N88-17247

EXPERT MISSION PLANNING AND REPLANNING SCHEDULING SYSTEM FOR NASA KSC PAYLOAD OPERATIONS

Roger Pierce
Payload Operations
Information Systems Office (CS-ISO)
National Aeronautics and Space Administration
Kennedy Space Center, Florida 32899

ABSTRACT

EMPRESS (Expert Mission Planning and REplanning Scheduling System) is an expert system created to assist payload mission planners at the Kennedy Space Center (KSC) in the long range planning and scheduling of horizontal payloads for space shuttle flights. Using the current flight manifest, these planners develop mission and payload schedules detailing all processing to be performed in the Operations and Checkout (O&C) building at KSC. With the EMPRESS system, schedules are generated quickly using standard flows that represent the tasks and resources required to process a specific horizontal carrier. Resources can be tracked and resource conflicts can be determined and resolved interactively. Constraint relationships between tasks are maintained and can be enforced when a task is moved or rescheduled. EMPRESS became operational in March 1986. It was developed jointly by NASA at the Kennedy Space Center and by the MITRE Corporation of Bedford, Massachusetts. EMPRESS-II, currently under development by the MITRE Corporation, is scheduled to be delivered to KSC in September 1987. This paper will briefly descibe the domain, structure, and functionality of the EMPRESS system. It will describe some of the limitations of the EMPRESS system as well as improvements expected with the EMPRESS-II development. Finally, the future of the project will be discussed.

INTRODUCTION

As the primary launch and landing site of the Space Transportation System (STS), the Kennedy Space Center (KSC) is responsible for the final checkout, preparation, and installation of payloads into the space shuttle orbiter vehicle prior to a mission. Upon return from space, KSC is responsible for the deintegration of these payloads. A payload is an experiment or set of experiments attached to a carrier structure, which is then placed into the shuttle payload bay. A mission can be thought of as a set of payloads that are flown into lower Earth orbit using the STS.

Payloads and payload operations at KSC are divided into two primary categories, vertical and horizontal. Vertical payloads are installed into the orbiter vehicle at the launch pad and include payloads that use the Payload Assist Module (PAM) or the Inertial Upper Stage (IUS) as their carrier structure. Most satellites launched from the space shuttle are considered vertical payloads. Horizontal payloads, on the other hand, are usually installed into the orbiter payload bay at the Orbiter Processing Facility (OPF). Carrier structures for these payloads include the Spacelab long module (LM), short module (SM), and pallet. The mission peculiar experiment support structure (MPESS) is also considered a horizontal payload carrier. Various payload carrier combinations may be flown on the same mission which increases the complexity of the work performed.

Processing of horizontal payloads occurs primarily at the Operations and Checkout building (O&C) in the KSC industrial area. This processing includes all of the steps or tasks necessary to assemble and install the experiments onto the carrier structure as well as the steps needed to perform the required experiment and subsystem functional verifications prior to transport of the payload to the OPF and installation into the orbiter. To moniter and control this processing activity, NASA generates and maintains a hierarchy of schedules illustrated in Figure 1.

At the top of the NASA schedule hierarchy is the flight manifest. Released periodically from NASA Headquarters in Washington D.C., this document assigns launch dates, orbiter vehicles, and payloads to each STS

PRECEDING PAGE BLANK NOT FILMED

mission and lists this information for missions slated for the next five years. An example of the manifest in its telemail format is found in Figure 2. With this data KSC then creates long range plans and schedules that describe how KSC will support the launch date milestones. In the payload processing community, one of the long range support plans is called the Multiflow Assessment or MFA. This schedule was previously known as the Master Multiflow or MMF.

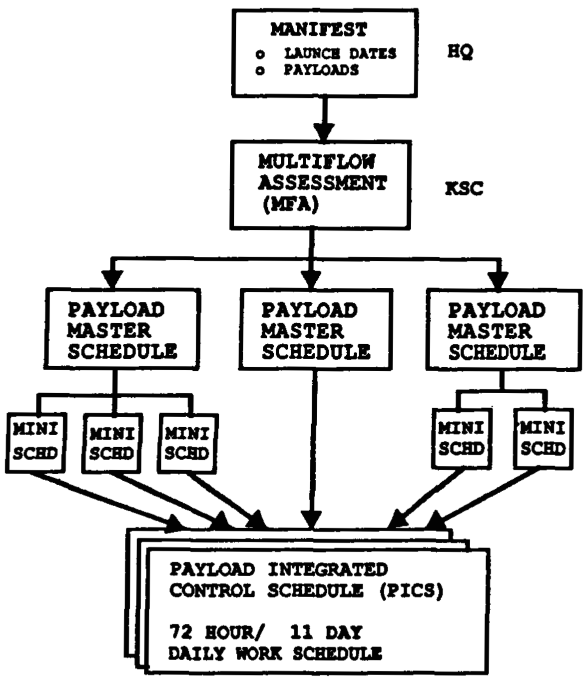

Figure 1 - KSC Schedule Hierarchy

The MFA consists of Gantt charts that illustrate the major processing activities needed for each payload listed in the manifest. More detailed information is given for payloads processed by the KSC Payload Directorate. For horizontal payloads, the major activities include experiment integration, carrier integration, and orbiter integration operations. These activities are referred to as Level IV, Level III/II, and Level I, respectively. An example of the MFA format is shown in Figure 3. In addition, the MFA contains information on some of the critical resource needs of these payloads. This enables early recognition of potential conflicts between limited resources. The MFA also serves as a baseline for the development of more detailed mission and payload schedules.

Because of the dynamic nature of shuttle operations, the need to respond quickly and effectively to changes in the processing schedule is important. Payload mission planners are often called upon to develop new MFA's quickly when the manifest is changed or to produce "what-if" schedules when examining unusual mission scenarios. This planning and replanning

*** SHUTTLE FLIGHT ASSIGNMENTS FOR PAYLOADS ***
JUNE 1987 PRELIMINARY BASELINE
POP 87-2 MAX DOD OFFLOAD CP-APR 8-JUN-87

FLT	DATE ORBTR	INCL ALT	CRW DUR	PAYLOAD	CARRIER	OTHER PAYLOADS	UF
34	89 10 5 COLUMBIA	28.5 160	5 4	GPS-1 GPS-2 MSL-3	PAM-D2 PAM-D2 MPESS		
35	89 10 23 DISCOVERY	28.5 110	5 4	GALILEO	IUS-3 STA		
36	90 1 11 ATLANTIS	0.0 0	0 0	DOD			
37	90 3 1 COLUMBIA	28.5 160	5 5	GPS-3 GPS-4 MSL-4	PAM-D2 PAM-D2 MPESS		
38	90 4 9 DISCOVERY	0.0 0	0 0	STARLAB	LM+1P		
39	90 4 30 ATLANTIS	0.0 0	0 0	DOD			
40	90 6 7 COLUMBIA	28.5 160	5 4	GRO			
41	90 7 12 DISCOVERY	28.5 160	5 4	TDRS-E	IUS-2 STA		

Figure 2 - NASA Manifest in Telemail Format

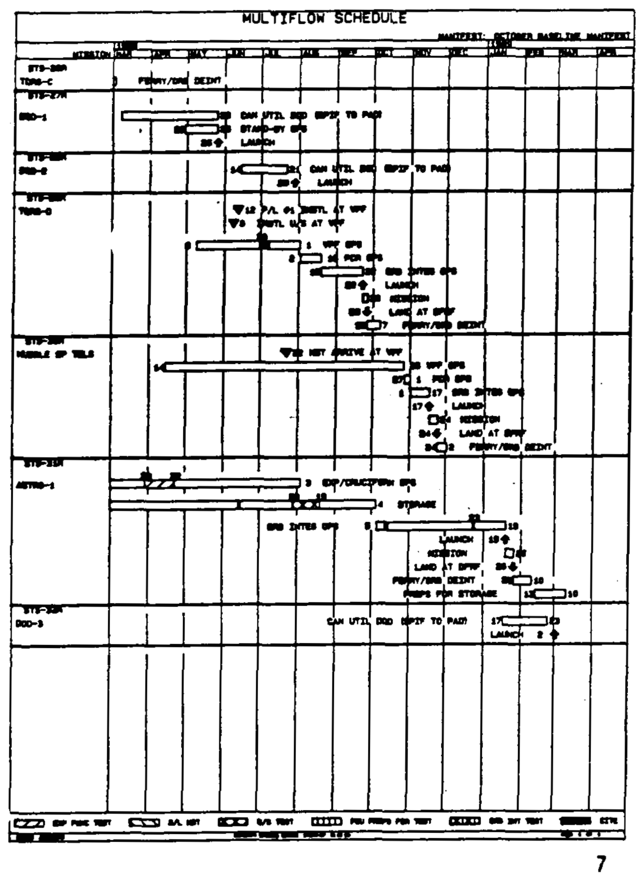

Figure 3 - KSC Multiflow Assessment example using Artemis.

ORIGINAL PAGE IS OF POOR QUALITY

must be as accurate and timely as possible. Unfortunately, the manual methods for developing and producing these schedules do not fulfill these requirements.

In an effort to automate the process of producing the MFA, NASA at the Kennedy Space Center initiated a project to perform this function using artificial intelligence (AI) techniques. The project was named EMPRESS (Expert Mission Planning and REplanning Scheduling System) and was developed jointly between NASA at KSC and the MITRE Corporation out of Bedford, Massachusetts. The goal of the project was twofold. First, EMPRESS was to demonstrate the application of AI to a real KSC problem, namely planning and scheduling. Second, the project was to be instrumental in building a NASA in-house AI capability for payload operations. Both of these objectives have been met with the current EMPRESS system.

HISTORY

The initial study for EMPRESS was begun by NASA and the MITRE Corporation in May of 1984. Dual implementation began in January of 1985 and the prototype was completed in February 1986. EMPRESS became operational in March of 1986. The software was developed on Symbolics 3600 series LISP machines using Symbolics' 6.1 operating system. EMPRESS was written in ZetaLISP using the Symbolics' object oriented programming paradigm called Flavors. A Flavor is a data structure used for symbolic representation of an object. In November 1986, EMPRESS was updated by NASA to the new Symbolics Genera 7.0 operating system.

EMPRESS is currently used by the KSC payload operations community as an aid in developing quick "what-if" mission scenarios and resource utilization studies. EMPRESS is not used to produce the MFA. This is due in part to a change in the responsibility for the MFA and in limitations of the EMPRESS prototype. In January 1987, responsibility for the MFA was transferred from NASA to the Payloads Ground Operations Contract (PGOC), which was awarded to the McDonald Douglas Aerospace Corporation (MDAC). MDAC uses the commercial Artemis scheduling system supplied by Metier to produce the MFA.

DESCRIPTION

EMPRESS was designed to allow a payload mission planner the ability to develop schedules quickly for horizontal payloads using the space shuttle flight manifest. When creating a schedule for a payload on a mission, EMPRESS first looks to see if a current schedule already exists for that particular mission or payload. If not, EMPRESS creates a default schedule using a standard flow, which is simply a list of all of the tasks, task constraints and resources required to process a particular horizontal carrier. With the default schedule generated, the planner can then modify the tasks and resources as required. EMPRESS gives the planner the ability to verify that resource conflicts have not occurred between parallel operations and to revise resources and tasks automatically if conflicts do exist. Constraint relationships between tasks are maintained and can be enforced when tasks are moved or rescheduled. The user interface is quite robust and gives the planner an excellent graphical representation of the schedule and detailed histograms of the resource utilization.

DOMAIN

The domain knowledge base for EMPRESS can be divided into three major areas - tasks, resources, and system heuristics. Task data include the various activities required to process a payload and the temporal relationships between these tasks. The resource knowledge encompasses the people, hardware, and facilities required to process a payload. The heuristics control scheduling, resource allocation, and conflict resolution.

All tasks in EMPRESS are represented as Flavor objects. Each task may have a set of subtasks and each subtask may have subtasks resulting in an overall tree structure for the task knowledge base. At the top of the EMPRESS task tree is the manifest. A manifest may have any number of mission subtasks and each mission may have any number of payload subtasks. Each mission task has at least a launch date milestone and a fly-mission task associated with it. In addition, each payload task has a series of processing tasks. These processing tasks are divided into the various integration activities, which include the Level IV, Level III/II, and Level I functions. The integration steps are finally reduced into the lowest task level which may include activities like experiment staging, integration, power-up, or testing.

Relationships between tasks are also maintained by the EMPRESS domain. This knowledge is represented by variables in the task Flavor structure. Using these variables, constraints such as task-subtask or parent-child relationships, as well as predecessor-successor or sibling relationships can be defined. EMPRESS refers to these relationships by the terms "contains", "contained-by", "before", or "after". A task may also have a set of milestones associated with it. Milestones are separate data structures in EMPRESS and signify one time events with no duration. Launch dates, as an example, are considered milestones. The relationships between a task and a milestone are referred to by "begins", "begun-by", "ends", or "ended-by". Using these eight task constraint mechanisms, EMPRESS has a powerful method for defining any task hierarchy.

The resource domain defines the facilities, hardware, and people available to process a payload. Table 1 outlines the resource class structure used in the development of EMPRESS. Like tasks, resources are stored as flavor objects in the working memory of the system. However, where task hierarchial relationships are maintained by variables, the resource relationships are maintained by the inheritence mechanism provided by the flavor paradigm. For example, a PCU is a type of Test Equipment, which is a type of Facility, which is an essential resource. Resource data includes the maximum number of individuals available, any possible alternative resources, and a list of the current users.

Along with the task and resource knowledge, the domain embodies the heuristics used to control the planning and scheduling. This includes the methods for scheduling the tasks, searching for existing schedules, assigning resource states, and maintaining task constraints. EMPRESS has a small set of rules used to resolve resource conflicts. These rules control how EMPRESS finds alternative resources, modifies resource utilization, or reschedules the resources within a task to correct a resource problem.

STRUCTURE

The EMPRESS architecture is divided into six primary modules. These modules are the Command Module, the Display Module, the Scheduling Module, the Resource Module or Resource Tracker, the Constraint Module, and the Data Module. Figure 4 illustrates this software structure. Each module performs a specific series of functions and the interaction between the modules provides for a flexible and powerful planning tool.

The Command Module contains most of the higher level system functions. It controls the system build and holds many of the system utility functions and application programs.

The functions for the user interface are performed by the Display Module. This includes the window processes, menus, mouse functions, and the graphics code seen by the user, including the calendar and histogram displays. The EMPRESS command loop and file system interface codes are also found here.

The Scheduler is responsible for building a schedule and maintaining the constraint relationships between tasks. The Scheduler pulls task information from the Data Module, calculates start and end times, and verifies that no temporal constraints have been violated. This module also defines the task flavor structure and controls many of the query and modification functions that can be performed on a task.

With a preliminary schedule created, the Resource Module allocates resources and maintains resource accountability. The resource flavor structure and class hierarchy are stored here. When a user decides to commit a resource to a task,

TABLE 1 - EMPRESS Resource Class Summary

Facility	Non-Flight Hardware
GSE	Alignment Equipment
EGSE	Brackets
MGSE	Cable
Offline Area	Cable Harness
Storage Area	Canister
Test Equipment	Covering
ATE	Crane
CITE	Fork Lift
HITS	Harness
HRDE	Long Trolley
PCU	Strong Back
PITS	Subsystem Hardware
PSTE	Support Structure
SPCDS	Transporter
Test Stand	Trunnion Support Fix
User Room	

Flight-Hardware	People
Carrier	Electrical Engineer
Experiment	Logistics
Flight Payload	Mechanical Engineer
Floor	Quality Control
GSA	Safety
IPS	Technician
Keel	Test Conductor
MPE	
Orbiter	
Rack	
Trunnion	

ORIGINAL PAGE IS OF POOR QUALITY

the Resource Tracker verifies the resource availability and modifies the resource usage if appropriate. If the resource is unavailable, the resource tracker contains the rules needed to resolve the resource conflict.

The Constraint Module defines the temporal relationships between tasks. It also has the capability to create precedence relationships between generic tasks. However, this capability has not been implemented.

Finally, the Data Module stores the majority of information used by the system. This includes the manifest, mission, payload, and standard flow files, as well as the resource and task knowledgebases. Any schedules saved by the user and the system files required to initialize EMPRESS are found here.

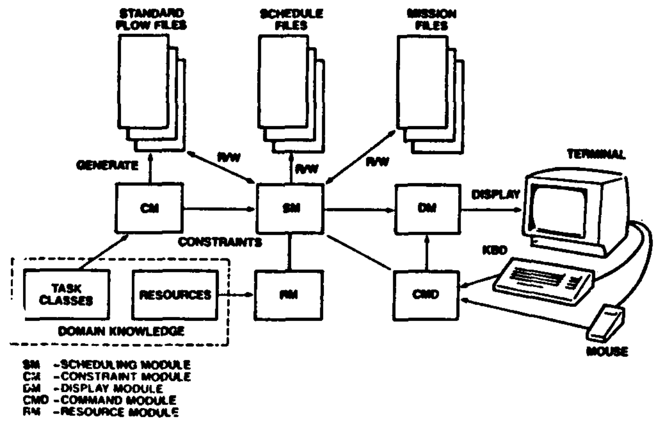

Figure 4 - EMPRESS Structure Summary

FUNCTIONALITY

EMPRESS contains a multitude of functions for creating a schedule. The best way to summarize these functions is to review a hypothetical session with an EMPRESS user when given a new flight manifest. In this example session, we will create a schedule from a manifest, describe some of the interface display options, explain resource handling, review conflict resolution, and finish by describing some useful tools. All functions in the EMPRESS system are menu or mouse driven.

When given a new manifest, usually in a telemail text format, the text file is first converted to a usable form by running an EMPRESS utility function. The operator then selects a "load manifest" menu option to begin the task of creating a new schedule. As each mission and payload on the manifest is read, EMPRESS searches to see if an existing schedule already exists. If so, then that schedule is used. Otherwise, EMPRESS creates a default schedule for the payload based on its carrier. These default carrier schedules are called standard flows and they contain the tasks, task constraints, and resources needed to process that type of carrier. In EMPRESS there are standard flows for the following horizontal carriers - Spacelab long module, pallet-igloo, MDM-pallet, and MPESS. For missions that do not contain horizontal carriers, EMPRESS creates a default fly-mission task using the manifest launch date.

After loading in each payload schedule, the payload tasks are scheduled to determine start and end times and to verify that no task constraints have been violated. This is done by making a backward CPM pass with the launch date as the starting point. The total process of converting a new manifest from the telemail format to creating a schedule for 50 missions takes less than 20 minutes. An example of the loaded manifest is given in Figure 5.

With schedules for all of the manifested missions loaded, the operator has many display options. He may keep the display in the manifest format or view an individual mission. The spreadsheet-like display can be moved in any direction and the calendar end times may be changed as required. In the "tree" function, the duration of the calendar is tied to a specific task or subtask and can display the activities on a weekly or daily level. The operator can also "open" or "close" any task to show any level of the task-subtask hierarchy. This is shown in Figure 6 for the planned STS-31 mission. If a task needs to be relocated, the operator may move the task graphically with the mouse, then reschedule the task. Resources are reallocated automatically and task constraints verified.

Using the standard flows, resource needs are identified for various tasks in the processing task tree. EMPRESS resources may be in one of three states: unspecified, requested, or committed. The operator can change the status of a task's resource needs to "requested" or "committed" and view the resource status in a histogram or utilization chart. A resource histogram is illustrated in Figure 7. When the resources are committed, EMPRESS will detect resource conflicts.

There are several ways to change a resource state. The operator has the option to request or commit all resources for all missions or to set the resource state at any level of the task tree. In addition,

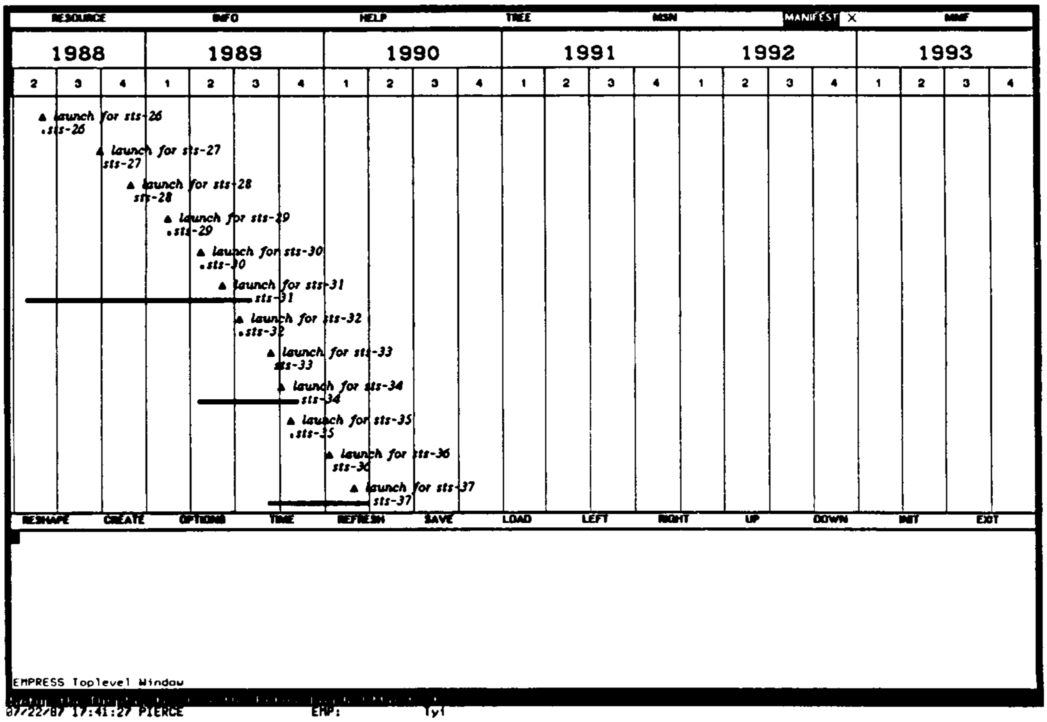

Figure 5 - Example of a flight manifest schedule using EMPRESS.

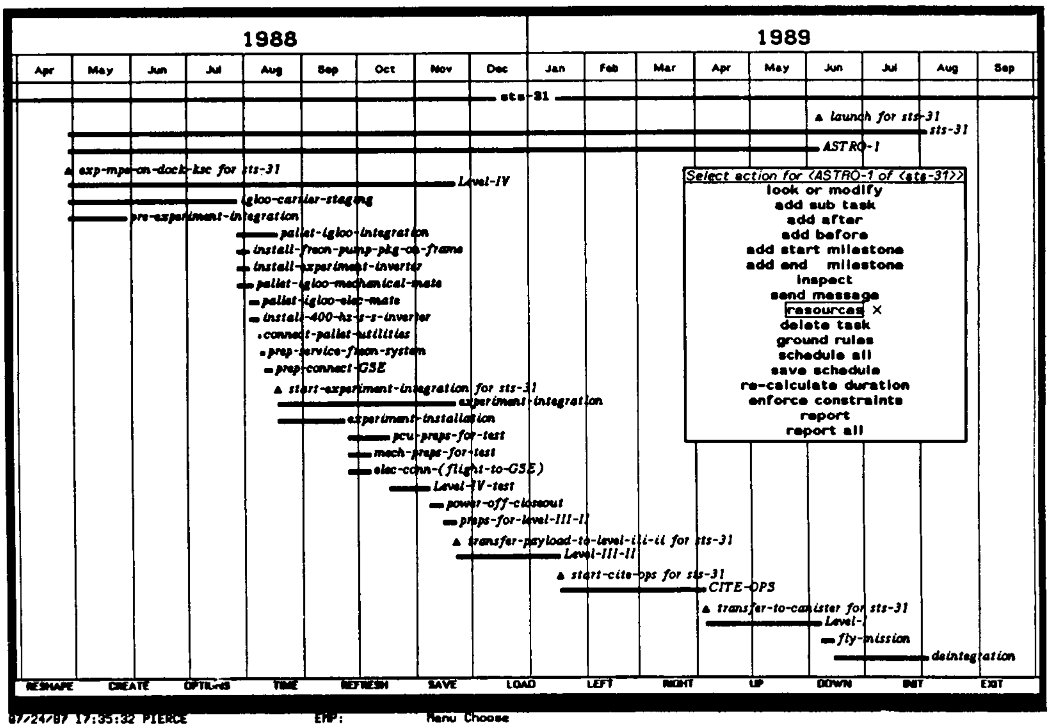

Figure 6 - Example of a task-subtask tree and task menu using EMPRESS.

ORIGINAL PAGE IS
OF POOR QUALITY

the operator may add or delete resources for any individual task in the schedule.

With an EMPRESS schedule created and resources committed, the operator can now detect and resolve any conflicts that may have arisen. Conflicts occur during task scheduling or resource allocation. If a task constraint is violated (ie. a child task is moved outside of its parent task), the operator has two options. The constraint can be enforced and the child task will be repositioned under the parent task, or the constraint can be relaxed and the parent task's end times will be recalculated to remove the contention. In a resource allocation conflict, a set of rules are fired that allow the operator to resolve the conflict. These rules allow the operator to substitute an alternative resource, to increase the workload of the resource (ie. add more shifts), or to reschedule the task that caused the problem. The operator may also choose to let EMPRESS resolve all resource conflicts automatically without operator input.

After all conflicts are resolved, the operator can save the schedule into the EMPRESS data directory or into his or her personal directory. This feature allows each user to create and store individual "what-if" files that can be recalled and revised later. With the schedule saved, the operator can then reinitialize the system and start anew.

There are a set of optional functions that further enhance the EMPRESS planning and scheduling capability. A "create" option allows the operator to create a "what-if" mission that does not exist on the flight manifest. If the user is interested in working on only a small subset of the manifest, there are "mark" functions that will reduce the mission set. There is also a complete set of query functions for reviewing resource accountability and the system knowledgebase.

LIMITATIONS

While EMPRESS provides an effective tool for payload mission planning, it has limitations. The primary limitation with the system is the lack of output. When a schedule is completed, the operator must use screen prints to hardcopy the display. For lengthy schedules, this is quite bothersome and ineffective. EMPRESS does not match many of the Artemis graphics capabilities used in the current MFA. These limitations must be corrected in order for EMPRESS to produce this document.

There are other limitations with the current EMPRESS system. EMPRESS does not store its data in a relational format and can not access a relational database. There is no justification mechanism to explain scheduling, rule firing or conflict resolution. There are also small problems with the current implementations for deintegration tasks, standard flow flavor structures, and the scheduler.

EMPRESS II

After completion of the original EMPRESS prototype in February 1986, the MITRE Corporation continued development on a new project. This project, called EMPRESS-II, uses a different approach to the payload planning and scheduling problem. EMPRESS-II is built upon a new planning and scheduling architecture developed by MITRE for the Air Force called CAMPS (Core of A Meta Planning System). CAMPS provides a more robust foundation for planning and scheduling than the original EMPRESS system and addresses many of its limitations. CAMPS provides for a full declarative representation of the knowledge base. It supports external relational databases, has improved scheduler and resource tracking capabilities and implements an effective justification and truth maintenance system. CAMPS will use strategies and agendas to facilitate automatic planning and replanning operations.

A pre-release version of EMPRESS-II was delivered to KSC in December 1986. The current development project concludes in September 1987.

THE FUTURE

Despite the limitations of EMPRESS, the future of the project looks bright. Work is about to begin on the development of a new version of the EMPRESS system. This work will be performed by NASA using the KSC Payload Operations Artificial Intelligence Application Laboratory with assistance from its PGOC contractor. A user group is being started and a system for configuration control will be implemented. The redevelopment of the EMPRESS system will focus on creating graphical output, improving the user interface and scheduler, and in enhancing the conflict resolution and justification capabilities of the system. The new EMPRESS will also access schedules from the Artemis database currently in use. With a concerted effort, KSC's goal to implement an operational AI system for payload planning and scheduling will be achieved.

CONCLUSIONS

EMPRESS has provided KSC with an exceptional prototype planning and scheduling tool. Using artificial intelligence techniques, schedules for horizontal payloads are created quickly and contentions for limited resources can be determined and resolved interactively. Development of a new version will address many of the limitations with the initial system and bring the project to a more operational state.

REFERENCES

1. Dumoulin, J.M., "EMPRESS Expert Mission Planning and Replanning Scheduling System", software design presentation, NASA, Kennedy Space Center, Florida, May, 1985.

2. Dumoulin, J.M., "Payload Operations and Management AI/Expert Systems Projects Status", management briefing, NASA, Kennedy Space Center, Florida, June, 1986.

3. Dumoulin, J.M., "Payload Operations Artificial Intelligence Lab Activities and Facilities", management briefing, NASA, Kennedy Space Center, Florida, February, 1987.

4. Hankins, G.B., Jordan, J.W., Katz, J.L., Mulvehill, A.M., Dumoulin, J.M., and Ragusa, J.M., "EMPRESS Expert Mission Planning and REplanning Scheduling System", M8533, Mitre Corp., Bedford, Massachusetts, September, 1985.

5. Mulvehill, A.M., "Cargo Expert System Feasibility and Evaluation Phase", MTR-9746, Mitre Corp., Bedford, Massachusetts, September, 1985.

6. Mulvehill, A.M., "A User Interface for a Knowledgebased Planning and Scheduling System", MTR-10175, Mitre Corp., Bedford, Massachusetts, November, 1986.

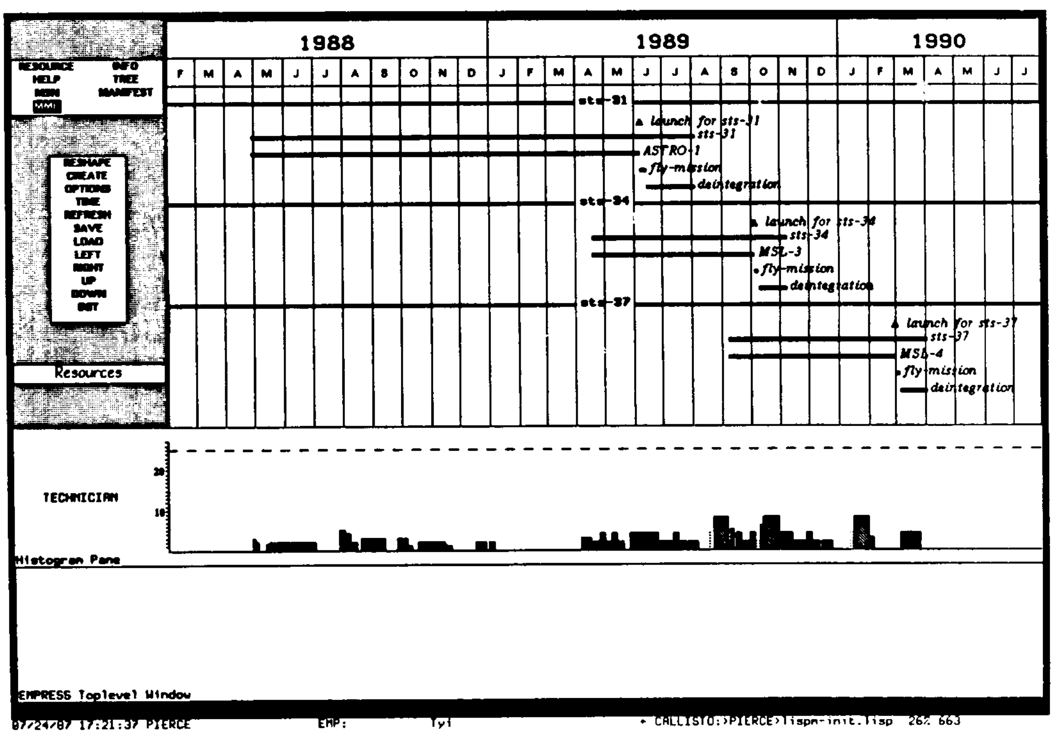

Figure 7 - Example of a resource histogram using EMPRESS.

ORIGINAL PAGE IS OF POOR QUALITY