

## USING AI/EXPERT SYSTEM TECHNOLOGY TO AUTOMATE PLANNING AND REPLANNING FOR THE HST SERVICING MISSIONS

L. Bogovich, J. Johnson, A. Tuchman, D. McLean, B. Page,  
A. Kispert, C. Burkhardt and R. Littlefield  
AlliedSignal Technical Services Corp. (formerly Bendix Field Engineering Corp.)  
Seabrook, MD 20706

W. Potter  
Goddard Space Flight Center  
Greenbelt, MD 20779

### ABSTRACT

This paper describes a knowledge-based system that has been developed to automate planning and scheduling for the Hubble Space Telescope (HST) Servicing Missions. This new system is the Servicing Mission Planning and Replanning Tool (SM/PART). SM/PART has been delivered to the HST Flight Operations Team (FOT) at Goddard Space Flight Center (GSFC) where it is being used to build integrated timelines and command plans to control the activities of the HST, Shuttle, Crew and ground systems for the next HST Servicing Mission. SM/PART reuses and extends AI/expert system technology from Interactive Experimenter Planning System (IEPS) systems to build or rebuild timelines and command plans more rapidly than was possible for previous missions where they were built manually. This capability provides an important safety factor for the HST, Shuttle and Crew in case unexpected events occur during the mission.

Keywords: HST Servicing Mission, AI, Expert System, Automation.

### INTRODUCTION

#### The IEPS Group

The IEPS group at Bendix has been building spacecraft ground support systems with embedded AI/expert system capabilities since 1985. The IEPS group in conjunction with the Spacecraft Control Programs Branch (Code 514) has built several powerful planning and scheduling systems using the C language and conventional hardware (PCs and UNIX-based workstations) rather than traditional AI languages and specialized AI

machines. It has been possible to quickly and efficiently change or enhance these knowledge-based systems to adjust to new scheduling conditions.

#### The IEPS Development Approach

The IEPS systems have been developed with an evolutionary prototyping approach. In contrast to the more traditional waterfall approach, the evolutionary prototyping approach starts with the assumption that a software application cannot be totally specified at the start of the development process. The evolutionary prototyping approach uses the basic cyclical paradigm: gather requirements, create/evolve a prototype, evaluate the prototype, and improve the prototype. In this approach, developing a system is considered to be a discovery process which results in continuously evolving specifications.

In contrast to rapid prototyping approaches, the evolutionary prototyping approach emphasizes the evolution and reuse of generic software tools. By more effectively reusing generic software tools developed in earlier systems and prototypes, the evolutionary prototyping approach reduces the overall system development time.

In the IEPS development approach, several prototypes are delivered to the customer for evaluation before the final system is delivered. This approach allows the customer to provide feedback about the prototypes and results in improved functionality for the final system with decreased risk for the customer. The final system is delivered only after the customer is satisfied with the system's performance.

## The ERBS System

In 1987, the ERBS-TDRSS Contact Planning System (McLean *et al.*, [1]) was delivered to the Earth Radiation Budget Satellite (ERBS) Flight Operations Team (FOT) at GSFC. This system is written in C and is implemented on an IBM PC/AT. The system automates the process of generating requests for communications support from the NASA Tracking and Data Relay Satellite System (TDRSS), and it was the first expert system at GSFC to provide ground-system support for an on-going mission.

The ERBS system uses scheduling environment data from the Flight Dynamics Facility at GSFC along with strategic planning knowledge from a Knowledge Base (KB) to build a 1-week schedule of TDRSS requests. The system uses alternative scheduling strategies and traditional conflict avoidance techniques to perform conflict resolution (McLean *et al.*, [2]).

The ERBS system uses the Planning and Resource Reasoning (PARR) shell to build timelines in batch and interactive scheduling modes. Using PARR, a schedule of requests can be built in a few minutes, compared with several hours by the manual method. After a schedule of requests is built in the batch mode, a graphical timeline can be displayed. Users can edit the timeline in an interactive mode, while obtaining "expert" help from PARR.

The ERBS system has been used steadily since its delivery. In addition, the system has been modified or enhanced several times to meet changing mission requirements. These changes were easily made because of the knowledge-based features of the system (McLean [3]).

## Explorer Platform Planning System

In 1991, the Explorer Platform Planning System (EPPS) was delivered to the Extreme Ultra-Violet Explorer (EUVE) FOT at GSFC (McLean *et al.*, [4]). EPPS uses AI/expert system technology from the ERBS system.

In addition, EPPS provides several enhancements to the ERBS system. First, EPPS runs on a UNIX-based workstation with X-Windows/Open-Look. Second, EPPS schedules several types of EUVE mission support activities in addition to TDRSS service requests. Third, EPPS provides knowledge acquisition tools so that EUVE FOT can modify the strategies and constraints in the KB and try "what-if" scenarios to adapt EPPS to handle new scheduling situations. Finally, EPPS uses an Ethernet to electronically receive resource data from the Flight Dynamics Facility at GSFC, TDRSS schedule data from the Network Control Center at GSFC, and planning data from EUVE Investigators at the University of California at Berkeley. This Ethernet is also used to send TDRSS schedule data to the Network Control Center and sequences of EUVE command procedures to the Command Management Facility at GSFC.

## The IEPS Software Toolkit

As IEPS systems were developed, many generic tools for building new IEPS systems were also developed. Eventually, these generic tools were formally organized into a software toolkit called the IEPS Software Toolkit (NASA-GSFC, [5]). This toolkit contains several types of system-building tools: data formatting and report generation tools, user interface tools, database tools, strategic planning tools and tactical planning tools.

To build a new system using the generic tools in the IEPS Software Toolkit, a software engineer first examines the basic requirements for a new system and identifies the IEPS tools that can be applied to the new system. Next, individual IEPS tools are configured to handle specific tasks, and script files are created to link the individual tools into a system that can be tested. Finally, the unified system is tested and iteratively refined until it meets all of the initial, plus discovered, requirements. Recently, IEPS tools have been used to build another planning and scheduling system, SM/PART:

## SM/PART OVERVIEW

HST Servicing Missions are Shuttle missions that are expected to occur about every three years to upgrade or replace failed HST components and to help the HST function to its fullest extent over its 15-year mission lifetime. SM/PART is a planning and scheduling expert system that automates the complex process of building or rebuilding integrated timelines and command plans for the HST Servicing Missions (Johnson, *et al.* [6]). Integrated timelines and command plans are used to coordinate the activities of the HST, Orbiter, Crew and ground systems during the servicing missions.

SM/PART is currently being used to prepare for the first HST Servicing Mission that is scheduled for launch in 1993. It is expected that SM/PART will also be used to support all the other future HST Servicing Missions. For each servicing mission, HST Servicing Mission engineers must provide SM/PART with detailed planning and scheduling data. The planning and scheduling data that is required includes scheduling environment (resource) data, event definitions, sequence definitions and command procedures. SM/PART provides powerful data and knowledge acquisition tools for users to enter this planning and scheduling data.

Before a timeline or command plan is built, a defaults file and a Data Set Configuration (DSC) file must be created. The defaults file provides basic display information for a timeline and command plan such as the mission launch time, the timeline start time and stop time, timeline and command plan header information, and colors to be used on the displays. The DSC file provides the names of the defaults file, Merged Resources file, Event Definition KB, Sequence Definition KB and Procedure Definition KB that are to be used for a particular timeline and command plan.

After all of the required files and KBs have been constructed, SM/PART uses

PARR to build a timeline in a batch (automatic) scheduling mode. In this process, PARR places each HST event on a timeline in accordance with pre-defined scheduling strategies and constraints in the Event Definition KB. The data and knowledge components that make up a timeline are shown schematically in Figure 1.

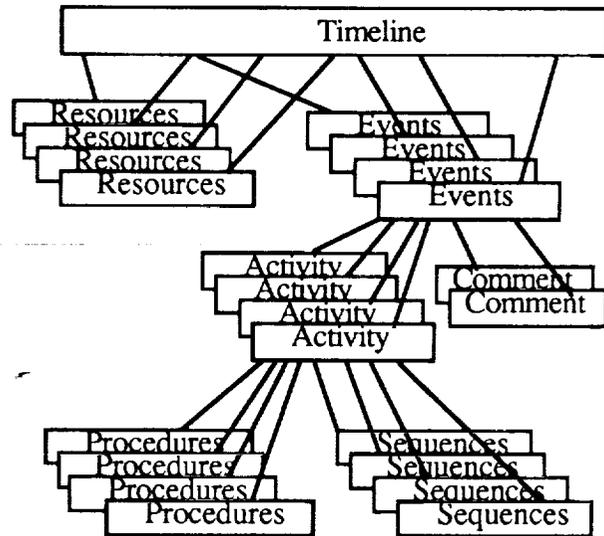


Figure 1. Timeline Components

The timeline that has been built in the batch mode can be graphically displayed with its scheduling environment data (resources) and scheduled HST events (activities and comments).

A section of an integrated timeline is shown in Figure 2.

Because the data objects displayed on an integrated timeline are actively connected to the Event Definition KB (via PARR), users are able to edit a timeline during an interactive scheduling session while they obtain "expert" scheduling help from PARR. For example, as an event is changed, the definition of the event in the Event Definition KB is automatically updated. If an event is dragged by mouse to a place where a scheduling constraint is violated, a prominent



In addition to reusing IEPS tools, SM/PART provides several new and enhanced features. For example, SM/PART is the first IEPS system that uses Motif software for its multitude of windows and pop-up/pull-down menus. Also, SM/PART features enhanced data and knowledge acquisition tools, as described below.

## **ENVIRONMENT DATA ACQUISITION**

One type of planning and scheduling information that must be acquired for SM/PART to function is scheduling environment, or resource, data. Scheduling environment data is part of the strategic planning information (data/knowledge) that is required by PARR to automatically build timelines and command plans during the tactical planning process.

Several types of scheduling environment data displayed on an integrated timeline are acquired electronically from external sources. For example, ORBIT#, DAY/NIGHT, SOUTH ATLANTIC ANOMALY, and TDRS data are received electronically from the Flight Dynamics Facility at GSFC via the HST Application Processor. This data is not generated or modified by HST Servicing Mission personnel, but just reformatted by SM/PART.

Other types of scheduling environment data on an integrated timeline are acquired directly from HST Servicing Mission personnel. Examples of this data include: HST ATTITUDE, ORBITER ATTITUDE, TELEMETRY FORMAT, CREW SCHEDULES, and GROUND SYSTEM ACTIVITIES. For acquiring this data, SM/PART provides several types of Motif-style data-entry forms.

Eventually, the various types of external and user-entered scheduling environment data must be merged into a single data file, the Merged Resources file. Later, data from this file is used by PARR, along with strategic planning knowledge, to automatically place HST events on a timeline.

## **KNOWLEDGE ACQUISITION**

Another type of planning and scheduling information that must be acquired for SM/PART is strategic planning knowledge. For SM/PART, strategic planning knowledge includes activity event definitions, comment event definitions, sequence definitions and command procedures. This knowledge is acquired from HST personnel and stored in various KBs. HST activity event definitions and comment event definitions are stored in the Event Definition KB, sequence definitions are stored in a Sequence Definition KB, and command procedures are stored in a Procedure Definition KB.

For acquiring strategic planning knowledge, SM/PART provides new knowledge acquisition tools. For example, to acquire complex scheduling strategies and constraints, event definition forms with linked push-button or pop-up menus and various options are provided. An Activity Event Definition Form, for AD# B508, is shown in Figure 3. This event is also seen scheduled on the timeline shown by Figure 2.

For acquiring the "start event" attribute of an event, linked push-button and/or pop-up menus are provided to allow the user to specify that an event start when a second event or resource starts or stops. Also, the user may specify a plus or minus offset for the "event start" relative to the start or stop time of the second event or resource.

For acquiring "constraints" for events, linked push-button and/or pop-up menus are provided to allow the user to specify that an event occur only when a second specified event or resource occurs. Alternatively, an event can be specified so that it avoids a second event or resource. In addition, the user can enter plus or minus offset times for the various options selected.

SM/PART also allows users to specify alternative scheduling strategies that can be tried when there is a scheduling conflict. One type of alternative strategy has SM/PART

### Activity Event Definition Form

AD #:  AD Title:  Seq. Num.:

Event Type:  Display Type:

Start Event:     Duration:

**Constraints:**

<input type="text" value="AVOID"/>	<input type="text" value="ZOE"/>	<input type="text" value=""/>	<input type="text" value=""/>
<input type="text" value="DURING"/>	<input type="text" value="ORBIT_DAYLIGHT"/>	<input type="text" value="Plus"/>	<input type="text" value="00:03"/>
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>

**Alternative Strategies:**

<input type="text" value="NEXT"/>
<input type="text" value=""/>

Note:  
 "BEFORE/AFTER" refer to an Activity or Comment  
 "PRIOR/NEXT" refer to a Resource type

Welcome to the Motif KB Editor  
 File /sm/part/r2.0/data/operwork.kb opened

Edit fields and hit Accept, or Cancel

Figure 3 . SM/PART Activity Event Definition Form

schedule an event just before or just after a conflicting event. Another type of alternative strategy has SM/PART schedule an event during the resource window that occurs just prior to or just after the resource window where the conflict occurs.

Sequence Definition Forms are provided for acquiring sequence information such as sequence number, sequence title, the activity events included in each sequence, and special ordering instructions for the activity events

within each sequence.

Procedure Definition Forms are provided for acquiring detailed command procedures associated with HST activity events. Procedure Definition Forms allow users to enter information describing the procedures to be performed by operations personnel for each activity event, the effects of each procedure, the duration of each step/substep, and the actions expected in space and throughout the ground system.

## **BUILDING A TIMELINE**

### **Batch Scheduling**

After scheduling environment data and strategic planning knowledge have been acquired, SM/PART is able to build a timeline in the batch (automatic) scheduling mode. This process is referred to as tactical planning. To build a timeline in the batch mode, PARR reads scheduling environment data from the Merged Resources file and strategic planning knowledge from the Event Definition KB, dynamically allocates an internal frame structure to represent each HST event, and uses the information to place events on the timeline. If resources are not available or if constraints are violated, then alternative scheduling strategies are used to try to resolve the scheduling conflicts.

If there are no scheduling conflicts, the event is put on the timeline and the Event Definition KB is updated. If there is a scheduling conflict that cannot be resolved then a prominent "VIOLATION" message is written in the Event Definition KB.

### **Interactive Scheduling**

A timeline that is built in the batch scheduling mode can be displayed graphically on the terminal screen with its scheduled events. Alternatively, a new timeline with scheduling environment data, but with no scheduled events, can be displayed graphically on the terminal screen.

In the interactive scheduling mode, the user can browse the timeline that is displayed and interactively add or change timeline events while receiving expert scheduling assistance from PARR. This expert scheduling assistance is possible because the timeline data objects are actively linked via PARR to the Merged Resource file and Event Definition KB.

As an example of editing a timeline in the interactive scheduling mode, a user may click on an HST activity event with the mouse and "drag" it to a new, valid position. In this

case, the Event Definition KB is automatically updated. However, if the activity is dragged to a place where a scheduling constraint is violated, then a pop-up window with a "VIOLATION" message that the user must respond to is displayed on the screen.

## **BUILDING A COMMAND PLAN**

After a timeline has been built, a detailed command plan corresponding to the timeline can be automatically built and displayed on the terminal. Building a command plan involves retrieving and combining scheduled timeline event information with sequence definitions and command procedures. Sequence definitions specify groups of HST activities while command procedures specify the detailed steps required to complete each scheduled HST event.

A command plan that is displayed on the terminal can be converted to an identical graphical command plan print. Command plan prints are used by HST Servicing Mission engineers at their control consoles during the HST Servicing Missions.

SM/PART is also able to automatically synchronize a command plan with a timeline. Synchronizing a command plan with a timeline is required whenever changes are made to either the command plan or its corresponding timeline.

## **REPLANNING**

An important capability of SM/PART is to quickly rebuild a timeline and command plan. This capability is particularly important if unexpected events or changes in the scheduling environment occur during a mission. In critical situations, this capability provides an important safety factor for the HST, Shuttle and Crew. Initial results from the HST Flight Operations Team indicate that SM/PART is able to reduce the time to rebuild a timeline and command by a factor of ten compared with the former manual method using a Macintosh (Potter *et al*, [7]).

## CONCLUSIONS

SM/PART has successfully reused and extended IEPS AI/expert system technology to build SM/PART and automate the complex task of building timelines and command plans for HST Servicing Missions. To automate this task, SM/PART initially provides capabilities for HST Servicing Mission personnel to acquire scheduling environment data and strategic planning knowledge. Next, SM/PART is able to use the acquired scheduling environment data and strategic planning knowledge to automatically place HST events on a timeline. During interactive scheduling sessions, SM/PART is able to provide "expert" scheduling assistance to users. Finally, SM/PART is able to combine timeline event data with sequence definitions and detailed command procedures to automatically generate command plans.

An evolutionary prototyping approach which emphasizes reusing and enhancing AI tools was successfully used to build SM/PART in an eight month period.

## ACKNOWLEDGEMENTS

The authors wish to thank Patricia Lightfoot and Dorothy Perkins at NASA-GSFC (Code-510) and Ellen Stolarik and David Warren at Bendix Field Engineering for their continual support and many contributions to the IEPS task. This work was supported by NASA Contract NAS5-27772.

## REFERENCES

1. McLean, D., Littlefield, R. and Beyer, D. (1987). An Expert System for Scheduling Requests for Communications Links between TDRSS and ERBS, *Proceedings of the 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics*, Goddard Space Flight Center, Greenbelt, MD.
2. McLean, D., Page, B., Tuchman, A., Kispert, A., Yen, W. and Potter, W. (1991). Emphasizing Conflict Resolution versus Conflict Avoidance during Schedule Generation, *Expert Systems World Congress Proceedings*, Vol. 1, pp. 385-394, Orlando, Florida.
3. McLean, D. R. (1990). Maintaining an Expert Planning System: A Software Tools Approach, Jay Liebowitz (Ed.). *Institutionalizing Expert Systems: A Handbook for Managers*, (pp. 91-98) Prentice Hall.
4. McLean, D., Page, B., and Potter, W. (1990). The Explorer Platform Planning System: an Application of a Resource Reasoning Shell, *Proceedings of First International Symposium on Ground Data Systems for Spacecraft Control*, pp. 195-200, Darmstadt, Germany.
5. NASA-GSFC/Code 514, (August 1992). IEPS Software Toolkit, Vols. 1-7.
6. Johnson, J., Tuchman, A., McLean, D., Kispert, A., Bogovich, L., Burkhardt, C., Page, B., Littlefield, R., Potter, W., and Ochs, W., (1992). HST Servicing Mission Planning and Replanning Tool, *World Space Congress*, Washington, DC.
7. Potter, W., Bogovich, L. and Johnson, J. (1992). Hubble Space Telescope Servicing Mission Planning and Replanning Tool (SM/PART), *1992 GSFC Research and Technology (R&T) Report*, (in press).

# ***Monitoring/Control/Diagnosis***

