SCHEDULING THE FUTURE NASA SPACE NETWORK: EXPERIENCES WITH A FLEXIBLE SCHEDULING PROTOTYPE

Nadine Happell [1], Karen L. Moe [2], Jay Minnix [1]

N94-23965

[1] Stanford Telecommunications, Inc.
1761 Business Center Drive
Reston, VA 22090 USA

[2] NASA Goddard Space Flight Center
Software and Automation Systems Branch, Code 522
Greenbelt, MD 20771 USA

ABSTRACT

NASA's Space Network (SN) provides telecommunications and tracking services to low earth orbiting spacecraft. One proposal for improving resource allocation and automating conflict resolution for the SN is the concept of flexible scheduling (Ref. 1). In this concept, each Payload Operations Control Center (POCC) will possess a Space Network User POCC Interface (SNUPI) to support the development and management of flexible requests. Flexible requests express the flexibility, constraints, and repetitious nature of the user's communications requirements. Flexible scheduling is expected to improve SN resource utilization and user satisfaction, as well as reduce the effort to produce and maintain a schedule. A prototype testbed has been developed to better understand flexible scheduling as it applies to the SN. This testbed consists of a SNUPI workstation, an SN scheduler, and a flexible request language that conveys information between the two systems. All three are being evaluated by operations personnel. Benchmark testing is being conducted on the scheduler to quantify the productivity improvements achieved with flexible requests.

Key Words: Scheduling, flexible request language, space communications, prototyping

1. INTRODUCTION

Providing communications and tracking support to low earth orbit spacecraft has long been one of NASA's primary objectives. NASA's SN, in conjunction with the various ground based communications networks, is designed to provide these necessary support functions.

1.1. The Space Network

The SN is comprised of three main elements: the space segment, the ground segment, and the control segment. All three elements are currently undergoing an evolution to meet user needs in the late 1990s and beyond.

The space segment consists of a constellation of three Tracking and Data Relay Satellite (TDRS) spacecraft that provide forward and return communications and tracking services for customer spacecraft. The future SN may consist of a mixed constellation of TDRS and TDRS II (the next generation TDRS) spacecraft, eventually evolving to a constellation solely of TDRS II spacecraft. While TDRS II is expected to provide some new functionality, the operations concept will be essentially the same as the TDRS operations.

The main element of the ground segment is the White Sands Ground Terminal (WSGT), which provides the communications link between the TDRS spacecraft and the ground. In the near future, a Second TDRS Ground Terminal (STGT) will join WSGT in providing space to ground link support. Over the next several years, both ground systems are planned to receive significant upgrades to address the changing SN environment.

The control segment is responsible for managing SN resources. The primary component of the control segment is the Network Control Center (NCC), which schedules communication events on the SN. In the next few years, the NCC will undergo several upgrades in response to upcoming programs (e.g., STGT). In the late 1990s, the NCC is scheduled for an upgrade, which will significantly enhance the management of SN resources.

1.2 Scheduling the Space Network

Current scheduling practices, which rely heavily on manual conflict resolution strategies, are adequate for the current SN environment. Future changes to the SN are expected to severely impact the capability of the current scheduling operation. These changes could include partial or complete failures of TDRS spacecraft, new services proposed for TDRS II and an anticipated increase in mission loading. Furthermore, the SN has a requirement to support both classified and unclassified customers, therefore, customer visibility into the composite schedule remains severely restricted, complicating the conflict resolution process.

As part of the NCC upgrade, a customer POCC interface called SNUPI is being proposed. One of the primary functions of SNUPI is to support the scheduling interface between the POCC and the NCC. SNUPI and the NCC will implement a concept called flexible scheduling in order to increase scheduling efficiency.

1.3. Space Network User POCC Interface

Current operations use multiple systems to support communication interfaces between user POCCs and the NCC. For example, the User Planning System (UPS) is used for scheduling the SN, while a different interface is used for real-time operations. SNUPI is proposed as a complete interface tool kit that the POCC can access for all of its interactions with the NCC. A key feature of SNUPI is its support of the flexible scheduling concept.

2. FLEXIBLE SCHEDULING

Current SN scheduling schemes involve the submission of individual requests for specific resources at specific times. Flexible scheduling, however, permits POCCs to express requests for communications support in terms of their flexibility, repetition characteristics, and service and event constraints (Ref. 2). Flexibility may be expressed in terms of:

- Service start times (e.g., between 1200 and 1215)
- Service duration (e.g., 15 to 20 minutes long)
- Requested resources (e.g., any SA antenna on any TDRS)
- Repetition characteristics (e.g., repeat once/orbit)
- Service and event constraints (e.g., 2 to 5 minutes after sunlight entry)

Repetition characteristics may include:

- Number of event repetitions (e.g., 10 repetitions)
- Event repetition cycles (e.g., repeat every orbit)
- Summed event duration (e.g., as many repetitions as needed to provide a total of 7 hours of support)
- Repetition restrictions (e.g., 10 minutes minimum gap between repetitions)

Service and event constraints may refer to:

- Orbital activities (e.g., at apogee)
- Calendar activities (e.g., only on weekdays)
- POCC defined activities including other requests (e.g., after request Y starts)

An example of a flexible request embodying these capabilities is a request to schedule a 15 to 20 minute type A event on any single access (SA) antenna anytime after 1200 hours, except while over the South Atlantic Anomaly (SAA), every day for the next 2 weeks. This request statement expresses flexibility in the start time and duration of the support and in the resource required. It also expresses the repetitive nature of the required support. The repetition instructions in the request allow the POCC to submit one request while receiving multiple scheduled instances of that request (14 in this example). The request statement also expresses constraints limiting the potential times during which this request can be satisfied, based upon orbital activities. Figure 1 illustrates the flexible and repetitive dimensions of the request. Figure 2 provides a sample of how scheduling can be affected by constraint statements.

Increasing Repeatability ↑

Event: Type A Duration: 20 minutes • Repetition: every day Resource: TDRS E SA1 Start Time: 15:12:36 • Period: next 2 weeks	Event: Type A • Duration: 15-20 minutes • Repetition: every day • Resource: any TDRS SA • Start Time: after 12:00:00 • Period: next 2 weeks
Event: Type A Duration: 20 minutes Repetition: once Resource: TDRS E SA1 Start Time: 15:12:36 Period: Tuesday	Event: Type A • Duration: 15-20 minutes Repetition: once • Resource: any TDRS SA • Start Time: after 12:00:00 Period: Tuesday

Increasing Flexibility →

Figure 1. Flexible Scheduling Request Example

The use of flexible requests should improve scheduling operations because the information contained in them concerning time and resource flexibility:

- Facilitates automated conflict resolution
- Increases resource utilization (by scheduling more events)
- Reduces number of requests to be submitted (since multiple events can be scheduled from one request)
- Reduces request-response iterations between the NCC and the POCCs
- Reduces the need to coordinate scheduling options verbally (backup events can be specified in case the primary event cannot be scheduled)

Although more options make conflict resolution more complex, algorithmic solutions exist and automated processes have been successfully used (Ref. 3).

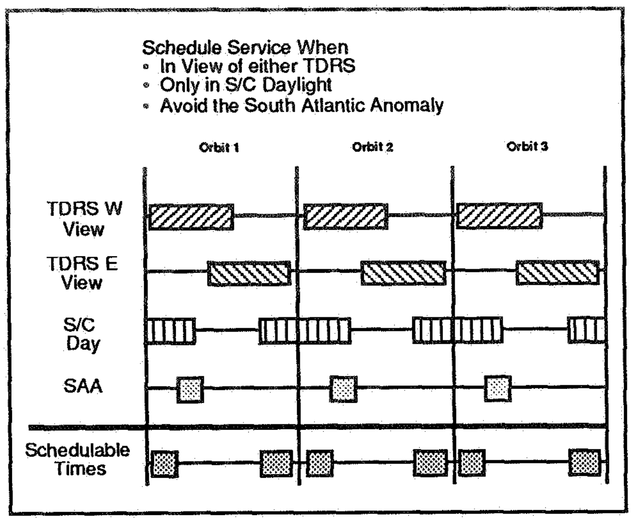

Figure 2. Constraint Relationships Example

3. THE SNUPI PROTOTYPE

The SNUPI prototype illustrates the flexible scheduling concepts as applied to the definition and management of POCC requests. SNUPI runs on a SUN SPARCstation and is written in C. The Transportable Applications Environment (TAE Plus), a NASA developed graphical user interface tool, supported user interface development. The initial version of the prototype was completed in seven man months. This prototype provided access to six top level functions:

- Request generation and editing
- Request validation
- Database management
- Request transmission
- Request status management
- Schedule management

3.1 Request Generation and Editing

3.1.1 The Flexible Request Language

Flexible requests are described to the scheduling system in a flexible request language. The flexible request language used in SNUPI prototype is based on the Flexible Envelope Request Notation (FERN) language (Ref. 4). FERN is English-like and human readable. As used in the SNUPI application, it has a hierarchical structure consisting of definitions of:

- Resources, time intervals, and configuration parameter sets
- Services (or steps)
- Events (or activities)
- Requests (or generics)

Resource definitions consist of the resources that may be scheduled and when they are available. Resource definitions also define pooled or equivalent resources. For example, the resource definition TDRS_East_SA_antenna represents a pooled resource composed of both SA antennas on the TDRS East spacecraft.

Time interval definitions describe relevant time intervals, usually related either to orbital events (e.g., when the spacecraft is in view of a particular relay satellite), or calendar events (e.g., a workday, defined as 09:00 to 17:00 EST Monday through Friday).

Configuration parameter sets indicate how to configure the TDRS spacecraft and ground terminal in order to support a desired service. Typical configuration parameters include data rates, frequency, polarization, and channel identification.

A service definition is comprised of the configuration parameter set and the resources that are required to provide the service. It may also include a list of constraints that apply to that service.

An event definition describes a collection of services as well as the duration of each service. A typical SN event consists of a forward service, a return service and a tracking service. Constraints that apply to the event as a whole, such as "must occur during spacecraft sunlight", are also stated in the event definition.

The request definition describes what events are to be scheduled and how often. Thus, the number of repetitions and the repetition cycles are part of the request definition. Requests also specify what events may be scheduled as backups, if the primary event(s) cannot be scheduled.

3.1.2 The Request Editor

Although the flexible request language is very readable, it still involves keywords and a structured syntax. In order to avoid forcing the POCC operator to learn tedious language syntax, a form fill request editor was developed for the SNUPI prototype. The editor allows the operator to create, edit, or delete the definitions noted above. Each definition is saved and can be retrieved for use in building other definitions. In general, there is a form for each definition, with an associated form for the specification of constraints. The forms support key-in data entry, menu selections, and check boxes. On-line help is provided as well.

The request editor takes the information entered in the forms and automatically formats it into the flexible request language syntax. At any point in the data entry process, the operator may elect to view the resultant definition in the request language. A text editor is provided for the knowledgeable operator who wishes to work directly in the request language. Syntax and error checking would be performed to the fullest extent possible in an operational system, but only limited error checking was implemented in the prototype.

3.2 Request Validation

Once a flexible request has been defined, the POCC operator should validate it to ensure that it will generate the type, quantity, and approximate placement of scheduled events desired. The request validation facility expands the request into its multiple individual instances and plots these instances on a graphical timeline. Selected constrains may also be plotted. This process allows the operator to see how many request instances will be generated from the request definition and where they are likely to be placed on the schedule. Missing, misplaced, and undesired request instances will be apparent. The operator then may modify the request definitions to correct these problems. Once the operator is satisfied with how the request expands, it is marked as valid.

Figure 3 shows a typical request validation screen. The operator may chose to expand the request along a relative timeline for general validation, or expand the request with respect to a specific time interval. The request may be expanded for one day, one week, or any arbitrary length of time the operator chooses. For example, if the repetition cycle is once a day, the operator may choose to expand the request for a week in order to view a number of repetitions.

3.3 Database Management

Validated definitions for requests and time intervals are entered into a database, and placed under configuration management. Once a definition is entered into the database, modifications to that definition would be tracked and controlled. The database manager provides the operator with the ability to add, delete, view, and sort definitions. The database manager would also support resource inventory updates from the NCC.

When a request is added to the database, the operator is prompted to enter the life span of the request. The life span of a request defines when the scheduling process should start to use the request, and when its use should be terminated. The life of a request may span multiple scheduling periods.

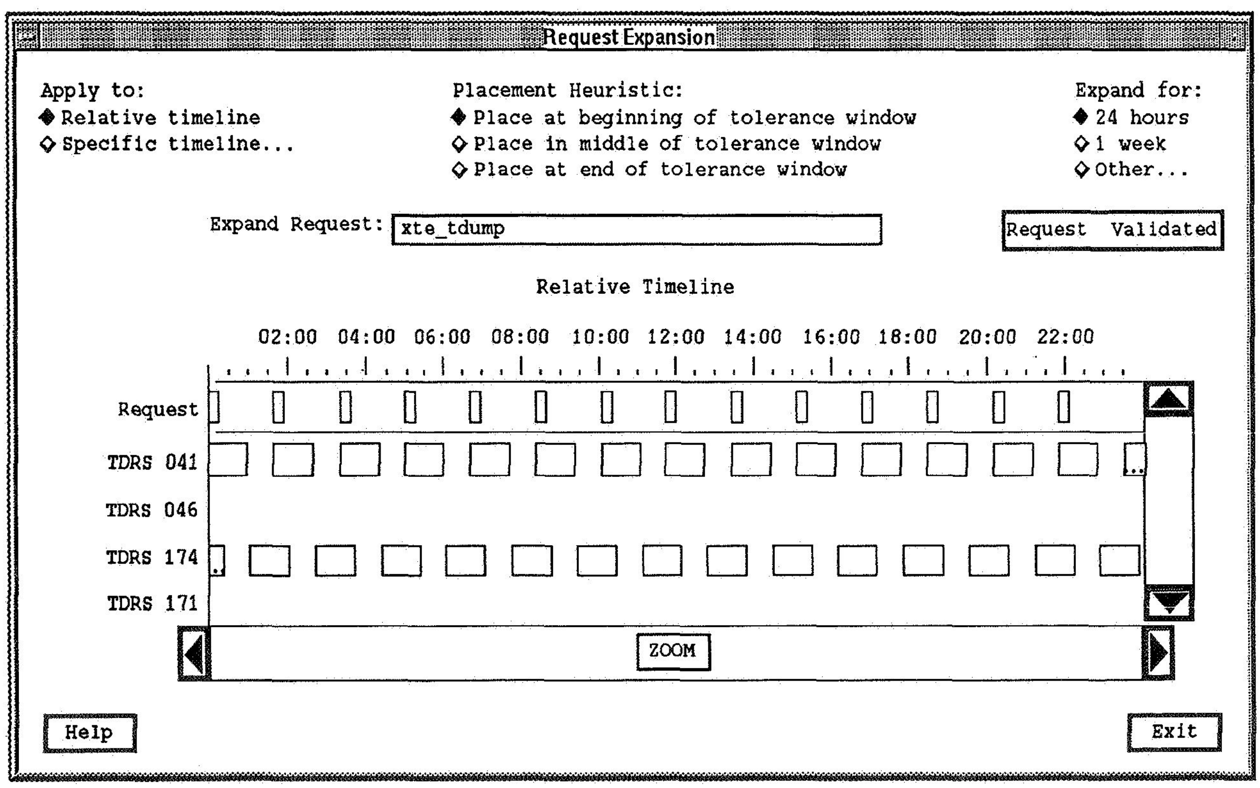

Figure 3. Flexible Request Validation Example

3.4 Request Transmission

The SNUPI prototype is part of a scheduling testbed which includes a flexible scheduling system called the Request-Oriented Scheduling Engine (ROSE) (Ref. 5) which prototypes the NCC scheduler. Requests residing in the SNUPI database can be transmitted via a TCP/IP connection to the scheduling system. When the scheduling system receives a request, an acknowledgment is transmitted back to SNUPI and displayed.

3.5 Request Status Management

SNUPI maintains and modifies the status of all requests from initial request generation, through request transmission, to schedule result. The status display prompts the operator to enter the scheduling period of interest, and then lists all requests relevant to that period and their current status for that period. A request status history is also available.

3.6 Schedule Management

After the NCC scheduling prototype has produced a schedule, each POCC's resulting events are transmitted to their SNUPI. The operator may display these scheduled events on a graphical timeline. Since SNUPI may be managing multiple schedules simultaneously, the operator must specify which scheduling period is of interest, and whether the primary or a contingency schedule is to be displayed. The operator may choose from a variety of display grouping options, such as group by resource scheduled. Details on any particular event may be viewed. The schedule may also be displayed in a tabular form. Information concerning unscheduled events and schedule summary statistics (e.g., total minutes allocated) is available as well.

4. SNUPI EVALUATION

The SNUPI prototype user interface and the Flexible Schedule Request (FSR) concept were evaluated by collecting subjective comments from potential users. The objectives of the evaluation were to determine the acceptability of the FSR concept from the mission operations perspective, and to gather feedback on the techniques used in the SNUPI prototype user interface. The evaluation plan addressed the process to be used for each evaluation session, identification of participants and evaluation materials.

Evaluations were generally conducted with one participant at time and were completed in about an hour. For each evaluation session, participants were provided with a brief overview of the SNUPI prototype, the FSR concept and the objectives of the evaluation. The SNUPI prototype was then demonstrated, following a strict scenario which highlighted features of the FSR concept (as described in Section 3), proposed capabilities to support the concept, and several user interaction techniques. Finally, reactions were obtained by having participants respond to an interview questionnaire.

Over fifty people participated in the SNUPI prototype evaluation. Participants were members of Goddard mission operations teams, managers and developers of operational systems. The evaluation material included a handout of the SNUPI overview, a brief description of the FSR concept, and copies of major SNUPI prototype display panels. The questionnaire requested information about the evaluator's background to determine their personal experience with systems similar to SNUPI in function, and asked whether they considered themselves to be managers, developers or operators. Statements were worded positively and then scored by evaluators as 5 for "strongly agree", 3 for "neutral" (or don't know) and 1 for "absolutely do not agree". The statements were intended to elicit subjective feedback regarding the perceived utility of the FSR concept. Words like "useful" were used when exploring the proposed operations concept, and "clear and easy to understand" for human-computer-interface concepts. Participants were also asked to comment on overall prototype strengths and weaknesses.

Responses to the questionnaires were numerically scored and then analyzed to calculate the average response to each statement by participant category and in total. Responses that were rated both high and low (20th percentiles) were further examined, with correlating comments from the participant groups.

The overall result of the evaluation indicated strong approval of the FSR concept, with scores ranging from 3.07 to 4.81, and averaging 4.21. In general, graphic displays of schedules were well received. Low rated statements tended to point to human-computer interface (HCI) difficulties. It was recognized that while requesting scheduled services is a complex task, SNUPI tools need to be more intuitive, perhaps prompting for the next required operator input. Participants recommended that future scenarios deal with some of the complexities currently experienced with SN scheduling, including handling rejected requests, handling more than a single request, and making modifications to scheduled events. For example, an infrequent activity may require that a single instance of a routine request have expanded service support.

The evaluation also indicated that users anticipate that extensive training would be required to develop detailed knowledge of the FERN language. Another key evaluation result was that scheduling SN resources is only part of the planning function for each GODDARD mission. Participants recommended integrating SNUPI scheduling functions with other POCC mission planning activities and tools.

5. CONCLUSIONS AND FUTURE EFFORTS

A key benefit of the flexible scheduling request concept is the shift of a significant conflict resolution effort from humans to computers. Whereas the current space network scheduling approach requires POCC operators to identify specific resource and time requirements in a fixed format request, the flexible scheduling request concept allows customers to describe their requirements in a readable language which accommodates flexibility. The FSR operations concept minimizes request-response iterations between the network scheduling system and the customer since multiple events can be scheduled from a single request (using repeatability specifications). Also, backup events can be identified and substituted in case the primary service is not available. The FSR concept supports automated conflict resolution strategies, since tolerances in start times and duration are provided. More events are scheduled, thus supporting effective TDRS resource utilization. The time to generate a week's worth of schedules can be reduced to hours instead of days. Finally, the potential exists to evolve the FSR concept into a standard for exchanging schedulig information on resource requirements and constraints with international space networks.

Currently a second version of the prototype is in process to address some of the weaknesses identified by the evaluators. One particular improvement is the addition of a new display illustrating the relationship of different services within an event. This display is included in the request editor, and later may be made available to both the request validation and schedule management facilities. A new graphical format is also planned to allow easy definition of time intervals.

To address the need for integrated tools in mission planning and SN scheduling which was identified by the evaluators, the prototyping effort will also explore ways to automate the link between mission planning tasks and the generation of flexible scheduling requests for SN resources. This prototyping effort will integrate SNUPI with an operational mission planning tool called the Explorer Platform Planning System (Ref. 6).

Smart HCI capabilities could be incorporated within the SNUPI prototype. Context sensitive defaults and intelligent error checking could be provided. For example, SNUPI may recognize that the majority of events defined in its database include a tracking service, and query the operator for the inclusion of a tracking service in any new event definition. The interaction style also could be extended to a demonstrational interface (Ref. 7). Demonstrational interfaces go beyond direct manipulation and allow the operator to abstract and generalize actions performed for repetition on similar objects. An intelligent demonstrational SNUPI would allow direct manipulation of event and service icons on a timeline in order to describe a request. Using heuristics, SNUPI then would determine the constraining factors and translate this information into the request language.

REFERENCES

1. Moe, Karen L., ed. Dec. 1990. Space Network Control Conference on Resource Allocation Concepts and Approaches, Conference Proc. NASA CP-3124.

2. Stanford Telecommunications, Inc. Jan. 21, 1992. Space Network Control System/Payload Operations Control Center Flexible Scheduling Concepts, Tech. Rept. TR-91-108, GSFC Contract NAS5-31260.

3. Munns, S., N. Goodman, T. Welden, and D. Crehan. Aug. 1991. Lessons Learned Prototyping Flexible Scheduling for the TDRSS Domain, CSC, Loral AeroSys and NASA/GSFC, DSTL-91-003.

4. Tong, G.M. Sept. 1991. Flexible Envelope Request Notation (FERN) User's Guide, NASA/GSFC, DSTL-91-001.

5. Grasso, B. and G. M. Tong. Jan. 1992. Request Oriented Scheduling Engine (ROSE) Software Maintenance Document, Loral AeroSys and NASA/GSFC, DSTL-92-025.

6. McLean, D., B. Page and W. Potter. 1990. The Explorer Platform Planning Systems: An Application of a Resource Reasoning Shell. In Proceedings of the First International Symposium on Ground Data Systems for Spacecraft Control. Darmstadt, Germany: ESA pp. 195-200.

7. Myers, Brad A. 1992. Demonstrational Interfaces: A step beyond Direct Manipulation. In Computer, 25, 8: 61-73.