N88-16427

The Resource Envelope as a Basis for Space Station Management System Scheduling

by
Joy Bush and Anna Critchfield
Computer Sciences Corporation, System Sciences Division
4600 Powder Mill Road
Beltville, Maryland 20705

Abstract

This paper describes the Platform Management System (PMS) Resource Envelope Scheduling System (PRESS) expert system prototype developed for space station scheduling. The purpose of developing the prototype was to investigate the resource envelope concept in a pratical scheduling application, using a commercially available expert system shell. PRESS is being developed on an IBM PC/AT using Teknowledge, Inc.'s M.1 expert system shell.

The research includes a proposed definition of the content of the resource envelope, and examination of the resource envelope's flexibility and limitations for scheduling. Our definition of the resource envelope includes time parameters, resource usage, and constraint considerations. The suggested format is exercised by the PRESS scheduler, which performs both initial planning and replanning, fulfilling two of the functions currently defined for the PMS: short-term planning and constraint conflict resolution.

I. Introduction

PRESS is a prototype expert system developed as part of an effort to study the feasibility of using expert system technology in the Space Station environment. PRESS implements some of the functions that have been defined for the PMS, and uses the not yet fully defined "resource envelope" concept that has been developed for Space Station applications.

In a previous article [1], presented following completion of the rapid prototype, the authors described in detail the PRESS system functionality for the rapid prototype, and planned full prototype capabilities. Since then, the full system prototype, with finalized functionality and key concept refinement, has been developed and demonstrated for NASA representatives. In the present article the existing PRESS capabilities will be briefly described for familiarization purposes, but the main emphasis will be placed on those assumptions, concepts, and technical approaches which were developed and/or finalized after the PRESS rapid prototype demonstration and the previous publication.

II. Discussion

The goals of PRESS are: to research the feasibility of expert system applications in providing PMS functionality; to use (and therefore help define) the resource envelope concept as a basis for automatic scheduling; to use realistic examples like the Cosmic Background Explorer (COBE) and the Upper Atmosphere Research Satellite (UARS) spacecraft scheduling needs as a proof-of-concept; and to evaluate the suitability of the system development environment for expansion of this prototype or developement of an operational system.

The PMS [2] is a software system that provides operational management services among payloads and platform systems for the space station. PMS consists of an automated on-board segment, the Platform Management Application (PMA), and a ground segment, the Platform Management Ground Application (PMGA).

PRESS addresses two of the PMS functions. The first of these is the Short-Term Plan Management function. This function involves the PMGA receiving a plan from the Platform Support Center, and uplinking appropriate portions to the PMA. The PMGA and the PMA may receive plan changes requested by operators, customers, subsystems, and payloads. The second function, Conflict Recognition and Resolution, involves the monitoring of resource usage, allocation, and margins. Conflicts for resource usage are to be resolved on a priority basis. This function will be used in deciding whether a given request may be scheduled. The PMA and the PMGA are required to modify the short-term schedule while maintaining a conflict-free plan that does not exceed the platform's resource capabilities or compromise its safety.

A final PRESS prototype system was completed and demonstrated for NASA on September 3, 1987. Some functions which were implemented in the rapid prototype were removed from the full prototype due to memory limitations. Both PRESS prototypes taken together serve as a proof-of-concept for all the defined functions. The following functions are implemented by PRESS.

- Perform initial scheduling, processing requests by priority
- Perform rescheduling
- Accept schedule modification requests as resource envelopes
- Accept schedule modification requests as changes to resource availability and/or constraints
- Resolve schedule conflicts based on assigned envelope priorities
- Perform conflict checking
- Accept multiple envelope requests
- Place scope of user interaction at operator's discretion
  - provide advice for modifications to the resource envelope requests that would permit successful scheduling (rapid prototype version only)
  - provide capability for operator to cease processing before completion of input file
- Perform input error checking
  - on user input data file via preprocessor
  - on legality of interactive query responses via M.1 capabilities
- Provide graphic and textual representation of system output

Brief definitions of some terms are presented here to avoid confusion with possible active usage elsewhere. An "activity" is the item being scheduled. It may be anything from a complex scientific experiment to a single use of a communications link. An activity may consist of more than one event, with each event represented as one "resource envelope". A request for scheduling an activity is represented to PRESS in the form of one or more resource envelopes, together with an activity header. Each "resource envelope" is equivalent to an event and represents a time period, one or more resources whose use is required, and a usage level for each resource. Resource usage is assumed to be stable for scheduling purposes. Examples of "resources" are power, an instrument, a communications link, etc. The "schedule" or "schedule timeline" is produced by the system to show what activities have been scheduled within a given time period. PRESS output shows the activities plotted against the resource used over time, so that the schedule timeline is actually a set of parallel timelines, one for each resource, with usage periods identified with an envelope identifier. PRESS accepts user input interactively or from a stored file, and is capable of both initial and rescheduling functions, including rescheduling as required by changes in resource availability or operational constraints. Requests are scheduled based on externally determined priorities. Higher-priority requests receive preferential treatment during scheduling. This includes the automatic deletion of already scheduled lower-priority requests if needed to successfully schedule a new higher-priority request. (See detailed description of PRESS functionality [3]).

The presumption at the outset was that the resource envelope concept would permit a global view of resource usage, which is important for shared resources like power or communication links. A global view provides protection against oversubscription, especially in concurrently used resources.

PRESS implementation of the the resource envelope concept included the following major points: (a) the activity is viewed in terms of resource consumption; (b) within a resource envelope, resource usage is considered constant, for scheduling purposes, over the time period of the envelope; (c) activities which vary in resources used, or dramatically in level of resource usage, must be broken into separate events, with each event represented by a resource envelope; (d) envelopes in one activity may not overlap in time, and events within an activity are defined by the user in chronological order; and (e) activities may overlap in time, and use the same resources, availability permitting.

PRESS is implemented as a production rule system in M.1 by Teknowledge. The knowledge base contains approximately 300 entries; about 145 of them are rules. PRESS represents resources and constraints as lists, but treats them as virtual objects. The representation includes: resource identifier; start and stop times; amount of resource; and an event identifier (or nil) field. Constraints are represented in a similar manner to the resources. The resource and constraint objects are dynamically created, deleted, divided or combined by PRESS during the process of scheduling. Resource and constraint identifiers are not "hard-coded" and are transparent to PRESS. Resource levels can be represented as absolute units or as percentages of use, as long as the convention is maintained consistently within one resource. These two characteristics afford a maximum flexibility; the intent was to make PRESS as generic a scheduler as possible, easily applied to a number of specific applications. Requests for scheduling include an activity header and one or more request envelopes. The activity header specifies the priority and type of the request. The request envelope is implemented as a list, containing the following fields: activity identifier; envelope identifier; start time and duration windows; resources and amount required; and constraints generated and avoided.

The initial test application of PRESS was to schedule COBE communication link usage. The communication links themselves were considered as the resources (uplinks, downlinks, and links via TDRSS or ground links were treated as separate resources). Line-of-sight times determined resource availability. The COBE example had no constraints. PRESS could easily perform scheduling in this case, but the example was too limited to exercise all of PRESS's capabilities.

The UARS observation instrument scheduling was selected as a more complex problem. UARS contains a platform with three separate instruments capable of performing solar or stellar observations. Although the instruments make independent observations, the attitude of the platform, which all share, determines what object may be observed at any one time. Resources included power, the instruments and the platform. Initially, in a manner analogous to the COBE approach, the sun and stars were also treated as resources to be "used" by being observed. However, it was found that the presence of the sun, for example, constrained stellar observation. This meant that even if the sun was viewed as a resource, it must be considered as a constraint as well. Alternatively, if solar availability was treated as a constraint solely, it became necessary to specifiy both the sun's presence and absence as constraint objects, since solar observations could only occur in the presence of the sun, and certain stellar observatons only in the absence of the sun. It was felt to be more consistent to represent solar availability as a constraint than to have it appear as both a constraint and a resource. Even so, it was difficult to create a test scenario to accurately reflect a realistic activity in UARS terms. Such an activity might include slewing the platform to obtain a sighting, opening instrument shutters, placing certain filters, and taking the observation. It was difficult to define this with resources limited to power, the platform, and the individual instruments, at least not in any obvious one-to-one correspondence with the actions taking place. And although PRESS can express concurrent use of resources, it cannot express truly shared use. An example is the need to say that if a scheduled activity already has the platform slewing toward the sun, a second activity requiring the same action need not use resources to do it, but rather can "share" the existing action.

III. Conclusions

PRESS is a prototype system, with certain conditions and simplifications assumed. However, conclusions, and problems the authors have encountered during system development may be applied to operational systems which use the resource envelope concept in a similar technical environment.

The authors feel that the resource envelope as a theoretical concept is very useful; it provides a global view and permits tight control over concurrent usage of shared resources. However, resource envelopes do not seem to be straightforward in practical application because of the need for additional definition and assumptions, such as determining what should be considered a resource and what should be considered a constraint. Another difficult decision involves breaking down an activity into envelopes, which must take into consideration "wastage" of resources due to the assumption of a constant level of use against the complexity of defining and processing the request.

From this perspective, the resource envelope concept may be needed in an environment similar to space station environment. It is not as easily used as the exclusive scheduling approach for individual payload activities. The optimal balance between the resource envelope approach and a more traditional commanding scheduling approach in a multi-payload application awaits further investigation.

Acknowledgements

The authors wish to acknowledge Ed Beach, Karen Leavitt, Gail Maury, and Audrey Loomis (CSC), Steve Tompkins, Steve Wadding, and Larry Zeigenfuss (GSFC code 511), and Brian Keeler (Bendix Field Engineering Corporation), for their encouragement and helpful discussions. This work was funded by the National Aeronautics and Space Administration, Goddard Space Flight Center for the Mission Operations Division (GSFC code 510) and the Data Systems Technology Division (GSFC code 520) under Contract NAS5-28620, task assignment 319.

References

1. Bush, J.L., A. Critchfield, and A. Loomis, "Space Station Platform Management System (PMS) Replanning Using Resource Envelopes", May, 1987.
2. Goddard Space Flight Center (GSFC),"The Platform Management System Definition Document", October, 1986.
3. Computer Sciences Corporation (CSC), "Space Station Platform Management System (PMS) Resource Envelope Scheduling System (PRESS): Prototype Expert Scheduling System for the Multisatellite Operation Control Center (MSOCC)", September, 1987.