Maximally Expressive Modeling

John Jaap, Elizabeth Davis, and Lea Richardson
Marshall Space Flight Center, Huntsville, AL

Abstract— Planning and scheduling systems organize "tasks" into a timeline or schedule. Tasks are logically grouped into containers called models. Models are a collection of related tasks, along with their dependencies and requirements, that when met will produce the desired result. 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.

1. Introduction

Marshall Space Flight Center has a long history of planning and scheduling manned space missions, including Skylab, Spacelab, Shuttle and the International Space Station. The Ground Systems Department is building on its experience in developing scheduling engines to develop a new scheduling system that is highlighted by a maximally expressive modeling schema.

The current state-of-the-art in modeling methodologies and scheduling engines results in a linear paradigm with knowledge contributed by task experts (scientists, etc.), vehicle experts and scheduling engine experts. This paradigm, depicted in Figure 1-1, requires significant effort and flow-time. The task experts often struggle to enter their requirements using a language that is limited – frequently resorting to notes to fully describe their requirements. The vehicle and hardware experts then convert and augment this knowledge to 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.

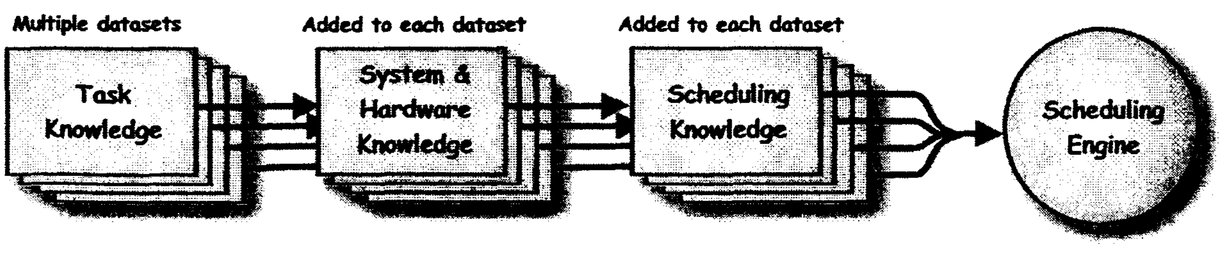


Figure 1-1 — State-of-the-Art Paradigm

This paradigm has begotten the "scheduling cadre," a group of people that 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 combination of an automatic scheduler and a schedule editor (a mixed-initiative approach to scheduling).

The modeling schema (and corresponding scheduling engine) presented in this paper allows a streamlined paradigm as depicted in Figure 1-2. The vehicle experts would enter the system and hardware constraints independently of the task knowledge. The task experts would enter the tasks' requirements. The modeling schema would allow them to specify all of the payload requirements without resorting to notes – 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.

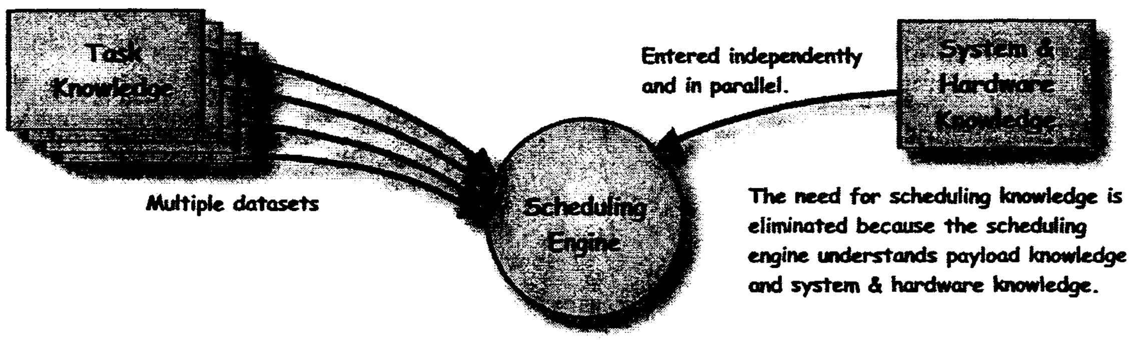


Figure 1-2 – Future Paradigm

The modeling schema described in this paper can significantly reduce the manpower and flow-time required to produce a schedule in a complex scheduling domain.

2. The Schema

Inadequate 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. But modeling even the simplest of tasks cannot be automated because 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 (expressing the requirements in the schema) is a human enterprise – both an art and a science. The modeling schema must be "maximally expressive" so that *all* the requirements can be captured without resorting to notes or awkward representations and the schema must be friendly enough to allow the user to enter the requirements without excessive labor or training. The schema which meets the "maximally expressive" criteria is a synergy of technological advances and domain-specific innovations.

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

The 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. In the schema sequences are actually networks; the term sequence is used for legacy reasons.

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.

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 antenna deploy).

Sequential — Items follow each other; which item starts first is to be defined. 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.

Avoid — An item to be scheduled may not overlap any instance of the avoided item. This constraint is also enforced on not-yet-scheduled instances of the avoided item. Minimum before and after buffers 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.

Fragmentable — When an activity may be broken 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 broken 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.

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.

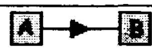
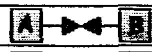
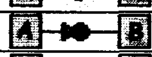
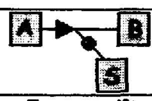
Repeated — An item in a sequence may be repeated; minimum and maximum repetition counts may be specified. When the minimum repetition count is 0, the item is considered optional. The temporal relationship between repetitions can be separated, overlapped, or cyclic (see below).

Cyclic — A frequency can be specified for repeated items. The frequency may be specified in hours, days, or weeks. 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.

Lock-In — If two activities in a sequence contain identical "one-of" selection groups, then specifying lock-in will force the same constraints to 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 performance of this item is

Table 2-1 — Temporal Relationships of the Schema

Relationship	Graphics	Order			Parameters
		A,B	B,A	either	
Sequential		✔	✔		Minimum & maximum separation
Separated				✔	Minimum & maximum separation
Avoid		Implied			Buffer before & after
During		✔	✔		Minimum & maximum start time separation
				✔	Minimum & maximum end time separation
Overlap		✔	✔		Minimum & maximum overlap
				✔	
Percent Coverage		✔	✔		Maximum number of occurrences Minimum duration of an occurrence Maximum separation between occurrences Minimum percent coverage
Fragmentable		✔	✔		Maximum number of fragments Minimum duration of a fragment Maximum duration of an interruption
Standby		Shown with a sequential relationship.			None
Repeated		When the minimum number of repetitions is 0, then the item is optional and the loop is dashed. All repeated relations include a temporal relationship such as separated, overlap or cyclic.			Minimum & maximum number of repetitions
Cyclic					Hourly: Hours between repetitions Time variance
					Daily: Days between repetitions Desired center time with variance
					Weekly: Weeks between repetitions Day(s) of the week Desired center time with variance

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.

Relationships to Internal Items of Embedded Sequences — A sequence can specify that temporal relationships to it are to be applied to an item within itself; this item is said to have a "hook." For example, a sequence to take a video might include video setup, video record, and video downlink activities; the video record would be specified as the item to which relationships are applied. In sequences that embedded the video sequence with a during relationship, the video record activity would be run during the specified activity; setup and downlink would occur before and after the activity to be recorded.

The parameters that may be specified for each temporal relationship are shown in Table 2-1. The temporal relationships defined by the schema are common-sense relationships, not the classical (and sometimes esoteric) temporal relationships. Sequential, separated, during, and overlap can be mapped directly to the classical relationships as shown in Table 2-2 and to Allen's thirteen relationships[2] as shown in Table 2-3. The schema introduces percent coverage, fragmentable, and standby relationships 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.

Relationships to Pre-existing Tasks

Relationships to pre-existing tasks are modeled like relationships between tasks to be scheduled.

Table 2-2 — Schema Relationships Compared to Classical Relationships

Interval	Endpoint Relations	Pictorial	Schema Equivalent
X < Y	$X_E < Y_S$	X Y	sequential: $wn \neq 0$
X meets Y	$X_E = Y_S$	XY	sequential or overlap: $wn = wx = 0$
X overlaps Y	$X_S < Y_S$ and $X_E > Y_S$ and $X_E < Y_E$	X Y	overlap: $wn \neq 0$ and $w < min(X_D, Y_D)$
X during Y	$X_S > Y_S$ & $X_E \leq Y_E$ or $X_S \geq Y_S$ & $X_E < Y_E$	X Y X Y X Y	during: $wn_S + wn_E > 0$
X = Y	$X_S = Y_S$ and $X_E = Y_E$	X Y	during: $wx_S = wx_E = 0$

- S and E indicate the start and end.
- wn and wx indicates the minimum and maximum separations (waits).
- D indicates the duration.
- All times are non-negative.

Table 2-3 — Schema Relationships Compared to Allen's Relationships

Allen's Relation	Symbol	Endpoint Relations	Pictorial	Schema Equivalent
X before Y	<, b	$X_E < Y_S$	X Y	sequential (X before Y): *wn* ≠ 0
X meets Y	m	$X_E = Y_S$	X Y	sequential or overlap (X before Y): *wn* = *wx* = 0
X overlaps Y	o	$X_S < Y_S$ & $X_E > Y_S$ & $X_E < Y_E$	X Y	overlap (X before Y): *wn* ≠ 0, $w < min(X_D, Y_D)$
X starts Y	s	$X_S = Y_S$ & $X_E < Y_E$	X Y	during (X during Y): $wn_S = 0$, $wn_E \neq 0$ or overlap (X before Y): *wn* ≠ 0, $w = X_D, < Y_D$ [1]
X during Y	d	$X_S > Y_S$ & $X_E < Y_E$	X Y	during (X during Y): $wn_S \neq 0$, $wn_E \neq 0$
X finishes Y	f	$X_S > Y_S$ & $X_E = Y_E$	X Y	during (X during Y): $wn_S \neq 0$, $wn_E = 0$ or overlap (Y before X): *wn* ≠ 0, $w = X_D, < Y_D$ [2]
X equal Y	=	$X_S = Y_S$ & $X_E = Y_E$	X Y	during: $wx_S = wx_E = 0$ or overlap: *wn* ≠ 0, $w = X_D, = Y_D$ [3]
X finished by Y	fi	$X_S < Y_S$ & $X_E = Y_E$	X Y	during (Y during X): $wn_S \neq 0$, $wn_E = 0$ or overlap (X before Y): *wn* ≠ 0, $w = Y_D, < X_D$ [4]
X contains Y	di	$X_S < Y_S$ & $X_E > Y_E$	X Y	during (Y during X): $wn_S \neq 0$, $wn_E \neq 0$
X started by Y	si	$X_S = Y_S$ & $X_E > Y_E$	X Y	during (Y during X): $wn_S = 0$, $wn_E \neq 0$ or overlap (Y before X): *wn* ≠ 0, $w = Y_D, < X_D$ [5]
X overlapped by Y	oi	$X_S > Y_S$ & $X_S < Y_E$ & $X_E > Y_E$	X Y	overlap (Y before X): *wn* ≠ 0, $w < min(X_D, Y_D)$
X met by Y	mi	$X_S = Y_E$	Y X	sequential or overlap/ (Y before X): *wn* = *wx* = 0
X after Y	>, a	$X_S > Y_E$	Y X	sequential (Y before X): *wn* ≠ 0

- S and E indicate the start and end.
- wn and wx indicates the minimum and maximum separations or overlaps
- D indicates the duration.

1. Overlap becomes *starts* when the overlap = the duration of the shorter earlier item.
2. Overlap becomes *finishes* when the overlap = the duration of the shorter later item.
3. Overlap becomes *equal* if the overlap and both durations are equal.
4. Overlap becomes *finished by* when overlap = duration of the shorter later item.
5. Overlap becomes *started by* when overlap = duration of shorter earlier interval.

Relationship Rules

Some of the relationships require that one of the items pre-exist, some relationships allow one of the items to pre-exist, and some relationships can only exist between to-be-scheduled tasks (see Table 2-4).

Table 2-4 — Relationship Rules for Pre-Existing Items

Relationship	Pre-Existence Rule
Avoid	One item must pre-exist
Either item in a fragmentable relationship, a repeated item, a standby item, either item in a percent-coverage relationship	Neither item can pre-exist
Sequential, Separated, During, Overlap	Only one item may pre-exist

Some relationships only apply to activities (see Table 2-5).

Table 2-5 — Relationship Rules for Activities

Relationship	Activity Rule
Fragmentable	The item to be fragmented must be an activity
Percent-Coverage	Both items must be activities
Sequential, Separated, During, Overlap, Repetition, Standby	No limitation

Additional Constraints on a Sequence

In addition to relationships, other temporal constraints can exist on sequences. A minimum and maximum separation between performances can be specified. This separation applies to the separation between to-be-scheduled performances and pre-existing performances. A minimum and maximum duration can be specified. A maximum duration is used when the aggregate flexibility of the sequence (network) would allow the sequence to be longer than desired. Multiple performance windows may be specified. These windows would limit when a sequence can be scheduled; examples, during the month of March, or during the third week of an expedition.

Graphical Interfaces

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

Hierarchies of groups of requirements best describe the constraints of most non-trivial activities. The outline interface is well-suited to modeling hierarchies of groups because it can be manipulated by a drag-and-drop interface and nested to any depth without ambiguity. Figure 3-1in the example discussed in Section 3 illustrates the use of the outline hierarchy. Constraints are selected from a list, added to the hierarchy canvas. A dialog box is used to fill in the mode or values.

Sequences use a "network" interface to define the relationships between tasks (activities and embedded sequences). The user selects from a list of task, placing the item onto a canvas. The user then positions the items as needed, creates relationships by connecting them together and differentiates between relationships to pre-existing tasks and to-be-scheduled tasks. A dialog box is used to specify which relationship and its details. Table 2-1 shows the parameters for each relationship and Figure 2-1 shows the features of an item that can be set by the user. Figure 2-2 shows the visual cues attached to a sequence embedded in the displayed sequence. The example in Figure 3-1 has several embedded sequences.

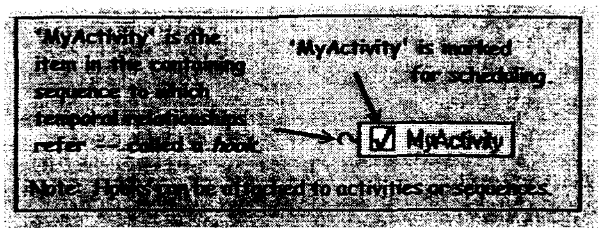


Figure 2-1 — Features of an Items in a Sequence

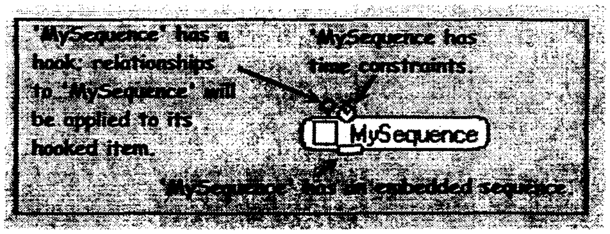


Figure 2-2 — Indicators on an Embedded Sequence

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 interface 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, 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.

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 timing in relationships, 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.

3. An Example

Payload Overview — The hypothetical 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 a 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. The models required to represent this experiment are discussed below beginning with a global model and then the details.

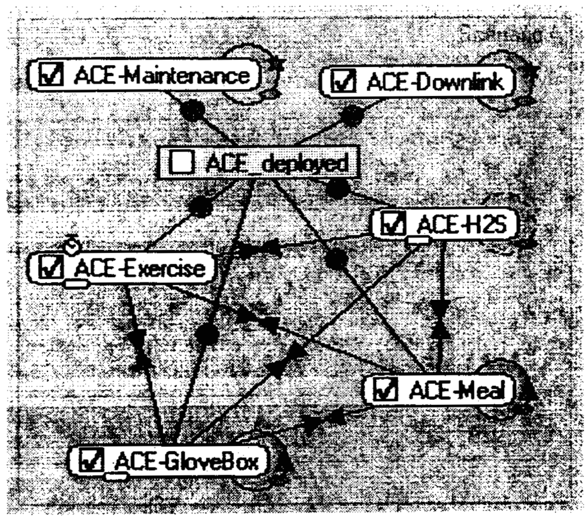


Figure 3-1 — The Main Model for ACE

The Sequence Model — The model shown in Figure 3-1 depicts the relationships of the various sequences of the ACE experiment to one another. In the sequence above, the ACE-Exercise, ACE-H2S, ACE-Meal and ACE-GloveBox tasks are separated (not allowed to overlap). The ACE-H2S sequence is optional, and the ACE-Meal, ACE-GloveBox, ACE-Maintenance and ACE-Downlink sequences are repeated multiple times, either with a cyclic relationship or a sequential relationship. It also shows that these tasks are to be done while the hardware is deployed (the ACE_deployed activity is not checked for scheduling, but is scheduled by another sequence.) The embedded sequence, ACE-H2S, is shown in Figure 3-2. This sequence contains two scenarios or

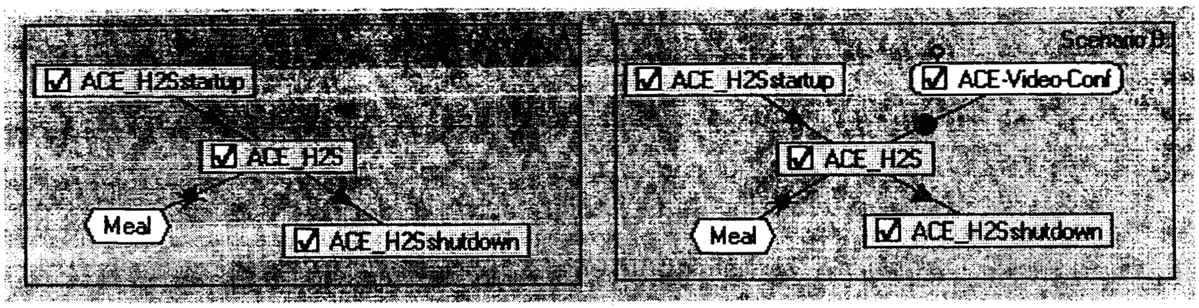


Figure 3-2 —ACE-H2S Model Showing Scenarios

alternate ways of accomplishing the tasks: one with a video conference and one without. It also has an avoid relationship with Meal. The video conference sequence is shown in Figure 3-3. ACE-Video-Conf has a hook on the ACE_video activity indicating that relationships to this sequence are treated as relationships to the ACE_video activity.

The Activity Model — The model for ACE_video-setup is shown in Figure 3-4. This activity shows a hierarchy of constraints; if the activity is done by Crew: SC2, the duration is 20 minutes; if done by Crew: SC3, the duration is 15 minutes. It also uses the USL Camcorder in checkout mode.

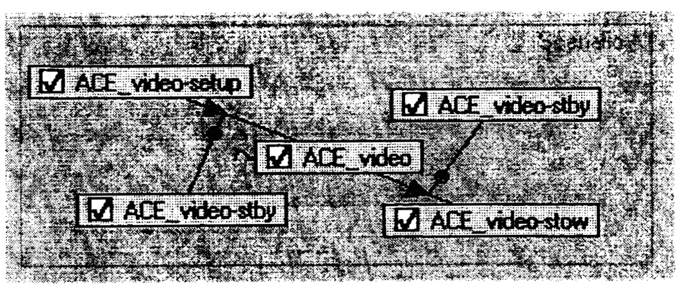


Figure 3-3 — ACE-Video-Conf Model

The Mode Model — The model (not shown) for USL Camcorder equipment has five modes: *setup* which uses no resources, *checkout* which uses no resources, *record* which uses no resources, *battery charge* which uses power, and *download* which uses power and the video channel (all modes implicitly use the camcorder itself and all have different procedures). Mode models may contain hierarchies of requirements similar to activity models.

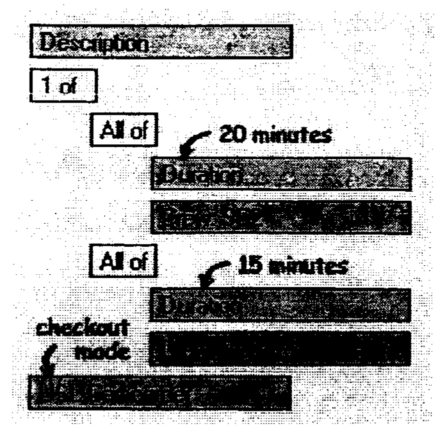


Figure 3-4 —ACE-video-setup Mode

4. Summary

The modeling schema presented in this paper is one component of a system that could enable new scheduling paradigms. The other components include a scheduling engine commensurate with the modeling [3]. An operations concept based on the modeling schema and scheduling engine has been proposed [4].

The modeling schema is simultaneously robust (can represent complex requirements) and easy to use (relies on everyday terminology, constructs, and interfaces). 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 can be directly represented. The schema is truly "maximally expressive."

References

[1] J. Jaap, P. Meyer and E. 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). online...

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

[3] J. Jaap and E. Davis, "An Enabling Technology for New Planning and Scheduling Paradigms," (SpaceOps 2004, 17-21 May, 2004) online...

[4] J. Jaap and K. Muery, "Putting ROSE to Work: A Proposed Application of a Request-Oriented Scheduling Engine for Space Station Operations," (SpaceOps 2000, 19-23 June, 2000) online...