MISSION AND SCIENCE ACTIVITY SCHEDULING LANGUAGE

Larry G. Hull
Software and Automation Systems Branch, Code 522
NASA/Goddard Space Flight Center
Greenbelt Maryland 20771, USA

ABSTRACT

To support the distributed and complex operational scheduling required for future National Aeronautics and Space Administration (NASA) missions, a formal, textual language, the Scheduling Applications Interface Language (SAIL), has been developed. Increased geographic dispersion of investigators is leading to distributed mission and science activity planning, scheduling, and operations. SAIL is an innovation which supports the effective and efficient communication of scheduling information among physically dispersed applications in distributed scheduling environments.

SAIL offers a clear, concise, unambiguous expression of scheduling information in a readable, hardware independent format. The language concept, syntax, and semantics incorporate language features found useful during 5 years of research and prototyping with scheduling languages in physically distributed environments. SAIL allows concise specification of mission and science activity plans in a format which promotes repetition and reuse.

Key Words: Scheduling, language, data representation, distributed mission operations

1. DISTRIBUTED OPERATIONAL SCHEDULING

Operational scheduling for NASA missions is becoming more geographically distributed and more operationally complex (Ref. 1). Increased geographic dispersion of investigators (including Japanese and European) is leading to distributed mission and science activity planning, scheduling and operations. The growing number of sophisticated instruments that generate very high data rates and volumes is leading to a requirement for more complex and more automated planning, scheduling, and resource management.

The focus in this paper is on the scheduling of NASA mission operations, specifically on the scheduling of spacecraft resources to support spacecraft operations and geographically dispersed investigators. Figure 1 is an end-to-end scheduling environment (Ref. 2).

From the point of view of the network, scheduling serves to allocate shared communications resources. From the point of view of the mission, scheduling serves to maintain the health and safety of the spacecraft and to allocate shared spacecraft resources.

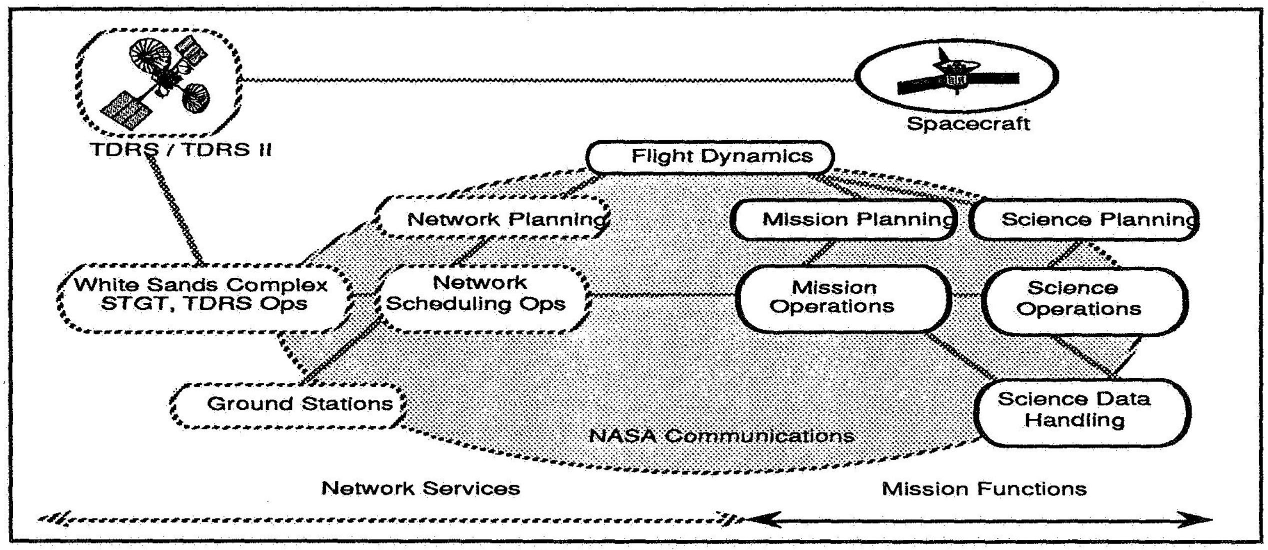


Figure 1. End-to-end Scheduling Environment

From the point of view of investigators, scheduling serves to maintain the health and safety of the science instruments and to support science data acquisition. Although all use the same or similar data for generating schedules which must eventually be consistent, objectives and techniques differ.

2. SCHEDULING DATA REPRESENTATION

As indicated, the scheduling of NASA mission operations is becoming an increasingly intricate and distributed process. To support future mission operations, scheduling information must be communicated effectively among physically dispersed applications in distributed scheduling environments. The scheduling information representation must allow expression of complex, variable scheduling requirements. For effective and efficient scheduling, the representation must also provide sufficient flexibility to accommodate changing requirements and scheduling alternatives. Finally, the representation should be concise, unambiguous, and in a hardware-independent format that can be processed by computer.

The Software and Automation Systems Branch at GSFC has addressed the problem of defining a scheduling data representation for two specific scheduling domains: communications support scheduling and spacecraft resource scheduling (Ref. 3). Each domain is a physically distributed scheduling environment with several different scheduling applications running on several different hardware configurations. Consequently, a suitable data representation could not presume a specific physical topology, a specific communications protocol, or a specific operations scenario. On the other hand, efficient and effective scheduling required reusable scheduling data representations supporting repetitive activities, sufficient flexibility to express often complex scheduling requirements and constraints, and the capability to communicate scheduling alternatives. The scheduling data representation had to be self-documenting, easily maintained, and physically and logically portable. Requirements analysis (Ref. 4) concluded that a formal, textual language was needed.

3. SAIL

Requirements analysis led to the definition of the Scheduling Applications Interface Language (SAIL). SAIL is a formal, textual language (Ref. 5) for expressing scheduling information communicated between nodes of a distributed, activity scheduling system. SAIL provides interoperability between different scheduling applications running on different hardware configurations. SAIL expresses scheduling information in a clear, concise, unambiguous format that can be processed by computer.

SAIL expresses the inputs to the scheduling process, the outputs of the scheduling process, and other transactions needed to maintain schedules in an operational environment. SAIL expresses nominal and contingency operations plans for instruments and spacecraft, the constituent activities of the operations plans, the resource needs of these activities, and the constraints that restrict when these activities can be scheduled. SAIL also expresses both the resources available to satisfy activity resource needs and the resources allocated to activities (the schedule).

SAIL is intended to be used in the repetitive scheduling process that occurs throughout the life of a mission. In a typical operational environment, users:

- develop plans that reflect high level goals and objectives for an instrument or facility,
- derive scheduling information from these plans, e.g., instrument activities and target or view times,
- transmit scheduling information and requests to a scheduling facility,
- schedule local resources in accordance with instrument schedules received from scheduling facilities, and
- transmit rescheduling information as needed.

Scheduling facilities typically will:

- distribute appropriate scheduling information to users,
- receive and store user-generated scheduling information and requests,
- coordinate with resource providers to build integrated schedules,
- disseminate schedules to users, and
- accept new inputs and reschedule to meet changing science needs.

4. A SCHEDULING NODE

Figure 2 shows a functional architecture for a scheduling node that accepts scheduling requests from multiple users. The scheduling node in the

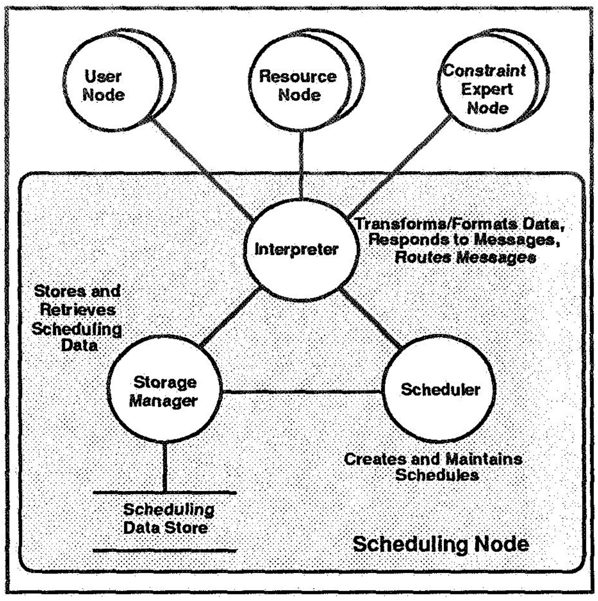

User Node
Resource Node
Constraint Expert Node
Interpreter
Transforms/Formats Data, Responds to Messages, Routes Messages
Stores and Retrieves Scheduling Data
Storage Manager
Scheduler
Creates and Maintains Schedules
Scheduling Data Store
Scheduling Node

Figure 2. Scheduling Node Functional Architecture

figure is somewhat representative of other nodes in the network, since many applications in a scheduling domain perform some scheduling functions (e.g., user nodes schedule local resources.)

The SAIL interpreter accepts messages and interacts with the scheduler and the storage manager. The interpreter transforms the incoming data for the scheduler and the storage manager. Interpreting a message may simply cause the receiving node to send back an acknowledgment message. In many cases, the message may have more substantial effects. A message that specifies a request for resources (an activity) for a future schedule results in this request being added to the data store at the receiving node. A message that requests a change be made to a current schedule results in a reallocation of resources.

The scheduler creates and maintains schedules, including the current on-line schedule. This function is typically performed by operations personnel with assistance from automated scheduling software. The scheduler interacts with the SAIL interpreter and processes messages that affect the current on-line schedule. The scheduler interacts with the storage manager to store schedules and to retrieve information needed to create and maintain schedules.

The storage manager services requests from the SAIL interpreter and the scheduler to store, update, or access information in its data store. The data store contains the information needed to create schedules and the schedules that result from the scheduling process. The types of data stored typically include requests for resources, resource availability information, constraint information, and schedules.

5. DATA MANAGEMENT

The scheduling problem is a data management problem. A scheduling node typically maintains several schedules. Each of these schedules consists of many pieces of data (requests for resources, constraint information, resource availabilities, etc.). Some data is relevant over a long period of time; other data is relevant for only one or a few schedules.

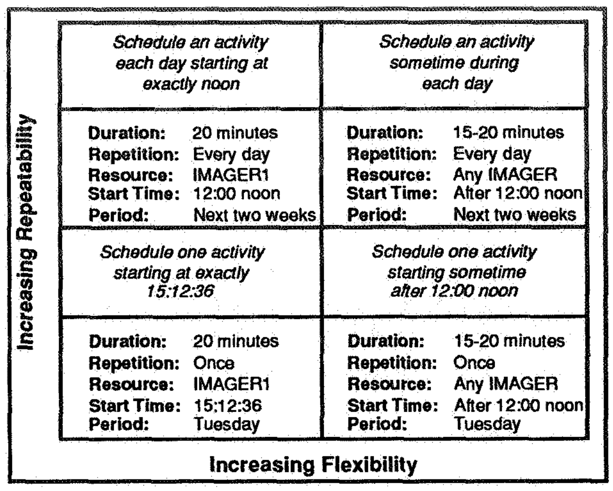
SAIL contains data structures that make it suitable for complex, multi-schedule environments. SAIL statements enable users to specify the schedule(s) for which the data is valid. Since multiple schedules can be specified, data that is used repeatedly does not have to be retransmitted.

SAIL handles the Flexible Scheduling Request Concept. This concept is an alternative to traditional methods of fixed format scheduling. With fixed format scheduling, users submit requests that are either approved or disapproved. Users frequently submit many more requests than may be necessary in anticipation that some of the requests will be rejected. With flexible scheduling, users specify single or repeated activities, and enough flexibility in activity execution time and resource requirements to enable the scheduler to more optimally handle the request.

For instance, an activity might specify that the duration of an operation must be "exactly 2 hours" or that the duration must be "longer than 20 minutes." Flexible requirements can potentially reduce the number of conflicts in schedules, and may enable a scheduler to produce a more optimal schedule. For example, suppose Activity A has been scheduled at 2:00 p.m. on Tuesday but its requirements specify that it really can be scheduled any time during the day on Tuesday. Next, suppose that the scheduler receives a request to schedule Activity B only at 2:00 p.m. on Tuesday. Given the flexibility specified in the requirements of Activity A, the scheduler is free to move Activity A to another time in order to accommodate Activity B. The resulting schedule is therefore more optimal.

Figure 3 shows the flexibility options SAIL is capable of representing. The vertical axis illustrates repeatability--in the upper left corner, more activities

C.9



Schedule an activity each day starting at exactly noon	*Schedule an activity sometime during each day*
Duration: 20 minutes **Repetition:** Every day **Resource:** IMAGER1 **Start Time:** 12:00 noon **Period:** Next two weeks	**Duration:** 15-20 minutes **Repetition:** Every day **Resource:** Any IMAGER **Start Time:** After 12:00 noon **Period:** Next two weeks
Schedule one activity starting at exactly 15:12:36	*Schedule one activity starting sometime after 12:00 noon*
Duration: 20 minutes **Repetition:** Once **Resource:** IMAGER1 **Start Time:** 15:12:36 **Period:** Tuesday	**Duration:** 15-20 minutes **Repetition:** Once **Resource:** Any IMAGER **Start Time:** After 12:00 noon **Period:** Tuesday

Figure 3. Flexible Scheduling Request Concept

will be scheduled per request. The horizontal axis illustrates flexibility--in the lower right corner, more flexibility is specified in a request providing the scheduler with more scheduling options. In the upper right corner, the request is both repeatable and flexible--many activities can be scheduled from a request with a lot of flexibility giving the scheduler many scheduling options.

Figure 4 illustrates possible scheduling time windows, selected by a scheduler, based on a flexible request. The request includes the constraints that must be satisfied when scheduling a viewing activity but does not specify specific start and end times. The scheduler is given the option of determining when to schedule the activity, taking into consideration the set of constraints for this activity. In the example, for the set of constraints specified, the scheduler has the flexibility to schedule the viewing activity any time during the time windows shown as shaded boxes.

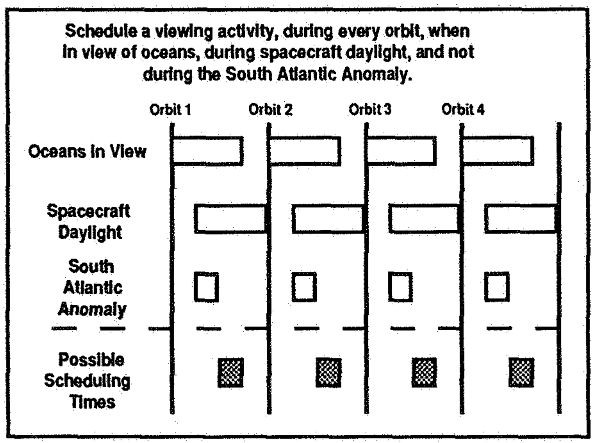

Schedule a viewing activity, during every orbit, when in view of oceans, during spacecraft daylight, and not during the South Atlantic Anomaly.
Orbit 1
Orbit 2
Orbit 3
Orbit 4
Oceans In View
Spacecraft Daylight
South Atlantic Anomaly
Possible Scheduling Times

Figure 4. Possible Scheduling Time Windows

6. AN INTRODUCTION TO SAIL

6.1 SAIL Messages

Nodes in a distributed environment communicate through SAIL messages. There are two types of messages in SAIL: acknowledgment messages and declarative messages. An acknowledgment message acknowledges the receipt of a declarative message. Declarative messages contain planning and scheduling information and are used to transmit this information from one node to another.

A declarative message contains a header and a sequence of statements. The declarative message header specifies such information as the sender of the message, the intended receiver, and the time the message was sent. Each statement within a declarative message specifies an action and some data. The statement can be any of the following:

- a request for the receiver of the message to perform the stated action on the specified data,
- a response that indicates the success or failure of a request, or
- a notification that the sender of the message has already performed an action on the specified data (no response action is required).

A declarative message may contain any number of request, response, and notification statements in any order.

6.2 SAIL Statements

SAIL has four request actions, four response actions, and four notification actions. Each of these actions, together with a SAIL structure (defined below), form a SAIL statement.

6.2.1 Request Statement

Request statements are the "action" statements in SAIL. Through request statements, the sender can cause changes at a receiver's (remote) node. A user node sends messages containing request statements to a scheduling node to accomplish each of the following:

- add an activity to a schedule,
- change an activity already on a schedule, and
- delete a scheduled activity from a schedule.

Request actions are CREATE, REPLACE,

DELETE, and DESCRIBE. The first three are data management actions that create, update, or remove SAIL structures located at another node. The fourth request action, DESCRIBE, allows a node to obtain the definition of a structure stored at another node.

6.2.2 Response Statement

Response statements consist of the reserved word RESPONSE_TO followed by the request action and an indication of the success or failure of the request. Response actions are the same as request actions.

6.2.3 Notification Statement

Notification statements consist of a notification action usually followed by a structure. Notification actions are: CREATED, REPLACED, DELETED, and DESCRIBED. Notification statements are used to notify other nodes that a change has occurred. Notification statements differ from response statements in that a notification statement is not the result of any request statement. For instance, if a scheduling node reschedules some events in response to a change in predicted resource availabilities, it notifies the affected users that a change has occurred.

6.3 SAIL Structures

Efficient planning and scheduling in a distributed environment requires the capability to express several types of information. For instance, user nodes must be able to communicate their resource needs to the provider of the resource or to the control center responsible for allocating the resource. Scheduling results (resource allocations) must be returned to the user nodes that requested the resources.

This information is contained in the SAIL declarative message. SAIL request and notification statements in a SAIL declarative message identify an action and a structure that is the object of the action. Structures express an input or output of the activity scheduling process. There are five structures defined in SAIL: activities, plans, allocations, profiles, and meta-schedules. Therefore, a facility may request another facility to create, describe, delete, or replace an activity, plan, allocation, profile, or meta-schedule.

6.3.1 Activities

Activities specify the requirements of spacecraft and instrument operations. These requirements consist of resource needs and other restrictions that limit the start time, end time, and duration of the activities. A specification includes the name of the activity, the priority of the activity with respect to other activities requested by the same node, any requirements to coordinate this activity with other activities and spacecraft events, and options that a scheduler should try in case the activity cannot be scheduled as initially specified. Requirements can be specific or flexible, as previously described.

Schedulers use activities in determining how to allocate available resources. When an activity is "scheduled," resources are allocated for the operation described by the activity. Resource allocations are expressed using the allocation structure.

6.3.2 Plans

The plans provided in SAIL represent a sequence of activities that some facility or system performs. SAIL plans are expressed as repeating sequences of activity structures and are often reusable. For instance, the plan "perform a tape recorder dump every fourth orbit using TDRS-East" is a reusable SAIL plan.

6.3.3 Allocations

Allocations represent the output of the scheduling process. An allocation expresses assignment of resources needed to perform an operation or a part of an operation with specific start and end times for each resource assignment. Collectively, the allocations for a given time period (e.g., a day) represent a schedule.

6.3.4 Profiles

Profiles are a general data representation structure primarily used to represent numeric data that varies over time. The data in profiles consist of words, numbers, and character strings; thus, profiles can express a variety of types of information, such as:

- the varying amount of a resource available over time as activities obligate and release the resource,
- user antenna view (UAV) times, and
- user spacecraft ascending node, descending node, apogee, and perigee.

Profiles that represent resource amounts, such as available power, often constrain activities. For instance, an activity might require 20 watts of power for 30 minutes. The power profile specifies the available power and restricts the scheduling of the activity. Figure 5 shows an example of an available power profile with a maximum value of 30 watts and a minimum value of 10 watts. The profile is defined for a 6-hour period. Using this available power profile, an activity that requires 20 watts of power could be scheduled only from 6:30 to 7:30.

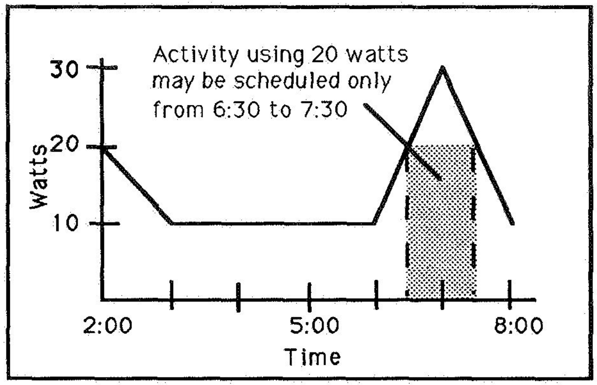


Figure 5. Power Profile

6.3.5 Meta-schedules

A SAIL meta-schedule structure contains a schedule (i.e., a list of allocations) and additional information about the schedule. This information consists of the inputs to the scheduling process (SAIL activity, plan, and profile structures) and metrics that express the quality of the schedule with respect to scheduling goals. A meta-schedule structure provides a record of the information used to create a particular schedule and a record of the results of the scheduling process. Structure headers specify the names of the schedules (more specifically, meta-schedules) to be used when performing an action on a structure.

SAIL meta-schedules are either active or inactive. An inactive meta-schedule does not contain specific assignments of resources (i.e., it does not contain allocations). Meta-schedules are initially inactive. During this period, a scheduling node that creates schedules collects requests for resources. At some point in time, the scheduling node "creates a schedule." To "create a schedule" is to make an active meta-schedule from an inactive meta-schedule (i.e., to assign resources to activities and create the allocation structures that specify the assignments).

7. SUMMARY

SAIL is a formal, textual language developed to support effective and efficient communication of scheduling information among physically dispersed applications in distributed scheduling environments. The language is an innovation which allows both for expression of complex, variable requirements and for concise specification of mission and science activity plans in a format which promotes repetition and reuse. SAIL's ability to express plan flexibilities and alternatives reduces the need for schedule iterations. SAIL does not presume a specific physical topology, communications protocol, or operations scenario.

Experience with predecessor languages in prototyping environments indicates that SAIL can express the control and data information that must be communicated between nodes in a distributed scheduling environment, and the complex scheduling requirements of instruments and spacecraft. SAIL is viewed as a relatively high-payoff, low-risk approach for operational implementation.

8. REFERENCES

1. Hull, Larry G. et al. 1990. Distributed Planning and Scheduling for Instrument and Platform Operations. In AIAA/NASA Second International Symposium on Space Information Systems. Washington, DC: American Institute of Aeronautics and Astronautics.

2. Moe, Karen. 1992. End-to-End Planning and Scheduling Systems Technology for Space Operations. In 43rd Congress of the International Astronautical Federation. Paris: International Astronautical Federation.

3. Buford, Carolyn. 1992. Scheduling Data Representation: Concept and Experience in Code 520, (DSTL-92-010). Greenbelt, MD: NASA/Goddard Space Flight Center.

4. Zoch, David. 1991. Requirements for the Scheduling Applications Interface Language (SAIL). Seabrook, MD: Loral AeroSys.

5. Sary, Charisse. 1992. The Scheduling Application Interface Language (SAIL) Reference Manual, (DSTL-91-021) Revision 1. Greenbelt, MD: NASA/Goddard Space Flight Center.