

MODELING OF COMPLEX OPERATION SEQUENCES

John M. Usher

Mississippi State University

John Jaap

NASA, Marshall Space Flight Center

ABSTRACT

This paper presents a modeling tool that is a part of the Request Oriented Scheduling Engine system being designed at NASA for the scheduling of payload space activities on board the International Space Station. The resident modeler provides a robust method for easily representing complex sequences of activities for use in planning and scheduling activities. Although directed toward space activity scheduling, the paper addresses other application areas for this technology.

KEYWORDS: Sequence modeling, project planning, project scheduling

INTRODUCTION

Many planning and scheduling activities with manufacturing enterprises make use of some form of sequence models. These models may be simple enough to be represented in the working memory of the person with the responsibility for generating a plan or schedule, or they may be of such a complexity that some documented form is required to adequately explore the planning and scheduling options. Many applications of planning and scheduling in today's industry make use of computerized tools to provide the speed and flexibility to generate optimal, near-optimal, or alternative solutions for evaluation.

This paper introduces a modeling tool that is a part of the Request Oriented Scheduling Engine (ROSE) system being designed at NASA for the scheduling of payload space activities on board the International Space Station. The resident modeler within the ROSE system provides a robust method for easily representing complex sequences of activities for use in planning and scheduling activities. Although ROSE is directed toward space activity scheduling, the services it provides are highly applicable to other areas as well. In the sections to follow, the modeling capabilities developed for ROSE will be introduced and other potential applications for this technology addressed.

ROSE

A request-oriented scheduling engine, better known as ROSE, is under development within the Flight Projects Directorate of the National Aeronautics Space Administration (NASA) for the purpose of planning and scheduling of the activities and resources associated with the payload science experiments to be performed aboard the International Space Station

(ISS). ROSE is being designed to incrementally process requests from payload developers (PDs) to model and schedule the execution of their science experiments on the ISS. The novelty of the approach comes from its web-based interface permitting the PDs to remotely define their request via the construction of a graphical model to represent their requirements. Additional information on the specifics of ROSE can be found in [1, 2].

Given the wide variety of experiments that can be performed and the complex nature of such experiments, the ROSE modeler must provide a general set of primitives that can be used to represent the tasks of an experiment and their relationships. This requirement is complicated by the fact that the experiments must share the use of a limited set of resources (labor, equipment, energy, etc.) with many of the tasks bound by temporal constraints (e.g., must occur within a certain time of lift-off, within a certain portion of orbit, etc.). The result is the need to represent not only a sequence of the tasks within an experiment and their precedence relationships, but to also define precedence relationship with other experiments. These relationships within or across experiments may also have a conditional or temporal constraint associated with them. As one can see, such requirements place a heavy burden on the job of modeling such requirements much less scheduling the resulting models.

ACTIVITY AND SEQUENCE MODELING

A principal strength of the current implementation of ROSE is in its modeling capability. ROSE provides a robust method of graphically representing complex requirements for planning and scheduling. These requirements are expressed in terms of activities and sequences of these activities.

Activities

An activity represents a single step in a sequence. Each activity is uniquely identified by an assigned name and further defined by a set of attributes and constraints. The attributes of the activity include a procedure, description, and an optional note. The procedure provides various fields for defining information to be made available directly to the entity (user, program, etc.) that will be performing the task. This attribute provides a means for communicating to the user specific details that will be pertinent to the successful performance of the task. Currently this attributes supports such information as the activity name, its sequence number, an execution note, as well as a detailed description of the task.

In addition to the attribute data, information about an activity's duration and location are used to model basic temporal and physical constraints. Due to the inherent variation that exists when performing a task and the stochastic nature of the environment within which they take place, the duration of an activity is modeled by specification of both a minimum

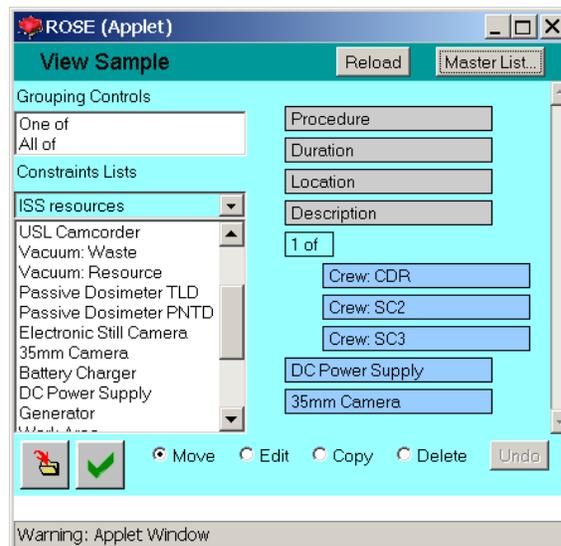


Figure 1. Defining an activity in ROSE.

and maximum time duration. It is also possible (optional) for the user to specify an expected value (referred to as *preferred* in the current version of ROSE) of the duration that lies within these limits. Each of these time values is specified in terms of days/hours:minutes:seconds (i.e., d/hh:mm:ss). If an activity must be performed at some specified place, a location is specified to define this requirement.

Other activity constraints may include the need to use specific resources to support the performance of the activity, or to impose some other conditions such as limiting the activity's execution to a given time of day (i.e., daylight, evening, etc.). ROSE permits the modeler to specify these constraints by choosing from a list of predefined types. Depending on the type of condition or resource selected, the modeler may be required to specify some additional information about its application. As Figure 1 illustrates, the resources for the *view sample* activity were taken from the *ISS resources* list and include the use of a DC power supply, a 35mm camera, and a crew person. Note the box appearing before the listing of the crew. It is a grouping control that indicates that only *one* of the three crew that are listed is required. Both the 'one-of' and 'all-of' grouping controls are available to use repeatedly. Use of these controls enhances the flexibility available for planning and scheduling.

Given that some multiples of some resources exist, ROSE provides a means for the definition and modeling of the use of pooled resources. These may represent such entities as the use of one tool from a set of six, a level of power from a finite source, or some volume of a liquid from a container. As well, ROSE models equipment *modes*. One of the usage concepts for ROSE has the payload developers (PD), who are intimately familiar with the payload but have only peripheral knowledge of the space station infrastructure, generate the scheduling requests. In order to assure that the interface with the station is modeled correctly the PDs and NASA will work together to define the equipment and the resources used by each operational *mode* of each piece of equipment. The PD contributes knowledge about the equipment (how much power) and the NASA expert contributes knowledge about the station (which power bus). Equipment models may use any listed resources including other equipment (in a specific operational *mode*). When an activity requires a piece of equipment, the equipment *mode* is also specified, and the implicitly used resources are automatically used. This approach to modeling is like the real world; an experiment in a laboratory doesn't directly use power, the experiment uses a piece of equipment that uses power.

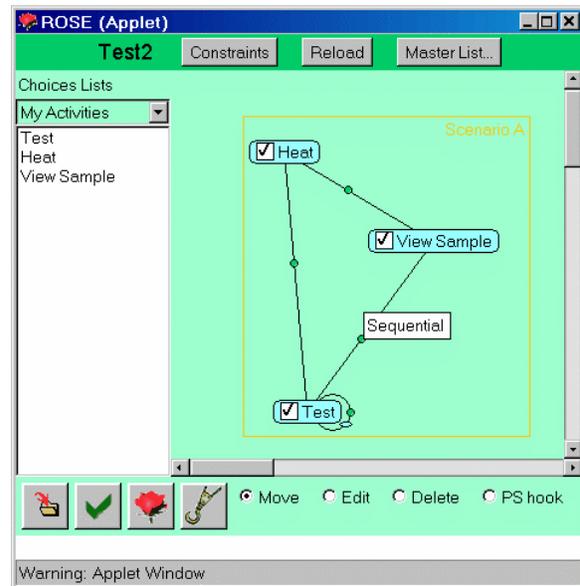


Figure 2. ROSE sequence editor.

Once the activities of a model are defined, a modeler can easily return to the system to edit any of the constraints or conditions listed for an activity via a double click on the item of

interest. The modeler also has the freedom to move and rearrange these elements using the mouse (via the drag and drop capability).

Sequences

Once the activities are defined it is then possible to model the tasks to be performed as sequences of activities. Creating a sequence involves arranging a set of activities into a meaningful procedure for executing them. ROSE provides a sequence editor where a user selects from the list of available defined activities to create the sequence. The relationships between the selected activities are defined by the links a user draws between them. The locations of the activities on the screen are not important to the scheduling engine, but a user should take into account that activity placement does convey a sense of precedence in the mind of a person viewing the resulting sequence. Figure 2 illustrates the user environment for constructing the sequence entitled *Test2*.

The new activities defined by the modeler are listed under the *My Activities* selection of the drop-down box. Other lists in this box that are available to the modeler include *Public Tasks*, *Public Services*, and *My Sequences*. For NASA, the *Public Tasks* and *Public Services* lists represent predefined activities that are common and may be of need to modelers in defining their payload requirements. What they represent is the ability for ROSE to be able to offer, in an organized fashion, a variety of tasks and sequences that are already defined and available to be incorporated into new models. The *My Sequences* are those sequences that have been previously created and can be inserted into another sequence as a subsequence of activities that must be completed. This feature provides the capability to quickly create nested sequences of activities representing the requirements of complex sequences.

Due to the complex nature of the payload related experiments performed on-board the ISS, ROSE was designed to model a large number of complex relationships between sequences and tasks in a sequence. The basic relationships between activities are those that define the temporal relationship. Possibilities provided by ROSE include sequential, during, overlap, and percent coverage. Examples of the first four of these are shown in Figure 3.

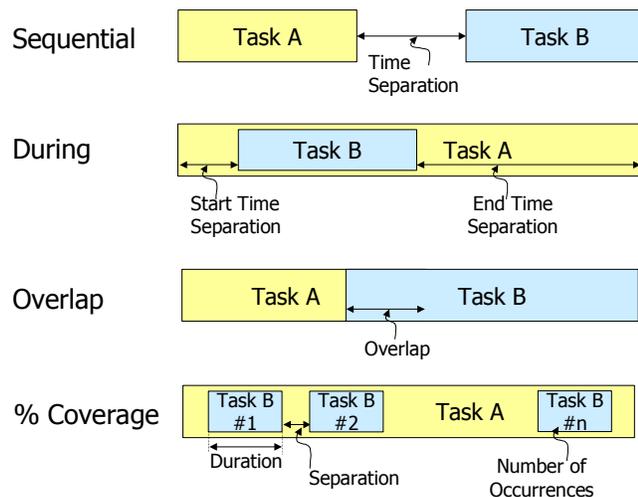


Figure 3. Example of temporal relationships.

The temporal relationship between two activities constrains the scheduling of those activities with respect to one another. The simplest and most common relationship is to require that one task be performed prior to another. This defines a *sequential* relationship between the two tasks. If desired, a user can enter an additional constraint to define the minimum and/or maximum time separation that should be permitted when scheduling the tasks. The *during* relationship describes the situation where a task must be performed

simultaneously while another task is being performed. ROSE permits a modeler to further constraint this relationship by providing minimum and maximum values for either the separation between the start of the two tasks, and/or the separation between the completions of the two tasks. Since it is possible for a modeler to specify variable times (a range of possible values) for a task, the problem is not over defined if the user constrains the separation of both the start and end times. If it is desired that one task only be partially performed while another task is being performed then the tasks are said to have an *overlap* relationship. The amount of overlap can be left blank or specified using minimum and maximum values. The *percent coverage* relationship provides the ability to specify that multiple instances of one task should be scheduled repeatedly during another task to provide a specified minimum of the percent coverage specified. The modeler can further constrain the scheduler by defining one or more of following constraints: the maximum number of times that the one task should be repeated, the minimum duration of the task, and/or the maximum separation between the repetitive tasks.

In addition to these temporal relationships, ROSE provides some additional features. The first is that it is possible to specify an *indeterminate* relationship between two tasks meaning that no temporal relationship exist, but if one task is scheduled the other must be scheduled as well. A second is the ability to specify that a task be repeated. Given this requirement the user will define the minimum and maximum number of repetitions. A *repetition* count relationship is indicated in the diagram as a circle on the side of an activity with a small blue oval. An example of such as activity can be seen in Figure 2 for the Test activity. A *one-to-one* relationship is specified when a task with multiple performances is related to another task with multiple performances and the relationship is to be applied exactly once per pair. A last important feature ROSE provides is the ability to *lock-in* the use of resources from one task to the next. This supports the situation that the resources used by one task match those of the second. This is capability is necessary since it is possible for the modeler to specify that the scheduling engine select from one of several resources to support the activity. If these same resources should be used on a subsequent activity, then a *lock-in* relationship is needed to guarantee this condition.

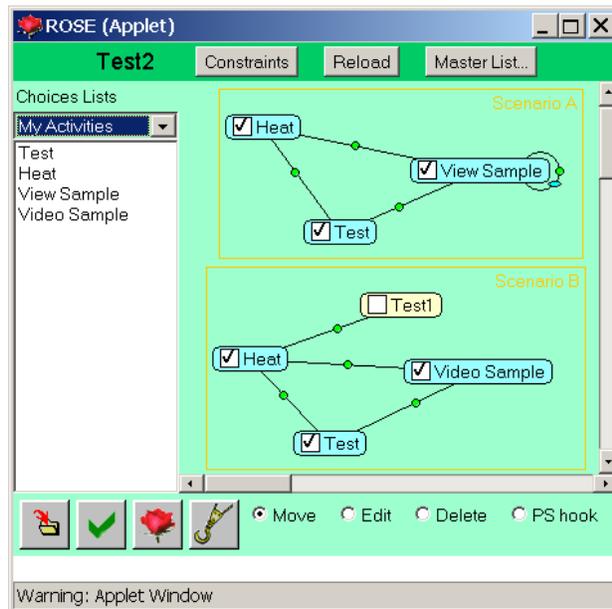


Figure 4. Alternative scenarios for the Test2 sequence.

As is often the case, there is always more than one way to structure a sequence. ROSE provides a mechanism that supports the ability to define alternative sequences. These alternative sequences are referred to as scenarios. All the user has to do is create the additional sequences in the same window (under the same sequence name) as the first sequence and ROSE will automatically label and identify the given sequences as alternative

scenarios (see Figure 4). Providing these alternatives enhances the flexibility available to the planner and scheduler.

Within the window used to define a sequence the user is free to move the activities into any graphical arrangement desired. To edit any relationship only requires a double-click within the circle on the link between the activities. When the mouse cursor is placed over the circle of a link, ROSE brings up a text box defining the relationship (see the box containing “sequential” in Figure 2 for example).

NESTED SEQUENCES

Activities are the primary building blocks for defining sequences. However, ROSE has the capability for the user to embed a sequence as a single step in another sequence. This ability permits a modeler to use ROSE to rapidly construct complex sequences. For example, it is possible to chain together sequences and use the relationships described above to define the constraints that may exist between the action sequences. In addition, a person could also use the scenario feature of ROSE to offer alternatives ordering of those sequences. A sequence is distinguished from an activity in the modeler by the color of the node. As can be seen in Figure 4, an activity node is denoted by its blue color, while the sequence nodes are yellow.

ADDITIONAL FEATURES

One other feature of the modeler in ROSE is the ability to specify what tasks or sequences the planning and scheduling engine should actually schedule. This need is conveyed by the presence of a check mark in the check box that appears on the left side of both the activities and sequence nodes. If a check mark appears in the box, then that activity (or sequence) is to be scheduled when planning and scheduling the sequence. If the box is not checked, then the task is not to be scheduled as part of this sequence and the relationships are applied to an already-scheduled instance of the unchecked task.

As was stated earlier, a modeler accesses the ROSE system through a web-based interface. This ability to use an internet to access the system to construct models provides a tool that surpasses the geographical and temporal constraints of many current methods. This approach provides the benefit of allowing users to construct models from anywhere (even while out of the office), make use of a variety of hardware (as long as it supports popular web browsers), and to make use of the activities and sequences that other have modeled since the activities and sequences are resident on the host.

POTENTIAL APPLICATIONS

Given the robust modeling provided by ROSE to express activities and sequences and its future ability to generate schedules from these models, there was an interest in determining what other potential applications would benefit from this tool. Preliminary considerations resulted in the identification of several potential applications worth further examination. These are the scheduling of hospital operating rooms, airline scheduling, new product development planning, construction planning, and manufacturing scheduling. A focused report on each of these can be found in [3] and [4]. Although similarity exist between each

of these applications and that of space activity planning and scheduling, it is believed that the best match would be with new product development planning and construction planning.

The domain of project planning for new product design and development has a lot in common with space activity planning and scheduling (SAPS). In new product development there are often a large number of tasks that must be completed and the desire is to complete these tasks as quickly as possible to reduce the required time to bring a new product to market. As in SAPS, these tasks are constrained by the availability of resources as well as a strong dependency on other tasks expressed using such temporal relations as defined in ROSE. As in ROSE, the resulting product of the planning and scheduling activity for new product development is a timeline.

When considering construction planning (CP) there exists a strong parallel between it and SAPS applications if one equates the planning and scheduling of subcontractor activities in CP with payload developers in SA. Planning and scheduling in both cases requires the coordination of a large number of interdependent activities. Given the nature of construction, these activities are highly interdependent requiring the expression of a variety of the relationships offered in ROSE. Another similarity between the two applications is that as in SAPS where early planning determines what PDs will participate in an increment's activities, early phases in CP involve determining which subcontractors will participate in the project. As well, in both domains it is not unusual for there to be modifications to the schedule as a project matures. In CP applications, rescheduling usually does not involve a change in resource assignments, only changes in the timing of execution. However, in SAPS applications rescheduling may involve a change in resource assignment (i.e., crew, satellite).

SUMMARY

The resident modeler within the ROSE system provides a robust method for easily representing complex sequences of activities for use in planning and scheduling activities. Although ROSE is directed toward space activity scheduling, the services it provides are highly applicable to other areas as well.

REFERENCES

1. Jaap, J., and Davis, E., "Can Customers Schedule Their Own Payload Activities?," Proc. of the 2nd Int. NASA Workshop on Planning and Scheduling for Space, (2000). (Online: <http://payloads.msfc.nasa.gov/ROSE/publications/>).
2. Jaap, J. and Muery, K., "Putting ROSE To Work: A Proposed Application of a Request-Oriented Scheduling Engine for Space Station Operations," Proc. of the Sixth Inter. Symposium on Space Mission Operations and Ground Data Systems (SpaceOps 2000), Toulouse France, (Online: <http://payloads.msfc.nasa.gov/ROSE/publications/>).
3. Usher, J.M., "Other Potential Applications for the ROSE Technology," Technical Report, 7 pp., (2001). (Online: <http://payloads.msfc.nasa.gov/ROSE/publications/usher/other.html>).
4. Usher, J.M., "Preliminary Review of the Airline Scheduling Domain," Technical Report, 4 pp., (2001). (Online: <http://payloads.msfc.nasa.gov/ROSE/publications/usher/airline.html>).