813-RD-011-001

EOSDIS Core System Project

Planning and Scheduling Prototype Results Report for the ECS Project Part C: Phase 3 Prototype

March 1995

Hughes Applied Information Systems
Landover, Maryland

Planning and Scheduling Prototype Results Report for the ECS Project Part C: Phase 3 Prototype

March 1995

Prepared Under Contract NAS5-60000

APPROVED BY

A. Anders /s/	3/31/95
Tony Anders, FOS Segment Manager EOSDIS Core System Project	Date

Hughes Applied Information Systems
Landover, Maryland

813-RD-011-001

This page intentionally left blank.

Preface

This document contains the prototype results report for phase 3 of the Planning and Scheduling prototype, which was performed between June 1994 and February 1995. Planning and Scheduling is part of the Flight Operations Segment within the EOSDIS Core System project.

This document is an informal document that is approved at the ECS Office Manager level and does not require Government approval. After this prototype has been completed, a final report will be documented in the Prototyping and Studies Final Report, DID 331/DV3.

For additional technical information pertaining to the Planning and Scheduling prototype, contact Bill Moore, Planning and Scheduling Section Manager at 301-925-0378 or via electronic mail BILLM@EOS.HITC.COM.

This page intentionally left blank.

Abstract

The purpose of the Planning and Scheduling (P & S) Prototype Results Report is to present the design and existing feature of the P & S phase 3 prototype, in addition to analyzing the science and flight operations community feedback to establish future priorities. Throughout the phase 3 prototyping effort, operational requirements and user feedback obtained from the phase 1, phase 2 and informal phase 3 demonstrations contributed to the prototype design. The design was subsequently developed into an evaluation package for presentation to the science and flight operations community at the ECS PRR, February 23, 1995. During this evaluation phase, feedback from the science and flight operations community was solicited through evaluation of informal demonstrations, meetings and audience issues and comments raised at the ECS PRR. The feedback was reviewed and incorporated into recommendations for phase 3 of the P & S prototype effort and helped establish priorities for the overall development effort.

Keywords: prototype, evolvability, scenario, IST, ASTER, P&S, CERES, MOPITT, ASTER, PI/TL, TDRSS, FOS, EOC

This page intentionally left blank.

Contents

Preface

Abstract

1. Introduction

2. Results

3. User Feedback

4. Summary

Figures

Tables

Appendix A. P & S Class Hierarchies

Abbreviations and Acronyms

1. Introduction

1.1 Identification

The purpose of the Planning and Scheduling (P & S) Prototype Results Report is to present the design and existing features of the P & S phase 3 prototype, in addition to analyzing the science and flight operations community feedback to establish future priorities. Throughout the phase 3 prototyping effort, operational requirements and user feedback obtained from the phase 2 Prototype Results Review (PRR) and informal phase 3 demonstrations contributed to the prototype design, which was developed into an evaluation package for presentation to the science and flight operations community at the FOS PRR, February 23, 1995. During this evaluation phase, feedback from the science and flight operations community was solicited through evaluation of informal demonstrations, questionnaires, meetings, and audience issues and comments raised at the FOS PRR. The feedback was reviewed and incorporated into recommendations for phase 3 of the P & S prototype effort, and helped establish priorities for the overall development effort.

1.2 Prototype Overview

The P & S prototype effort is an evolutionary process of software design and development for the operational P & S system. Each phase of the prototype can be considered an incremental development towards the final release packages. Below is the schedule of activities related to phases 1-4 of the P & S prototype development effort.

Phase 1

Prototype Requirements Definition	5/93 - 6/93
Prototype Design	5/93 - 7/93
Ops Concept Development	5/93 - 8/93
Prototype Development	5/93 - 10/93
Formal Demonstration	11/93
Prototype Results Report	1/94

Phase 2

Prototype Requirements Definition	11/93 - 1/94
Prototype Design	11/93 - 1/94
Ops Concept Development	11/93 - 3/94
Prototype Development	12/93 - 5/94
Formal Demonstration	6/94
Prototype Results Report	8/94

Phase 3

Prototype Requirements Definition	6/94 - 7/94
Prototype Design	6/94 - 8/94
Ops Concept Development	6/94 - 9/94
Prototype Development	6/94 - 12/94
Formal Demonstration	1/95
Prototype Results Report	2/95

Phase 4

Prototype Requirements Definition	3/95 - 4/95
Prototype Design	3/95 - 5/95
Ops Concept Development	3/95 - 6/95
Prototype Development	3/95 - 7/95
Formal Demonstration	8/95
Prototype Results Report	9/95

By providing incremental prototypes, the science and flight operations community can offer feedback that influences the design and architecture throughout all stages of project development. For example, the user feedback obtained from the Phase 2 PRR was evaluated and helped establish the areas of focus for the phase 3 design.

1.3 Objectives

One of the primary objectives of the development effort was to refine the framework of the P & S system. This framework includes an architecture for providing distributed planning and scheduling. The software design would establish an evolvable system that allows future spacecraft and operational concept changes.

Analyzing the science and flight operations community feedback has the objective of establishing the guidelines and priorities that need to be incorporated into later phases of prototype development. Comments and issues raised by the science and flight operations community were acquired in various ways, including informal demonstrations, evaluation surveys, meetings, and documented questions and comments from the FOS PRR.

1.4 Applicable Documents

The following documents are applicable to the material in this document.

193-317-DV1-001	ECS Prototype Plan
193-318-DV3-005	ECS Prototype Plan & Progress Report
193-707-PP1-002	ECS Prototype Results Review
193-216-SE1-001	ECS Requirements Specification
193-604-OP1-001	ECS Operations Concept Document for the ECS Project

194-207-SE1-001	ECS System Design Specification
194-813-SI4-002	Planning and Scheduling Prototype Results Report for the ECS Project, Part A: Phase 1
194-813-SI4-006	Planning and Scheduling Prototype Results Report for the ECS Project, Part B: Phase 2
	P & S Prototype Design Specification
	EOS Distributed Planning and Scheduling Prototype Lessons Learned Working Paper
	Technical White Paper on the Hughes Inter-Process Communication Library (HIPC)

This page intentionally left blank.

2. Results

2.1 Prototype Driver Analysis

Throughout all phases of the prototyping effort, operational requirements and mission needs drive the design and development of the system. For the P & S prototype, the drivers were derived from several sources and were used to shape the overall prototyping strategy. The drivers originated from the following sources:

a) **System Requirements**

There exist several key requirements that impose a critical risk or significant cost impact, such as the distributed P & S needs of the EOC and ISTs.

b) **Phase 2 Prototype Results Review Feedback**

During the phase 2 PRR, user feedback was gathered from the science and flight operations community through evaluation surveys, meetings and documenting questions and comments. Topics focused on user interface, scheduling analysis capabilities, and primary interfaces (e.g. Command Management Subsystem, ASTER ICC).

c) **GSFC EOS Distributed P & S Prototype Lessons Learned**

The GSFC EOS Distributed P & S Prototype Lessons Learned document represents the findings from a coordinated study between the University of Colorado, JPL and GSFC that investigated issues related to distributed P & S. Based on the lessons learned, the primary topics included the necessity for a clear understanding of activity definitions and parameters, as well as difficulties in developing a scheduling system with external P & S components (e.g. ASTER ICC).

d) **Hughes Mission Planning IR&D**

The Hughes Mission Planning IR&D offers lessons learned from previous prototyping experience, such as human-machine interface (HMI) approach, architecture and interactive scheduling.

e) **Previous Operations Experience**

Based on previous experience with operational systems, evolving P & S concepts have been established from operation support. For example, experience on long-term projects has shown that the operations concept is constantly evolving as the operators learn new ways to use the system. Therefore, there is a need to support an operations concept that will change throughout the mission life.

f) **Previous Mission Planning Development Experience**

Through development of previous mission planning systems, design elements have been identified that are critical to mission performance. For example, the interface with command management will have a direct impact on the P & S system.

The result of the driver analysis and the projected response phase is summarized in Table 2-1.

Table 2-1. P & S Prototype Phases Response Approach (1 of 2)

Prototype Driver	Origin	Response Plan Phase I	Phase II	Phase III	Phase IV
Provide the capability to support an evolving baseline of spacecraft and instruments	System requirements	Develop a flexible, extensible object class structure that will support future system upgrades and changes. Evaluate alternative structures.	Evaluate and refine model for AM-1 to demonstrate approach.	Develop initial comm system model and evaluate extensibility.	Refinement of activities that will be able to model commanding requests for various instruments in a more generic manner.
Provide a system that can support an evolving Operations Concept	Experience, system requirements, Phase 1 PRR Feedback	Design and evaluate alternatives that support ease of upgrade, ease of run-time reconfiguration, and ease of code modification.	Demonstrate and evaluate operations flexibility with AM-1 spacecraft and instrument manifest.	Demonstrate and evaluate evolvability via initial end-to-end flow through the system.	Present the end-to-end operations concept with most tools and interfaces prototyped.
Provide a mission planning system that supports a distributed P & S Operations Concept	System requirements, GSFC Distributed Test Bed Lessons Learned, Phase 1 PRR Feedback	Demonstrate global visibility of constraints and plans based on a distributed system architecture	Refine distributed resource model concepts	Further refine the distributed resource model concepts and evaluate distributed P & S alternatives	Evaluate distributed resource models.
Provide for simplicity of operation and common look-and-feel. Minimize training and maintenance cost.	Experience, system requirements, Hughes Mission Planning IR&D lessons learned, Phase 1 PRR Feedback	Develop a set of common planning tools that can be used across the EOC and IST. Initial phase 1 prototype will have a basic timeline and resource model.	Include additional functionality into the timeline and add other toolset components to simplify scheduling analysis.	Refine spacecraft and instrument resource models. Include additional toolset components.	Add functionality to tools initially prototyped in the Phase 3 prototype to simplify scheduling.

Table 2-1. P & S Prototype Phases Response Approach (2 of 2)

Prototype Driver	Origin	Response			
		Phase I	Phase II	Phase III	Phase IV
Provide an early P & S test bed to mitigate external interface risks (e.g. ASTER interface, command management interface)	Experience, GSFC Distributed Test Bed Lessons Learned	Develop activity generator to be used as a basis for an ASTER interface test driver.	Develop and refine interface issues and concepts. Establish ASTER interface test driver	Develop and refine FOS internal interfaces (e.g. Command Management System).	Develop and refine FOS internal and initiate FOS external interfaces (e.g. FDF, NCC)
Provide a framework for the P & S development	Experience, proposed evolutionary development methodology	Build prototype to operations code standards and documentation. Phase 1 will include a simple end-to-end thread that will provide the basis for subsequent phases to build upon.	Incremental prototype design and development baseline	Incremental prototype design and development baseline	Refine software infrastructure given results of interface with CMS and performance tests with distributed resource models.

Additional drivers and modifications to the planned responses may be identified upon further analysis and feedback .

2.2 Prototype Design Analysis

Based on the prototype drivers (see Section 2.1), several analyses were performed to address issues related to the ECS P & S design, architecture and development. The findings of these investigations were based upon several sources, including:

- Informal prototype demonstrations to the user community
- Conversations with the user community
- ECS Level 3 and Level 4 Requirements
- Phase 1 and Phase 2 PRR Feedback
- GSFC EOS Distributed P & S Prototype Lessons Learned
- Previous Hughes experience in developing mission planning systems

Table 2-2 breaks down the primary design decisions related to the ECS P & S system. The overall details of the design and architecture are explained in Section 2.3.

Table 2-2. P & S Prototype Design Issues (1 of 4)

Design Issue	Criteria	Decision	Rationale
Evolvability	Establish a mission management foundation where extensions can be built for an evolving ECS problem domain. This mission management foundation would incorporate concepts and design elements from other developed P & S systems.	Mission Planning Heritage Code Usage	The mission planning heritage code is a collection of C++ class libraries which provide baseline classes for display, collection, inter-process communication and P & S. These class libraries are developed by Hughes and embody concepts and design elements from other Hughes developed mission management systems. By allowing ECS extensions to be built on top of the tested code, development and test cycles are shortened, thereby reducing overall life cycle costs.
	Ability to accept changes in the mission requirements and operational concepts. An evolvable system in which each prototype phase is considered an incremental development towards the operational system.	C++ Programming Language	C++ is an object oriented programming language that allows the P & S problem domain to be separated into software objects that encapsulate the attributes and behavior of the physical mission elements (e.g. instruments, subsystems). Encapsulating an object's attributes and behavior minimizes the software impact due to changing requirements. C++ also offers inheritance which allows system evolvability by incorporating mission attributes and behavior as they become known.
Distributed P & S	Accommodate a wide range of interfaces and information between distributed processes. Provide peer-to-peer communication between concurrently executing processes that are running on the same computer or distributed across a heterogeneous computer network.	Object Oriented IPC with evaluation of evolution to DCE/CORBA	The Hughes IPC (HIPC) library is a set of C++ software objects that provide peer-to-peer communication between concurrently executing processes. If used with C++ processes, HIPC allows objects to be passed between processes. For external processes outside of P & S that are written in other languages (e.g. C, Ada or FORTRAN), interface widgets will facilitate the communication.

Table 2-2. P & S Prototype Design Issues (2 of 4)

Design Issue	Criteria	Decision	Rationale
	Allow two local or remote processes the ability to locate each other in order to establish a connection.	Nameserver Process	A nameserver is a separate process that allows process-to-process connections in a distributed environment. Within a local workgroup, processes register with their assigned nameserver. When a process wishes to establish connections with another process, it asks their nameserver for the process' address. Nameservers can be constructed to establish connections with other nameservers for remote process communication.
	To allow distributed P & S, local or remote processes should be able to send schedule requests and receive schedule changes from different resource models.	Resource Model Process	A resource model process can perform constraint checking of various mission resources, such as instruments and subsystems. Resource model processes can reside at the EOC and ISTs. Processes such as a scheduler can send requests to a local or remote resource model, while other processes like the timeline display can receive schedule updates from a resource model.
	In order to allow simultaneous phases of schedule development over a distributed network, scheduling access should be controlled by time periods (initial scheduling, final scheduling, etc.) and resource type (CERES, MODIS, etc.).	Plan Window Manager Process	The plan window manager will control read/write access to the mission planning database, allowing a distributed instrument community to work concurrently on different phases of schedule development. For example, initial and final scheduling represent different phases that can be performed concurrently.
	Establish distributed process-to-user communication of scheduling status, information and critical messages.	FOS Event Handler Process	The event handler will receive messages from the P & S processes (scheduler, timeline, etc.) and display the information to the user. By existing as a separate process, the message handler can receive scheduling status, information and critical messages from remote network processes.

Table 2-2. P & S Prototype Design Issues (3 of 4)

Design Issue	Criteria	Decision	Rationale
Global Visibility into Scheduling	Capability to visualize any instrument and subsystem state over time. State information includes resource consumption, start/stop times, durations, constraints and other related details.	Timeline Display Process	A timeline display will provide a temporal representation of instrument and subsystem states. As a separate process, it will allow a distributed nature where schedule updates from a distant resource model can be sent over a network by object oriented messages.
	Capability to visualize an instrument or subsystem state over large periods of time, specifically 16 day orbit cycles.	Segmented Timeline Process	A segmented timeline can be configured to provide schedule comparisons for varying spans of time (e.g. days, orbits). The segmentation also allows for greater resolution into large planning cycle schedules.
	Provide ability to obtain hard-copies of an instrument or subsystem schedule similar to the Timeline Display Process.	PostScript Timeline	The PostScript timeline enables users to obtain hard-copy printouts of the timeline display for evaluation.
Scheduling Methodolog y	Ability to define the activities to be scheduled using the Planning and Scheduling Tools.	Activity Definer Process	The activity definer provides users with an interface for creating activities, scheduling requests for configuring an instrument or subsystem through time, and storing them within the project database.
	Capability to define Baseline Activity Profiles for instruments.	Baseline Activity Profile Definer Process	The BAP definer enables users to define the set of activities that will be scheduled together to configure instruments over time to follow a baseline mode of operations.
	Provide an interface that allows the user to interactively or automatically edit the mission schedule. Capability to submit activity requests to a local or remote resource model for constraint analysis.	Instrument Activity Scheduler Process	The instrument activity scheduler will allow the user to schedule, modify and delete activities from the mission schedule. The instrument activity scheduler will work in conjunction with the local and remote resource model processes to ensure that no constraints are violated.
	Capability to schedule on multiple ("what-if") plans.	Plan Tool Process	The plan tool allows schedulers to create multiple plans for "what-if" analysis.

Table 2-2. P & S Prototype Design Issues (4 of 4)

Design Issue	Criteria	Decision	Rationale
	Ability to schedule NCC communication contact periods as well as determine playback durations for the Solid State Recorder buffers.	Communication Contact Scheduler Process	The communication contact scheduler incorporates an algorithm to determine optimal NCC communications contact periods and evaluate Solid State Recorder buffer usage in determining playback durations.
Mitigate interface and system risks	Provide a P & S test bed for interfaces and system processes to minimize potential software complications.	Interface Handler Process	The interface handler is a test driver that receives activity lists for impact scheduling. The interface handler is used as a scheduling tool to test P&S processes and forms the basis of an ASTER ICC. Specific interface handlers will be established for the command management interface.
	Capability to interact with the Instrument Support Terminal	IST Interface	The interface with the Instrument Support Terminal provides the user the ability to quickly toggle between Planning and Scheduling and instrument and spacecraft telemetry monitoring.
	Provide the capability to receive Real-Time updates in order to keep the resource model up to date.	Interface with the Analysis software	The analysis interface allows update of the status of the SSR buffers on-board the spacecraft. This capability ensures accurate, up to date modeling of the amount of space available in the data buffers.

For the phase 4 prototype, additional design issues will be investigated, including:

- Instrument microprocessor load uplink scheduling
- Activity command list editing
- FDF Orbit Information Processing
- TDRSS contact rescheduling impact
- Inter-instrument constraint determination and visualization
- Automatic rescheduling of activities removed from the schedule

2.3 Prototype Architecture

2.3.1 P & S Architecture Design

One of the primary objectives of the phase 3 prototype was to continue refining the framework for the P & S system. Based upon the analysis of the requirements and user needs (see

Sections 2.1-2.2), an architecture was developed consisting of independent, distributed C++ processes that communicate by passing object oriented messages between each other. Figure 2-1 shows the overall P & S architecture for the phase 3 prototype. The architecture embodies concepts and design elements from the EOS Distributed Prototype, other Hughes developed mission management systems and the phase 2 PRR feedback. Each of the current P & S processes are described below:

2.3.1.1 Resource Model

The resource model is the portion of the P & S system that represents those aspects of the real world necessary for mission planning. The process models the behavior of mission resources, such as the instruments, spacecraft and subsystems. Because it simulates the behavior of these mission elements, the resource model performs a level of constraint checking based upon any physical limits. In addition, the resource model maintains the state of the instruments, subsystems and other elements. Other processes, such as the instrument activity scheduler, can modify and update the states by interfacing with the resource model.

2.3.1.2 Activity Definer

The activity definer is a tool which provides mission planners with the ability to define the types of activities they will schedule for a particular instrument or subsystem. Activity definitions contain the mode of the resource during and after the activity as well as any necessary commands and parameter defaults. The user will later be able to schedule one of the activities over a given time range using the instrument activity scheduler (see Section 2.3.1.7).

2.3.1.3 BAP Definer

In order to simplify routine scheduling tasks, users define Baseline Activity Profiles (or BAP's) using the BAP definer. A BAP is a group of activities which can be scheduled as a single activity using the instrument activity scheduler (see Section 2.3.1.7). The BAP definer builds the BAP's that the user will need by adding activities to and removing activities from a specified BAP definition. The user sets start and stop times for the activities in that BAP using times relative to the BAP's scheduled times or relative to orbit events.

2.3.1.4 Timeline

The timeline process presents a graphical representation of spacecraft and instrument activities as a function of time. Activities are the building blocks from which mission plans are constructed, representing the state of an instrument, subsystem or other schedulable entity. The timeline displays information regarding activity start/stop times, resource consumption and constraints. By receiving and displaying schedule updates from the resource model, a timeline represents the current state of the overall mission plan.

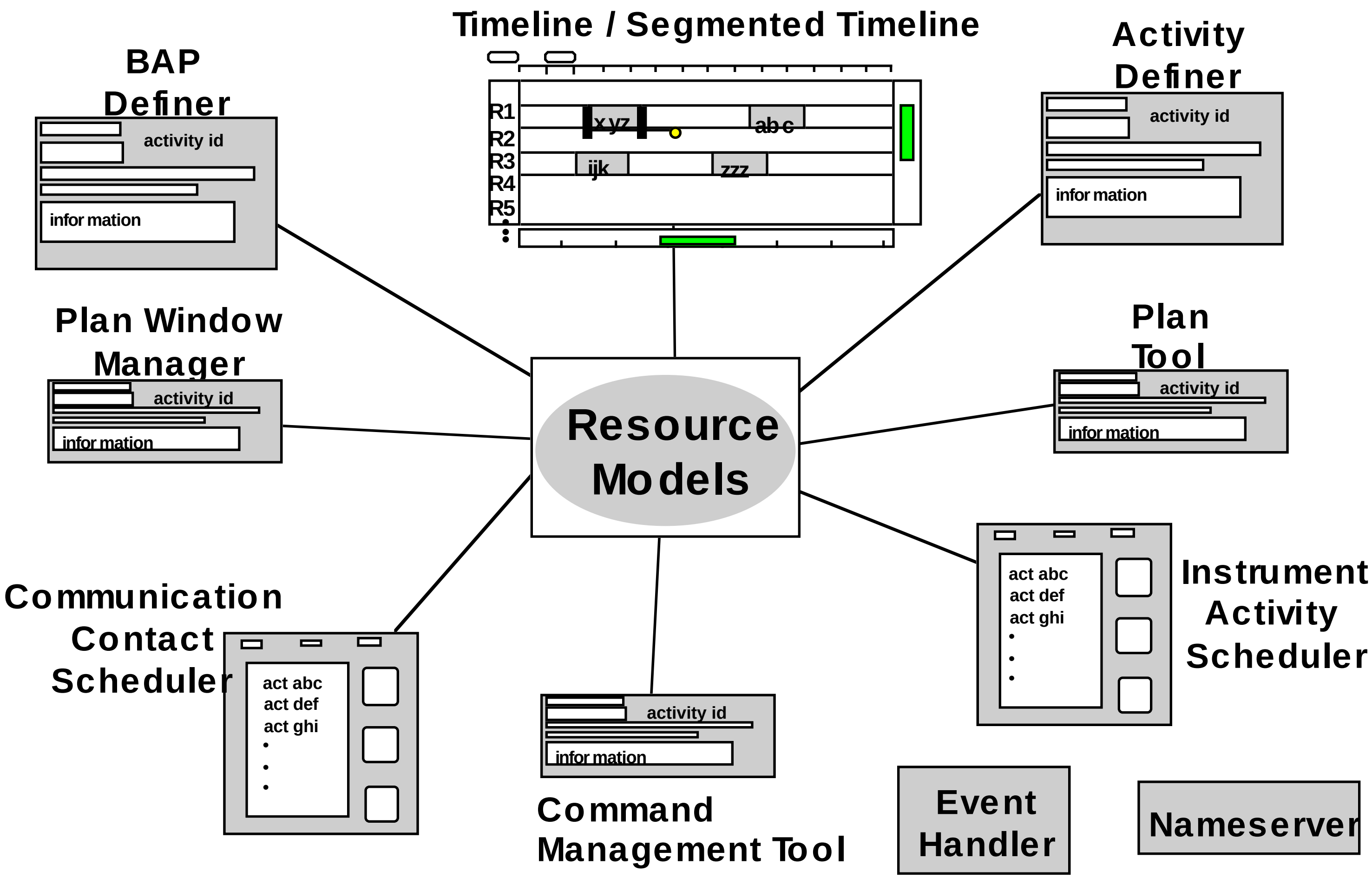


Figure 2-1. P & S Architecture

2.3.1.5 Segmented Timeline

The segmented timeline process behaves similarly to the regular timeline in that it displays a graphical representation of spacecraft and activities as a function of time. However, unlike the regular timeline, the segmented timeline allows the user to view resources for consecutive days on a single display. This allows the user to view activities that may be tied to particular orbit events that may occur on a daily basis. For example, if an instrument planner wishes to view a repetitive activity that may be triggered by a daily orbit event, the segmented timeline will be able to give him the capability to view the daily repetition by specifying a certain number of days to be displayed in 24 hour sections.

2.3.1.6 PostScript Timeline

The PostScript Timeline tool is a version of the Planning & Scheduling timeline that produces PostScript output suitable for printing on black & white or color PostScript printers, or for displaying using a PostScript Viewer. The PostScript timeline uses the same configuration files as the regular timeline, so the resources, colors and text are similar. The PostScript timeline does not need the resource scroll bar, time scale scroll bar, or window borders. Printed PostScript timelines are constrained by the resolution of the printer (typically 300 dpi) as opposed to the resolution of the video screen (typically 72 dpi). Screen snapshots also include unnecessary scrollbars and window borders. The prototype version prints on a single page in landscape mode, scaling the resources & time scale to fill the page. This tool produces superior quality output that can be printed or displayed at the EOC, IST's or any other site with PostScript capabilities.

2.3.1.7 Instrument Activity Scheduler

Planning and Scheduling needs to provide the ability to schedule activities for instruments. Activities that can be scheduled are either pre-defined activities or pre-defined lists of activities called baseline activity profiles (BAPs). Once these activities or BAPs have been defined, the user needs to enter the appropriate data and press the schedule button. An instance of an activity will be created and given to the resource model to be scheduled. The scheduling algorithms schedule activities sequentially and depending on user needs, an impact or non-impact scheduling algorithm is used. The software uses different types of scheduling algorithms which can easily be changed from one type to another. Once the resource model performs the scheduling, the results can be seen on the timeline.

2.3.1.8 Communication Contact Scheduler

The Communication Contact Scheduler tool is used for establishing TDRSS and ground station contact periods. The Communication Contact Scheduler chooses desirable contact requests from a list of availability times. The availability times are the inview times minus previous rejections. The scheduler can be used for either the initial list of requests to send to the NCC or for subsequent iterations with NCC. For the PRR demonstration, inviews for TDRSS were supplied by the NCC and read in by the Communication Contact Scheduler. The data used in the demonstration was for TDRS-1, but the scheduler could handle data for any relay or ground resource with appropriate inview periods.

The algorithm the prototype scheduler uses to determine the best contacts is a score-based depth first search. The scheduler reads the score ranges from a configuration file for properties such as separation time (time between end of previous contact and start of this contact), contact duration, predicted data volume at start of contact, amount of data loss if this contact is missed and the communication path used (i.e., TDRS-E, DSN, WSGT). For the current prototype, only the separation, duration and communication path properties were implemented. The algorithm accepts several command line arguments that affect the depth of the search, the granularity of the search and a pruning parameter. The depth controls how many contacts to look ahead when scheduling a contact. No look-ahead (depth=1) works fine as long as there are no large gaps in the availability periods. Using depth=3 does a good job of placing contacts immediately before & after large gaps. The granularity specifies the difference between subsequent attempts. The demonstrations used a granularity of 60 seconds. The last parameter is a pruning parameter and it controls whether the algorithm searches paths that are worse than the best. The pruning parameter is a percentage between 0 (do an exhaustive search) and 100 (only use Look-ahead to break ties). Values around 70-90 seem to be a good mix between execution speed and finding optimal contacts.

2.3.1.9 Plan Tool

The plan tool is a user interface program that allows the user to create multiple plans. This capability will be needed if an instrument planner wishes to create a "what-if" plan to validate how changes to his instrument schedule will affect other instruments or spacecraft subsystems. In addition to plan creation, the plan tool allows the user to delete plans, copy plans (or portions of a plan) and snap to plans. The snap function is used to cause other tools that display information about a plan to automatically change to the specified plan. For example, if a user wishes to snap to a master plan and visibility process such as a timeline was currently displaying activities on a "what-if" plan, the timeline would automatically switch and immediately display the activities and spacecraft resources on the master plan.

2.3.1.10 Plan Window Manager

In order to allow simultaneous phases of schedule development over a distributed network, the plan window manager will control access to the mission planning database. The ability to make changes to the schedule will be controlled in two different ways. First, the plan window manager will manage schedule development over time periods. This will allow the planners and schedulers to simultaneously work on different time sections of the plan which is necessary to perform functions related to initial scheduling and final scheduling. Second, the plan window manager will control schedule development over specific resources, such as the CERES and MODIS instruments. Resource control will allow the distributed instrument community to simultaneously schedule their instruments over the same time period without impacting one another. Resource control will also prevent one instrument team from changing another team’s schedule.

2.3.1.11 Activity Interface (Filter)

The Planning and Scheduling Subsystem interacts with nine interface elements. The context diagram in figure 2-2 represents these elements, along with their corresponding data flows. To handle the different types of data transactions, interface processes will exist between the P & S subsystem and the interface elements. For the purpose of prototyping, a general purpose filter was developed to allow the ingest of ASCII formatted information to the resource model. The current prototype uses an activity interface to ingest an ASCII representation of an ASTER ICC activity list. Each entry in the ASCII list represents an ASTER activity to be scheduled.

2.3.1.12 CMS Interface

The interface between planning and scheduling and command management provides constraint checking during activity scheduling and generates a detailed activity schedule, an ATC load and a ground script. When an activity is scheduled, the information is passed to command management and is expanded into commands and put into a schedule of commands. Once the commands have been expanded into the schedule, constraint checking is performed and a message is sent back to planning and scheduling notifying them if any constraints have been violated. When the user wants to generate the detailed activity schedule, the user will enter a start and stop time and then press the Generate Load button. A message with the detailed activity schedule is then sent to command management where an ATC load and ground script are generated from their schedule of commands.

2.3.1.13 IST Interface

During this phase of the prototype, a first attempt to integrate the P & S system into the Instrument Support Toolkit (IST) was made. The IST provides users with the ability to enter "rooms" where different tasks are performed. For example, the user may evaluate real-time data from a TDRSS contact in one room and schedule new contacts in the P & S room. In this prototype, both the IST and the P & S system were modified in order that the P & S system would be presented in one of the rooms of the IST.

2.3.1.14 Analysis Interface

The interface between planning and scheduling and analysis provides a mechanism through which analysis sends planning and scheduling information concerning the real time status of the solid state recorders. Planning and scheduling shows the status of the solid state recorders on the timeline by a line graph. The initial data that planning and scheduling uses to show this status is predicted data. To show the users the most accurate data, planning and scheduling displays as-flown data such as actual playback times and buffer contents provided by real-time analysis.

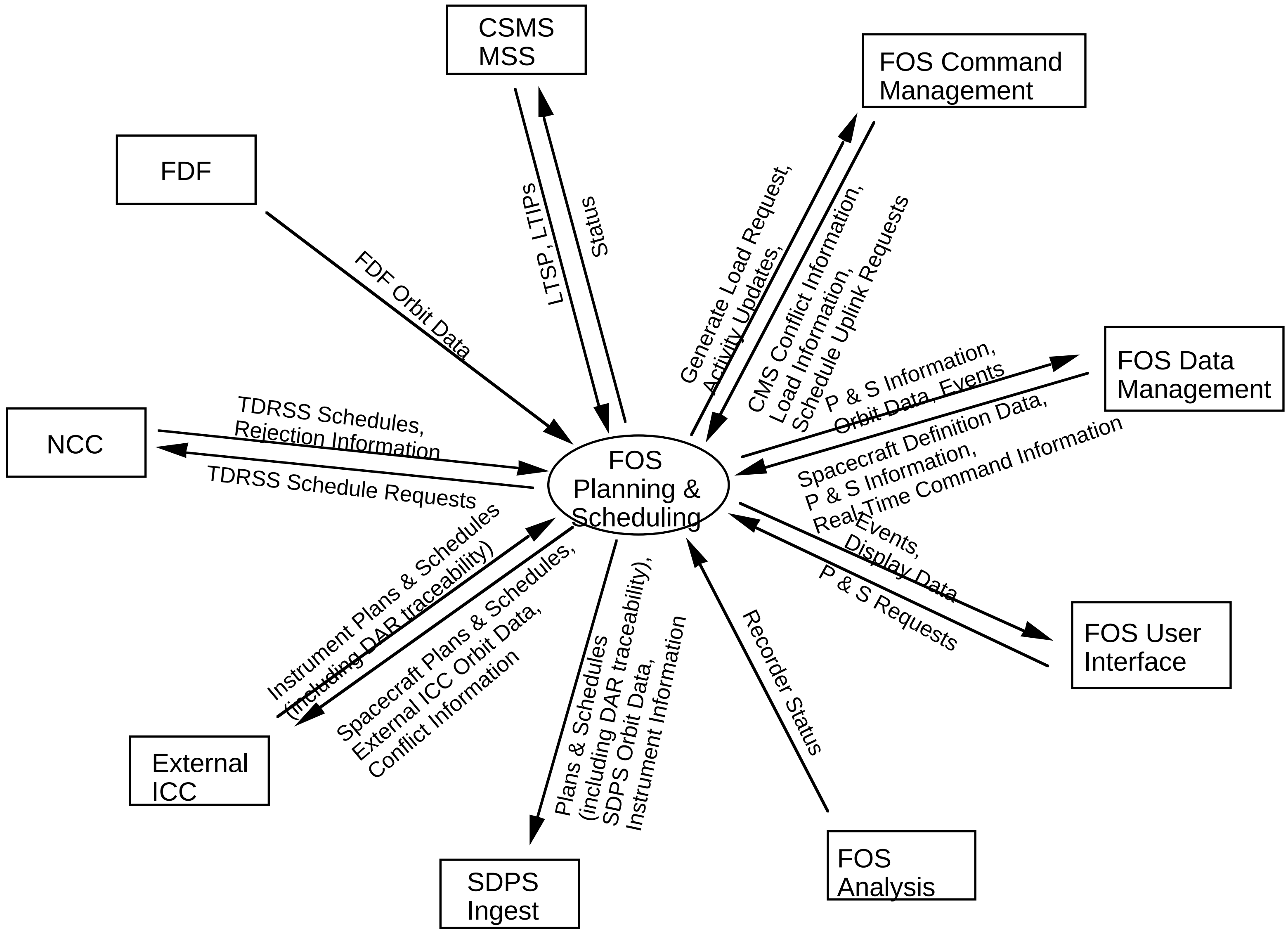


Figure 2-2. P & S Subsystem Context Diagram

Analysis creates a connection with planning and scheduling process and sends planning and scheduling a message containing a start time associated with the start of a playback for the solid state recorder buffers and the number of EDU blocks that were played back for each buffer. Then the information can be displayed on the timeline, giving the user a graphical view of the solid state recorder status.

2.3.1.15 Name Server

Since the P & S architecture consists of separate processes that communicate by message passing, these processes must have the ability to locate one another. A name server performs this function, allowing processes to exist at remote locations for distributed planning and scheduling. When a process, such as a timeline, is started within a work group, it registers address information with the name server. A process that wishes to establish communication will ask the name server for the necessary address information. Name servers can be constructed to establish connections with other remote name servers, allowing work groups to establish communications between themselves. For the ECS P & S system, the name server has been extended to provide hooks for supporting DCE.

2.3.2 Future P & S Architecture

Although not included in the phase 3 prototype, other processes were identified based upon the design analysis (see Section 2.2) and incorporated into the P & S architecture. The following processes are necessary for EOS mission planning, and may be incorporated into later prototype developments.

2.3.2.1 Command Group Builder

During the activity defining process, a user will need the capability of creating a group of commands associated with the activity. The Command Group Builder Tool will allow users to specify the command names, relative time offsets and any related parameter information (e.g. default values). Once the command group is established, a check will be performed as part of the validation process to ensure proper sequencing and timing between the commands. The user can initiate the Command Group Builder Tool from their Activity Definer display.

2.3.2.2 Uplink Scheduler

The FOS will provide the capability to schedule the uplink times for the various spacecraft and instrument loads, such as microprocessor loads and table loads. The Uplink Scheduler Tool will schedule an uplink window that indicates the time period the user would prefer the load to be sent. The time period may correspond to a specific communication contact (e.g. one TDRSS contact) or a longer time duration (e.g. a 24 hour time period). The FOT will use the uplink window for choosing a communication contact upon which the load will be sent. The user will be notified of the chosen communication contact through the timeline display.

2.3.2.3 Autoscheduler

The Autoscheduler Tool will allow the FOT/IOT to perform queries on the mission plan for specific activity types (e.g. MODIS Calibration Activities). The Autoscheduler will display the list of activities and, if desired, save the list for future reference. This allows the user to make changes to the mission plan and, for any reason, reschedule the saved list of activities at a later time. The Autoscheduler will also receive the list of activities that were removed during the generation of the conflict-free Detailed Activity Schedule. Once a saved list is no longer needed, the user may delete it with the Autoscheduler.

2.3.3 Inter-Process Communication

The distributed P & S architecture consists of separate processes that communicate by message passing. To accomplish inter-process communication, a set of software objects known as the Hughes Inter-Process Communication (HIPC) class libraries are utilized to provide peer-to-peer communications between concurrently executing processes. These processes may be running on the same computer or distributed across a heterogeneous computer network. If used with C++ processes, HIPC allows objects to be passed between processes. For external processes outside of P & S that are written in other languages (e.g. C, Ada or FORTRAN), interface handlers will facilitate the communication. For a more detailed description on the inter-process communication approach, refer to the technical paper on the Hughes Inter-Process Communication Library referenced in section 1.4.

2.3.4 Mission Planning Class Libraries

The P & S architecture is based upon the Hughes Mission Planning Class Libraries, represented by the software usage hierarchy diagram shown in Figure 2-3. The mission planning heritage code is a collection of C++ class libraries which provide baseline classes for display, inter-process communication and P & S. These class libraries are developed by Hughes and embody concepts and design elements from other Hughes developed mission management systems. As represented in Figure 2-3, the ECS P & S system is built as an extension to the Mission Planning Class Libraries. By taking advantage of tested code that encompasses mission management experience, development and test cycles for the ECS P & S system are shortened, thereby reducing overall life cycle costs (see Section 2.2) .

At the foundation of the software usage hierarchy are the Hughes Class Libraries (HCL). HCL is a collection of C++ class libraries that are used as general purpose programming utilities. HCL includes class libraries for handling collections, displays, and inter-process communication. The Hughes Mission Planning Class Libraries are built on top of HCL. These libraries include:

- Timeline Class Libraries (TCL) - framework for displaying time ordered information on a timeline graph
- Scheduling Class Libraries (SCL) - framework for development of system domain specific scheduling algorithms
- Resource Class Libraries (RCL) - framework for modeling space based, ground based and related resources

- Planning Class Libraries (PCL) - framework for the planning and scheduling of mission resources (built on top of TCL, SCL and RCL)

For configuration management, all of the heritage class libraries, along with the extensions built for the phase 3 P & S prototype, use the Source Code Control System (SCCS). Because ECS is currently establishing a project-wide configuration management system under a new product, ClearCase, SCCS will be replaced during the Phase 4 prototype.

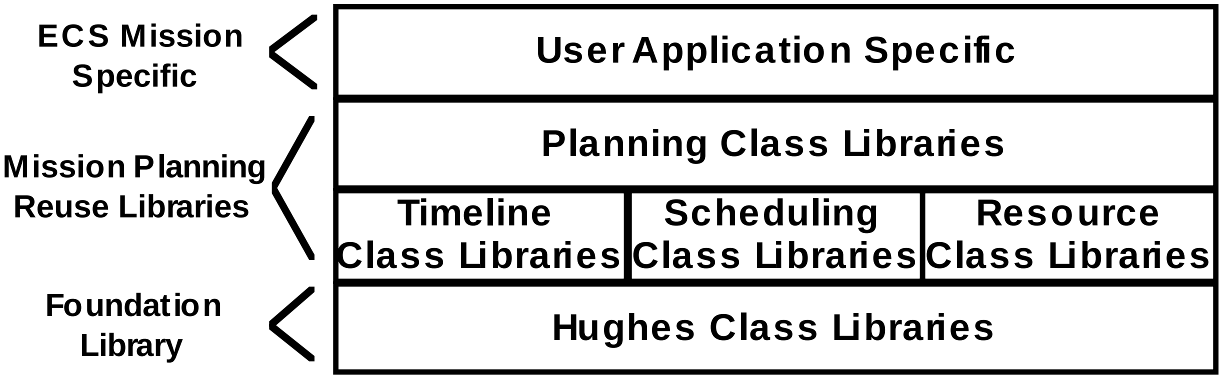


Figure 2-3. Mission Planning Class Libraries

For a more detailed description of HCL and the Hughes Mission Planning Class Libraries, refer to the P & S Prototype Design Specification.

2.3.5 Evolvability

The P & S development effort is an evolving process, where each prototype phase is considered an incremental development towards the operational system. Throughout the development process, the P & S system must be able to accept changes in the mission requirements and operational concepts. To accomplish this, the C++ programming language was utilized to take advantage of an object oriented development approach.

Object oriented programming allows the P & S problem domain to be separated into software objects (instances of classes) that encapsulate the attributes and behavior of the physical mission elements. Figure 2-4 shows the P & S class structure representing an EOS spacecraft and its components. At the bottom of the class structure are software objects that represent the physical spacecraft elements, such as the AM-1 Payload and the CERES instrument. These software objects offer a direct mapping into the problem domain, so when the science and flight operations community speaks of a spacecraft element, it corresponds to a software object that encapsulates its attributes and behavior. Encapsulation minimizes the impact due to an evolving operational concept, since changes to the requirements of a physical mission element will only affect its corresponding software object.

In addition to encapsulation, object oriented development allows inheritance of attributes and behavior. For example, the CERES class shown in Figure 2-4 inherits the attributes and behavior of its parent class ECSSimpInstr. The CERES class represents an extension to the more general parent class. As specific mission requirements become known, inheritance allows P & S extensions to be built on top of the existing knowledge contained in the class structure.

Figure 2-5 shows the object oriented representation of the AM-1 spacecraft. The AM-1 spacecraft object consists of a payload, instrument and subsystem objects. To encapsulate behavior, each instrument and subsystem object control their own allocation. The AM-1 payload object controls the interfacing between the instrument objects and the subsystem objects. For instance, when the MODIS instrument object needs subsystem resources (power, data, etc.), it asks the AM-1 payload object, which in turn asks the necessary subsystems. By encapsulating the instrument and subsystem objects in this manner, changes to instrument or subsystem operations will only affect the corresponding software object. In addition, hooks were put in place for future elements by creating objects without incorporating the specific attributes and behavior (e.g. DAS).

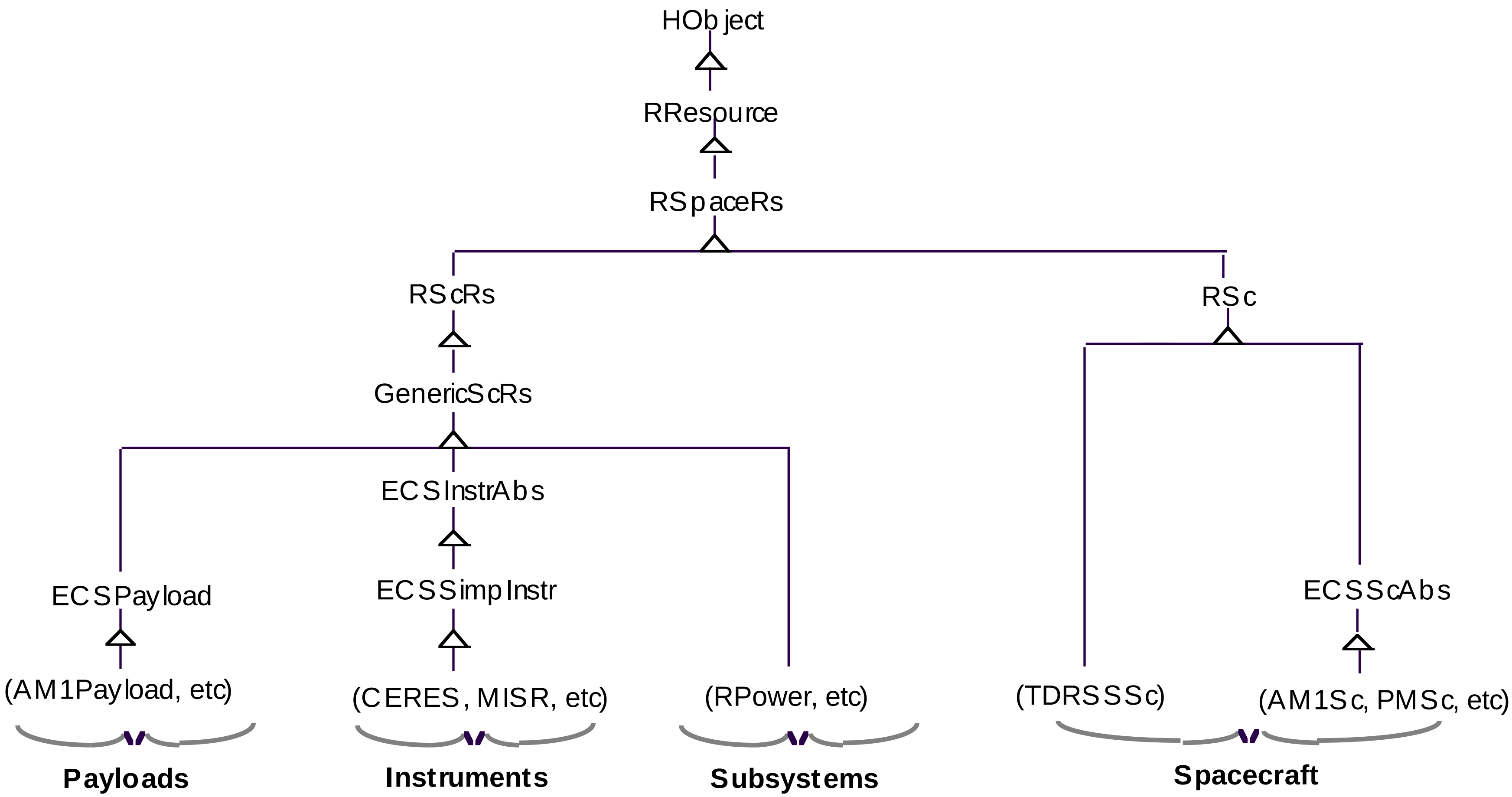


Figure 2-4. P & S Object Oriented Spacecraft Representation

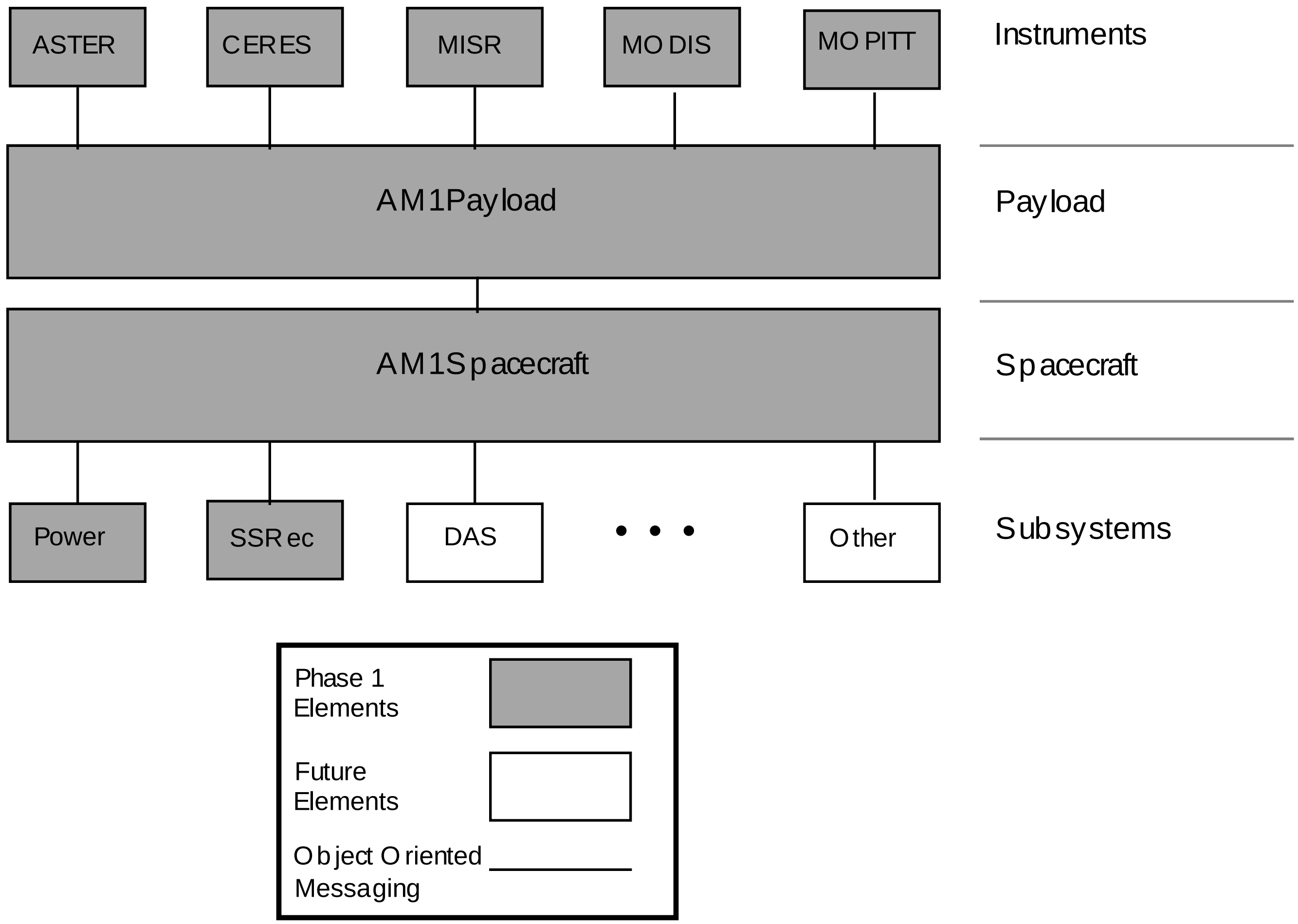


Figure 2-5. AM-1 Spacecraft Representation

2.4 Prototype Features

2.4.1 Phase 3 P & S Prototype Scenario

The phase 3 P & S prototype focused on FOS internal interfaces and further refinement of the scheduling tools that aid scheduling. The hardware setup for the FOS was based on initial operations concepts and helped in demonstrating the interfaces that were developed during the phase 3 prototype. Figure 2-6 shows the hardware and high-level software configuration.

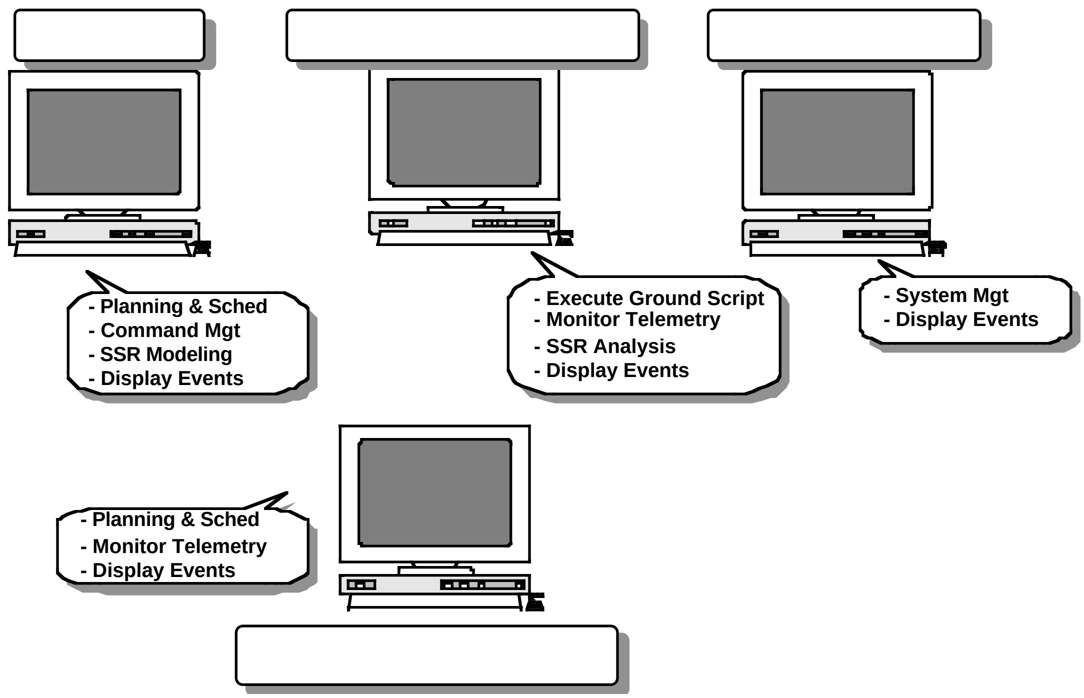


Figure 2-6. Prototype Software and Hardware Configuration

For the FOS, the AM-1 elements represented in the prototype scenario included: an EOC Scheduler, an EOC Command Activity Controller, an EOC Ground Controller and a CERES IST. Interfaces were developed between the P & S software and the FOS Command Management System, the FOS Real-Time System and the FOS Instrument Support Toolkit. These elements provide a thread through the AM-1 scheduling process, including pre-scheduling, initial scheduling, final scheduling, and generation of the Command Load and Ground Script. During the phases of scheduling, as the EOC Scheduler incorporates the instrument activities, schedule updates are broadcast from the EOC Scheduler to the CERES IST in order to provide global visibility into the mission plan. The EOC Scheduler and the CERES IST each have a basic set of tools for performing their scheduling functions. The P & S prototype scenario follows the scheduling of the CERES instrument and the basic scheduling functions performed at the EOC. Table 2-3 illustrates the P & S prototype scenario and on which hardware platform the different scheduling functions are performed.

Table 2-3. P & S Prototype Scenario

Location	Phase	Scheduling Task	Tool
	Pre-Scheduling	Goal: Create Activities from Commands and Define Baseline Activity Profiles for the CERES instruments	
EOC or CERES IST		Create a "Ceres Biaxial Scan" activity definitions	Activity Definer
EOC or CERES IST		Create a "Ceres Biaxial Scan" Baseline Activity Profile definition	BAP Definer
	Initial Scheduling	Goals: 1) To Schedule the CERES instruments for 7 days 2) To Schedule TDRSS requests from the initial schedule	
CERES IST		Create an access to schedule onto the Master EOC plan	Plan Window Manager
CERES IST		Schedule a seven day "Ceres Biaxial Scan Baseline Activity Profile"	Instrument Activity Scheduler
CERES IST		Display seven day period at one time	Segmented Timeline
EOC		FOT displays CERES FORE and AFT instruments on EOC Timeline	Timeline
EOC		FOT displays typical planned 24 hour period	Timeline
CERES IST		Schedule TDRSS Playback requests	Communication Contact Scheduler
	Final Scheduling	Goal: Schedule a Deviation from the CERES Baseline Activity Profile	
CERES IST		Copy the "Master EOC" plan to a "Calibration plan"	Plan Tool
CERES IST		Schedule a "Ceres Solar Calibration" on the "Calibration plan"	Instrument Activity Scheduler
CERES IST		Discard the "Calibration Plan"	Plan Tool
	Command Management	Goal: Create the ATC Load and Ground Script from the final Schedule	
EOC		Release the Detailed Activity Schedule and generate the ATC Load and Ground Script	Command Management Tool
EOC		Display the ATC Load and Ground Script	Text Editor

2.4.1.1 Pre-Scheduling

The majority of pre-scheduling activities take place prior to mission launch. During the pre-scheduling phase, instrument teams and/or EOC FOT members define activities that will be stored in the project database and later used for scheduling. Activities are defined as a user request for a resource (e.g. CERES instrument) to be placed in a particular configuration or mode over a specified time. Activity definitions include information such as the commands necessary to place a resource into a certain configuration and any additional information needed by the P & S software to perform modeling and constraint checking.

In addition to defining activities, certain instrument teams and/or FOT members need to create Baseline Activity Profiles (BAPs) definitions. BAP definitions are a collection of activities with additional information necessary for scheduling. The CERES, MISR, MODIS and MOPITT instruments will have a Baseline Activity Profile (BAP) that defines a repetitive sequence of activities over the spacecraft orbit cycle. The PI/TL at the IST has the responsibility of defining the BAP for their respective instrument during pre-scheduling.

2.4.1.2 Initial Scheduling

The P & S prototype provides the EOC and ISTs with initial scheduling capabilities for building a mission plan. The initial scheduling process begins approximately 3-4 weeks before the target week. During initial scheduling, CERES, MODIS, MOPITT and MISR instrument teams submit BAPs for their instruments. ASTER creates a list of activities representing the resource utilization (e.g. power, data buffer usage) that will be required for its eventual schedule submitted during the final scheduling phase.

At the CERES IST, prior to scheduling any activities, a CERES instrument scheduler will first create access to a plan on which they wish to schedule. In order to do this, the plan window manager process is used to specify a "plan window" which captures a resource for a given time period within which they wish to schedule. An access is then created which locks out other users from making modifications to that portion of the plan. Figure 2-7 shows the plan window manager process.

For scheduling, the P & S prototype provides the CERES instrument scheduler with an instrument activity scheduler process that uses the baseline activity profile definition to schedule a week of instrument activities. Figure 2-8 shows the instrument activity scheduler process. For initial scheduling, a CERES instrument scheduler would schedule a CERES BAP (e.g. Biaxial Scan Baseline Activity Profile) on one of the AM-1 CERES instruments. Once the CERES IST completes the scheduling of the BAPs, the activities are sent to the EOC and the updates are subsequently broadcast over the network to all interested parties. Figure 2-9 shows an example of the segmented timeline display that could reside at the CERES IST. With the segmented timeline, the CERES PI/TL can view the scheduled modes of operation for both the CERES FORE and CERES AFT instruments, in addition to power consumption and the data volume levels for a number of consecutive days. The CERES PI/TL can also view the schedules of the other instruments, if desired. For a more detailed description of all the prototype displays and features, refer to section 2.4.2.

Complex instruments do not have BAPs because their schedule tasking is less predictable. For the ASTER ICC, a list of activities will be submitted over the network for schedule integration at the EOC. In the prototype, a test driver representing the ASTER ICC was developed. By developing test drivers before the interface requirements are established, design risks can be minimized. Using ASTER scenarios developed by the Jet Propulsion Laboratory, an ASCII representation of an activity list was created. The format of this activity list was based upon the Upper Atmosphere Research Satellite (UARS) Daily Activity Plan format, in which keywords are utilized for representing scheduling requests.

The ASTER ICC activity list is read by the interface handler process and incorporated into the mission plan. During initial scheduling, no requests are rejected by the scheduling system. The resource model allows over-subscription of data volume since TDRSS contact times have not yet been established. If constraints are violated (e.g. spacecraft power limit is exceeded), the area of conflict is highlighted as red, but the activities are not flushed from the mission plan. This allows the instrument community to resolve constraints among themselves before the EOC needs to be involved.

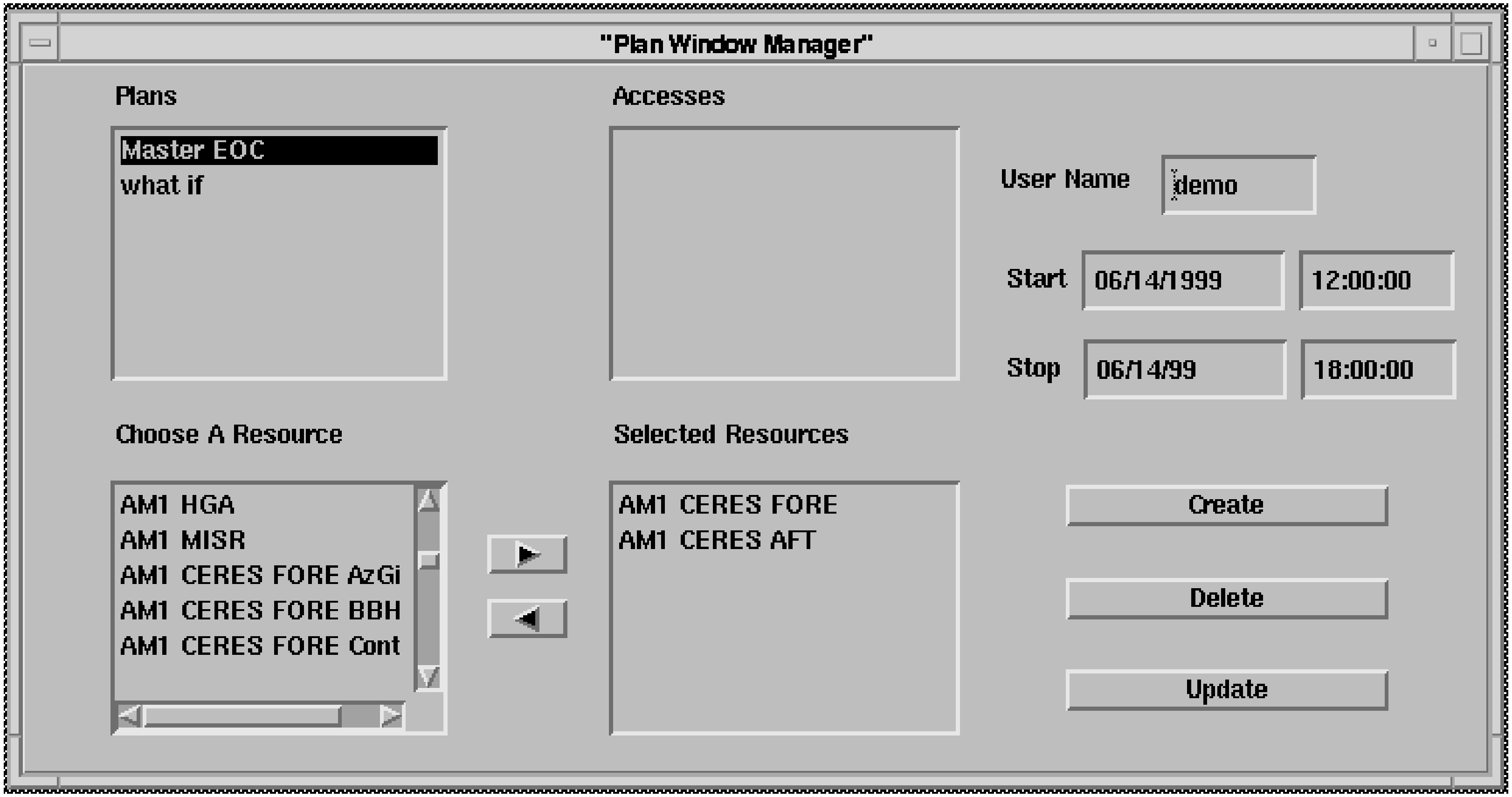


Figure 2-7. Plan Window Manager

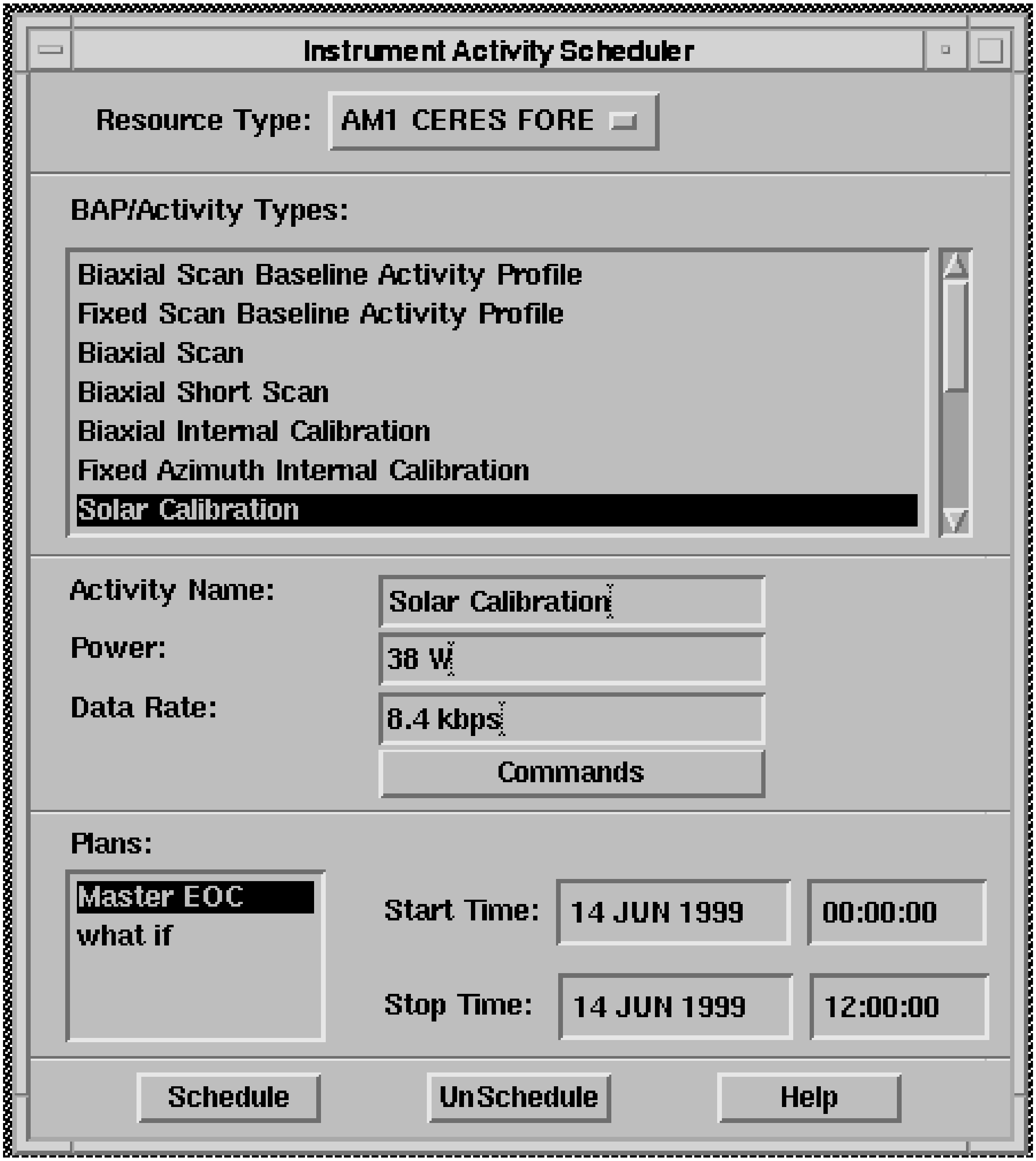


Figure 2-8. Instrument Activity Scheduler

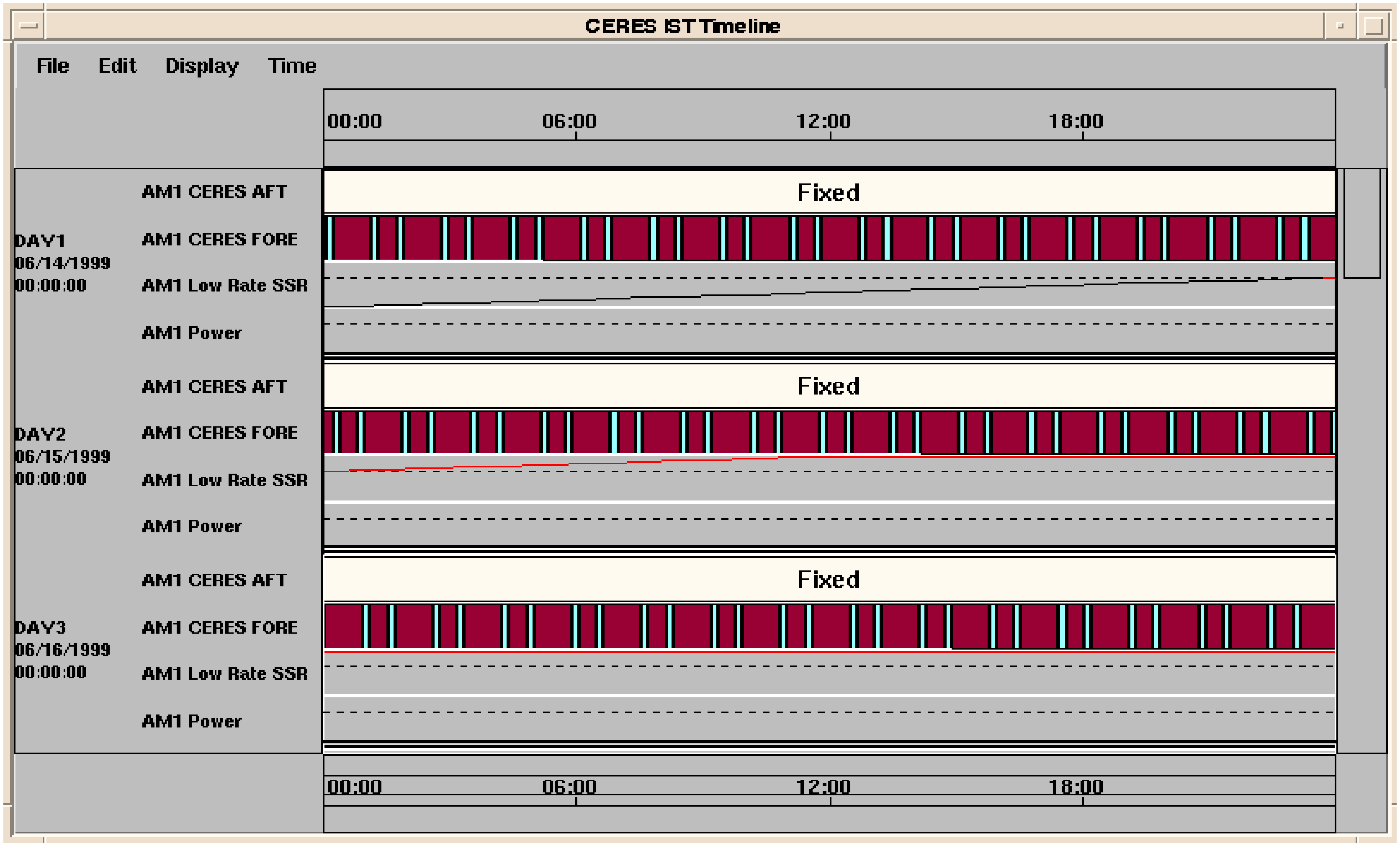


Figure 2-9. Segmented Timeline Display

Three weeks before the start of the target week, the EOC has received all initial scheduling inputs from the ISTs and ASTER ICC. Using the integrated schedule, the EOC scheduler can use their timeline display to analyze data volume limits. Figure 2-10 shows a timeline view where the EOC scheduler can view the data volume needs of the CERES instrument. "AM1 Low Rate SSR" represents the solid-state recorder buffer where CERES stores its data. The dashed line corresponds to the data storage limit that is available in the buffer while the two-dimensional line plot shows the data volume as a function of time. When the two-dimensional line plot exceeds the dashed line, the plot color turns red, indicating that data will either be overwritten or lost.

To avoid any data loss, the EOC scheduler must negotiate with the Network Control Center for available TDRSS contact times. During the TDRSS contacts, the solid-state recorder buffers are played back at a rate of 150 kbps, thereby reducing the amount of data volume. In the prototype, a communication contact scheduler was developed. The communication contact scheduler creates a schedule that represents nominal contact times in which each AM-1 orbit has two 10 minute contacts that are equally spaced. From these contact times, playbacks for each of the solid state recorder buffers are scheduled to model the downlink of instrument collected data. As the playbacks are scheduled, the two-dimensional line plots corresponding to the data volume adjust accordingly.

At one week before the target week, the final TDRSS schedule is ingested into the mission plan. All schedule updates are made available to the ISTs and ASTER ICC. The PIs/TLs can view the mission plan through their timeline displays. If TDRSS contact times do not satisfy the data volume needs of an instrument, the timeline data volume plot will exceed the dashed line and have a red color, indicating that data will either be overwritten or lost. This notification allows instrument schedulers to resolve constraints amongst themselves without involving the EOC. However, in nominal TDRSS scheduling, data volume conflicts should not be an issue.

2.4.1.3 Final Scheduling

Between seven and two days before the target day, the final scheduling process takes place. During this time period, deviations to instrument schedules can be incorporated into the mission plan as long as they stay within the instrument's resource envelope. The final scheduling process is very similar to initial scheduling.

Both at the EOC scheduler and at the CERES IST, a planner has the flexibility to schedule on alternate plans in order to analyze the impact of deviations to the schedule. Instead of scheduling deviations directly to the Master EOC plan, a CERES instrument scheduler can use the plan tool to copy the Master EOC to a new plan (e.g. "what-if"). Deviations can then be scheduled on the new plan, and any impacts investigated before submitting them to the EOC.

Similar to scheduling BAP and initial activities, the PI/TL has three ways of submitting schedule deviation requests to the EOC:

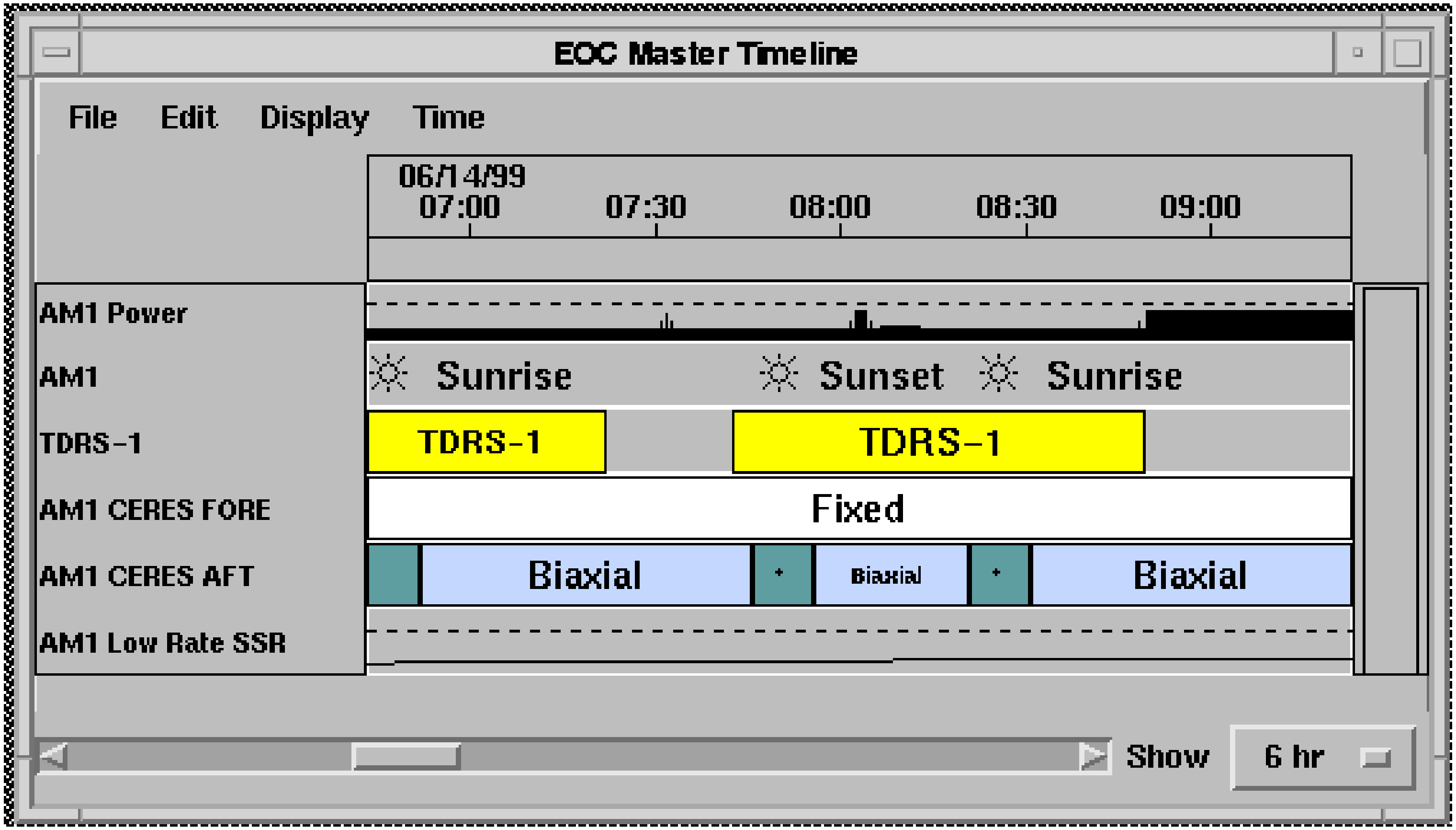


Figure 2-10. Timeline Display

1) Utilize the instrument activity scheduler tool for scheduling the deviation. For example, the CERES PI/TL can use the instrument activity scheduler shown in figure 2-8 to schedule a solar calibration activity over a specified time interval. This activity would represent a deviation to the CERES BAP.

2) Submit an e-mail request to the EOC specifying the activities, start times and stop times. The EOC scheduler would receive this e-mail message and utilize the instrument activity scheduler tool for scheduling the deviation to the BAP.

3) The PI/TL can build an activity deviation list in an ASCII format. For the prototype, this format is based upon the UARS Daily Activity Plan, where keywords are used to specify scheduling requests. This PI/TL submits this deviation list to the EOC where it is incorporated into the mission plan through a scheduling filter.

Likewise, activity deviations from the ASTER ICC are represented in an ASCII keyword file. An interface handler process at the EOC incorporates ASTER activity deviations into the mission plan.

All schedule deviations that occur in the mission plan are broadcast by the EOC to interested parties. This allows the geographically distributed ISTs to view the current state of the mission plan through their timeline displays.

2.4.1.4 Command Load Generation

At two days before the target day, the EOC makes the integrated schedule available to the Command Management Subsystem for command load generation. In order to accomplish the generation of the Absolute Time Command (ATC) load and ground script, the command management tool is used by an EOC scheduler. Figure 2-11 shows the command management tool. An EOC scheduler can specify the size of the command load to be generated by entering start and stop times and then release the schedule within this time window to the command management subsystem. When a plan is released, all activities that are in conflict are removed from the schedule and the remaining activities are sent to CMS to be expanded into commands. The activity definitions, originally stored within the project database via the activity definer process, are used for determining the expansion.

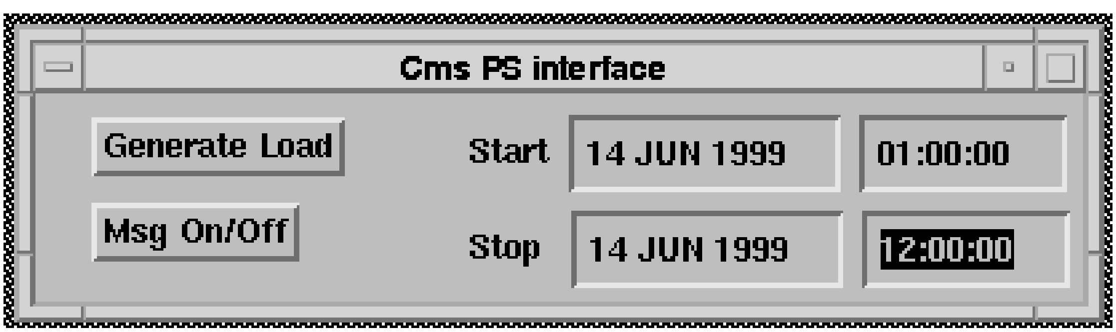


Figure 2-11. Command Management Tool

2.4.2 Prototype Displays and Features

2.4.2.1 Activity Definer

The activity definer provides the user with a graphical interface for creating and modifying activity definitions. Figure 2-12 shows an example of the activity definer display.

In order to create or modify an activity definition, the user must first select the instrument or subsystem for which they wish to create and modify definitions. The list of pre-existing activity definitions appear in a scrolled list. If the user wishes to modify previously defined activities, they simply click on the activity they are interested in and the definition's information appears in the display. The user may modify the parameters and save the changes to the project database by hitting the Save button. The changes may be discarded by hitting the Reset button.

If the user wishes to create a new activity, they may do so by one of two ways. First, they can select a previously defined activity and change the name field to the name of the new activity. Secondly, they may create a new activity from scratch. The user can modify any of the definition's parameters or its command list. The command list is modified by hitting the Commands button, which brings up a command list editing window. In this window, the user can add and delete commands in the activity, set the times the commands will execute (relative to the start or stop time of the activity), and change the default values of the command parameters. User modifications can be saved or discarded by hitting the Save or Reset buttons, respectively.

2.4.2.2 BAP Definer

The BAP definer provides the user with a graphical interface for creating and modifying Baseline Activity Profile definitions. Figure 2-13 shows an example of the BAP definer display.

The user selects from a list of instruments and subsystems in order to create or modify a BAP for that resource. The activities for that resource appear in a scrolled selection list called the Activity Types list. The user may add activities to the BAP by selecting an activity in the Activity Types list and hitting the Add button. When an activity is added to the BAP, it appears in the BAP Activity list. The user may delete activities from the BAP Activity list by selecting the activity in the list and hitting the Delete button. The user may also edit parameters for the activity and provide relative start and stop time for that activity by hitting the Edit button.

The user must also specify the name they wish to assign to this BAP definition. The user's changes may be saved by hitting the Save button. The changes may be discarded by hitting the Reset button.

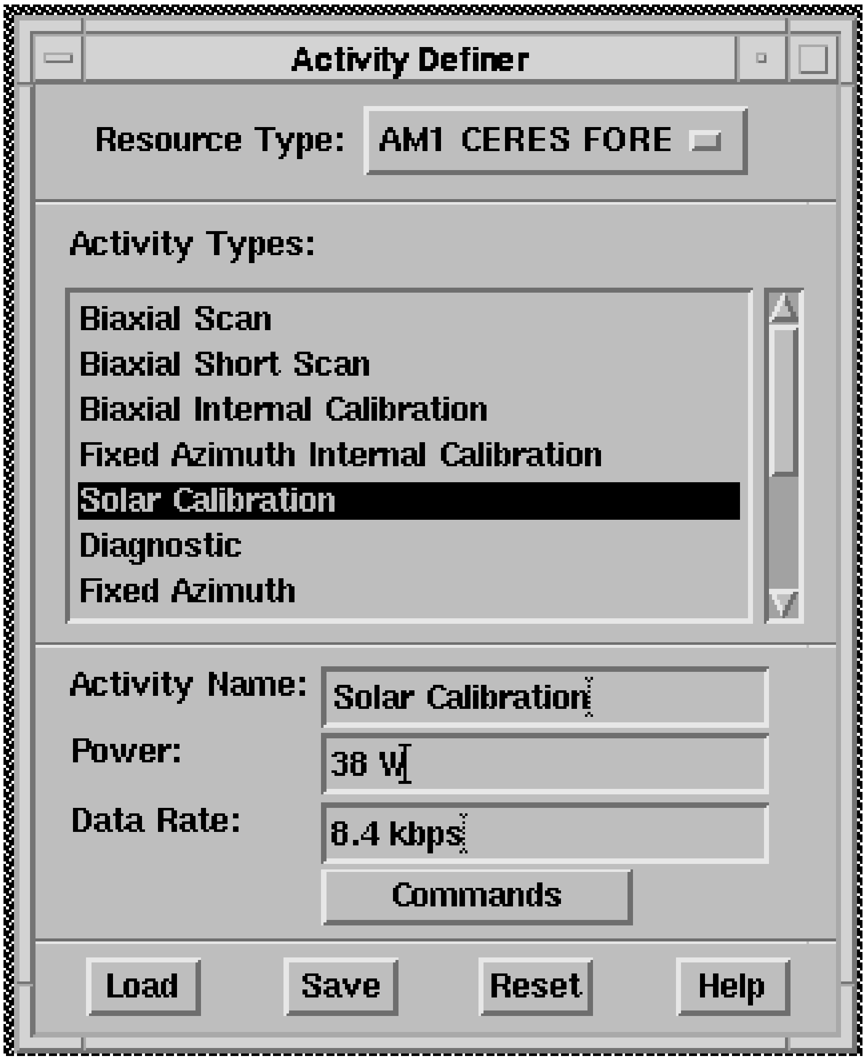


Figure 2-12. Activity Definer

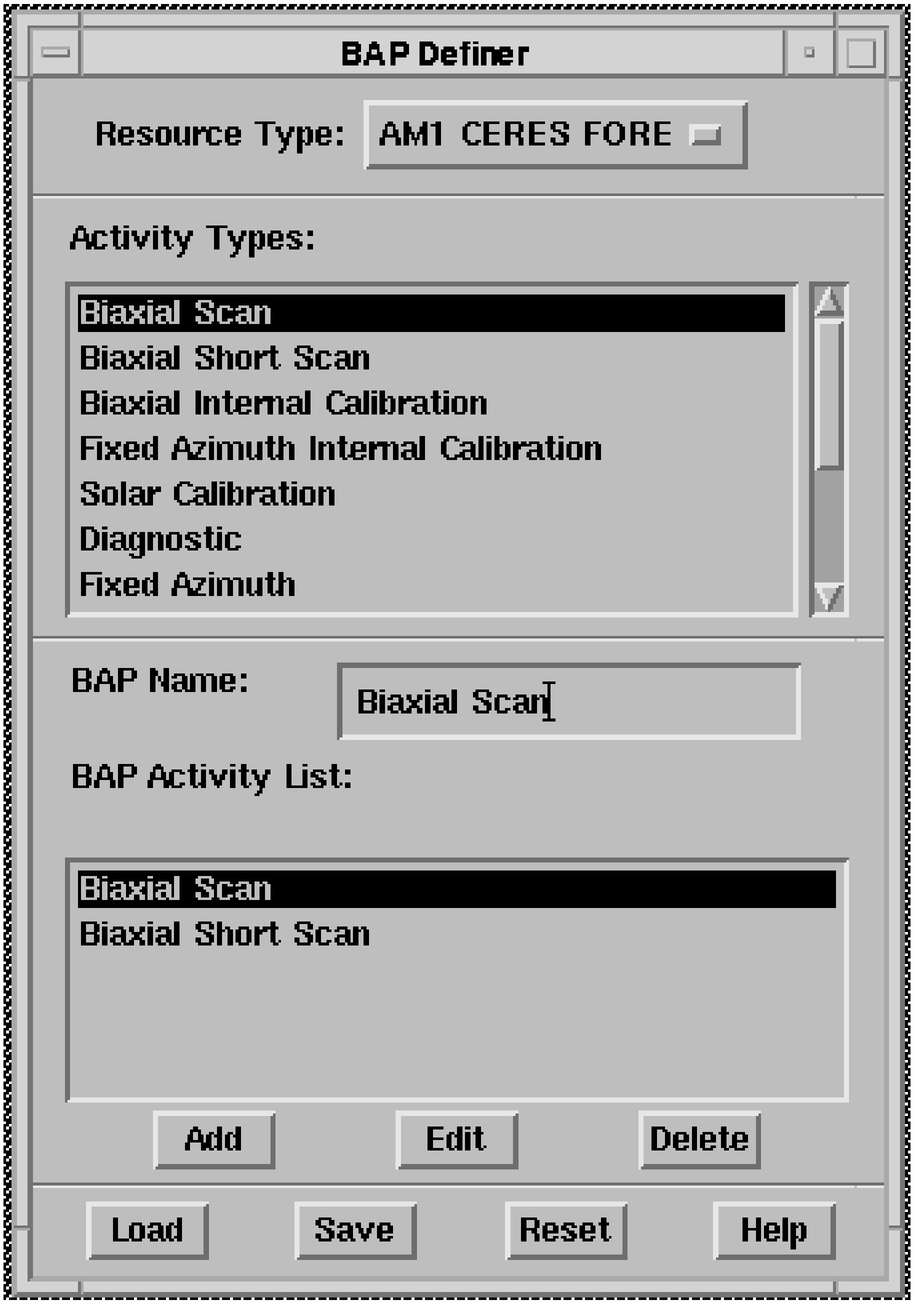


Figure 2-13. Baseline Activity Profile Definer

2.4.2.3 Timeline Display

The timeline process presents, a graphical color representation of instrument and subsystem activities as a function of time. Figure 2-14 gives an example of the timeline display. A time axis appears on the top of the display showing the current time range. A scroll bar appears at the bottom of the display that allows the user to scroll back and forth through the current time window. To scroll through time, the user holds down the leftmost mouse button and drags the scroll bar either right or left, depending on the range of time that the user wishes to view. As the user drags the scroll bar, a popup text field gives the time range to which the display will adjust after the user releases the mouse button. The user can also click anywhere in the scroll bar trough and the time range will be adjusted by a default amount of time.

To adjust the range of the current time window, the user can use the menu entitled "Time" and choose the option "Change Plan Window". This action will present the user with a display that will allow the user to enter start and stop times to change the timeline window boundaries. The user can then choose one of the option buttons to either confirm, apply or cancel the time range change.

To give the user further control of display views, the timeline also gives a method of limiting the time range that is currently displayed. The "Show" option menu in the lower right-hand corner of the timeline controls the range of time that can currently be displayed. This will allow users to zoom in on a particular region of interest. The user can click on the menu and choose the length of time to be displayed.

The left hand side of the timeline process shows the various resources that can be scheduled. The data in the center represent the resource states. The rectangular boxes are scheduled activities, each with the name of the activity as the label. The timeline process displays an asterisk when an activity label is too small to appear. Activity types can be represented with a specific color. This aides the planner and scheduler by providing easy recognition. A description of the different types of resources are shown below:

> Instruments - All of AM-1's instruments are represented on the prototype timeline display (refer to figure 2-10). A simple list of activities were developed based upon the instruments' modes of operation. For activities that have a data rate and consume power, scheduling will impact the associated SSR buffer and spacecraft power subsystem.
>
> Solid State Recorder Data Buffers - These resources represent data storage areas for instrument science and housekeeping data. The dashed line corresponds to the data storage limit that is available in the buffer, while the two-dimensional line plot shows the data volume as a function of time. When the two-dimensional line plot exceeds the dashed line, the plot color turns red, indicating that data will either be overwritten or lost. Whenever a user schedules an instrument activity which consumes buffer space, the amount of consumption is reflected on the corresponding buffer. When a user schedules a TDRSS contact (also known as a TDRS playback request), the data volume plot reflects the contact use by sloping downward indicating that data is being downlinked and thus space is made available in the buffer for future data collection.

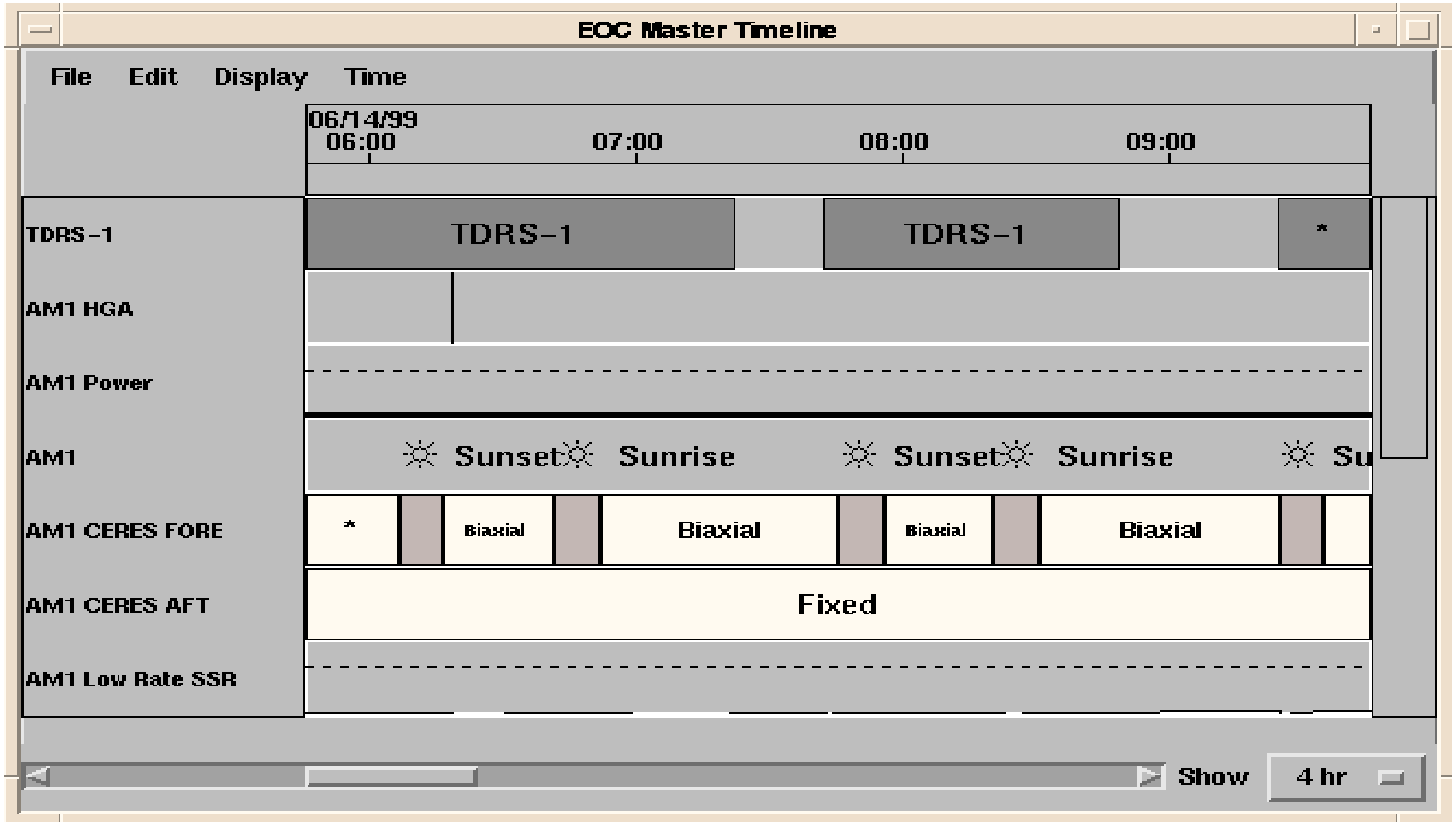


Figure 2-14. Timeline Display Example

AM1 Power - This resource corresponds to the power subsystem on the AM-1 spacecraft. The AM-1 power subsystem permits multiple resources to utilize its power output. The dashed line corresponds to the power consumption limit that is available for the entire spacecraft while a histogram represents the power that is currently being consumed. When the histogram exceeds the dashed line, it turns red, indicating a power consumption constraint. Whenever a user schedules an instrument activity which consumes power, the amount of power consumption is appended to the power histogram.

TDRS-1 - This resource represents an example of a Tracking and Data Relay Satellite that will be used to downlink data from the EOS spacecraft. Shown on the timeline are activities which represent scheduling windows for TDRS that indicate when the EOS spacecraft is in view of the TDRS spacecraft. The scheduling windows show available periods for requesting TDRS contacts in which to uplink loads or downlink data. The scheduling periods are provided by the Flight Dynamics Facility (FDF).

AM1 HGA - This resource corresponds to activities related to the EOS spacecraft's high gain antenna. In the current planning and scheduling prototype, this resource only shows contact activities between AM1 and TDRS. The contact activities consist of transfer of data to TDRS (or uplink of loads from TDRS to AM1) and only occur during a TDRS scheduling window described previously.

AM1 - This resource displays orbit event information such as sunrise and sunset times. This will allow a flight operations planner to schedule instrument and spacecraft subsystems based on spacecraft activities. For example, the CERES scheduler will schedule the CERES biaxial short scan activities at each sunrise and sunset time. Event information is usually provided by the FDF.

Another feature of the timeline allows the user to change the display configuration. A menu in the upper left-hand corner titled "Display" contains two options, one to change colors and another to change resources. When the user chooses the "Change colors" option, a color palette window is presented to the user along with a scrollable list of activity names. The user may click on an activity name and then choose a color. As soon as the user clicks the confirm or apply button the chosen type of activity will appear in the chosen color on the timeline resource line. Activities of the same type that are scheduled later will also show up in the chosen color. The "Change resource" option allows the user to selectively choose the resources that they would like to view on the timeline. If the user selects this option, a window is displayed that allows the user to add or remove resources to be viewed from the list of available resources. After the user clicks the confirm or apply button, the timeline is redisplayed showing the selected resources.

The timeline also has a file menu in the left-hand corner that contains options for the user to either load a timeline configuration, save a configuration or get a hard copy of the timeline. If the user selects the save configuration option, a file dialog panel is displayed that allows the user to save the current configuration of the timeline. The timeline configuration consists of the colors and resources that are displayed to the user and that are dynamically modified by the submenus of the Display options described previously. The load configuration option allows the user to load a previously saved configuration. Note that the definition of a configuration in this context only contains the types of resources and activity colors, not the actual activities and their occurrences on the timeline.

2.4.2.4 Segmented Timeline Display

The segmented timeline process presents a graphical color representation of instrument and subsystem activities as a function of time in a similar fashion to the regular timeline. Figure 2-9 gives an example of the segmented timeline display. A time axis appears on the top of the display showing the current time range. A rectangular scroll bar appears at the bottom of the display that allows the user to scroll back and forth through the current time window. By sliding the scroll bar back and forth with the mouse, the user can change the time period of the displayed activities. In addition, a user can shrink or enlarge the scroll bar, effectively zooming in and out of time.

The left hand side of the timeline display shows a set of resources for each segment. Each segment is a pre-determined length of time defined in the segment timeline database. This feature allows the user to view multiple segments on one display in order to give him the capability to see repetitive cycles of scheduled activities. Activities are displayed as rectangular boxes similar to the regular timeline.

Other features such as displayed resources, configuration changes and loading and saving configurations are similar to the capabilities of the regular timeline that was described in section 2.4.2.3

2.4.2.5 Instrument Activity Scheduler

The instrument activity scheduler is a process that schedules individual activities or lists of activities, called baseline activity profiles (BAPs), over a given time range. Figure 2-8 shows an example of the scheduling display. The user schedules an activity or a baseline activity profile by first selecting the resource on which to schedule the activity. Once the resource is selected, a list of activity types the user can schedule will appear. The user then selects one of the activities or one of the baseline activity profiles. A plan upon which to schedule must be selected and a start and stop time must be entered, if the default values are not desired. Once all of the above has been performed, the user hits the Schedule button to schedule the activity or the baseline activity profile.

2.4.2.6 Communication Contact Scheduler

The communication contact scheduler is a tool which allows the user to schedule requests for TDRSS and ground contacts. This phase of the prototype does not provide a mechanism to process the responses from the NCC concerning those requests.

Figure 2-15 shows an example of the Communication Contact Scheduler display. The user provides the following information: the plan they wish the contact requests to be scheduled on and the start and stop times for which contact times are needed. When the user hits the Schedule button, the scheduler determines TDRSS availability based on the times that the TDRSS is in view of the spacecraft and schedules contact requests over the given time range. By hitting the Unschedule button, the user may remove contacts over a given time range on the given plan.

2.4.2.7 Plan Tool

The plan tool interface is used to create, delete, and copy plans. It also has a function to snap to a plan. Figure 2-16 shows an example of the plan tool display.

To create a new plan, the user hits the Create button and a window is displayed that prompts for a plan name. The user enters the plan name and then hits the Apply button. The new plan name appears in the plan list on the main window.

Another function allows the user to delete a plan. The user selects the plan from the list and then the user hits the Delete button. When a plan is deleted, all activities and associated resources on that plan are also deleted. The plan name that the user selected is removed from the list indicating a successful plan deletion.

The third function offered by the plan tool allows the user to copy one plan to another. The user first selects the plan to be copied from the plan list and hits the Copy button. A window requesting information about the destination plan is presented. The user then enters the destination plan name and (optional) start and end times for the portion to be copied. If no start and end times are specified, the entire source plan is copied to the destination plan. If the destination plan already exists, the source plan overwrites the destination plan. After the user enters the destination information, the Apply button is hit. The destination plan name is added to the plan name list in the main window.

Another function provided by the plan tool lets the user "snap" to a plan. "Snapping" to a plan allows the user to broadcast the plan to any tool that requires knowledge of a current plan such as a timeline. When the user "snaps" to a plan, visualization tools, such as a timeline, will immediately display the activities and resources associated with the "snapped" plan. This allows all tools that have visibility into a plan to display views of the same plan. The user clicks the snap button on the main plan tool display and a window appears, prompting the user for a time range in which to snap. If no time range is selected, the snap will cover the entire plan. The user then clicks the apply button to initiate the snap and all tools having visibility into the snapped plan will change to that plan.

2.4.2.8 Plan Window Manager Display

The plan window manager display tool is a graphical user interface that manages user access to plan resources. User accesses are required if a user wishes to update a plan (i.e. schedule activities on a plan). This will prevent users from attempting to update the same portion of a plan at the same time. Figure 2-7 shows an example of the plan window manager display.

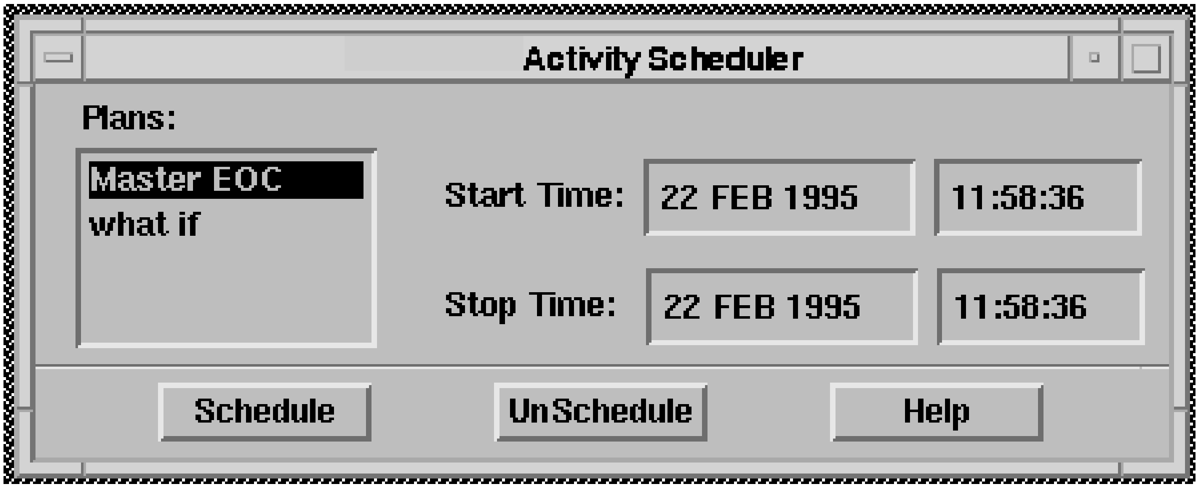


Figure 2-15. Communication Contact Scheduler

Figure 2-16. Plan Tool

The window contains areas that show a plan list for which accesses may be created, a resources list, an accesses list and a selected resources list. To create an access, the user first selects a plan from the plan list. Then the user selects resources from the resource list and clicks the add arrow to add resources to the selected resources list. They then enter the start and stop times of the portion of time on which they wish to make modifications to the plan. Finally, the user hits the Create button and the new access is added to the accesses list. By choosing an existing access from the access list, the user may call up the information about the selected access (i.e. the resources and time span). Then the user may make modifications to the resources and timespan (by either changing the time span or selecting different resources) and hit the Update button to make an update to an access. Also, a user may delete the access by choosing an existing access and hitting the Delete button.

2.4.2.9 CMS Interface

The command management interface is a graphical user interface for releasing the detailed activity schedule to Command Management. The interface allows the user to enter a time period for the detailed activity schedule and press a Generate Load button with the mouse. When this button is pressed the detailed activity schedule is passed on to command management which generates an ATC load and a ground script. Figure 2-11 shows an example of the command management tool.

2.4.3 Multi-Platform Capability

During the phase 3 prototype, all P & S code was ported to four major platforms including Sun, DEC, SGI and HP. Because the P & S software is built upon the existing Hughes Class Library foundation, no ECS specific code needed modification in order to run on any of the other platforms. For the phase 3 prototype scenario, all processes were executing on Sun workstations, but when operational, the P & S system will function in a heterogeneous computer environment. Different platforms and operating systems may comprise the P & S infrastructure. In order to establish flexibility, the ECS P & S software must be maintained for different platforms and operating systems. In order to accomplish this, nightly builds are run on each of the major platforms to catch any platform dependencies immediately.

This page intentionally left blank.

3. User Feedback

3.1 User Feedback Approach

During the phase 3 P & S prototype, comments and issues raised by the science and flight operations community were acquired in various ways. During the PRR demonstration, as well as informal demonstrations of the phase 3 P & S prototype, raised questions and comments were recorded. In addition, issues that surfaced during splinter meetings with the ECS PRR attendees were documented for later review.

3.2 Results To Phase 1 User Issues and Responses

Table 3-1 summarizes the primary issues pertaining to the phase 1 P & S prototype. In the table, an originator number corresponds to the source of each issue. The following originator categories are identified:

1 - Instrumenter
2 - Spacecraft Manufacturer
3 - Flight Operations Team
4 - Project

After review and analysis, a response was demonstrated through the P & S phase 2 prototype for each of these primary issues.

Table 3-1. Phase 1 User Feedback Results (1 of 3)

Issue	Orig	Response	Rationale
Establish initial command management interface	3	Developed the interface design with command management. Incorporate initial interface concepts into the phase 3 prototype.	Command management is one of the primary interfaces with P&S. By prototyping the initial concepts of this interface, potential risks can be minimized.
Provide a preliminary detailed activity schedule	4	Established initial format and contents of a detailed activity schedule by the phase 2 prototype, based upon discussions with command management. Provide initial detailed activity schedule in Phase 3 prototype.	The detailed activity schedule is the primary P & S product. By establishing an initial format and content, early feedback can be acquired from command management and other FOS elements.

Table 3-1. Phase 1 User Feedback Results (2 of 3)

Issue	Orig	Response	Rationale
Represent timeline activities at the command level	4	Developed design concepts with command management. Establish initial timeline command representation for a single instrument by the phase 3 prototype.	Prototyping the timeline command representation for a single instrument will provide the instrument community a conceptual demonstration.
Establish the preliminary ASTER interface concepts	1	Established initial ASTER interface test driver for phase 2 prototype.	The ASTER interface presents a critical risk for the P&S system. Providing an early P & S test bed for the ASTER interface will help mitigate these risks.
Architecture flexibility in allowing an external element to provide their own P & S components	4	Section 2.3.2 refers to the P&S IPC approach for handling external element message passing over the network. Ongoing discussions are taking place with instrument groups (primarily MISR and ASTER) for defining external P & S components. In addition, the phase 2 prototype incorporated an initial ASTER interface test driver.	Because instrument groups may be providing their own P&S components (e.g. ASTER), the P & S architecture must have the flexibility to incorporate them. As details are established, evolving interface test drivers will minimize the impact of these external components.
Incorporate the complete instrument manifest into the P&S prototype	1	For the phase 2 prototype, all AM-1 instruments were represented in the P & S system	Establishing an initial P & S representation for all AM-1 instruments will allow scheduling features to evolve as details are established from the instrument community feedback.
Response of the system to sudden TDRSS contact changes	2	Established a preliminary operational scenario that demonstrates the concepts and potential software tools for handling TDRSS contact times in the phase 2 prototype. Additional concepts and preliminary tools to respond to sudden TDRSS contact changes to be incorporated into phase 3 prototype.	TDRSS contact changes will affect data management, potentially resulting in loss of data. Providing tools for operator recovery will minimize impact. Early conceptual approach will provide science and flight operations community feedback into design.

Table 3-1. Phase 1 User Feedback Results (3 of 3)

Issue	Orig	Response	Rationale
Provide 2-D plots on the timeline display for representing SSR data volume and power consumption.	3	Developed 2-D line plots on the timelines for the phase 2 prototype to represent SSR data volume. In addition, developed histogram plots on the timelines for power consumption (shown at PRR).	Presenting data volume, power and other resources as graphical information (e.g. 2-D plots) will aid operator interpretation of mission resources.
Incorporate a graphical map analysis tool into the P & S system	1	Initial map analysis tool for the phase 2 prototype was presented at the PRR.	Displaying mission resources over space and time will benefit user analysis. By incorporating an early map tool, features will evolve based on input from the instrument groups.
Enhance the timeline display to reflect the scheduling needs of the instrument community	1	Incorporated timeline concepts for AM-1 instruments for phase 2 prototype. Currently, there are ongoing interaction with the instrument groups for defining their individual scheduling needs. Based on feedback, incorporate display concepts into phase 3 timeline prototype.	The varying scheduling needs of the instrument community will require a flexible timeline process to incorporate evolving display features.
Allow user control in tailoring the look and feel of the timeline display	1	For phase 2 prototype, incorporated simple tailoring of the timeline display through configuration files. Allow interactive user control for phase 3 timeline prototype.	Simple tailoring of the P & S displays (e.g. timeline) represents one of the primary user needs for performing day-to-day operations. By avoiding the hard coding of display features, the user can control their environment through interactive display selections and configuration files.
Transition the graphical user interface from Openlook to MOTIF	4	Ported P&S prototype to MOTIF for phase 2.	As established by the Common Open Systems Environment, MOTIF represents the primary GUI standard for running in a heterogeneous workstation environment.
Utilize paper scenarios to provide a complete thread through the P & S system	3	Scenario demonstration for presenting P&S concepts (e.g. timeline displays) was presented at the phase 2 PRR.	Providing scenarios of display concepts will allow early user feedback without the need for software prototypes.

3.3 Phase 2 User Issues and Responses

Based on the feedback provided by the science and flight operations community, Table 3-2 shows the issues pertaining to the phase 2 P & S prototype. In the table, an originator number corresponds to the source of each issue (refer to section 3.2 for originator categories). After review and analysis, a response was determined by the P & S team for each of these primary issues.

Table 3-2. Phase 2 User Issues and Responses (1 of 3)

Issue	Orig	Response	Rationale
Focus prototypes on risk areas.	2	Phase 2 prototype addressed several areas of risk (Section 2.1 and 2.2) including the ASTER and NCC interface (Section 2.3.1), distributed scheduling (Sections 2.3.1, 2.3.2, and 2.3.3), and evolvability (Section 2.3.4). The phase 3 prototype plans will address additional areas of risk including the CMS interface and baseline activity profile development.	The ASTER/NCC interface presents a critical risk for the P&S system. Providing an early P & S test bed for the ASTER/NCC interface will help mitigate these risks. Because instrument groups may be providing their own P & S components (e.g. ASTER), the P & S architecture must have the flexibility to incorporate them. As details are established, evolving interface test drivers will minimize the impact of these external components. Command management is one of the primary interfaces with P&S. By prototyping the CMS interface concepts, potential risks can be minimized.
Provide a flat map with target pointer to output a list containing orbits with ground track across target point.	1	The Phase 2 map analysis tool prototype provides a 2-dimensional map projection and displays geographically related data (Refer to sections 2.3.1.3 and 2.4.2.3). In addition, the scheduling analysis tool provides the capability to overlay spacecraft related data (such as ground tracks and instrument FOVs) onto the map projection. Currently, a user can determine visibility opportunities to targets by manually changing view times.	Displaying mission resources over space and time will benefit user analysis. By incorporating an early map tool, features will evolve based on input from the instrument groups.

Table 3-2. Phase 2 User Issues and Responses (2 of 3)

Issue	Orig	Response	Rationale
Investigate impact of the proprietary nature of DELPHI software.	4	DELPHI is a licensable product, which is maintained by Hughes and currently used on approximately 10 programs.	The DELPHI class libraries embody concepts and design elements based on several mission management systems. This code has been thoroughly tested, thus reducing overall life cycle P&S costs.
Identify any needs for multi-mission resource management and implement the necessary hooks	3	The phase 2 prototype provides the necessary hooks to re-use common behavior among multi-mission resources (e.g. instrument types, power subsystems) (for further details, refer to section 2.3.4)	A flexible, extensible object class structure that will support future system upgrades and changes is necessary for an evolvable system.
Establish some examples of the interface with the command management subsystem	4	Currently defining the interface with command management. Incorporate initial interface concepts into the phase 3 prototype.	Command management is one of the primary interfaces with P&S. By prototyping the initial concepts of this interface, potential risks can be minimized.
Integrate P & S prototype with User Interface IST prototype.	3	Coordination is currently taking place with the User Interface IST for integrating the prototypes by the phase 3 prototypes.	Establishing the interface between P & S and User Interface IST is necessary for providing the IST with an integrated set of FOS tools.
Transition to DCE for distributed communication and study role of CORBA.	4	Coordination is currently taking place with CSMS for integrating DCE into the phase 3 prototype. We are also investigating the role of CORBA to provide distributed P&S processes.	Establishing the DCE interface in P & S for distributed communication is necessary for providing the IST with an integrated set of FOS tools using CSMS as the infrastructure. The migration path to CORBA may play an essential part with the CSMS infrastructure.
Map display colors are difficult to look at.	1	The phase 2 map analysis tool prototype provides the capability for the user to select display colors.	Simple tailoring of the P & S displays (e.g. map) represents an important user need for performing day-to-day operations. By avoiding the hard coding of display features, the user can control their environment through interactive display selections and configuration files.

Table 3-2. Phase 2 User Issues and Responses (3 of 3)

Issue	Orig	Response	Rationale
Provide information sharing between ISTs and EOC.	1	The phase 2 prototype presented a common toolset at local (EOC) and distributed (IST) nodes that graphically depict the state of the mission plan, thus providing global visibility into spacecraft subsystem and instrument plans/schedules.	Global visibility into the P & S information represents one of the primary user needs for performing day-to-day operations. Visibility into all schedules allows for early conflict resolution.
Provide local control of locally owned resources.	1	The phase 2 prototype provides a common toolset at all nodes that allows for local control of locally owned resources. For example, a PI/TL at their IST can modify their instrument's schedule by building activity deviations (refer to sections 2.4.1.1 and 2.4.1.3). For resources not locally owned, the user can only view (read-only) the plans and schedules.	Local control of locally owned resources with global visibility into the P & S information represents one of the primary user needs for performing day-to-day operations. Visibility into all schedules allows for early conflict resolution.

3.4 Phase 3 User Issues and Responses

Based on the feedback provided by the science and flight operations community, Table 3-3 shows the issues pertaining to the phase 3 P & S prototype. In the table, an originator number corresponds to the source of each issue (refer to section 3.2 for originator categories). After review and analysis, a response was determined by the P & S team for each of these primary issues.

Table 3-3. Phase 3 User Issues and Responses (1 of 3)

Issue	Orig	Response	Rationale
Focus prototypes on risk areas.	4	Phase 3 prototype addressed several areas of risk including FOS internal interfaces (Sections 2.3.1.12, 2.3.1.13, 2.3.1.14).	The FOS internal interfaces present a critical risk for the P&S system because of early release requirements. Providing an early P & S test bed for the FOS internal interfaces will help mitigate these risks.

Table 3-2. Phase 3 User Issues and Responses (2 of 3)

Issue	Orig	Response	Rationale
Provide the capability to handle a general list of constraint types since many of the spacecraft and instrument constraints may not be known until spacecraft integration and test.	2	The Phase 4 prototype will explore different methods for handling constraint checking including using pre-existing systems such as BPARR, WIND/POLAR, CLIPS and other COTS products as well as developing code in-house to handle mission constraints already identified.	Exploring external scheduling, rule-based and expert system software packages will reduce the risk of developing constraint checking code from scratch.
Various instruments may schedule activities based on orbit event triggers that include lat-lon regions and orbit cycle numbers.	1,2	The Phase 3 prototype began to address the scheduling of activities using FDF obtained orbital events. Three events were modeled, TDRSS availability, sunrise and sunset events. The Phase 4 prototype will utilize that data in scheduling most of the instrument activities.	Given feedback over the first 3 prototyping phases, it was determined that the majority of instrument teams and the FOT schedule activities triggered by a variety of orbital events. Prototyping scheduling methods using orbital events helps provide refined feedback from the user community.
Provide performance checks on the distributed architecture of the P & S system.	2	The phase 4 prototype will investigate distributed resource models. The current design has one main resource model located at the EOC. By distributing the resource model process, greater performance should be realized. In addition to distributing the resource model, limiting the data that is distributed to the ISTs (e.g. the "Master" plan only) should also increase the overall performance of the distributed system.	The distributed nature of the P & S software objects could exact an ugly toll on the performance of the system since IST sites are constantly receiving updates. Distributing the resource model and limiting the data flowing throughout the system will solve any performance problems encountered.
Provide the capability to view commands.	1,2,3	The phase 4 prototype will establish various methods for viewing commands. Some of the methods include viewing commands on the timeline, allowing users to view commands contained within activities by bringing up the command editor (Section 2.3.1.8.1) and the command list generated by CMS.	By providing a number of methods for displaying commands, user feedback can be obtained as to their usefulness. In addition, many groups expressed interest in seeing the actual commands at three times during scheduling: before, during and after.

Table 3-2. Phase 3 User Issues and Responses (3 of 3)

Issue	Orig	Response	Rationale
Establish a design for handling the archiving of the data used by the P & S system.	4	A number of designs will be explored for storing data used by the P & S system. These designs may include tools for determining the partitioning of the schedule.	Because the P & S software deals with continuous plans, risk mitigation involving the partitioning of the schedule must be studied. The manner in which the loads are partitioned by CMS will play a key part in determining the partitioning. In addition, archiving off old portions of the plan will aid in performance of the system.
Transition to DCE for distributed communication and study role of CORBA.	4	Initial coordination took place with CSMS for integrating DCE into the prototype. Phase 4 will test the integration. We are also investigating the role of CORBA to provide distributed P&S processes.	Establishing the DCE interface in P & S for distributed communication is necessary for providing the IST with an integrated set of FOS tools using CSMS as the infrastructure. The migration path to CORBA may play an essential part with the CSMS infrastructure.
Determine the method for scheduling instrument microprocessor load uplink periods.	4	A new uplink scheduling tool will be developed to provide instrument teams who are generating instrument microprocessor loads with a tool for requesting uplink periods.	Prototyping a tool that enables instrument teams to request uplink periods will help in eliciting feedback into how those teams will want to schedule uplink times.
Establish a operations concept and corresponding supporting software design for handling the reception of FDF planning aid data.	3	During the phase 4 prototype, a method for handling the reception of FDF data will be determined. If necessary, software tools developed to handle importing the data into the P & S system and the possible shifting of activities to correspond to any changes in the refined orbit event times.	Since most activities will be scheduled relative to orbit events, the ingest of FDF planning aids could have a significant impact on the absolute times within those activities. A method for handling this needs to be established.
Refine the procedure for rescheduling activities that have been bumped from the schedule.	4	An automatic scheduling tool will be prototyped which collects activities that have been unallocated and the capability to reschedule those activities using an uncomplicated algorithm.	A software design needs to be established to be able to manage activities that have been removed from the schedule because of simple unallocation or impact scheduling.

4. Summary

4.1 Summary

The P & S Prototype Results Report presented the design, architecture and features of the phase 3 P & S prototype effort. A distributed P & S system was refined on the already existing architecture that consisted of independent, distributed C++ processes that communicated by passing object oriented messages between each other. In order to accept changes in the mission requirements and operational concepts, the object oriented infrastructure was further refined, taking advantage of the P & S experience contained in the Mission Planning Class Libraries.

For the FOS, the AM-1 elements represented in the prototype scenario include the EOC Scheduler, and the CERES IST. A number of new scheduling tools were introduced including the activity definer, the BAP definer, the segmented timeline, the postscript timeline, the communication contact scheduler, the plan tool and the plan window manager. In addition to the new tools, a number of interfaces were prototyped including the interface with CMS, the Instrument Support Toolkit and Real-Time Analysis. All of these elements provided a thread through the AM-1 scheduling process: pre-scheduling, initial scheduling, final scheduling and command load generation.. The phase 3 prototype tools were demonstrated at the ECS PRR, where discussions were held to gather input from the science and flight operations community.

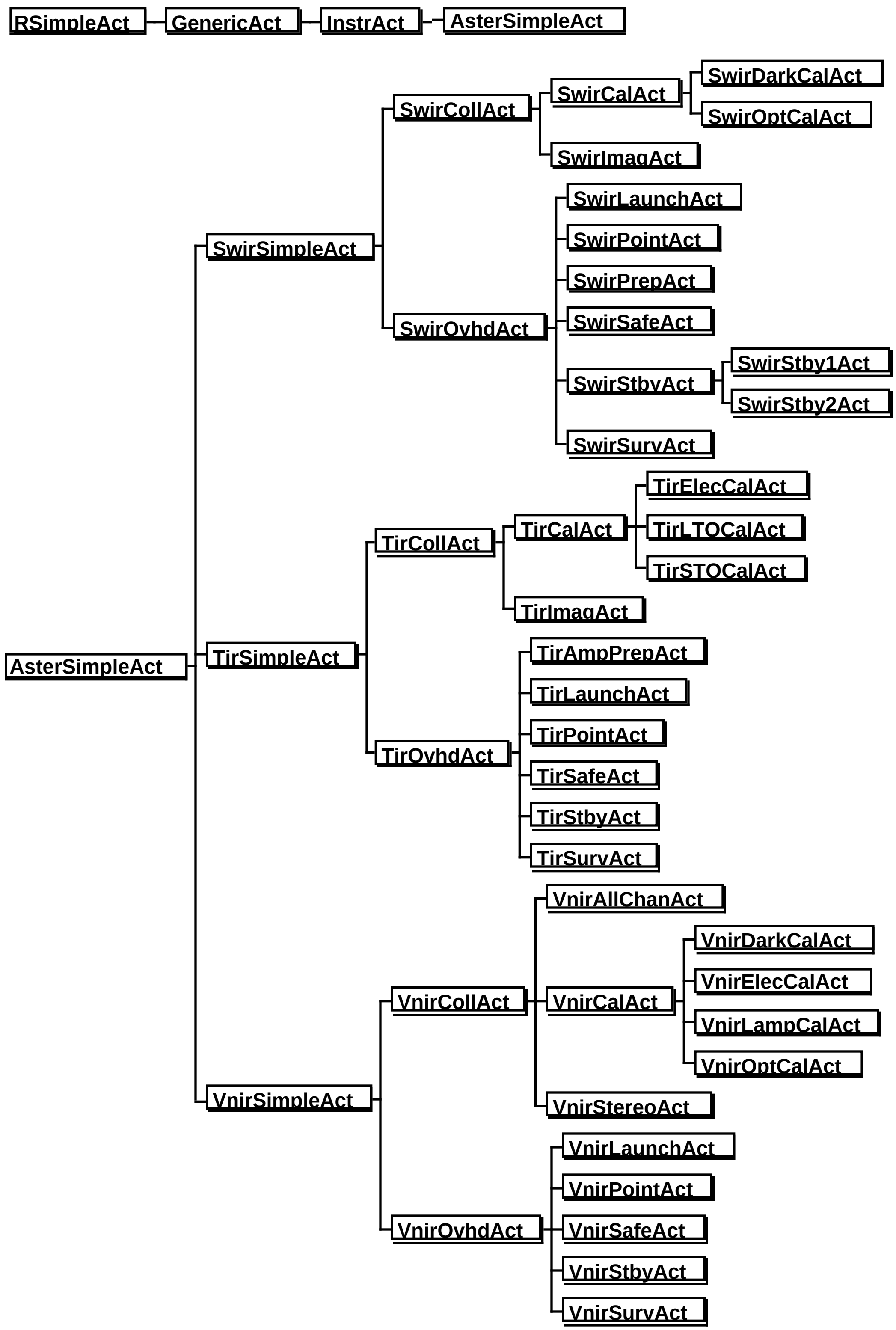
Based upon analysis of the science and flight operations community feedback (see Section 3.2), the following features are planned for the phase 4 prototype:

- Command Editor for modifying and viewing commands associated with activities
- Uplink Scheduler for scheduling uplink times for instrument microprocessor loads
- Instrument activity scheduling using orbital event triggers
- Performance checks for a fully integrated P & S and CMS subsystem
- Distributed resource models
- Methods for archiving portions of the continuous plan
- Automatic Scheduling tool for storing bumped activities and rescheduling them
- Process for ingesting FDF planning aids and adjusting the schedule

This page intentionally left blank.

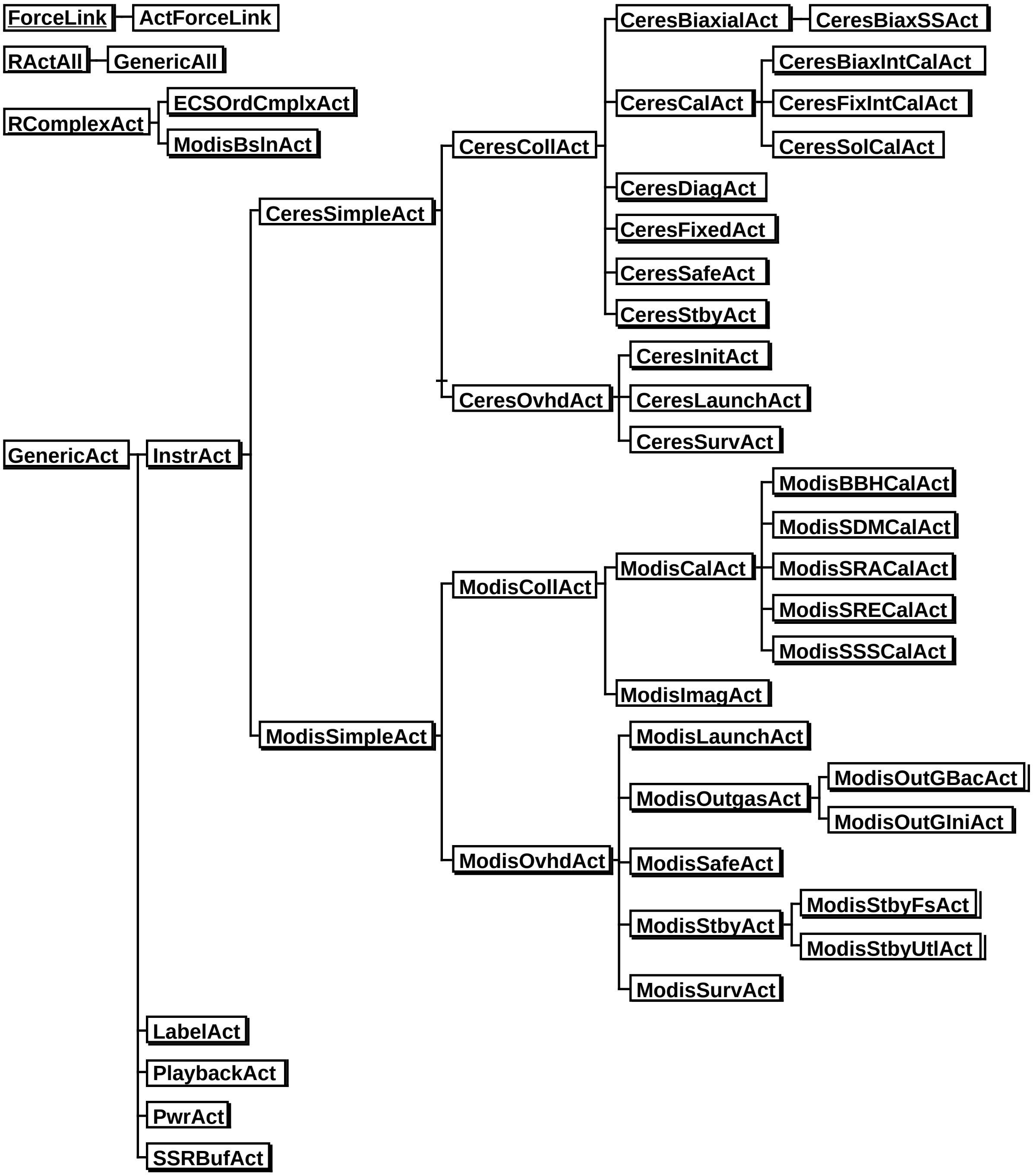
Appendix A. P & S Class Hierarchies

This page intentionally left blank.



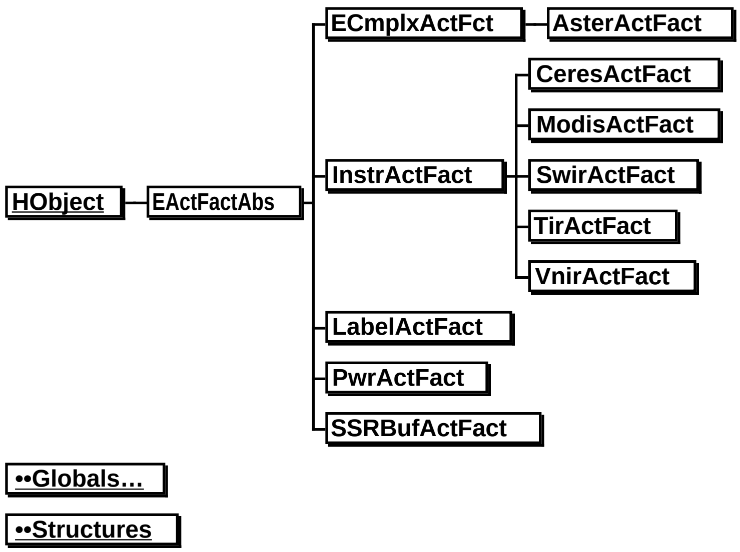
ECS FOS Planning and Scheduling:

Activity Class Library (eact) - Part 1 of 2



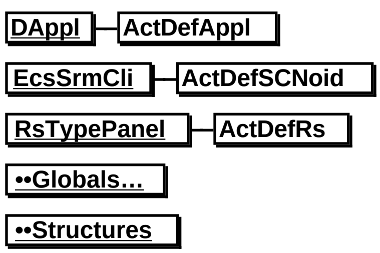
ECS FOS Planning and Scheduling:

Activity Class Library (eact) - Part 2 of 2



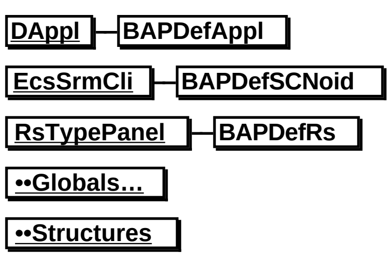
ECS FOS Planning and Scheduling:

Activity Configuration Class Library (eactcfg)



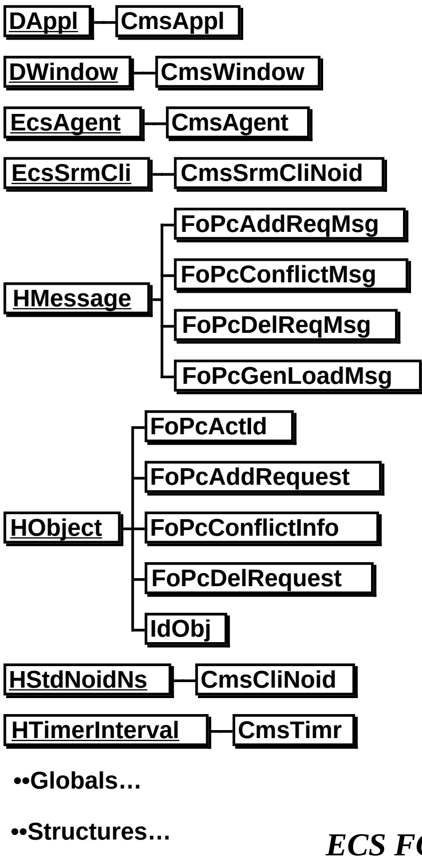
ECS FOS Planning and Scheduling:

Activity Definer Class Library (ead)



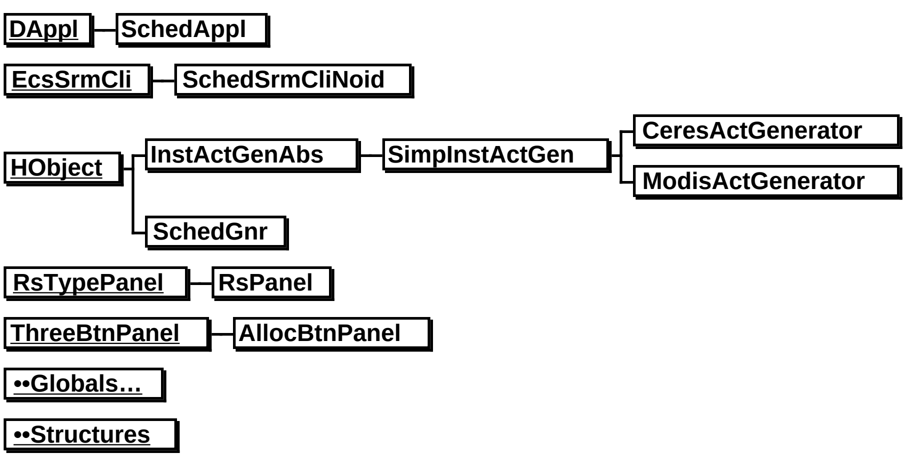
ECS FOS Planning and Scheduling:

Baseline Activity Profile Definer Class Library (ebd)



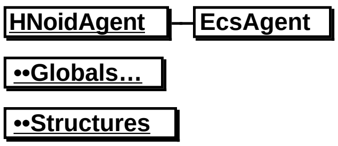
ECS FOS Planning and Scheduling:

Command Management Interface Class Library (ecms)



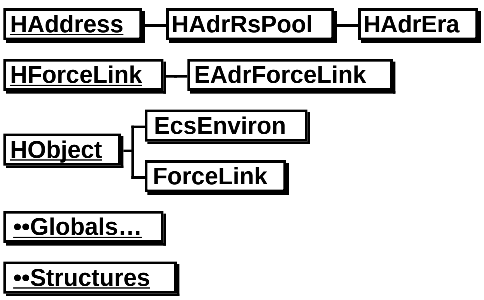
ECS FOS Planning and Scheduling:

Instrument Activity Scheduler Class Library (eias)



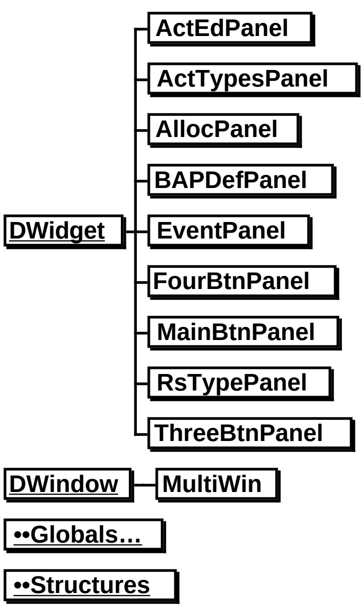
ECS FOS Planning and Scheduling:

Interprocess Communication Class Library (eipc)



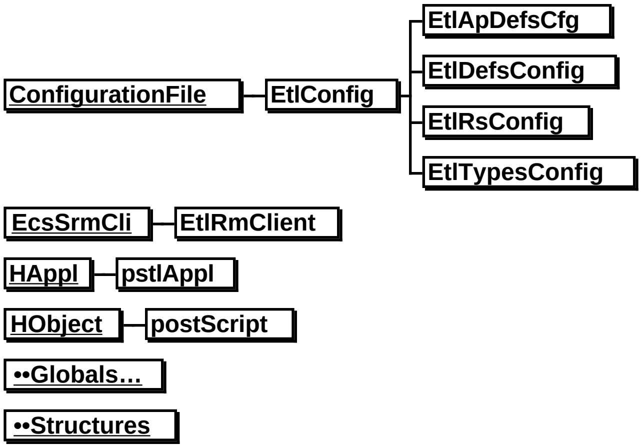
ECS FOS Planning and Scheduling:

Miscellaneous Class Library (emisc)



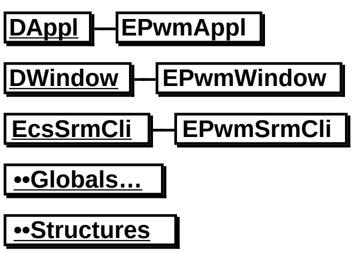
ECS FOS Planning and Scheduling:

Definer Panels Class Library (epanel)



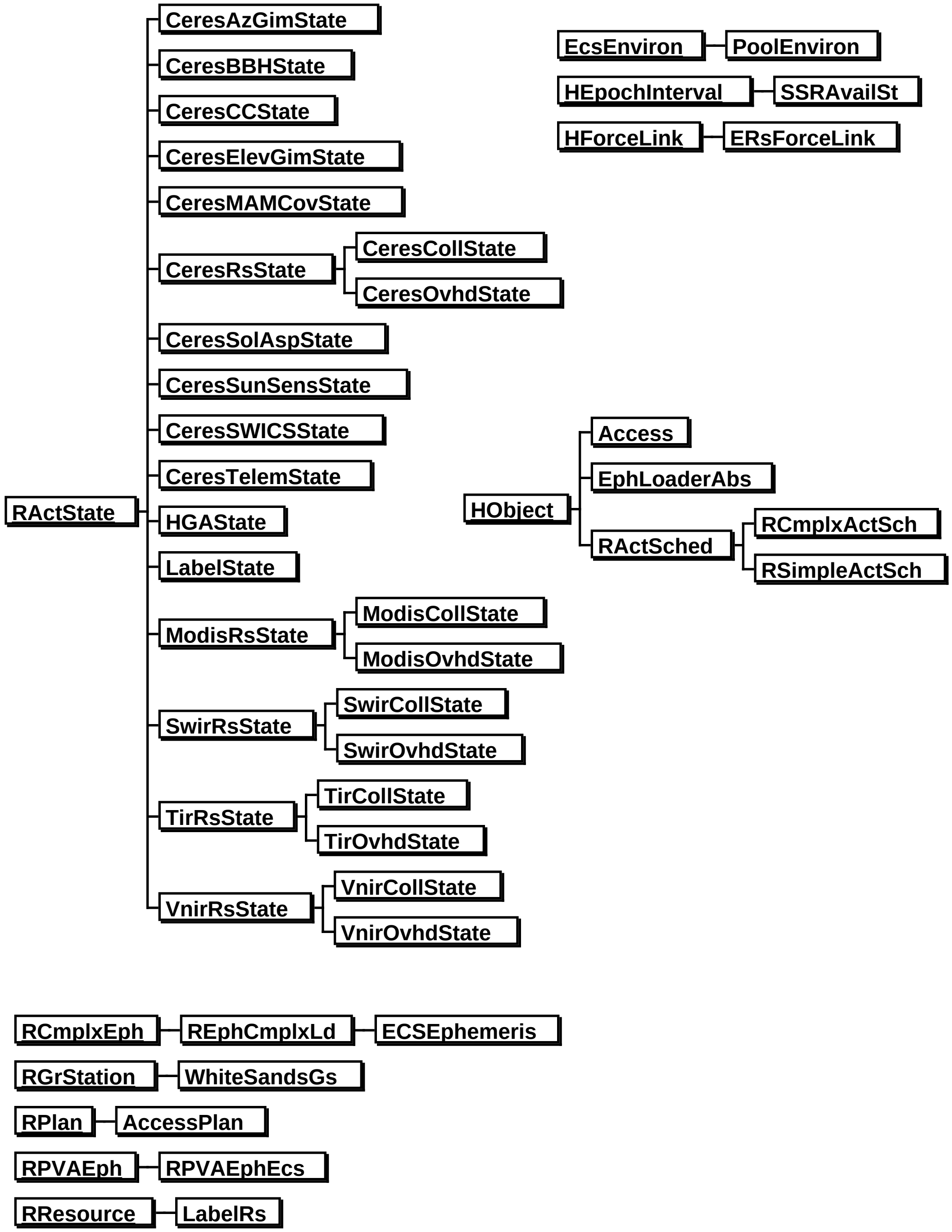
ECS FOS Planning and Scheduling:

Postscript Timeline Class Library (epstl)



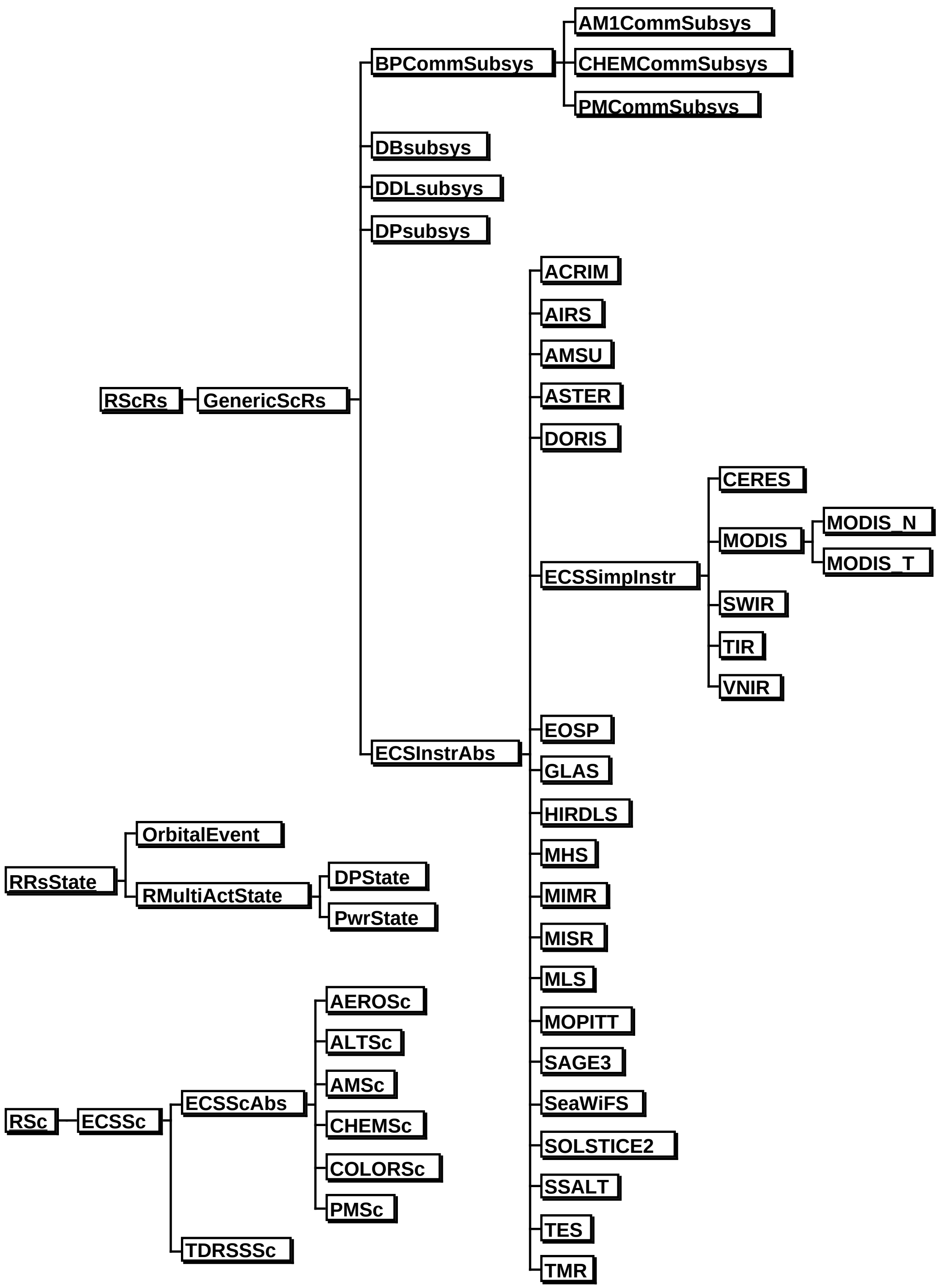
ECS FOS Planning and Scheduling:

Plan Window Manager Class Library (epwm)



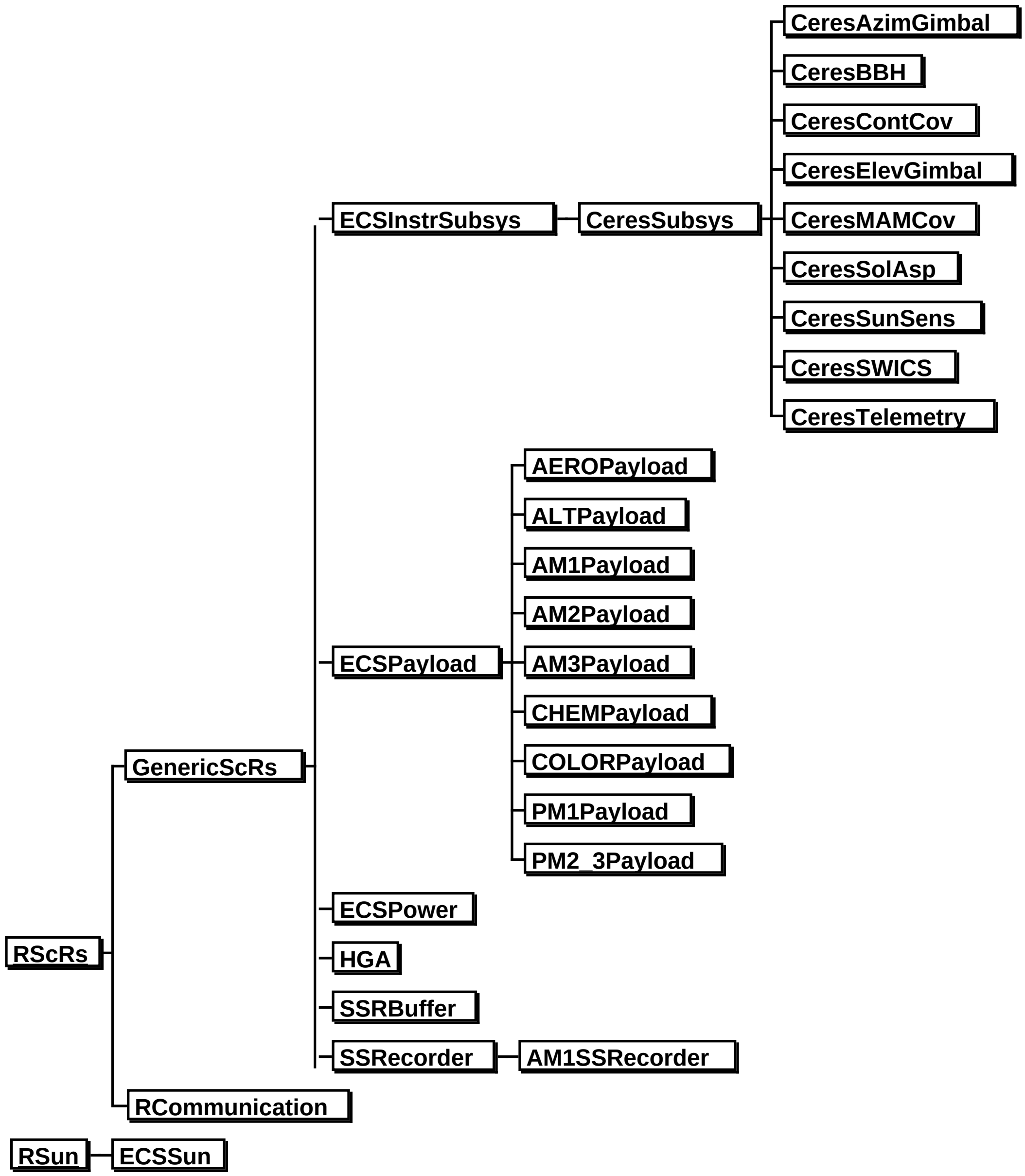
ECS FOS Planning and Scheduling:

Resource Class Library (ercl) - Part 1 of 3



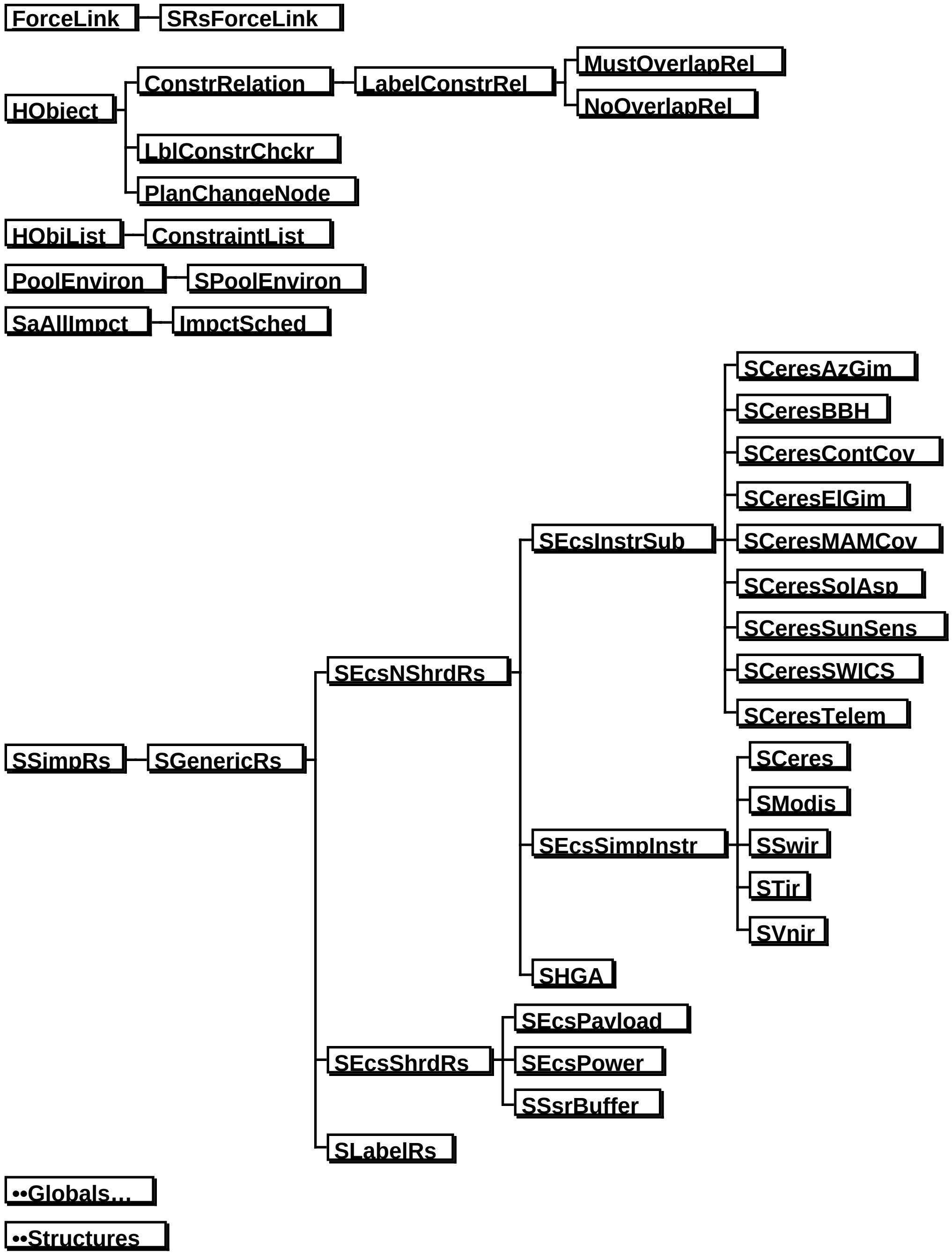
ECS FOS Planning and Scheduling:

Resource Class Library (ercl) - Part 2 of 3



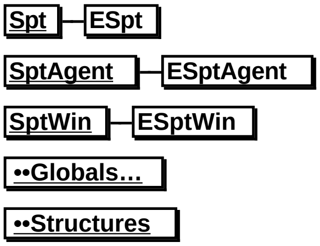
ECS FOS Planning and Scheduling:

Resource Class Library (ercl) - Part 3 of 3



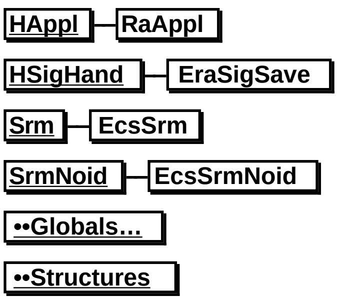
ECS FOS Planning and Scheduling:

Scheduling Resources Class Library (escl)



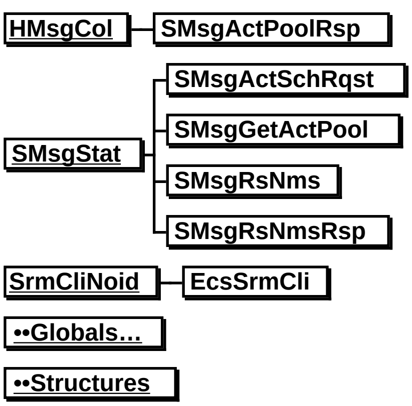
ECS FOS Planning and Scheduling:

Plan Tool Class Library (espt)



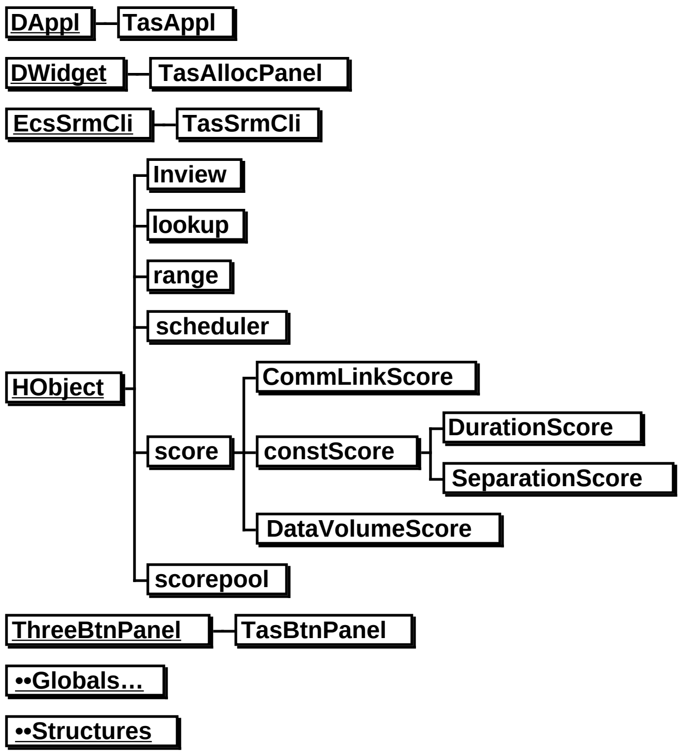
ECS FOS Planning and Scheduling:

Resource Model Class Library (esrm)



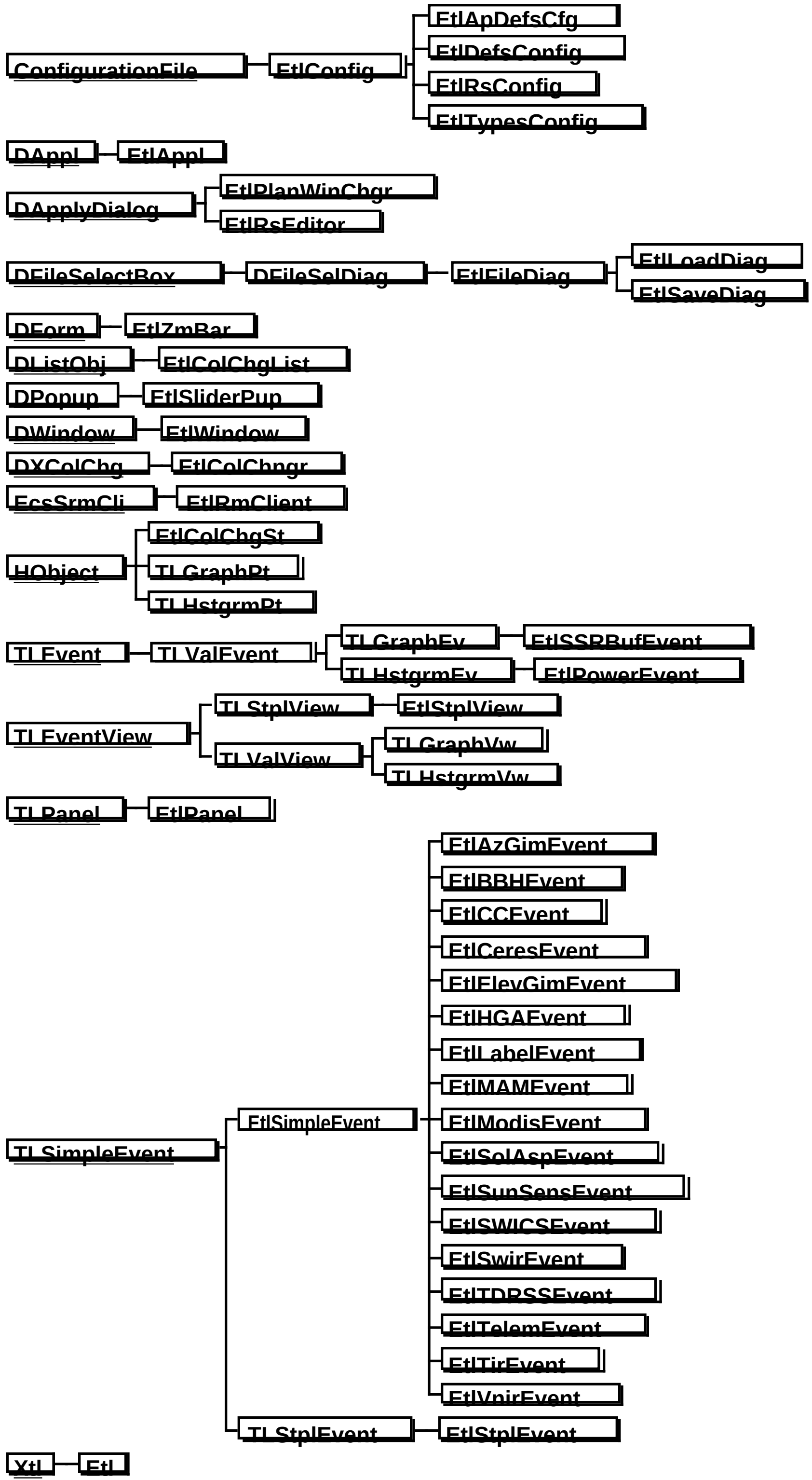
ECS FOS Planning and Scheduling:

Resource Model Client Class Library (esrmcli)



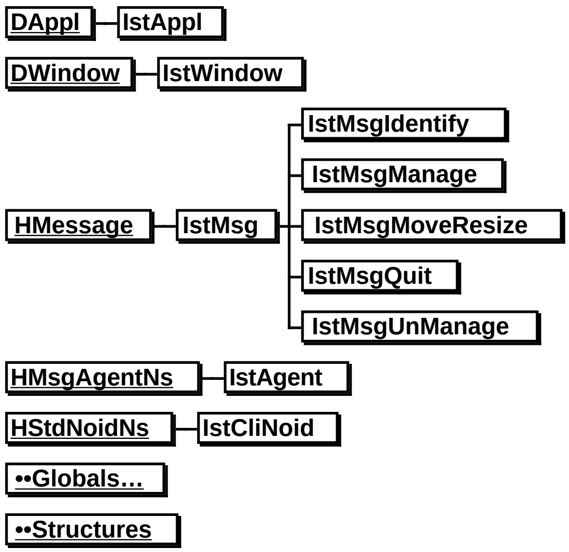
ECS FOS Planning and Scheduling:

Communication Contact Scheduler Class Library (etas)



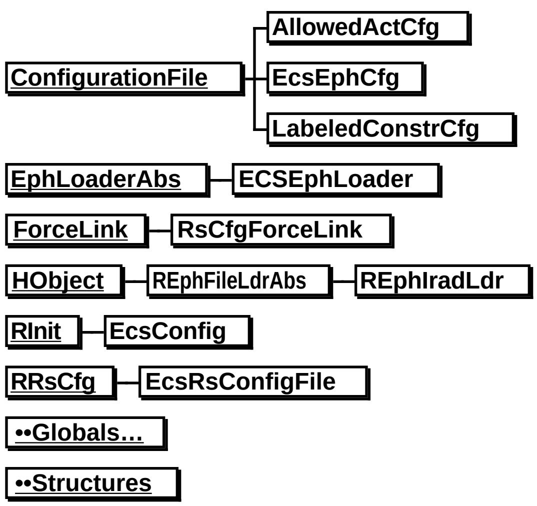
ECS FOS Planning and Scheduling:

Timeline Class Library (etl)



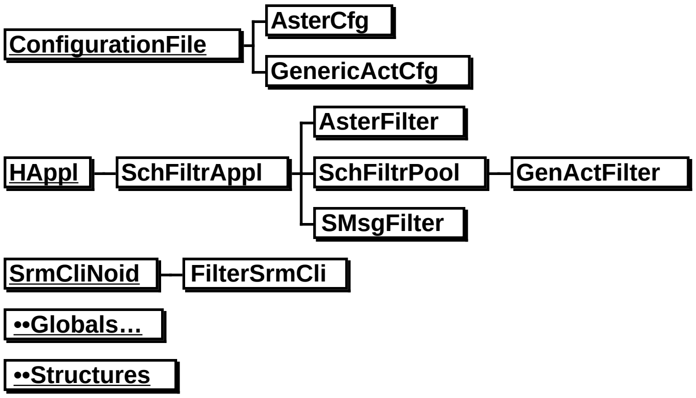
ECS FOS Planning and Scheduling:

Instrument Support Terminal Interface Class Library (ist)



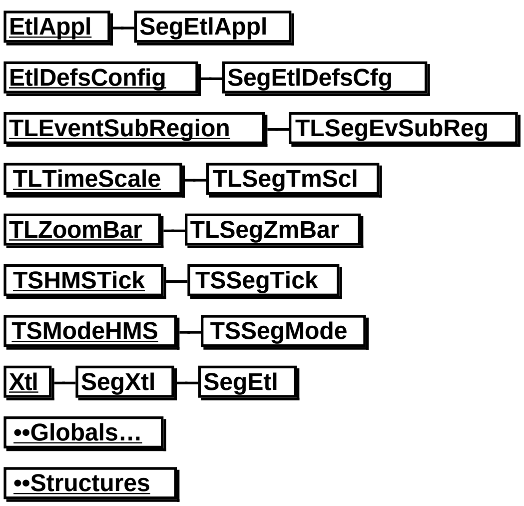
ECS FOS Planning and Scheduling:

Object Database Class Library (odb)



ECS FOS Planning and Scheduling:

Schedule Filter Class Library (schfiltr)



ECS FOS Planning and Scheduling:

Segment Timeline Class Library (segetl)

This page intentionally left blank.

Abbreviations and Acronyms

ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
BAP	Baseline Activity Profile
CERES	Clouds and Earth's Radiant Energy System
CMS	Command Management System
DAR	Data Acquisition Request
DAS	Detailed Activity Schedule
DCE	Distributed Computing Environment
ECS	EOSDIS Core System
EOC	EOS Operations Center
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
FOS	Flight Operations Segment
GSFC	Goddard Space Flight Center
HCL	Hughes Class Libraries
HIPC	Hughes Inter-Process Communication
HMI	Human-Machine Interface
ICC	Instrument Control Center
IR&D	Internal Research and Development
IST	Instrument Support Terminal
JPL	Jet Propulsion Laboratory
MISR	Multi-angle Imaging Spectro-Radiometer
MODIS	Moderate Resolution Imaging Spectrometer
MOPITT	Measurements of Pollution in the Troposphere
NCC	Network Control Center
OPS	Onboard Payload Support
P & S	Planning and Scheduling
PCL	Planning Class Libraries
PI/TL	Principal Investigator/Team Leader

PRR	Prototyping Results Review
RCL	Resource Class Libraries
RPC	Remote Procedure Call
SCCS	Source Code Control System
SCL	Scheduling Class Libraries
SSR	Solid State Recorder
TCL	Timeline Class Libraries
TDRSS	Tracking and Data Relay Satellite System
TOO	Target Of Opportunity
XDR	External Data Representation