical relationships. The schema introduces percent coverage, fragmentable, and standby that are not in the classical set of temporal relationships. The schema also includes a cyclic relationship that is not in the classical set but can be found in virtually every calendar program.

3. Extended Example

(Note: The following example is hypothetical; any similarity to a real experiment is accidental and unknown.)

Payload Overview

The Atmospheric Contamination Experiment (ACE) is an International Space Station payload that is designed to monitor both ionic and particulate contamination of the air inside the International Space Station. The hardware will be brought up on a Shuttle visit and returned to earth about three months later. The hardware consists of a base unit and six remote sensors. The base unit is attached via Velcro at a well-exposed location inside the main module and connected to both the power output receptacle and a data input receptacle. The base unit records data from the sensors in flash memory and periodically dumps the data to the ground. The six remote sensors are attached at various locations within the module. The remote sensors are battery-operated and communicate with the base unit via infrared signals. The base unit has cradles for recharging the remote sensors; it contains changeable filters and a small fan to force air through the filters as each is exposed. There is a hydrogen sulfide (H_2S) generator for a special test. Additional requirements of the experiment are discussed as the model is developed.

Equipment Mode Models

The base unit has three modes: checkout, record, and downlink. The modes are modeled this way:

- Checkout mode
  - Power (RL7 bus) – 157 watts
  - Data rate – 120 mbps
  - Real-time downlink (D22 bus) – required
- Record mode
  - Power (RL7 bus) – 42 watts
- Downlink mode
  - Data rate (D22 bus) – 400 mbps
  - Real-time downlink – required

The modes of the remote ion sensors are modeled this way:

- Passive mode
  - (no constraints)
- Recharge mode
  - Power (RL7 bus) – 80 watts

The mode of the particulate collector is modeled this way:

- Collection mode
  - Power (RL7 bus) – 12 watts

Activity Models

Selected activity models are shown below; others are referenced by the sequences but are not defined in this paper. The models shown below are screen captures that have been annotated to simulate some of the mouse-over pop-ups that are part of the graphics interface.

ACE_setup (activity model) — This activity unstows, deploys, and checks out the ACE base unit; see Figure 3. Crewmember SC3 is experienced on this activity and can do it in 20 minutes; crewmember SC2 is not experienced but can do a minimally acceptable job in 30 minutes, but 45 minutes would be better.

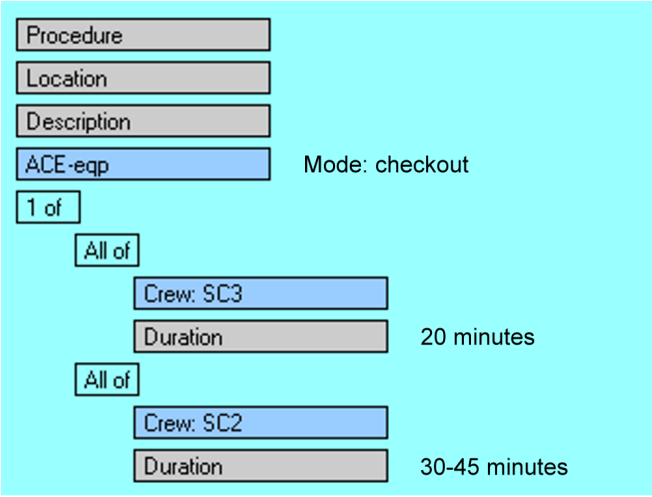

Figure 3 – ACE_setup Activity Model

The procedure box contains a reference to the book and page number where the procedure for this activity is documented. The location box contains the location where this activity occurs. The description box contains a brief description of this activity. The dialog box for entering the duration is activated by double clicking on the duration box. The dialog is shown in Figure 3a.

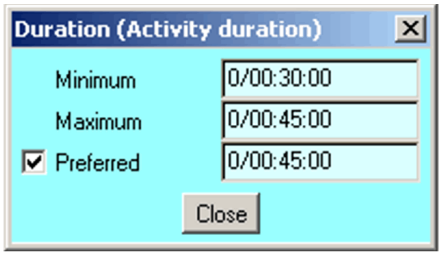

Figure 3a – Dialog for the Duration of an Activity

ACE_video (activity model) — This model, shown in Figure 4, expresses the requirement that one of the three explicitly listed crewmen be scheduled.

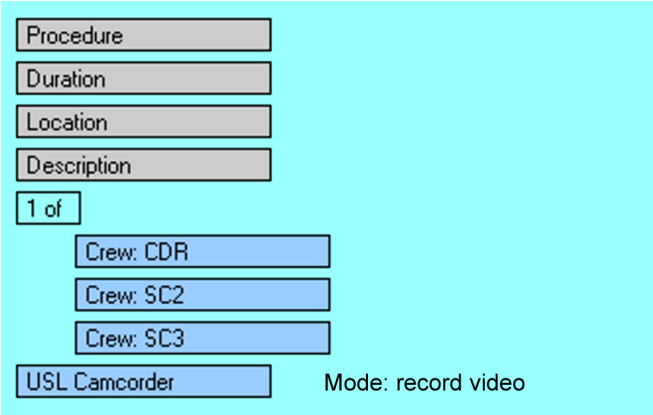

Figure 4 – ACE_video Activity Model

Sequence Models

Selected sequence models are shown in this section. On each diagram, the symbols between the boxes indicate what type of relationship has been defined. The legend for the symbols is shown in Figure 5a.

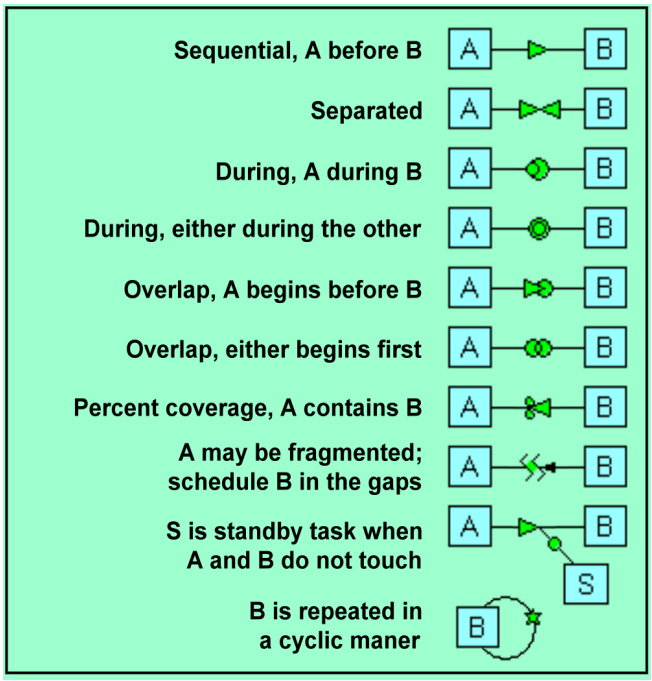

Figure 5a – Symbols for the Temporal Relationships

For each activity or sequence included within a sequence, the user can indicate two things. If the item is to be scheduled when scheduling the sequence, the user checks the box; otherwise the item must pre-exist in the schedule and relationships are applied relative to the pre-existing item. The user can also indicate that relationships to the containing sequence are interpreted as relationships to a given item in the sequence by affixing a *hook* to that item; see Figure 5b. A model cannot request that some items (such as sleep or docking) be scheduled, but a model can define temporal relationships to existing instances of these items.

When a sequence is included within another sequence, it is adorned with markings showing information about it. These are defined in Figure 5c.

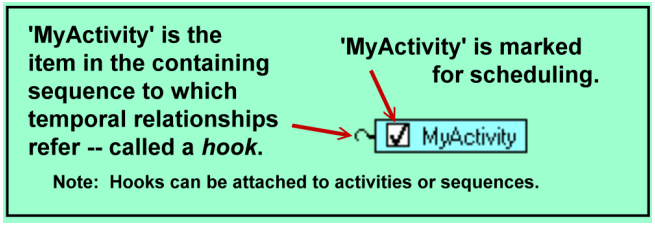

Figure 5b – Features of an Item in a Sequence

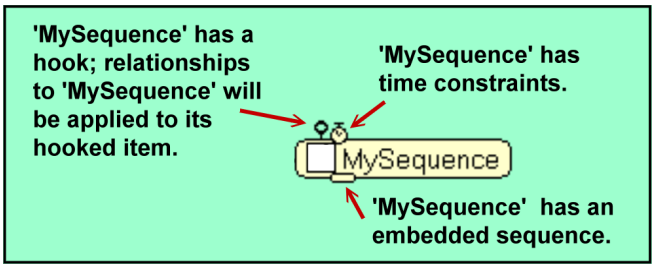

Figure 5c – Indicators on an Embedded Sequence

ACE-Deploy (sequence model) — This model shows the sequence for the deployment and stowing of the hardware; see Figure 6. The time duration of this sequence is about 90 days. The temporal relationships are all sequential. There is a lock-in relationship between the setup and stow activities – notice the black dot on the line between setup and stow. Lock-in means that whichever crewmember sets up the hardware is also requested to stow the hardware. One of the mouse-over pop-ups for a temporal relationship is shown in the box containing the word "Sequential." Placing a check

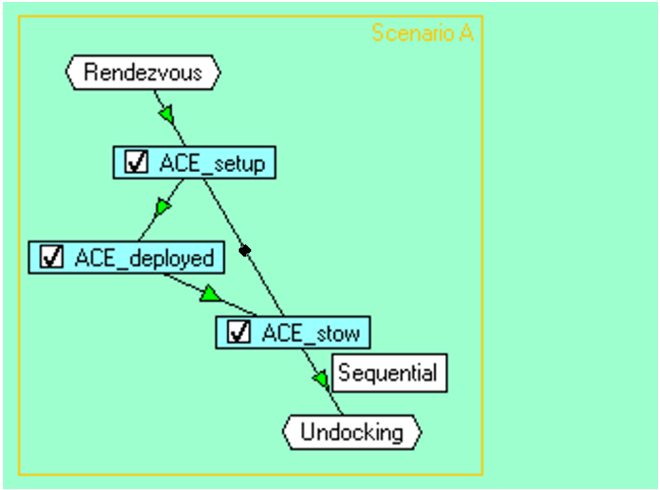

Figure 6 – ACE-Deploy Sequence Model

mark in a box indicates that the activity or sequence is to be scheduled when scheduling the containing sequence. Some slack time is allowed between Rendezvous and ACE_setup and between ACE_stow and Undocking. Notice that public tasks Rendezvous and Undocking are shown in a hexagonal box and cannot be marked for scheduling. The dialog for one of the relationships is shown in Figure 6a. The items are sequential with a separation ranging from 12 hours to 5 days.

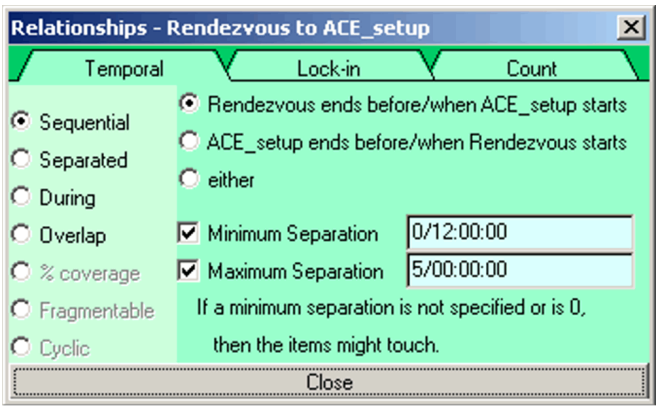

Figure 6a – Dialog for a Sequential Relationship

ACE-Meal (sequence model) — This model, shown in Figure 7, represents the requirement for a crewmember to remove one of the mounted sensors and hold it at a location in the galley during a meal. This is to be done for 10 minutes (duration is specified on the ACE_meal activity). The start of the activity is to begin between 15 and 25 minutes after the meal starts. The activity is to be done no more than once during any meal; i.e., there is a one-to-one relationship. The one-to-one relationship is indicated by the blue oval on the connecting line.

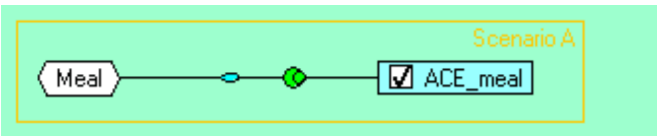

Figure 7 – ACE-Meal Sequence Model

The dialog box for defining the during relationship is shown in Figure 7a. Notice the specification of minimum and maximum separation between start times. The dialog for specifying the one-to-one relationship is shown in Figure 7b.

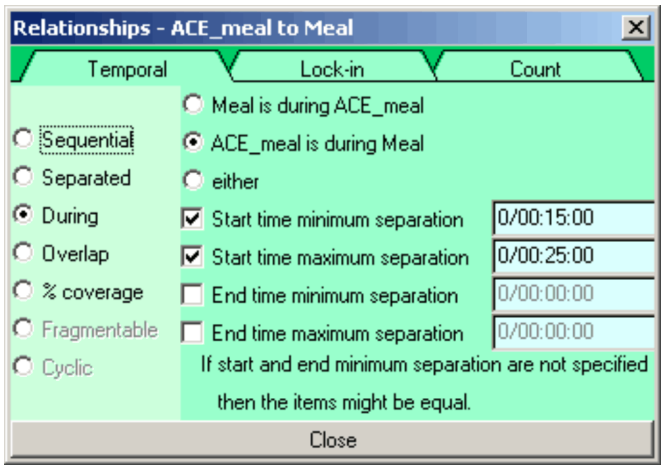

Figure 7a – Dialog for a During Relationship

ACE-Exercise (sequence model) — This model, shown in Figure 8, represents the requirement for a non-exercising crewmember to remove one of the mounted sensors and hold it at several different locations in the proximity of the exercising crewmember. Exercise lasts about an hour, but

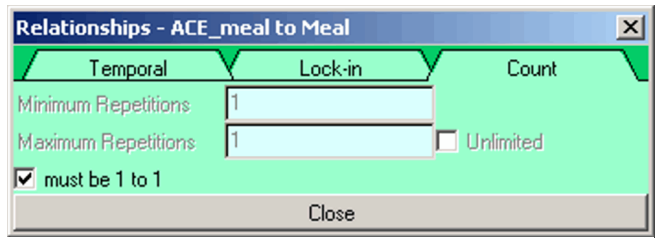

Figure 7b – Dialog for a One-to-One Relationship

the ACE-Exercise task does not need to be done continuously; it can be done in as many as ten segments, each at least 5 minutes long and separated by no more than 10 minutes. However, the sensor must be held for 65% of the duration of Exercise. In addition, video of the exercise is requested but not required. The video is only required for the first 10 minutes of the exercise; i.e., it overlaps by 10 minutes. This model also shows a typical use of an embedded sequence. The *eye-bolt* attached to the ACE-Video-Conf model indicates that it has a hook (as in ACE-Video-Conf model shown in Figure 9) and that the relationship to the sequence will be applied to the hooked item.

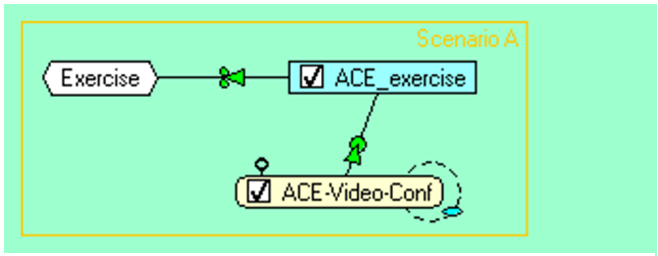

Figure 8 – ACE-Exercise Sequence Model

The dialog box for defining the percent-coverage relationship is shown in Figure 8a.

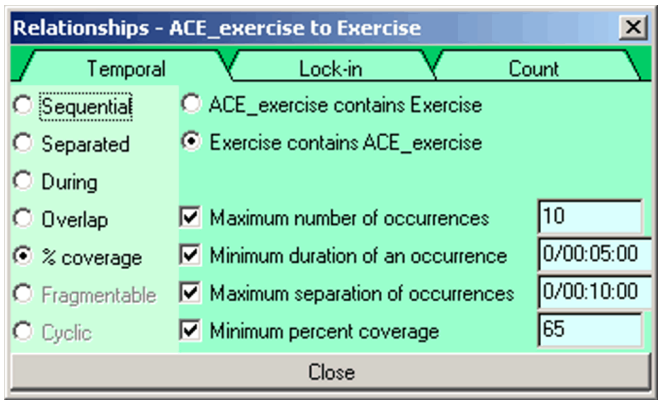

Figure 8a – Dialog for a Percent-Coverage Relationship

The dialog box for defining the overlap relationship is shown in Figure 8b. The video conference is to be optional; an item is optional when the minimum repetition count is zero – optional items are depicted by a dotted repetition loop. The dialog shown in Figure 8c is used to specify the repetition count.

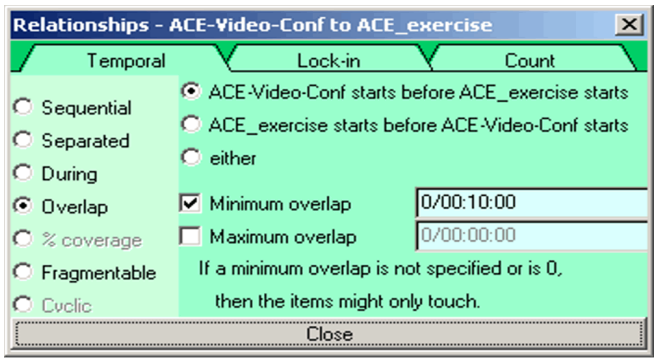

Figure 8b – Dialog for an Overlap Relationship

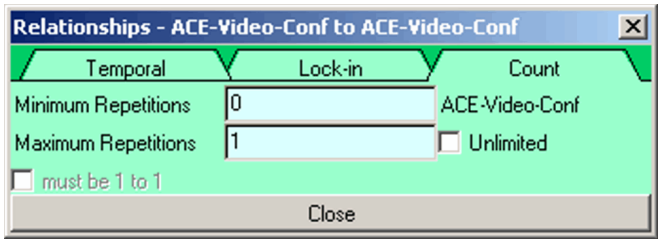

Figure 8c – Dialog for a Repetition Count

ACE-Video-Conf (sequence model) — This model shows the videoconference requirements; see Figure 9. The International Space Station camera (a shared resource) is configured in the ACE_video-setup activity. A gap is allowed between the ACE_video-setup and the ACE_video activity. During this gap the camera is not available to other activities; this fact is modeled by the standby relationship to the activity called ACE_video-stby, which reserves the camera. Likewise, a gap is allowed between the ACE_video activity and ACE_video-stow; this gap is also filled with the ACE_video-stby activity. If the scheduler shrinks one of the gaps to zero duration, then ACE_video-stby is not scheduled. Additionally, this sequence shows the use of the hook to designate that when including this sequence in another sequence, relationships to this sequence are actually relationships to the ACE_video activity. In the ACE-exercise model, the requirement for a 10-minute overlap of the ACE_Exercise activity and the ACE-Video-Conf sequence would result in an overlap with the ACE_video activity; if the hook were not employed, the ACE_exercise would probably overlap ACE_video-stow.

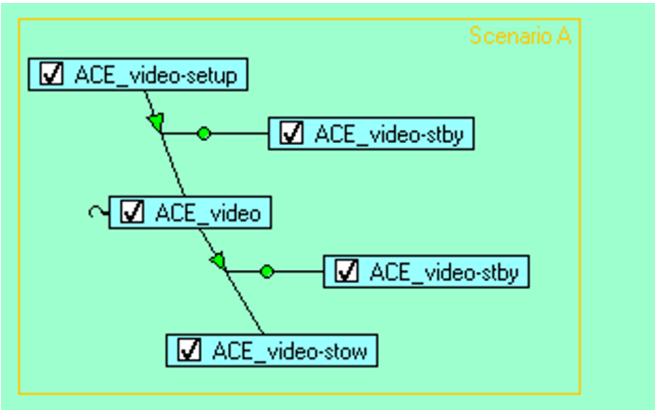

Figure 9 – ACE-Video-Conf Sequence Model

ACE-H2S (sequence model) — This model, shown in Figure 10, specifies two options, called scenarios, for accomplishing the same objective. The ACE-Video-Conf model is included in one scenario but not the other. A dialog box allows the user to specify the strategy for choosing between scenarios. Strategies are chosen from a list of available strategies.

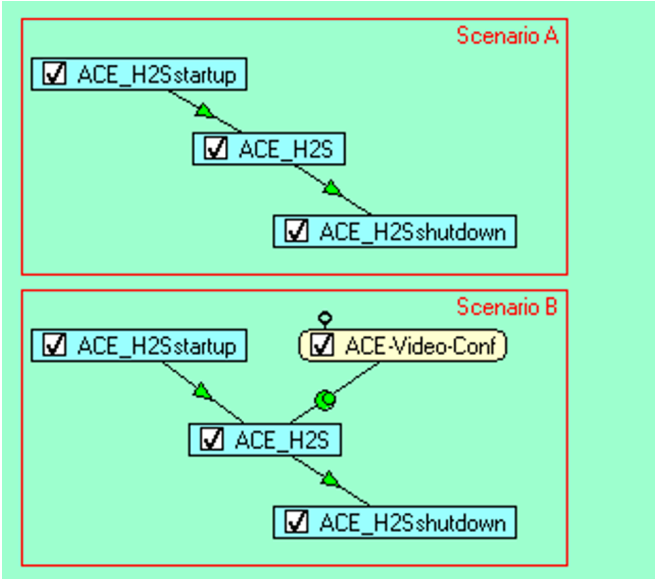

Figure 10 – ACE-H2S Sequence Model

ACE-Maintenance (sequence model) — This model shows that the maintenance activity can be fragmented or interrupted; see Figure 11. If it is interrupted, the ACE_stby activity is scheduled during the interruption. The ACE_stby activity is not the activity that caused the interruption but shows the state of the experiment and the resources used by and during the interruption.

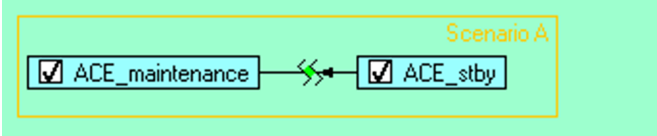

Figure 11 – ACE-Maintenance Sequence Model

The limits on the interruption are entered in the dialog shown in Figure 11a. In this case, maintenance can be broken into six fragments, none of which can be shorter than 10 minutes and the interruptions cannot exceed 15 minutes.

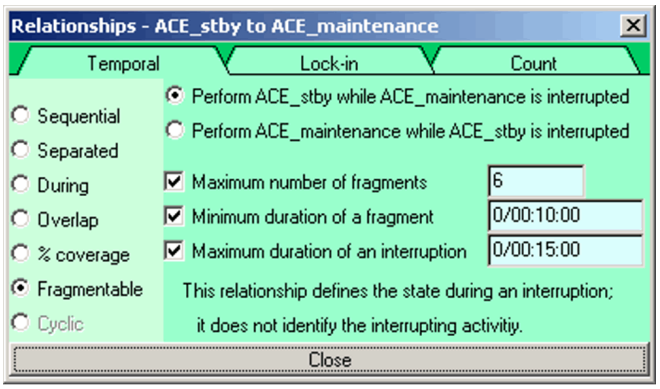

Figure 11a – Dialog for a Fragmentable Relationship

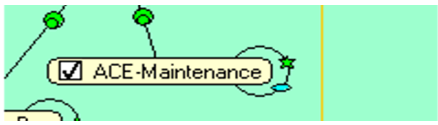

Figure 11b – Repeating ACE-Maintenance (Partial Model)

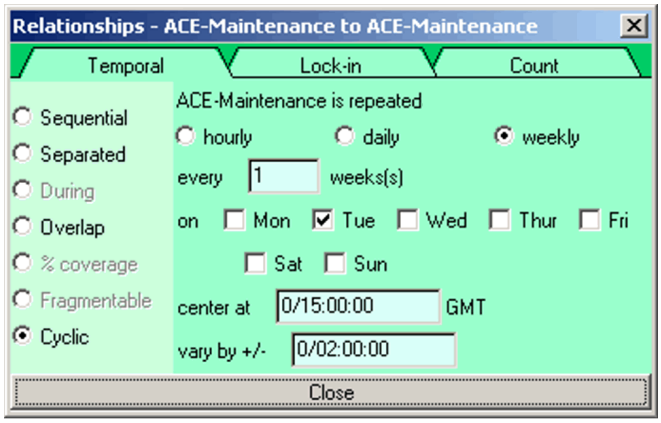

Figure 11c – Dialog for a Cyclic Relationship

The ACE-Maintenance model will be embedded in another model as shown in Figure 11b. A repetition relationship will be created specifying an unlimited number of repetitions (as in previous Figure 8c). A repetition will be utilized and a cyclic relationship modeled (see Figure 11c) so that ACE-Maintenance is done on Tuesday afternoons. Mousing over the oval on the repetition loop will pop up a box showing the number of repetitions requested.

ACE-GloveBox (sequence model) — This model, depicted in Figure 12, shows how to request the operation of another experiment or facility. The ACE experiment has a requirement to calibrate one of the remote sensors in a controlled environment that is available in the glovebox facility. The person modeling the ACE experiment outlines the ACE objective to the person who models the glovebox. The person modeling the glovebox then builds a glovebox model to support the ACE objective and provides the model as a *public service* model. The ACE model then only needs to include the public service in a sequence. The person modeling the ACE experiment does not need to know how to model the glovebox.

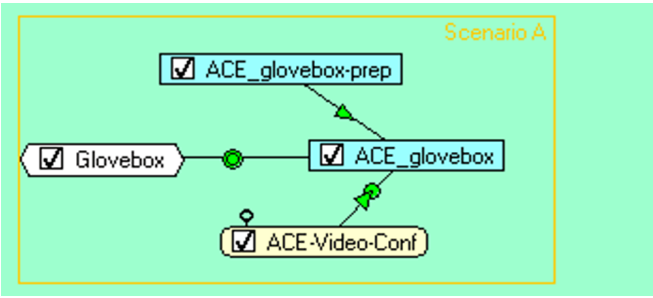

Figure 12 – ACE-GloveBox Sequence Model

Glovebox (public service sequence) — The glovebox facility is a vacuum chamber including gloves with which a crewmember can manipulate an article within a vacuum. Preparation requires glove inspection (change out when required) and chamber cleaning. Evacuation requires the exclusive use of the vent lines, etc. An operation of the facility is modeled in Figure 13. Since this model will be listed as a public service and will be included in other sequences, a hook is provided. When a sequence defines a relationship to this public service sequence, the relationship is applied to the hooked activity.

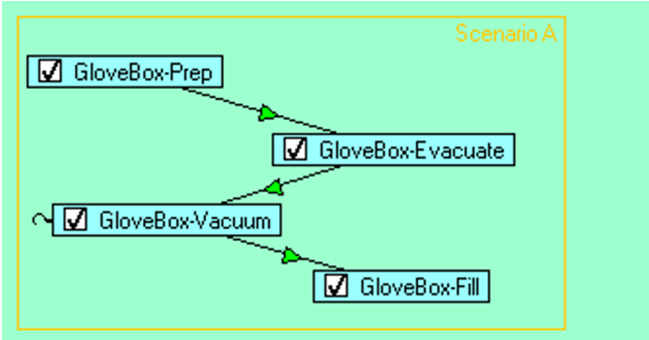

Figure 13 – Glovebox Public Service (Sequence Model)

ACE-Master (sequence model) — This model (see Figure 14) shows the relationships of the various sequences of the ACE experiment to one another. It also shows that these tasks are to be done while the hardware is deployed. Notice that the ACE_deployed activity is not checked for scheduling; it is scheduled by the deploy sequence. In this sequence, the ACE-Exercise, ACE-H2S, ACE-Meal and ACE-GloveBox tasks are separated. The ACE-H2S sequence is optional, and the ACE-Meal, ACE-GloveBox, ACE-Maintenance and ACE-Downlink tasks are repeated multiple times.

The astute reader will recognize that the entire ACE experiment could be scheduled by sending to the scheduling engine only the ACE-Deploy model followed by the ACE-Master model.

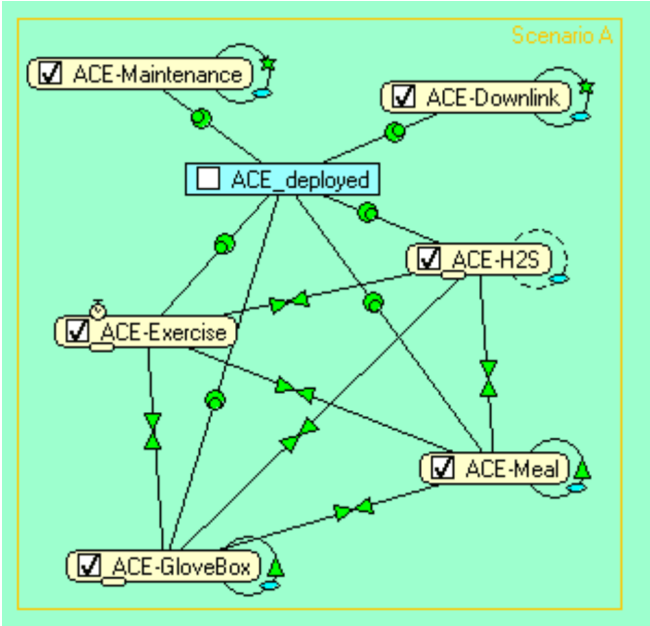

Figure 14 – ACE-Master Sequence Model

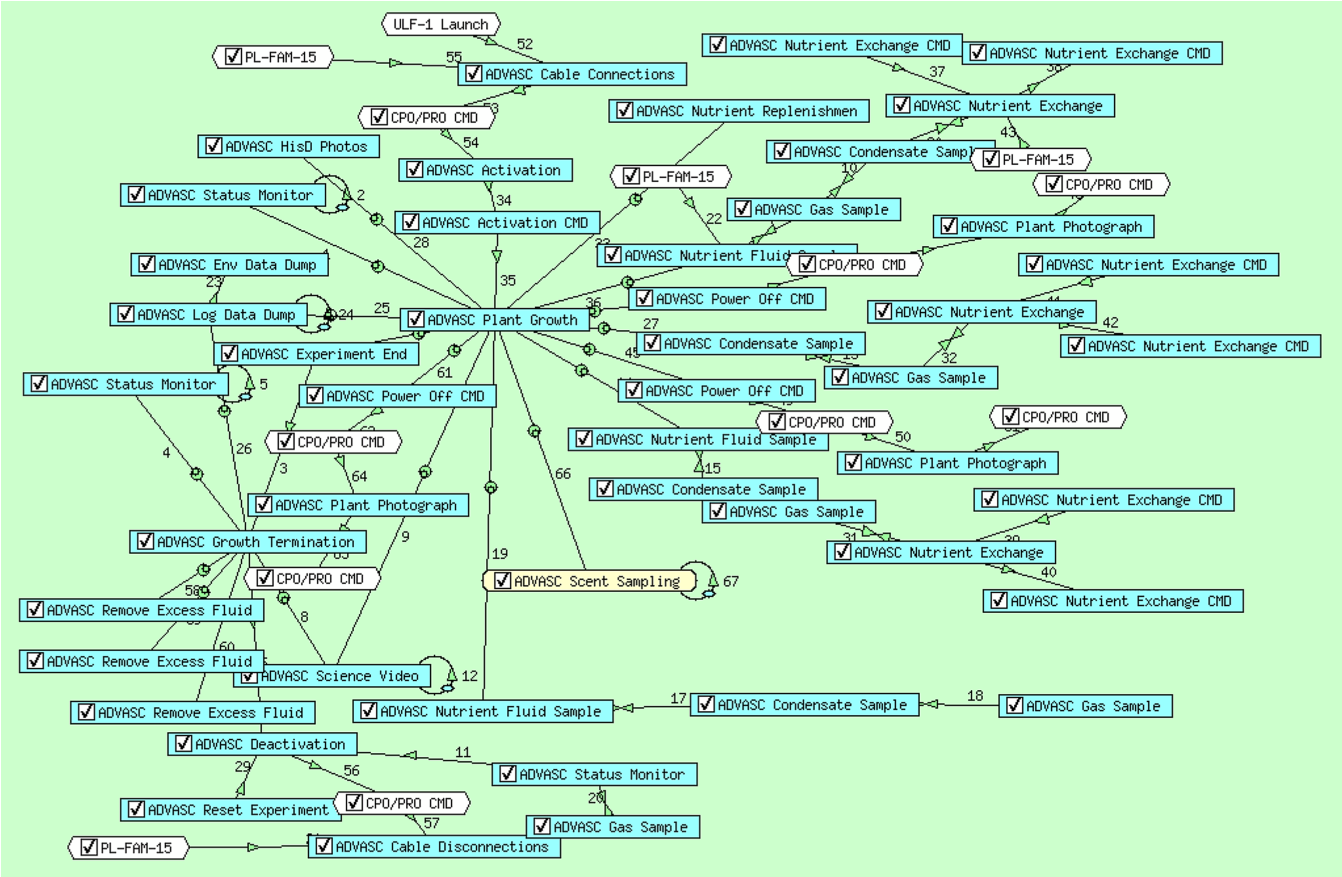

Figure 15 – Actual Operations Sequence for an Internalional Space Station Payload

An Actual International Space Station Model

Figure 15 shows a model for some of the Advanced Astroculture (ADVASC) payload tasks to be done on the International Space Station during Expedition 7, scheduled during the spring and summer of 2003. The modeling schema available to the ADVASC modeler does not have all the features of the schema presented in this paper and the user was required to enter several constraints as "notes." Because of the limitation of the scheduling engine, the scheduling cadre decomposed this sequence into about twenty separate sequences and, in the end, resorted to scheduling most of the tasks manually.

4. Potential Benefits

The current state-of-the-art in modeling methodologies and scheduling engines results in a linear paradigm with knowledge contributed by payload experts, vehicle experts and scheduling engine experts. This paradigm, depicted in Figure 16, requires significant effort and flow time. The payload experts often struggle to enter their requirements using a language that is limited – often resorting to notes to fully describe their requirements. The vehicle and hardware experts then convert and augment this knowledge to further prepare the models for scheduling. The scheduling team then feeds the models to the scheduling engine. Since the models are incomplete, they often have to "steer" the scheduler to produce an acceptable schedule.

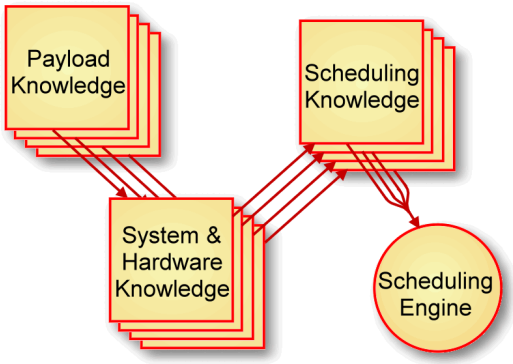

Figure 16 – State-of-the-Art Paradigm

The modeling methodology presented in this paper allows a streamlined paradigm as depicted in Figure 17. The vehicle experts would enter the system and hardware constraints independently of the payload knowledge. The payload experts would enter the payload requirements. The maximally expressive modeling schema would allow them to specify all of the payload requirements without resorting to notes for the scheduling team – these models would be ready for the scheduling engine. Having models that express *all* the constraints allows the scheduling engine to operate automatically without human intervention.

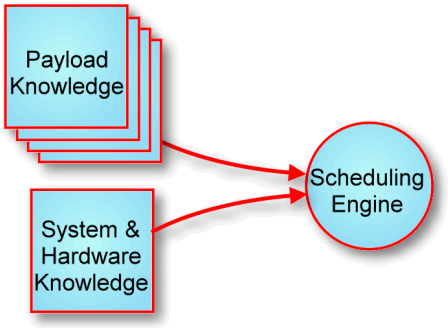

Figure 17 – Future Paradigm

The advent of a maximally expressive modeling schema can significantly reduce the manpower and flow time required to produce a schedule in a complex scheduling domain. An operations concept encompassing both the model schema presented in this paper and the corresponding scheduling engine has been proposed [3].

5. CONCLUSION

There are more esoteric representations of requirements, particularly the temporal relationships. However, the objective of the maximally expressive modeling schema is to allow a person who has detailed knowledge of the experiment and minimal knowledge of scheduling to build usable models. Toward this end, simple everyday relationships, like during, overlap, etc., are employed; graphics paradigms are used to enter and display the information; and all nuances of the tasks are directly representable. The schema is truly "maximally expressive."

REFERENCES

[1] John Jaap, Patrick Meyer and Elizabeth Davis, "Using Common Graphics Paradigms to Represent Complex Scheduling Requirements," *Workshop Notes - International Workshop on Planning and Scheduling for Space Exploration and Science*, 28-30 October, 1997.

[2] James F. Allen, "Maintaining Knowledge about Temporal Intervals," *Communications of the ACM*, November 1983, Vol 26, No 11.

[3] John Jaap and Kim Muery, "Putting ROSE to Work: A Proposed Application of a Request-Oriented Scheduling Engine for Space Station Operations," *SpaceOps 2000*, 19-23 June, 2000.

BIOGRAPHIES

John Jaap *has been employed by NASA's Marshall Space Flight Center since 1980. He has worked virtually all tasks related to planning and scheduling, serving as lead developer, designer and user of scheduling tools. He has designed and developed web applications since 1995. He has received NASA's Silver Snoopy, the NASA Exceptional Achievement Medal, the Space Act Award, and is a Space Flight Awareness Honoree. He received a Bachelor of Arts degree in Mathematics from Mississippi State University in 1966.*

Lea Richardson *has been employed at the Marshall Space Flight Center since 1990. She has worked numerous console positions during space-flight operations, collected and modeled science requirements for Spacelab missions, and developed space-flight software. She has been developing mission planning and web application software since 1996. She graduated cum laude with a Bachelor of Science degree in Mathematics with a minor in Computer Science from the University of Alabama in Huntsville.*

Elizabeth Davis *began her employment by NASA at the Marshall Space Flight Center in 1980. She has developed mission planning software since 1977. In 1978, she began concentrating on space activity scheduling software. She has been involved in web application development since 1997. She is the recipient of numerous awards, including the Silver Snoopy and the Space Act Award. She graduated summa cum laude with a Bachelor of Science degree in Mathematics from Middle Tennessee State University.*