The Mission Operations Planning Assistant

James G. Schuetzle

Ford Aerospace & Communications Corporation
Space Station Programs
4920 Niagara Road
College Park, Maryland 20740

ABSTRACT

The Mission Operations Planning Assistant (MOPA) is a knowledge-based system developed to support the planning and scheduling of instrument activities on the Upper Atmospheric Research Satellite (UARS). The MOPA system represents and maintains instrument plans at two levels of abstraction in order to keep plans comprehensible to both UARS Principal Investigators and Command Management personnel. The hierarchical representation of plans also allows MOPA to automatically create detailed instrument activity plans from which spacecraft command loads may be generated.

The MOPA system was developed on a Symbolics 3640 computer using the ZetaLisp and ART languages. MOPA's features include a textual and graphical interface for plan inspection and modification, recognition of instrument operational constraint violations during the planning process, and consistency maintenance between the different planning levels. This paper describes the current MOPA system.

INTRODUCTION

The UARS is a multi-instrument orbiting observatory scheduled to be launched by the Space Shuttle in 1991. The UARS will provide experimenters at remote locations with data on the temperature, composition, and dynamics of the earth's upper atmosphere.

Mission planning for a satellite such as the UARS is a complex and knowledge intensive process. There are ten instruments whose activities must be defined and coordinated daily for the estimated 18 month life of the mission. The mission planners need to be cognizant of the functions and capabilities of each instrument as well as the spacecraft itself. In addition, there are constraints and interdependencies among the instruments themselves and between the instruments and the spacecraft. Mission planners are also required to reason at different levels of abstraction during the process of translating high level descriptions of instrument activity into detailed command sequences.

BACKGROUND

The instrument planning scenario for the UARS is divided into three phases with each phase resulting in the formulation of a different plan. The three plans (in chronological order) are the Long Term Science Plan, the Daily Science Plan, and the Activity Plan.

The Long Term Science Plan is developed by the UARS instrument scientists (principal investigators) prior to the launch of the spacecraft. This plan describes how each of the instruments will be utilized in order to achieve the scientific objectives of the mission. After UARS launch and instrument checkout, the principal investigators will update the Long Term Science Plan to reflect changes in instrument performance or to include the study of new items of scientific interest. It is expected that this plan will change infrequently.

The Mission Planning group, based at GSFC, is responsible for developing a Daily Science Plan for each day of the UARS mission. This plan is developed using the operational strategy developed in the Long Term Science Plan. The Daily Science Plan specifies the operations of each instrument in terms of scientific objectives.

The final plan, the Activity Plan, is also developed daily by the Mission Planning Group. The Activity Plan specifies the operation of the instruments in terms of the detailed activities that each instrument will perform. These instrument activities at are at a level of detail suitable for the generation of spacecraft commands. This command generation is performed by the UARS Command Management System using the Activity Plan as input.

PROBLEM

The problem addressed by the MOPA system is to support the Mission Planning Group in the process of generating Daily Science and Activity Plans. In order to create these plans, the Mission Planning Group requires some of the knowledge possessed by the principal investigators. This knowledge falls into three general categories:

1. Instrument operations to achieve scientific objectives

   Each instrument can operate in different modes depending on the particular objective to be satisfied. For example, most of the instruments have several modes for data collection, calibration, and safing. The Mission Planning Group must know these modes and the conditions under which they should be selected in order to achieve the scientific objectives of the mission.

2. Instrument operational limitations and restrictions

   There are certain operating constraints placed upon the instruments. The improper operation of an instrument can result in damage to itself or another instrument, or in degradation of the data being captured by the instrument. The Mission Planning Group must be aware of these restrictions during the process of generating Daily Science and Activity Plans.

3. Translation of scientific objectives into detailed command sequences.

   The process of creating the Activity Plan from the Daily Science Plan requires that the Mission Planning Group decompose activities specified in terms of scientific objectives into the detailed activities which must be executed by the instrument to carry out the objective.

MOPA CAPABILITIES

The MOPA system provides many features to aid the mission planner in the creation of the various plans. This section provides an overview of MOPA features.

Generic Plans

In the MOPA system, the Principal Investigators knowledge of instrument operations to achieve scientific objectives is represented in the form of Generic Plans. MOPA Generic Plans attempt to capture much of the information present in the Long Term Science Plan. These plans are specified at such a level of abstraction as to be readily interpretable by the UARS Principal Investigators. The Generic plan specifies for each instrument, the conditions under which an instrument operation should be performed. Typically the conditions are expressed in terms of spacecraft orbital events such as sunrises or equator crossings. Additionally, these conditions may be expressed in terms of the operations of the spacecraft or other instruments. The instrument operations specified in the Generic plan identify abstract functions, usually scientific objectives, to be accomplished by the instrument. MOPA provides the mission planner with the ability to choose from a collection of previously created Generic Plans, selecting one that may be appropriate for the particular planning day.

Daily Science Plan generation

The MOPA system uses the selected Generic Plan together with orbital and other events scheduled for the day in order to automatically generate a Daily Science Plan. The Daily Science Plan is graphically represented by the MOPA interface in a timeline format. MOPA can also, at the option of the user, resolve conflicts between instrument operations in the plan. The user interface provides the planner with the ability to quickly view and modify the Daily Science Plan created by MOPA.

Activity Plan generation

Once the Daily Science Plan has been generated and reviewed, MOPA can automatically derive the Activity Plan from it. The Activity plan can then be displayed and modified through the user interface in either a graphical or textual manner. As with the Daily Science Plan, MOPA can resolve any conflicts between instrument operations.

Constraint checking

MOPA provides the capability to constraint check instrument operations specified in either the Daily Science or the Activity Plans. The MOPA system can represent and check a wide variety of operational constraints.

User Interface

The main purpose of the MOPA user interface is to provide the user with mechanisms to create, examine, and modify Daily Science and Activity Plans. MOPA has two different types of displays for these plans; A textual display and a graphical timeline display. MOPA reduces the amount of typing the user must perform by making extensive use of the mouse. Typical uses of the mouse are to select commands from menus, manipulate instrument activities and events, and scroll through displays.

Reverse Plan Maintenance

The Reverse Plan Maintenance feature of MOPA allows the user to make modifications to either the Activity Plan or the Daily Science Plan and "reflect" these changes back to the high level plan for the day, the Generic Plan. The "tailored" Generic Plan can then be saved for later use. This feature saves the user from having to edit the Generic Plan and then regenerate both the Daily Science and Activity Plans.

USER INTERFACE

In order to take advantage of the Symbolics windowing system features and object oriented design methodology, we have implemented the user interface portion of MOPA entirely in ZetaLisp. This design decision requires us to map ART schemata into ZetaLisp flavor objects which can then be used by the user interface subsystem. This mapping has been made almost transparent by using the flavors facility of ZetaLisp. We have defined a flavor called LART (Lisp to ART) from which all references to ART schemata from Lisp are made. Using the LART flavor, we can attach procedures (methods) to the schemata to perform such functions as accessing and updating attributes, and drawing activities and events on the screen.

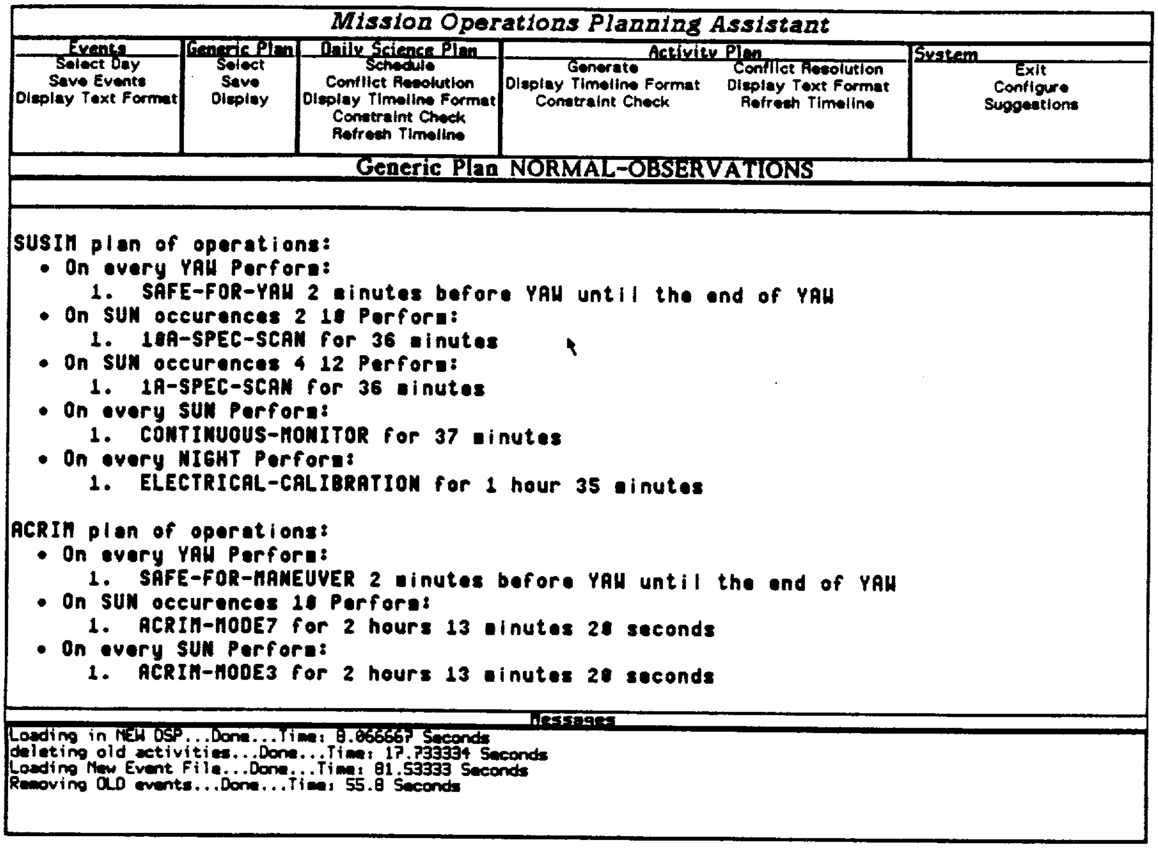

Figure 1

Figure 1 illustrates the MOPA display of a sample Generic Plan. The menus at the top of the screen allow the user to select different MOPA functions by pointing the mouse at the items within the menus. The menus are divided into functions pertaining to Events, Generic Plans, Daily Science Plans, and Activity plans. A System menu is available to modify certain system parameters and to enter suggestions or comments about experiences with the MOPA system.

At the center of the MOPA screen is a sample Generic Plan. Within the Generic Plan are entries for each UARS instrument which describe when certain instrument operations should be performed. This display may be vertically scrolled using the mouse cursor. At the bottom of the screen is a window area for informational messages. The messages within this display may also be scrolled using the mouse cursor.

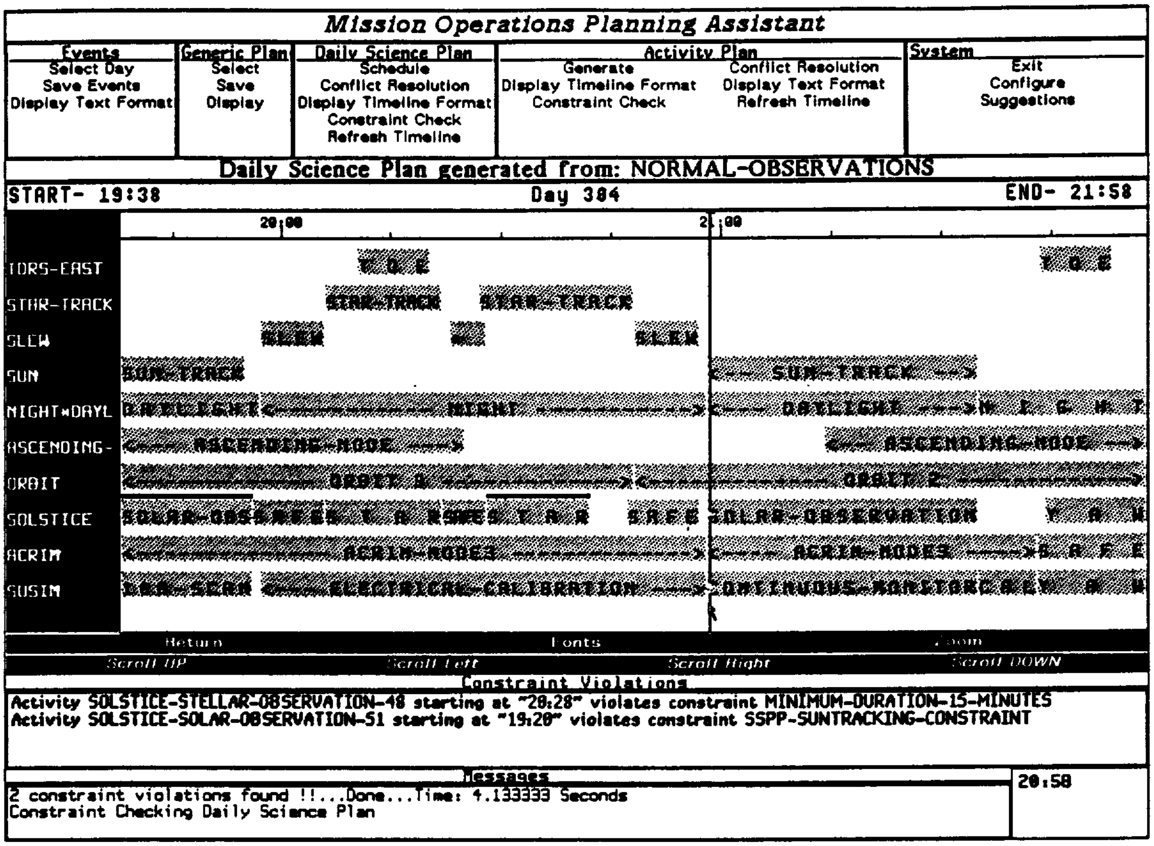

Figure 2

A timeline display of a Daily Science Plan is shown in figure 2. At the top of the display is a title which indicates the plan level and the name of the plan. On the next line are displayed the start time of the interval being displayed, the Julian date, and the interval end time. The interface provides both horizontal and vertical scrolling of the display. Horizontal scrolling can be accomplished in several ways according to user preference. The character font for the display may also be adjusted. Each timeline in the display can also be manipulated. The user can rearrange timelines, delete them from the display, or merge two timelines together. By pointing the mouse cursor over activities or events, the user can obtain a menu of several actions that may be performed; delete, edit, and replace. Constraint violation messages appear in the display below the timeline display. Activities with constraint violations are highlighted in the timeline display by drawing a thick line above them. The constraint violation messages have functions associated with them; the user may either display the constraint structure itself, or locate the activity with the violation.

Mission Operations Planning Assistant

Events	Generic Plan	Daily Science Plan	Activity Plan	System
Select Day Save Events Display Text Format	Select Save Display	Schedule Conflict Resolution Display Timeline Format Constraint Check Refresh Timeline	Generate Display Timeline Format Constraint Check Conflict Resolution Display Text Format Refresh Timeline	Exit Configure Suggestions

Activity Plan for day 304

Start Time	Instrument	Name	End Time	Priority
19:57	SOLSTICE	WAIT	20:06	4
19:57	SUSIM	ELECTRICAL-CAL*	20:58	2
20:06	SOLSTICE	OPEN-APERTURE-*	20:07	3
20:07	SOLSTICE	ADJ-WAVE-LENGTH	20:21	3
20:22	SOLSTICE	CLOSE-APER-DOOR	20:23	4
20:23	SOLSTICE	WAIT	20:28	4
20:28	SOLSTICE	OPEN-APERTURE-*	20:29	3
20:29	SOLSTICE	ADJ-WAVE-LENGTH	20:41	3
20:47	SOLSTICE	CLOSE-APER-DOOR	20:48	4
20:48	SOLSTICE	WAIT	20:57	4
20:58	ACRIM	ACRIM-MODE3	21:43	3
20:58	SOLSTICE	SUN-OBS-START	21:34	3
20:58	SUSIM	CONTINUOUS-MON*	21:35	3
21:34	SOLSTICE	CLOSE-APER-DOOR	21:35	3
21:35	SUSIM	ELECTRICAL-CAL*	21:43	2
21:43	ACRIM	SHUTTERS-CLOSE	21:45	5
21:43	SUSIM	CLOSE-DOOR	21:44	5
21:44	SOLSTICE	CLOSE-APER-DOOR	21:45	5
21:44	SUSIM	POWER-DOWN	21:45	5
21:45	ACRIM	WAIT	11/1 00:16	5

Messages

73 activities created
Creating Activity Plan from DSP...Done...Time: 140.4 Seconds
2 constraint violations found !!...Done...Time: 4.266667 Seconds
Constraint Checking Daily Science Plan
2 constraint violations found !!...Done...Time: 4.133333 Seconds
Constraint Checking Daily Science Plan

<u>Figure 3</u>

The Activity Plan can be displayed in a textual format as shown in figure 3. The attributes of the activities are displayed at the top of the display and their values for each activity are displayed below. Before the display appears, the user is presented with a menu of all activity attributes. The user may then select those attributes which he would like displayed as well as the order he would like them displayed in. In this way, the activities can be presented to the user sorted by any activity attribute, for example by start time or instrument. This display may be scrolled vertically using the mouse. Each activity entry in the display is mouse sensitive. The user can delete, add, or edit an activity simply by pointing the mouse over the activity.

KNOWLEDGE REPRESENTATION

There are several classes of entities which MOPA must reason about; Events, Activities, Instruments, Plans, and Constraints. We have chosen to represent all of these entities using ART's frame based representation facility referred to as schemata. In this section we discuss the representations of two important entities: instrument activities and constraints.

Activities

Central to the design of MOPA is the representation of instrument activities. Figure 4 illustrates a sample instance of an activity. All activity schemata have certain basic attributes associated with them. These attributes include the activity start time, end time, and duration. An activity is linked to the instrument which performs it via the activity-of relation. There are two relations of an activity which relate the activity to constraints. The has-pre-condition relation has as its value the constraint schema which limit the performance of the activity. The has-constraint-violation relation is used during the constraint checking process. The value of this relation is either NIL, meaning this activity violates no constraints, or the names of the constraint schema that this activity violated.

Activity Abstraction Hierarchies

Activities are represented at different levels of abstraction in the Daily Science plan and Activity plan. In the Daily Science Plan, the activities to be performed by an instrument are specified in terms of the scientific objective to be obtained. At the Activity Plan level, the activities are the detailed operations necessary to carry out the objective. MOPA's representation of activities allows for the construction of activity hierarchies to provide for both levels of specification.

For example, one function of a particular instrument might be to collect data about carbon dioxide level at some point in the earth's atmosphere. In order to make the collection of this type data, the instrument is powered on, a filter of the instrument must be set to a certain position, the instrument's microprocessor must cycle in this configuration for some duration, and then the instrument is powered off at completion. We might define a hierarchy of activities in which "Collect CO2 Data" could be an abstract activity composed of the primitive activities power-on, set-filter, cycle, and power-off. The planner could then specify "Collect CO2 Data" as an activity to be performed by the instrument in the Generic or Daily Science plans.

Two relations within the activity schema facilitate the construction of activity abstraction hierarchies: has-sub-activity and sub-activity-of. The has-sub-activity relation links an activity to its detailed components, the sub-activity-of relation provides a link in the opposite direction. MOPA allows these activity hierarchies to be constructed in an arbitrarily complex manner, with as many levels of activities as are desired. Different abstraction hierarchies may also share common components if necessary.

The activity shown in Figure 4 has a link to three constraint schemata via the has-pre-condition relation. This activity is associated with two more detailed activities via the has-sub-activity relation.

Activity instance schema

```
(defschema solstice-solar-observation
  (is-a activity)
  (name solar-obs)
  (short-name sun)
  (has-pre-condition minimum-duration-10-minutes
                     maximum-duration-38-minutes
                     sspp-suntracking-constraint)
  (activity-of solstice)
  (duration 9000)
  (has-sub-activity (sun-obs-start
                     sun-obs-stop)))
```

Figure 4

Constraints

The activities of the UARS instruments are often restricted by operational limitations called constraints. In MOPA, we define a constraint as a condition which must be true in order for an activity to be performed. Constraints, represented as ART schemata, are linked to the activities which they constrain via the has-pre-condition relation. These constraints take a variety of forms. This section describes the classification of constraints used by MOPA and their representations.

The schema definition of a generic constraint is shown in Figure 5. Each constraint has a severity attribute associated with it. The value of this attribute is an indication of the severity of violating this constraint.

In the MOPA prototype the two possible values for this attribute are LOSS-OF-DATA and INSTRUMENT-DAMAGE. The result attribute indicates the specific result of violating the constraint. The description attribute of the constraint is a text representation of the error message given the user if the constraint is violated.

Generic Constraint schema

```
(defschema constraint
  (severity)
  (result)
  (constraint-type)
  (description))
```

Figure 5

Constraint Classification

We have identified two major classes of constraints: value constraints and relational constraints. The value type of constraint, shown in Figure 6, restricts the range of values that an attribute of an activity schema can have. This type of constraint is often used in MOPA to enforce maximum and minimum durations of an activity. The slot-name attribute specifies the attribute of the activity whose value is to be restricted. The restriction-value attribute identifies the constraining value of the attribute. The predicate-name attribute specifies a Lisp predicate to be invoked to compare the restriction-value to the value of the activity slot-name attribute.

Slot Value Constraint Class schema

```
(defschema slot-value-constraint
  (is-a constraint)
  (slot-name)
  (restriction-value)
  (constraint-type slot-value-constraint)
  (predicate-name))
```

Figure 6

The other general class of constraint, the relational constraint, is shown in Figure 7. This class of constraint specifies a relation which must hold between two activities. For example, the relational constraint allows us to specify the constraint that one activity must occur during another activity.

The predicate-name attribute of the relation-constraint schema has the same meaning it did for the value constraint. The schema-name attribute of the relation-constraint schema, points to an activity schema. The relation defined in the predicate-name attribute must be true between the schema-name attribute value, and the activity to which this constraint is attached.

Relational Constraint Class schema

```
(defschema relation-constraint
  (is-a constraint)
  (schema-name)
  (predicate-name))
```

Figure 7

An example of a relational constraint used in MOPA is the sspp-suntracking constraint shown in figure 8. This constraint is attached to activities such as Solstice-solar-observation to indicate that the SSPP must be in Sun tracking mode during these activities. In this case, the schema-name attribute of the sspp-suntracking-constraint schema has the value sun-track, which is the SSPP Sun Tracking event. The predicate-name value is overlaps, a predicate which determines if one activity or event overlaps another in time.

Relational Constraint instance schema

```
(defschema sspp-suntracking-constraint
  (instance-of relation-constraint
  (schema-name sun-track)
  (predicate-name overlaps)
  (severity loss-of-data)
  (result "Poor data quality")
  (description "This activity requires the SSPP
               to be in sun-tracking mode"))
```

Figure 8

DESIGN

The section discusses the design of two important processes in MOPA; The generation of Activity Plans from Daily Science Plans, and the constraint checking process. These two processes operate on the activity and constraint schemata representations described in the previous section.

Activity Plan Generation

The MOPA prototype system automatically generates an Activity Plan from a Daily Science Plan. This process involves a decomposition of abstract activities into their detailed component activities. The generation process contains two main components: Activity Decomposition and Activity Instantiation.

Activity Decomposition - The Activity Plan generation process begins with a collection of scheduled activities at the Daily Science Plan level. The activities specified at this level may have has-sub-activity links to other more detailed activities, or they may be "primitive" activities (i.e. They have no has-sub-activity links).

The generate process iterates through a list of all activities specified at the DSP level. Each activity is processed individually. If the activity is "primitive", then decomposition is unnecessary, and the activity is added to the list of activities at the Activity Plan level. If the activity is linked to more detailed activities via the has-sub-activity relation, the decomposition process performs a recursive depth-first search of the activity hierarchy. This search collects a list of activities found at the "bottom" or lowest level of the activity hierarchy. This list of activities will be included in the list of Activity Plan level activities after each new activity is instantiated.

Activity Instantiation - If the decomposition process is performed, a list of detailed activities is created. These detailed activities are the schema names of the generic activities. The generic activities provide default values for the activities which are actually instantiated (scheduled). The Activity Instantiation process creates instances of these generic activities, filling in specific values for attributes of the particular instance. The attributes whose values must be assigned are: start-time, duration, end-time, sub-activity-of, priority, and has-event.

Constraint Checking

There are three components involved in the checking process: Generic constraint processing, Constraint schema definitions, and Constraint evaluation functions. The constraint schema definitions were described in the previous section on Knowledge Representation.

Generic Constraint Processing - Constraint checking can be performed at either the Daily Science or the Activity Plan levels. In the MOPA user interface, these two options are selectable from the menus at the top of the MOPA displays. MOPA maintains separate lists of activities created at each of the two plan levels. When constraint checking is selected, MOPA examines only those activities created at the appropriate plan level.

The Generic Constraint processing functions iterate through activities examining their has-pre-condition attribute. If the value of this relation is not NIL, MOPA proceeds to evaluate each constraint attached to the activity. As described in the Knowledge Representation section, MOPA has two general classes of constraints: Value and Relational. MOPA checks the constraint type of the constraint and invokes one function to evaluate Value constraints, and another to evaluate Relational constraints. The arguments to these functions are the constrained activity and the constraint itself.

The evaluation of the Value constraint types is fairly straightforward. The predicate-name, slot-name, and restriction-value values are first retrieved from the constraint schema. Then, using the slot-name value, the appropriate attribute value from the activity schema is retrieved. Finally, the predicate-name function is invoked with the attribute value of the activity and the restriction-value as arguments. If the function returns the value NIL (false), then a constraint violation is detected and a function is invoked to attach the constraint schema to the has-constraint-violation attribute of the activity. A message is also sent to the Constraint Violations window.

The Relational constraint evaluation process is similar to the evaluation of the Value constraints. The difference between the two lies in the method of invoking the predicate-name function. The semantics of the Relational constraint type indicates that the specified relation must hold between the constrained activity, and at least one occurrence of the activity specified in the constraint. For example, the sspp-suntracking-constraint specifies that the solstice-solar-observation activity can only occur while the SSPP is in sun-tracking mode. In the schema representing this constraint, the schema-name attribute has the value sun-track. The Relational constraint evaluation function finds all occurrences of the sun-tracking activity, and applies the predicate-name function to each activity. If the predicate-name function returns T for any of the occurrences, then the constraint is satisfied and no violation is detected. If no occurrence satisfies the constraint (i.e. returns T) then a constraint violation is detected and the constraint notification process begins.

CONCLUSION

The Mission Operations Planning Assistant has proven the applicability of the knowledge-based approach to mission planning. Several important concepts were demonstrated in the prototype; Activity abstraction hierarchies to facilitate multi-level planning , Reverse Plan maintenance, a general mechanism for the representation and evaluation of constraints, and contextual activity priorities.

MOPA has also demonstrated the effectiveness of a frame-based knowledge representation capability. In the early stages of the development of MOPA, we had implemented a large portion of the prototype using a rule-based approach. We discovered that this approach had several weaknesses: lack of structure, difficult to test and verify, difficult to maintain, and it hinders generic development. We believe the frame-based approach used in MOPA is key to developing maintainable and deliverable systems.

ACKNOWLEDGEMENTS

Thanks are due to NASA Goddard Space Flight Center for supporting this work; and to Robert Brekke, Edward Connell, Carolyn Dent, Larry Hull, Allan Jaworski, Patricia Lightfoot, William MaCoughtry, and Dorothy Perkins, for their support and guidance.

References

1. M. Fox, "Constraint Directed Search: A Case Study of Job-Shop Scheduling". Ph.D. Thesis.,Computer Science Dept, Carnegie Mellon University, Pittsburg, PA 15213, 1983.

2. W. Gevarter, "Artificial Intelligence". Noyes Publications, 1984.

3. I. Goldstein and B. Roberts. "NUDGE, A Knowledge-based Scheduling System". The Fifth Internation Joint Conference on Artificial Intelligence, IJCAI, 1977.

4. Inference Corporation, "ART Reference Manual Version 2.0". Inference Corporation, 1986.

5. J. King, "RPMS: Resource Planning and Management System". JAI PCC 1984.

6. J. Schuetzle and D. Zoch, "A Hierarchical Planning System for Mission Support". Proceedings of the IEEE Expert Systems in Government Symposium. 1986.

7. Symbolics Inc. "Reference Guide to Symbolics-Lisp". Symbolics Inc. 1985.

8. D. Zoch, "Maintaining Consistency Between Planning Hierarchies: Techniques and Applications". Proceedings of the NASA Conference on Artificial Intelligence Applications. 1987.