Automating Mid- and Long-Range Scheduling for NASA's Deep Space Network

Mark D. Johnston[1],Daniel Tran[2], Belinda Arroyo[3], Sugi Sorensen[4], Peter Tay[5]
Jet Propulsion Laboratory – California Institute of Technology, Pasadena CA 91109

and

Butch Carruth[6], Adam Coffman[7], and Mike Wallace[8]
Innovative Productivity Solutions, Inc., Bulverde, TX 78163

NASA has recently deployed a new mid-range scheduling system for the antennas of the Deep Space Network (DSN), called Service Scheduling Software, or S^3. This system is architected as a modern web application containing a central scheduling database integrated with a collaborative environment, exploiting the same technologies as social web applications but applied to a space operations context. This is highly relevant to the DSN domain since the network schedule of operations is developed in a peer-to-peer negotiation process among all users who utilize the DSN (representing 37 projects including international partners and ground-based science and calibration users). The initial implementation of S^3 is complete and the system has been operational since July 2011. S^3 has been used for negotiating schedules since April 2011, including the baseline schedules for three launching missions in late 2011. S^3 supports a distributed scheduling model, in which changes can potentially be made by multiple users based on multiple schedule "workspaces" or versions of the schedule. This has led to several challenges in the design of the scheduling database, and of a change proposal workflow that allows users to concur with or to reject proposed schedule changes, and then counter-propose with alternative or additional suggested changes. This paper describes some key aspects of the S^3 system and lessons learned from its operational deployment to date, focusing on the challenges of multi-user collaborative scheduling in a practical and mission-critical setting. We will also describe the ongoing project to extend S^3 to encompass long-range planning, downtime analysis, and forecasting, as the next step in developing a single integrated DSN scheduling tool suite to cover all time ranges.

I. Introduction

The scheduling of NASA's Deep Space Network (DSN) presents a variety of challenges. As a unique and mission-critical capability serving an increasingly large user community, it is essential that the network be efficiently scheduled and utilized. At the same time, DSN users represent an enormous range of types of scheduling requirements and constraints, that vary substantially by type of mission, with time as spacecraft enter different mission phases, and seasonally as visibility intervals overlap in different ways. Against a backdrop of increasing budget pressure, the DSN has undertaken a project to replace its existing aging scheduling software with a new and unified tool suite intended to improve all aspects of the DSN planning and asset allocation and scheduling processes.

[1] Principal Scientist, Planning and Execution Systems Section

[2] Member of Technical Staff, Artificial Intelligence Group, Planning and Execution Systems Section

[3] Deep Space Network Planning and Scheduling Manager

[4] Systems Engineer, Deep Space Network Planning and Scheduling Section

[5] Mission Operations Engineer II

[6] President/Architect, IPS Inc.

[7] Senior Programmer, IPS Inc.

[8] Programmer, IPS Inc.

In this paper we first give a general overview of the DSN and the nature of its scheduling problem, followed by a brief description of the scheduling process and software systems (Section II). We then describe the Service Scheduling Software (SSS, or S^3) system, covering some key design elements as well as operational lessons learned from its initial deployment phase (Section III). This is followed by a description of the ongoing extension of S^3 to encompass long-range planning and forecasting functionality (Section IV). Finally we summarize progress to date and plans for future work in our conclusions (Section V).

II. Deep Space Network Scheduling Overview

NASA's Deep Space Network[1-3] (DSN) is comprised of a set of large (34m and 70m diameter) antennas and the associated equipment required to communicate with spacecraft, from those in high Earth orbit to the most distant man-made objects. These antennas are situated at three Deep Space Communications Complexes (DSCC), as listed in Table 1. The complexes are spaced roughly equally in longitude, to provide round the clock coverage for missions anywhere in space. Although capabilities vary from one antenna to another, and one complex to another, overall the DSN provides a range of S-, X-, and Ka-band up- and downlink services to all of NASA's and a number of international partner missions. These services include support for spacecraft telemetry, command, and tracking, as well as radio science, radio astronomy and Very Long Baseline Interferometry (VLBI), radar, and calibration.

Currently the DSN supports 37 spacecraft or service users, counting all those with regular requirements for scheduled time on any antenna. The mission users span a wide range of distance and orbit type: high earth orbit, lunar orbit, solar orbit, probes at Mercury, Venus, Mars, and Saturn (and en route to Jupiter and Pluto/Charon), and the asteroids, out to the two Voyager spacecraft in interstellar space. Ground-based users conduct radio science and radio astronomy using the antennas, including coordinated programs with international partners. Other activities that must be scheduled include routine and special maintenance, and calibration, engineering, and test activities. The collected set of DSN users presents a very wide range of usage requirements due to their different designs and operating modes. Some require occasional contacts representing only a few hours per week, ranging upwards to continuous coverage during certain mission phases, such as post-launch and during critical mission events. At the present time, a typical week includes about 500 scheduled activities on the antennas of the three DSN complexes.

Complex	GDSCC	CDSCC	MDSCC
Location	Goldstone, California, USA	Canberra, Australia	Madrid, Spain
Longitude	117° W	149° E	4° W
Latitude	35° N	35° S	40° N
Antennas	1 - 70m 5 - 34m	1 - 70m 2 - 34m	1 - 70m 3 - 34m
Capabilities	S, X, Ka	S, X Ka downlink only	S, X Ka downlink only

Table 1. Deep Space Network communications complexes and some of their characteristics

A. Phases of the DSN scheduling process

The DSN scheduling process consists of three phases, which do not have sharply defined boundaries. Below we describe these phases as they exist today; later in this paper we discuss plans for how they may change in the future.

Long-Range Planning and Forecasting. In today's system, long-range planning is based on user-provided high-level requirements, specified in the form of a spreadsheet that is interpreted by analysts and entered into a database at JPL. The forecast software employs a statistical allocation method[4, 5] to estimate when these requirements translate into DSN loading over various time frames. Long-range planning has several major purposes:

- studies and analyses: periods of particular interest or concern are examined to determine where there is likely contention among missions, for example around launches or critical mission events (maneuvers, planetary orbit insertion or landings), or when construction of a new DSN antenna is under investigation
- downtime analysis: identifying periods of time when necessary antenna or other maintenance can be scheduled, attempting to minimize the impact on missions
- future mission analysis: in proposal phase, missions can request analysis of their proposed DSN coverage as part of assessing and costing proposals for new missions

The time range for long-range planning is generally 6 months or more into the future, sometimes as much as years.

Mid-Range Scheduling. The mid-range scheduling phase is when detailed user requirements are specified, integrated, negotiated, and all tracking activities finalized in the schedule. Starting at roughly 4-5 months before execution, users specify their detailed scheduling requirements on a rolling weekly basis. These requirements include:

- tracking time and services required
- constraining time intervals and relationships (e.g. minimum and maximum gaps)
- visibility constraints
- flexibilities

A more detailed discussion of these requirements and flexibilities is included below in Section III.C; more detail is provided elsewhere[6, 7].

Once the deadline passes and all requirements are in, the full set is integrated into an initial schedule where conflicts are reduced by taking advantage of whatever flexibilities have been specified. There follows an optimization step where an experienced DSN scheduler interactively edits the schedule and further reduces conflicts by taking advantage of unspecified flexibilities and making further adjustments. At the conclusion of this phase, the schedule usually contains a fewer than 30 conflicting sets of activities. It is then released to the scheduling user community who negotiate to reduce conflicts and further optimize coverage for their missions. This phase generally lasts 7-8 working days, after which the schedule is conflict free or has only waived conflicts for specific reasons. This is considered the "negotiated schedule" that missions use to plan their integrated ground and spacecraft activities, including the development of onboard command loads based in part on the DSN schedule.

Following this point, changes to the schedule may still occur, but new conflicts may not be introduced. There is a continuing low level of no-impact changes and negotiated changes that occur all the way down to real time.

Near Real-time Scheduling. The (near) real-time phase of DSN scheduling starts roughly eight weeks from execution and includes the period through execution of all the scheduled activities. Late changes may occur for various reasons (sometimes impacting the mid-range phase as well):

- users may have additional information or late changes to requirements for a variety of reasons
- DSN assets (antennas, equipment) may experience unexpected downtimes that require adjustments to the schedule to accommodate
- spacecraft emergencies may occur that require extra tracking or changes to existing scheduled activities

For many missions that are sequenced well in advance, late changes cannot be readily accommodated.

B. Status of DSN Scheduling Software

The DSN scheduling software systems represent a collection built over many years and interfaced in a very heterogeneous manner[8-10]. At the present time, the different stages of the scheduling process are supported by different tools and databases as indicated in Table 2.

The DSN has undertaken an overall unification and simplification of the scheduling software systems[6, 7, 11-13], of which the first increment has been operational since June 2011. This is called the Service Scheduling Software (SSS, or S^3) and has initially been applied only to the mid-range process. S^3 is described in more detail below.

Process Phase	Software tools (software/database)	Characteristic activities
Long-range	TIGRAS (RAP version) + MADB database	• identify and resolve periods of contention • plan for extended downtime • assess proposed missions • assess long range asset options
Mid-range	S^3 webapp/database	• schedule normal science operations • schedule pre-planned s/c activities (maneuvers, unique science opportunities) • generate negotiated schedules for s/c sequencing • schedule network maintenance
Near Real-time	TIGRAS (SPS version) + Service Preparation System (SPS) database	• predict generation for execution • reschedule due to unplanned resource unavailability • respond to spacecraft emergencies • activate pre-planned launch contingencies

Table 2. DSN software tools that support the different phases of the DSN scheduling process.

At the current time, DSN is deploying the first major update to S^3, as well as developing the initial stage of the replacement software for long-range planning and forecasting. In the following sections we provide an overview of S^3, including lessons learned from its initial deployment and how these have been addressed. We then discuss the current development effort for the long-range planning software, and how it leverages the mid-range software design to provide an improved set of long-range planning functionalities.

III. DSN Service Scheduling Software

The DSN Service Scheduling Software (SSS or S^3) project has had a long gestation period, but implementation of the mid-range stage started in December 2008. The system was developed in iterations which were made available to users as prototypes in order to get early feedback. The initial operational deployment came in June 2011, and the system has been in use for requirements entry and integration since December 2010, and for negotiation since April 2011.

The mid-range scheduling process was described briefly in Sect. II.A above and has several unique characteristics worth elaborating on as the basis for several key architecture and design decisions about S^3.

A. Mid-Range Scheduling Process Implications on S^3

Unlike other NASA network scheduling processes, DSN does not have a centralized scheduling authority that develops and publishes the schedule. Instead, the schedule is developed collaboratively via a negotiation process, on a rolling weekly basis. About 20 individual schedulers (not all of whom are full time), representing 37 DSN users, provide scheduling requirements to start the process each week. They then participate in negotiation where these requirements come into contention for oversubscribed assets. This negotiation process consists of a series of proposals and counter-proposals (typically up to around 40) that eventually converge on a mutually agreed schedule without conflicts.

Before S^3, requirements were provided via email or spreadsheets or word processor documents, and integrated manually by a team dedicated to this purpose. Negotiation took place via email or face-to-face meetings or telecons. There was no generally accessible schedule database for this phase of the process.

S^3 provides support for all the key elements of the mid-range process, based on a modern web application and integrated, accessible database[14] (see Figure 1). Users can directly enter their own scheduling requirements[6] and verify their correctness before the submission deadline. The database in which requirements are stored is logically divided into “master” and “workspace” areas. There is a single master schedule representing mission-approved requirements and DSN activities (tracks). Each user can create an arbitrary number of workspace schedules, initially either empty or based on the contents of the master, within which they can conduct studies and 'what if' investigations, or keep a baseline for comparison with the master. These workspaces are by default private to the individual user, but can be shared as readable or read-write to any number of other users. Shared workspaces can be viewed and updated in realtime: while there can only be one writer at a time, any number of other users can view a workspace and see it automatically update as changes are made. These aspects of the web application architecture and database design support the collaborative and shared development nature of the DSN schedule.

In addition, S^3 offers some additional features to facilitate collaboration, including an integrated wiki for annotated discussion of negotiation proposals, integrated chat, notifications of various events, and a propose/concur/reject/counter workflow manager to support change proposals. Details on the design and use of the S^3 collaboration features are provided elsewhere[14].

Underlying the web application and database is a scheduling automation component, the DSN Scheduling Engine[13] (DSE). The DSE provides a range of functions based on the semantics of the DSN scheduling domain, including:

- expanding scheduling requirements into tracking or other activities
- checking for and identifying conflicts in the schedule, i.e. situations that violate any DSN scheduling rules
- checking for and identifying requirement violations in the schedule, i.e. situations where activities in the schedule do not meet the user's specified requirements and constraints
- deconfliction algorithms that attempt to reduce conflicts or violations while preserving satisfied requirements

The DSE is based on a distributed session-oriented infrastructure running the ASPEN planning system (refs) with a DSN domain adaptation layer (refs).

In the following we focus on some specific design elements of S^3 and the DSE, as the basis both for “lessons learned” as well as the extension to long-range planning functionality.

B. S^3 Key Design Elements

Figure 1 shows a block diagram illustrating the main elements of the S^3 design. Users interact with the system as indicated on the left, via a web browser that communicates securely with the web application running on a server at JPL. A wiki is used to record general information about scheduling process status, as well as details relevant to

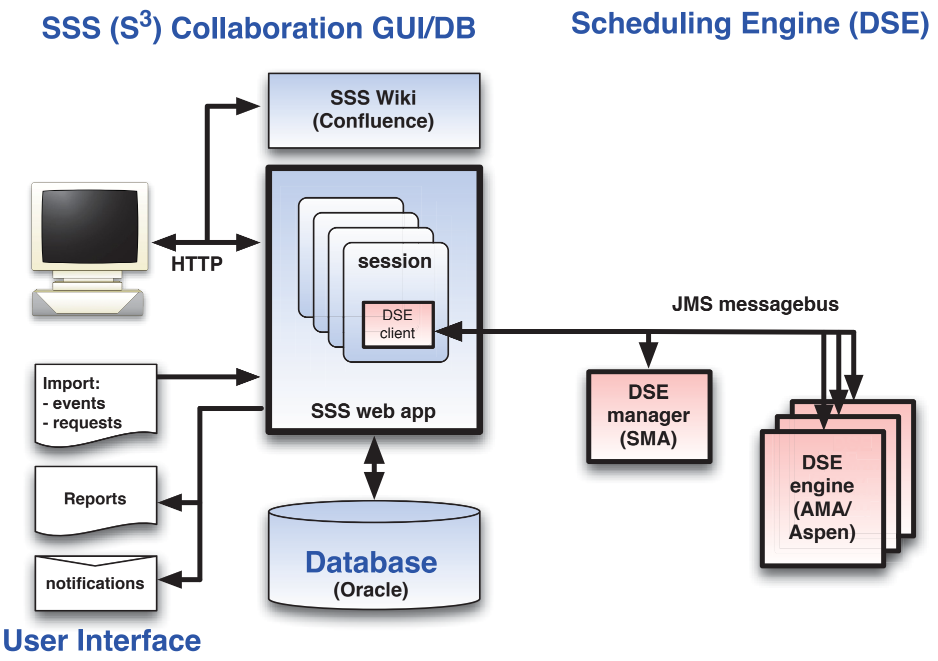


Figure 1. Block diagram of S^3 software elements and how the user interacts with the system.

negotiations in progress and to individual user scheduling requirements. File imports and exports are directly via the browser to/from the web application. Notifications of various events (proposal status changes, shared workspaces, new conflicts) are provided via an in-application mechanism, but at least twice a day an email notice is sent as a reminder of current negotiation status.

The central database is Oracle and stores schedule and requirement data for both the baseline master schedule as well as a large number of user workspaces. The scheduling engine (DSE) runs in a distributed manner on separate hosts and communicates via a Java Messaging System messagebus with the web application. The DSE Schedule Manager Application (SMA) mediates requests for new scheduling sessions on behalf of webapp users, and each user session is dedicated to an Aspen instance via the Aspen Manager Application (AMA). While there is only one instance of the S^3 web app running at a time, there are about 100 AMA/Aspen instances available, to eliminate any delays when users require the services of the scheduling engine.

C. DSN Scheduling Requests

One of the key innovations at the heart of S^3 is the notion of *request-driven scheduling*, that is, the schedule is determined not by specific track allocations (which are an output product of the process) but by scheduling requests, requirements, and constraints that represent a service-oriented approach to scheduling. The intent is to move towards a more abstract basis for scheduling, where users are allocated services that can be more flexibly provided by the network. Scheduling requests as implemented in S^3 include the following types of information. Note that much of this is optional, i.e. either by defaulting or as only relevant if some dependent choices require elaboration.

DSN users represent their needs to the S^3 software system as *scheduling requests*. The main elements of a scheduling request are:

- *Service specification*. S^3, via the DSE, provides an abstraction level on top of DSN asset specifications that may be referenced by users much more simply than specifying all of the possible options. At the physical level, the spacecraft onboard electronics (frequency band, data rates, encoding), radiated power, distance, along with the DSN antennas, receivers and transmitters and other equipment, determine what space and ground configurations are feasible. The abstraction level provided in S^3 is called a "service alias" such that a single service alias encapsulates a wide range of options, preferences, and associated information that is required to schedule the network. For example:
  - some users need the added sensitivity of more than one antenna at a time and so must be scheduled as antenna arrays using two or more antennas at once[15] (as many as four at a time)
  - for navigation data, there are special ranging scenarios that alternate the received signal between the spacecraft and a nearby quasar, over a baseline that extends over multiple DSN complexes

- for Mars missions, there is a capability for a single antenna to communicate with several spacecraft at once (called Multiple Spacecraft Per Aperture, or MSPA): while more than one at a time may be sending data to Earth, only one at a time may be receiving an uplink

A more detailed description of service alias functionality is provided below.

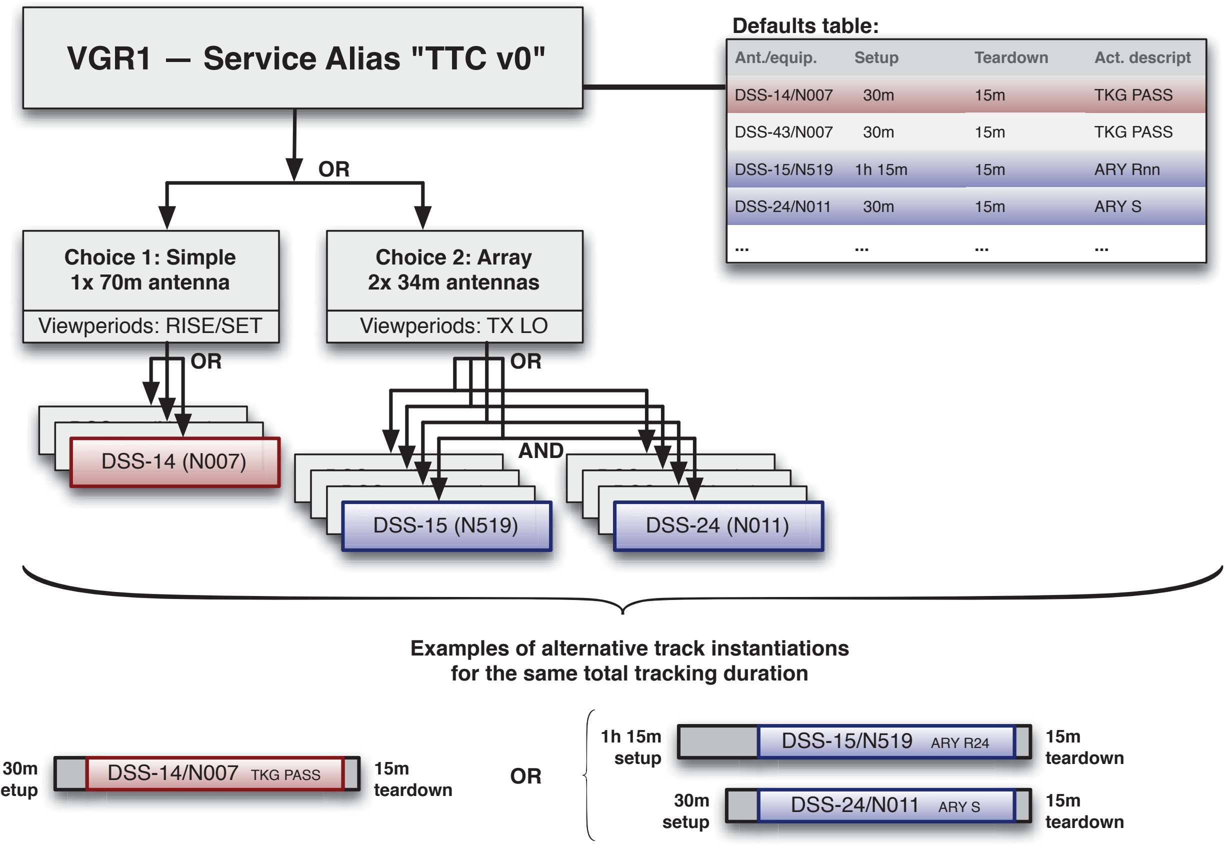
- *Timing constraints*. Users need a certain amount of communications contact time in order to download data and upload new command loads, and for obtaining navigation data. How this time is to be allocated is subject to many options, including whether it must be all in one interval or can be spread over several, and whether and how it is related to external events and to spacecraft visibility. Among the factors that can be specified in a schedule request are:
  - reducible: whether and how much the requested time can be reduced, for example to resolve conflicts
  - extendable: whether and how much the request time can be extended, should the option exist
  - splittable: whether the time must be provided in one unbroken track, or can be split into two or more separate tracks
  - split duration: if splittable, the minimum, maximum, and preferred durations of the split segments; the maximum number of split segments
  - split segment overlap: if the split segments must overlap each other, the minimum, maximum, and preferred duration of the overlaps
  - split segment gaps: if the split segments must be separated, the minimum, maximum, and preferred duration of the gaps
  - viewperiods: periods of visibility of a spacecraft from a ground station, possibly constrained to special limits (rise/set, other elevation limits), and possibly padded at the boundaries
  - events: general time intervals that constrain when tracks may be allocated; examples include: (a) day of week, time of day (for accommodating shift schedules, daylight, ...), (b) orbit/trajectory events (occultations, maneuvers, surface object direct view to Earth, …). Different event intervals may be combined (with optional inversion), and applied to a request.
- *Track relationships*. In some cases, contacts need to be sufficiently separated so that onboard data collection has time to accumulate data but not overfill onboard storage. In other cases, there are command loss timers that are triggered if the time interval between contacts is too long, placing the spacecraft into safemode. During critical periods, it may be required to have continuous communications from more than one antenna at once, so some passes are scheduled as backups for others.
- *Priority*. The DSN currently has a priority scheme which ranges from 1-7 with 7 being nominal tracking, and 1 representing a spacecraft emergency. Priority is relatively infrequently used, but it does have the effect that the scheduling engine will try to avoid conflicts with higher priority activities if possible. Depending on their degree of flexibility, missions trade off and compromise in order to meet their own requirements, while attempting to accommodate the requirements of others. As noted above, one of the key goals of S^3 is to facilitate this process of collaborative scheduling.
- *Preferences*. Most preferences are incorporated in the service alias and timing requirements described above, but some are directly representable in the scheduling request. For example, users may choose to schedule early, centered, or late with respect to the viewperiod or event timing interval.

One characteristic of DSN scheduling is that, for most users, it is common to have repeated patterns of requests over extended time intervals. Frequently these intervals correspond to explicit phases of the mission (cruise, approach, fly-by, orbital operations). These patterns can be quite involved, since they interleave communication and navigation requirements. S^3 provides for repeated requests, analogous to repeated or recurrent meetings in calendaring systems, in order to minimize the repetitive entry of detailed request information.

D. S^3 Service Configurations

One of the challenges of modeling the DSN scheduling domain is the wide range of options available for making use of the network. As previously described, one of the primary attributes of a scheduling request is the specification of the DSN services that are needed, which must be transformed into a set of specific resource reservations to satisfy the request. It has been a key element of the S^3 design that users can specify their needs at a more general and abstract level, and that the system will translate that into the details, ensuring the right antennas and equipment are scheduled. This has the obvious advantage that there is flexibility in the implementation of a request that can be used by the DSN systems, e.g. to optimize the schedule or to re-schedule on short notice in case assets go down. At the same time, the scheduling system needs to handle a very detailed specification of requested tracking time, down to the selection of individual antennas and equipment types to be reserved. A design to accommodate this spectrum of possibilities has been developed and implemented in the DSE, and is illustrated in Figure 2.

Each DSN service user or mission must define one or more service configurations, which are referred to by a name or “alias”. Each configuration specifies the following information:



Defaults table:

Ant./equip.	Setup	Teardown	Act. descript
DSS-14/N007	30m	15m	TKG PASS
DSS-43/N007	30m	15m	TKG PASS
DSS-15/N519	1h 15m	15m	ARY Rnn
DSS-24/N011	30m	15m	ARY S
...	...	...	...

Figure 2. An illustration of the structure of of a service alias representing a choice between a single antenna and multiple antenna (array) implementation of the same tracking duration. *Red highlights the information related to a single-track choice (left) and blue that related to a two-antenna array choice (right). More complex aliases are used to represent up to 4-station arrays, specialized ranging tracks (DDOR), separate uplink and downlink options for multiple spacecraft tracked all at once, and maintenance activities that affect an entire complex or the entire DSN at once.*

- one or more choices for how antennas and equipment can be allocated to meet the user's DSN requirements
- for each choice, which sets of antenna and equipment are acceptable
- for each antenna/equipment combination, what are the default values for associated tracking parameters:
  - setup and teardown time before and after the track
  - the 16 character activity description for the track
  - a standardized work category used to identify the kind of activity
  - if applicable, a code for a specific sequence of events that define all steps that occur during the track

A "choice" within an alias represents a high-level configuration option. For example, some missions may require either a single 70m antenna, or two or more arrayed 34m antennas. Each of these possibilities corresponds to very different antenna selections, while still satisfying the requirements of the overall service specification. Within a choice, all acceptable sets of antennas/equipment combinations must be specified, in preference order (if applicable). Antenna/equipment combinations within a single antenna choice are in the form of a single list, while those in array choices contain multiple such lists. The same antenna may play different roles within these options, for example as a reference or slave antenna depending on how the equipment is to be configured.

Depending on the nature of the activity, different times must be scheduled for the activity setup (before tracking starts) and teardown (after it completes). Typical setup times are 30 to 90 minutes, while teardown times are usually shorter. The alias definition specifies the default (minimum) setup and teardown time for each antenna/equipment option. In special circumstances these times may be lengthened, but may not be shortened without violating DSN operational rules (and causing a setup or teardown conflict).

Once aliases are defined and validated, their usage in S^3 is straightforward. Whenever a user creates a scheduling requirement, a service alias must be specified, simply by name. The selected alias then determines all the remaining DSN asset requirements and options, while the remainder of the requirement goes on to specify parameters such as timing, duration, and relationships to other tracks. By separating the definition of aliases from their usage, it becomes easier to validate them to ensure that any selection is a legal DSN configuration for that service user.

Most DSN service users will define at least several aliases corresponding to their commonly used scheduling configurations. For example, one alias might specify downlink-only configurations, while another might be used for both downlink and uplink: the latter requires the allocation of transmitters as well as receivers and decoders.

In addition to specifying which service alias applies to a given requirement, S^3 provides a capability for overriding the definition of that alias in any requirement in which it is used. An alias override can only *restrict* the full set of choices allowed by the alias, not add additional ones. As a result, validating the alias is sufficient to ensure that only legal configurations can be generated by the scheduling system. Examples of possible alias overrides include the following, to limit to –

- a single antenna vs. an arrayed configuration
- one or more DSN complexes (Goldstone, Canberra, or Madrid)
- a specific antenna subnet (70m, 34m, ...)
- a single specific antenna and equipment combination

As well as filtering the set of antenna and equipment choices, users can also override the default values associated with any choice. For example, a particular requirement might need an extended setup time, or customized activity description string that differs from the default. These can be specified using alias overrides.

In addition to antenna and equipment options, certain other attributes of any corresponding activities are also specified by the alias. These include:

- which kind of viewperiod must be used for scheduling, i.e. geometrical rise and set vs. higher elevation transmitter limits
- whether the activity is downlink or uplink only, which is used when scheduling MSPA activities as described in the next section
- special activity description suffixes that must be included to indicate certain types of activities
- an effective date and time range, which allows for phasing alias definitions in or out of service as ground or spacecraft configurations change

Service alias definitions are currently captured in XML files that specify all properties of the alias. A key design feature of the service alias concept in S^3 is that the same XML files are used by the DSE as the domain-specific model of DSN activities and assets, and in the S^3 web application GUI as the set of all legally selectable choices. Any changes to assets, aliases, or other mission parameters are immediately reflected in the DSE as well as the GUI, without code changes.

E. Lessons Learned from Initial S^3 Deployment

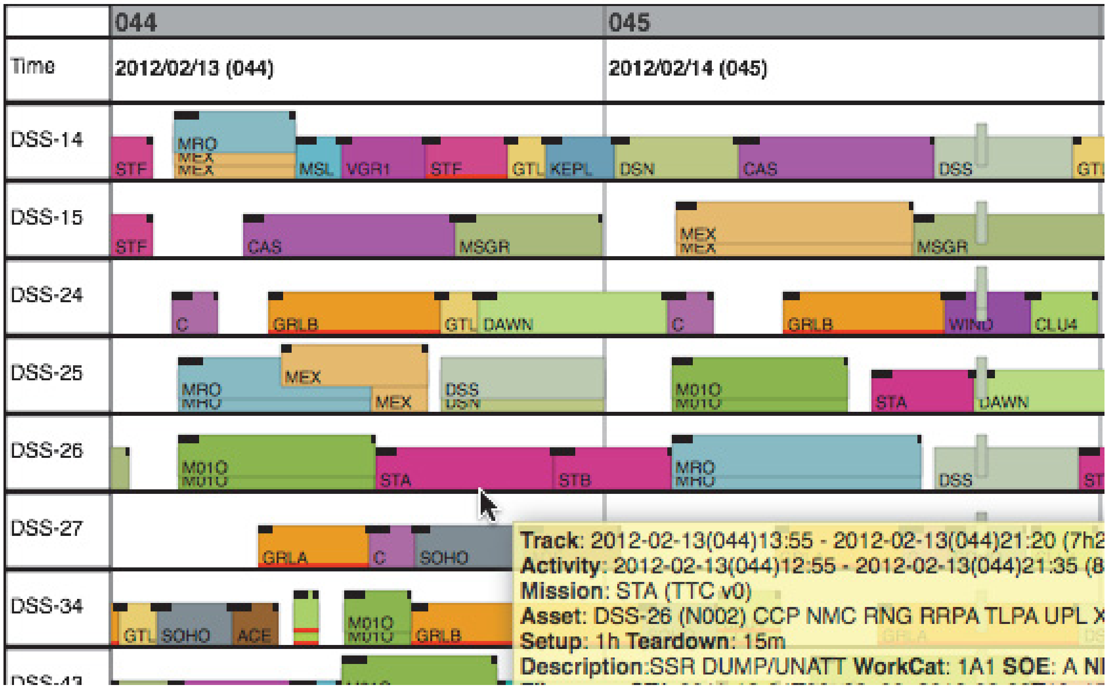
The initial deployment of S^3 to operations occurred in June 2011. Since that time, a number of areas have been identified as most in need of revision and updating. In this section we highlight the most important of these and what has been done leading up to the first major update to the software

Performance. Performance has proven to be the single largest issue. A typical week of DSN schedule consists of approximately 500 tracking activities. All versions of all requirements and activities are retained in the database. In addition, all past versions of conflicts and requirement violations were also retained. The access patterns for this volume of data, with its complex linkages, proved to be significantly slower than originally anticipated. In addition, the graphical scheduling components of the user interface were implemented in Adobe Flash, which requires a web browser plugin as well as specialized data pathways back to the host web server. As a result of all this, the system was insufficiently responsive when working on populated schedules.

Following an analysis and incremental change to the database storage approach, certain tables were denormalized, leading to a significant speedup in database queries for retrieving and displaying data. Following these changes, the largest contributor to slow performance was determined to be the Adobe Flash component. Adobe Flash is not intrinsically slow, but for large volumes of graphical data (thousands of DSN tracking activities), it requires a great deal of Flash software design and optimization to provide a responsive graphical user interface. This would have required the investment of substantially more resources than available. The decision to use Adobe Flash was made at a time several years ago when there were few alternatives. However, since that time, the evolution of web standards has made tremendous strides, and in particular the development of HTML5 and its adoption as a standard by the major web browsers has enabled several alternative approaches, including the HTML5 canvas and SVG features.

Following a feasibility investigation in August 2011, it was determined that the HTML5 canvas feature would be sufficiently performant and flexible to support all the scheduling graphics needs of S^3, and so the replacement of Flash by HTML5 canvas was initiated. Among the benefits of this approach are the following:

- simpler architecture: no plugin required, canvas is managed by javascript that co-resides with other elements of the webapp GUI, thus simplifying integration



Track: 2012-02-13(044)13:55 - 2012-02-13(044)21:20 (7h2
Activity: 2012-02-13(044)12:55 - 2012-02-13(044)21:35 (8
Mission: STA (TTC v0)
Asset: DSS-26 (N002) CCP NMC RNG RRPA TLPA UPL X
Setup: 1h **Teardown**: 15m
Description:SSR DUMP/UNATT **WorkCat**: 1A1 **SOE**: A NI

Figure 3. Example of an HTML5 canvas view of a portion of the DSN schedule. *Mousing over a track brings up a transient window with detailed information about the activity (lower right). In this view, different missions are color coded, and setup/teardown is indicated by the black segments at the top left and right of each activity. Each timeline represents one of the DSN antennas.*

- no vendor lock-in: the HTML5 canvas element is supported by all major browser platforms
- performance improvements driven by browser vendors: all of the browser platforms have made large javascript performance improvements, which can be leveraged by a canvas GUI scheduling component

As a rough measure, the HTML5 canvas component (Figure 3) turned out to provide about 10x faster performance when rendering schedules in the web GUI. As a representative point, a one-week schedule containing about 500 activities can be displayed in about 2 seconds, while an 8-week schedule with 4,000 activities takes less than 8 seconds. These times can be reduced when running over a slower network connection, or on a computer where the browser is competing for resources with other applications.

Traceability. Although S^3 retains the history of every track and requirement in the database, there were no features initially included to report this information out to users. From a usability perspective, users wanted to know not just what the current proposed or approved schedule contained, but how it got that way. In response, several features were added in the latest update to make it possible for users to see this information:

- all of the actions taken in a workspace (shared or private), with timestamp, details about the activity, and who performed it
- the history of previous versions of an activity in a workspace or the master schedule
- the history of all activities derived from a specified requirement, in a workspace of the master

These history features make it possible to tell in detail how activities got to their final states in the schedule, arbitrarily long after the fact.

Undo. As with many web-based database applications, changes to the S^3 schedule were committed to the database when made, without a capability to “back up” in time and revert recent changes. As with traceability, this is a usability concern that has been addressed to some degree in the current S^3 release.

The use case of particular concern for “undo” was schedule editing, where users could potentially make a succession of track changes while trying to resolve a conflict, but then need to back up should the approach taken lead to a dead end. To address this, the HTML5 canvas revision of the schedule editor GUI provides for a client-side “change list” that is only committed to the database when specified by the user (or in certain other circumstances described below). Items in this change list are constantly displayed in the GUI and can be undone in reverse order, just as in a conventional desktop application such as a word processor. The format of the change list makes it easy to spot exactly what as changed, and multiple attribute changes to the same track are merged into a single item in the change list.

The tradeoff with a client-side change list is that work can be lost in case the browser crashes, network connectivity is lost, or the user switches context and forgets to commit their pending changes. To mitigate this, S^3 has implemented an “auto-save” feature that commits changes after 20 items are put into the change list, or after 30 minutes of unsaved changes. In addition, the user is prevented from refreshing the page or switching to another S^3 context without confirming that the pending changes will be lost.

There is a second use case for “undo”, not yet addressed, that of reverting back past explicit user “save” operations. S^3 already provides a “save point” function that allows the user to make a snapshot of the state of a workspace, then revert back to that state at any time in the future. A future feature under consideration is a way to create automatic save points before each user save operation, thus allowing the option to return to any previous state with minimal effort.

IV. Long Range Planning and Forecasting

Long range planning is the most extended of the three phases of the DSN planning and scheduling process. The existing software tools used for this purpose were developed some years ago and have a number of shortcomings:

- they require users to provide requirement inputs in a week-by-week spreadsheet form, not necessarily straightforward to map to a mission planner’s activity timeline
- they require a translation step by trained analysts in order to enter user input into the forecast database
- the statistical method used, that of approximately allocating time equally across legal times in view, has accuracy limitations, especially at fine time granularity
- DSN capabilities developed since the tools were written are not modeled, such as Multiple Spacecraft Per Antenna (MSPA), which is commonly used at Mars to maximize efficient network utilization

As a result, the DSN has started on the first phase of a project to replace the current long-range planning tools by a new capability, designated Loading Analysis and Planning Software (LAPS), building on the functionality provided by S^3 for mid-range scheduling. In this section we discuss how the S^3 capabilities described above are being leveraged to provide long-range planning functionality, and where some additional needs are being met by infusion of other new technologies.

A. Similarities and differences between DSN long-range and mid-range processes

By necessity, there are many similarities between the mid- and long-range processes. Underlying both is the set of current and future DSN assets, including antennas and equipment, some coming into service and others being decommissioned. Both are based on DSN usage requirements from a varying mission set with a wide range of time-dependent tracking and navigation needs. Both are charged with arriving at an ultimately feasible allocation of DSN resources by balancing user needs and resolving periods of resource contention.

However, long-range planning has some significant differences from mid-range:

- long-range planning has to deal with numerous and sometimes intrinsic sources of uncertainty, including:
    - unpredictable spacecraft locations for some missions and trajectory types, leading to uncertainties in visibility times from the different DSN antennas
    - unknown science targets beyond some time horizon in the future
    - uncertainties in the mission set, due to funding changes, launch date changes, or mission extensions
- optimization criteria and scenarios differ from mid-range, where the main objectives are to minimize conflicts in the schedule and violations of user requirements; for long-range a variety of other objectives may come into play, including:
    - identifying best times to schedule extended downtime for preventive maintenance, minimizing the impact on active missions
    - identifying best times to schedule special flexible but resource intensive operations, such as reference frame calibration activities
    - maximizing the satisfaction of requirements where, due to contention, not all requirements can be satisfied across the entire DSN user base

In addition, long range planning needs to provide information to mission planners about where contention with critical events may occur, so that this can be taken into account as early as possible in each mission’s planning process. In many cases this needs to be provided during the mission proposal phase when, for both feasibility and costing, is is necessary to map out DSN allocation needs to some preliminary level of accuracy. Such proposal studies also impose a requirement for protection of proprietary or competition-sensitive information, whereas the midrange process for DSN provides general access to scheduling requirements and to the schedule itself.

Finally, long-range planning needs to support specification of a more abstract type of requirement with less detail than would be acceptable in mid-range. This serves two purposes: it represents at a coarse level some of the uncertainty in requirements, and it makes it easier to specify “what if” alternative scenarios.

B. Leveraged development of long-range planning and forecasting tools for DSN

Building first on the similarities noted above, the first phase of development of LAPS tools will make direct use of a number of capabilities already deployed operationally in the mid-range S^3 software, including:

- the model of DSN asset availability including antennas and equipment, with time-varying availability for new construction or new types of equipment, and out-of-service dates for retired assets
- the model of DSN user and mission types, including
  - ground- and space-based users, schedulable on non-interference basis or not
  - multi-spacecraft constellations
  - Multiple Spacecraft Per Antennas (MSPA) groupings and their special scheduling rules
- the service alias model, which defines what asset sets are allowable and preferable for a user, depending on the service desired (described in more detail in Section III.C)
- the viewperiod model, specifying legal visibility intervals of various types, calculated by the Service Preparation System and imported in a form optimized for scheduling
- the scheduling requirement model, allowing (but not requiring) allocation needs to be specified to the same level of detail as mid-range requirements (Section III.B), should such detail be both available and necessary for the type of study to be undertaken
- the DSN Scheduling Engine algorithms used in the mid-range process, which would allow for fully detailed "what if" generation of hypothetical mid-range schedule periods in those cases where sufficient detail is available to warrant this level of analysis

Re-use of the S^3 software base in these areas provides a large degree of leverage in the development of LAPS, but several other areas are also being addressed with additional capabilities:

- a planning request representation to allow for more abstract and high-level specification of allocation needs than the scheduling requirement model allows (for example "3x 8hr tracks/week on 34m BWG for the 6 months of interplanetary cruise"); at the same time, planning requests will be convertible automatically into mid-range scheduling requests into order to minimize duplicate data entry and speed up the mid-range process
- the capability to define and run planning scenarios in an automated way, such as:
  - to assess a range of options for downtime placement
  - to evaluate nominal and fallback requirement options for resource contention periods
  - to quantify the impact of a mission's alternative launch dates on projected resource loading
- a multi-objective optimization mechanism to automatically generate a portfolio of candidate plans/schedules optimizing the tradeoffs among multiple quantitive objectives

The incorporation of multi-objective optimization into LAPS offers a new way to optimize DSN resource allocations, taking into account that there is no single objective that captures all of the disparate goals and objectives that are important. Multi-objective optimization has been employed in a wide variety of problem domains, including scheduling for science missions[16-19] and generating some requirements inputs to the DSN mid-range process[20].

The initial phase of LAPS development will encompass the modeling and optimization noted above. The second phase will extend the user interface elements of the software to allow end users, such as mission planners and schedulers, to directly enter their own planning requirements and conduct "what if" analyses using a baseline DSN asset and mission model. It will also include an extended report generation mechanism to generate a wider variety of tabular and graphical output formats.

V. Conclusion

In this paper we have described the DSN scheduling process and software, including the initial operational deployment of the Service Scheduling Software (S^3) system, and its ongoing extension to support long-range planning and forecasting. S^3 represents a new approach to scheduling the DSN, embodying a request-driven approach to scheduling along with a collaborative peer-to-peer negotiation environment using modern web application and database technology. Future work is expected to address a number of areas including:

- *extension to real-time scheduling* – as described above, this third phase of the DSN scheduling process covers the period from execution out to about several weeks in the future. Extending S^3 to support this phase involves some challenging technical problems of integration with existing systems and support for contingency scheduling (e.g. launch slips, unplanned asset downtime); at the same time, bringing the information model of S^3 into the real-time domain will allow for decision making across options that are not now possible
- *cross-network scheduling* – NASA has recommended[21] integrating access to the capabilities provided by its three major networks: DSN, the Space Network (SN), and the Near Earth Network (NEN). For those users requiring services from two or all three of these networks, such integration would be a source of significantly improved efficiency and cost savings. S^3 has the potential to serve as a common scheduling platform in this regard – it is interesting to note that nowhere on the S^3 scheduling request editor main UI is there any indication that the user is working with the DSN; this is apparent only when drilling down into the detailed visibility and event intervals, and service alias definitions.

Acknowledgments

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. We gratefully acknowledge the support of the DSN scheduling community over the course of this work.

References

[1] "NASA's Deep Space Network," http://deepspace.jpl.nasa.gov/dsn/, 2010.

[2] Imbriale, W. A. *Large Antennas of the Deep Space Network*: Wiley, 2003.

[3] Berner, J. B., and Statman, J. "Increasing the cost-efficiency of the DSN," *SpaceOps 2008*. Heidelberg, Germany, 2008.

[4] Lacey, N., and Morris, D. G. "JPL RAPSO Long-Range Forecasting," *12th AAS/AIAA Space Flight Mechanics Meeting*. Vol. AAS 02-223, San Antonio, Texas, 2002.

[5] Morris, D., and Burke, E. "The JPL Resource Allocation Planning and Scheduling Office (RAPSO) process," *12th AAS/AIAA Space Flight Mechanics Meeting*. Vol. AAS 02-219, AAS, San Antonio, Texas, 2002.

[6] Johnston, M. D., Tran, D., Arroyo, B., Call, J., and Mercado, M. "Request-Driven Schedule Automation for the Deep Space Network," *SpaceOps 2010*. AIAA, Huntsville, AL, 2010.

[7] Johnston, M. D., and Tran, D. "Automated Scheduling for NASA's Deep Space Network," *6th International Workshop on Planning and Scheduling in Space (IWPSS)*. Darmstadt, Germany, 2011.

[8] Smith, J., and Wang, Y. "E-scheduling the Deep Space Network," *Aerospace Conference, 2001, IEEE Proceedings*. Vol. 7, 2001.

[9] Werntz, D., Loyola, S., and Zendejas, S. "FASTER - A tool for DSN forecasting and scheduling," *Proceedings of the 9th AIAA Computing in Aerospace Conference*. San Diego, CA, 1993, pp. 230-235.

[10] Loyola, S. J. "PC4CAST: A Tool for DSN Load Forecasting and Capacity Planning." Jet Propulsion Laboratory, 1993, pp. 170-184.

[11] Johnston, M. D., and Clement, B. J. "Automating Deep Space Network Scheduling and Conflict Resolution," *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. ACM, Hakodate, Japan, 2006.

[12] Clement, B. J., and Johnston, M. D. "Design of a Deep Space Network Scheduling Application," *International Workshop on Planning and Scheduling for Space (IWPSS-06)*. Space Telescope Science Institute, Baltimore, MD, 2006.

[13] Johnston, M. D., Tran, D., Arroyo, B., and Page, C. "Request-Driven Scheduling for NASA's Deep Space Network," *International Workshop on Planning and Scheduling for Space (IWPSS)*. Pasadena, CA, 2009.

[14] Carruth, J., Johnston, M. D., Coffman, A., Wallace, M., Arroyo, B., and Malhotra, S. "A Collaborative Scheduling Environment for NASA's Deep Space Network," *SpaceOps 2010*. AIAA, Huntsville, AL, 2010.

[15] Luong, I., and Bliss, D. "DSN Antenna Arraying; Its Past, Its Contributions to Achieving the Spitzer Space Science Mission Objectives, and Its Future Promise," *SpaceOps 2008*. Heidelberg, Germany, 2008.

[16] Johnston, M. D. "Multi-Objective Scheduling for NASA's Deep Space Network Array," *International Workshop on Planning and Scheduling for Space (IWPSS-06)*. Space Telescope Science Institute, Baltimore, MD, 2006.

[17] Johnston, M. D. "An Evolutionary Algorithm Approach to Multi-Objective Scheduling of Space Network Communications," *International Journal of Intelligent Automation and Soft Computing* Vol. 14, 2008, pp. 367-376.

[18] Johnston, M. D., and Giuliano, M. "MUSE: The Multi-User Scheduling Environment for Multi-Objective Scheduling of Space Science Missions," *IJCAI Workshop on Space Applications of AI*. Pasadena, CA, 2009.

[19] Giuliano, M., and Johnston, M. D. "Multi-Objective Evolutionary Algorithms for Scheduling the James Webb Space Telescope," *International Conference on Automated Planning and Scheduling (ICAPS)*. Sydney, Australia, 2008.

[20] Johnston, M. D., and Giuliano, M. "Multi-Objective Scheduling for the Cluster II Constellation," *6th International Workshop on Planning and Scheduling in Space (IWPSS)*. Darmstadt, Germany, 2011.

[21] SCAWG. "NASA Space Communications and Navigation Architecture Recommendations, Final Report." Space Communications Architecture Working Group (SCAWG), 2006.