SpaceOps 2010 Conference<br> <b><i>Delivering on the Dream</b></i><br> <i>Hosted by NASA Mars
25 - 30 April 2010, Huntsville, Alabama

AIAA 2010-2284

A Collaborative Scheduling Environment for NASA's Deep Space Network

Butch Carruth*1, Mark D. Johnston†2, Adam Coffman‡1, Mike Wallace§1, Belinda Arroyo**2, and Shan Malhotra††2
[1]*Innovative Productivity Solutions, Inc., Bulverde, TX 78163*
[2]*Jet Propulsion Laboratory – California Institute of Technology, Pasadena CA 91109*

A number of factors are driving the improvement of NASA's communications network scheduling, including reducing cost, scaling to support a larger number of increasingly complex missions, and maintaining round the clock availability and reliability. This paper describes the Deep Space Network (DSN) Service Scheduling Software (S^3), an innovative design to provide a unified scheduling framework for DSN scheduling services. Among the challenges addressed by the S^3 design are user-managed *workspace schedules* for scenario and contingency schedule development, automated *change proposals* for mutual peer-to-peer concurrence of schedule changes, and tight integration with a high performance *scheduling engine* that checks for conflicts and scheduling requirement violations, and performs automated conflict resolution. S^3 is architected as a web application that accesses a database and that can be run by any DSN scheduling user via a standard web browser. It makes extensive use of modern web technologies to provide a rich client user experience, comparable to popular social networking applications. Among the features provided by the web framework are the online status of other users, instant unobtrusive notification of events of interest, peer-to-peer and multi-user chat, document sharing, and integration with a wiki for a persistent running record of scheduling-related textual information. This approach is appropriate for DSN scheduling due to its highly collaborative nature: the DSN schedule is developed not by a central scheduling authority, but by individual DSN users working together to fit requests into the schedule, making tradeoffs and compromises where appropriate. S^3 is scheduled for operational deployment in 2010. This paper presents an overview of the innovative features of the S^3 design and implementation, and describes its usage as the next generation DSN scheduling system.

I. Introduction

NASA's Deep Space Network (DSN) is operated by the Jet Propulsion Laboratory in support of a wide range of space missions and science users, both domestic and international. The DSN comprises three complexes of antennas, in Goldstone, CA, Madrid, Spain, and Canberra, Australia. Each complex houses one 70m and multiple 34m antennas, with electronics that support a range of S, X, and K band capabilities[1, 2]. About 35 distinct users share access to the DSN and participate in its scheduling process. Missions that use the DSN as their primary means of communications include all the planetary exploration missions, as well as a number of spacecraft in high Earth or other orbits.

The DSN is developing a new scheduling system, called the *Service Scheduling Software*, or S^3. The use of the term "service" in the name is an indication of a shift from the allocation of specific assets to the provision of a more abstractly specified set of services such as telemetry, tracking, commanding, radio science, and various others. Flexibility in how the service is ultimately implemented by the DSN as the service provider can be used to make the network more efficient and more robust, while simplifying the end user's interaction with the DSN. Although not a

* President/Architect, IPS Inc.
† Principal Scientist, Artificial Intelligence Group, Planning and Execution Systems Section.
‡ Senior Programmer, IPS Inc.
§ Programmer, IPS Inc.
** Group Supervisor, Customer Integration Services Group, System Verification, Validation and Operations Section.
†† DSN System Engineer, DSN Planning and Execution Software Systems, Planning and Execution Systems Section.

full service management[3] implementation, S^3 provides a significant step in that direction at the level of DSN scheduling.

S^3 has been conceived, designed, and developed to provide a collaborative infrastructure and set of integrated capabilities for DSN scheduling. The explosive growth of web technologies has enabled a system design that not only provides the necessary algorithmic functionality, but also enables a high degree of person-to-person interaction, as exemplified by wikis and modern social networking sites. In this paper we describe the overall scheduling concept for DSN (Section II), followed by the S^3 architecture and technology elements (Section III) that make up the S^3 collaborative environment (Section IV). We follow this with an example scenario for reaching mutual agreement on a conflicted portion of the schedule (Section V), and conclude with a brief summary (Section VI).

II. Deep Space Network Scheduling

Scheduling the DSN is not a centralized function, but instead involves the active participation of missions or other users along with their scheduling representatives. In many cases, the details of a mission's scheduled DSN allocations are intimately linked to the mission's science data return, and so resolution of scheduling resource contention can require both scheduling and mission expertise and sometimes scientific judgment as to acceptable compromises. Recognizing this as a key capability to be supported by the scheduling system, S^3 provides a range of features that together provide a *collaborative environment* to facilitate the definition, elaboration, and negotiation of the DSN schedule. In addition, since changes to the schedule can occur for many reasons all the way up to the last minute, S^3 provides features for schedule change management, including a current baseline schedule at all times, and online and offline change concurrence by affected users. S^3 provides a high level of automation, but in a decision support context, not as a "black box". The software technology and associated components behind these features are described in Section III. The remainder of this section contains a description of the major elements of the DSN scheduling operations concept around which S^3 has been implemented, including definitions of the key terms.

Figure 1 illustrates the main stages in the end-to-end DSN schedule development process (Note that many details

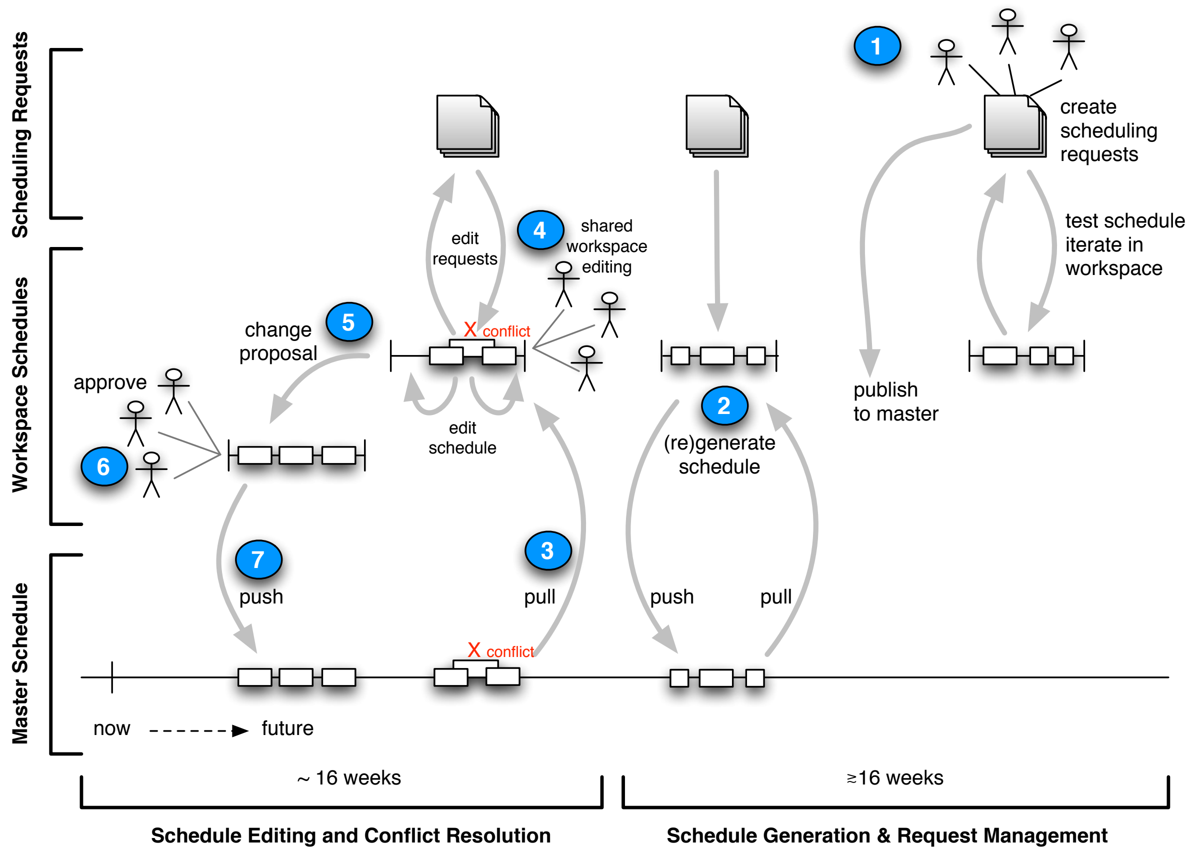


Figure 1. Major stages in the DSN schedule development process.

of the new DSN scheduling process using S^3 remain to be defined, so the time frames and examples used in this section are notional). Time runs from now (real-time) to the future, left-to-right in the diagram. Vertically, the divisions indicate operations primarily related to the master schedule (bottom), workspace schedules (middle), and scheduling requests (top). Generically, S^3 users are denoted as "service users" in the following: they include not only the missions, but also science users such as radio science and radar, along with maintenance, engineering, and calibration users.

- The **master schedule** is the current baseline DSN schedule, including scheduling requests and events, over a conceptually unbounded time frame; in practice some portions of the schedule could be very firm from months to over a year in advance (e.g. scheduling for a planetary encounter), while the remainder could be much less well-defined until 8 to 12 weeks from execution, at which point many missions need specific allocated tracks in order to develop their command sequences for upload.
- A **workspace** schedule (or **private workspace**) is a time-bounded schedule, including scheduling requests and events, created by an S^3 service user for one of several purposes. It could be a subset of the master schedule over some time period. It could be an initially empty schedule, to be used for what-if analysis. Or it could represent a contingency schedule, such as a variation of the master schedule that might be activated if a scheduled launch is delayed. Workspaces are owned by one service user, but can be shared with others, and support various access permissions. Changes to the master schedule are permitted *only when pushing changes from a workspace*, thus ensuring transactional integrity and traceability. Should a workspace become out of date with respect to the master schedule, the user can *pull changes from the master* to synchronize the workspace.
- **Scheduling requests** are specifications created by service users of their tracking requirements, including constraints and flexibilities. For example, a request could represent a need for three 8±1 hour tracking passes on a 70m antenna over the course of a week, with a minimum separation between each of 24 hours and a maximum of 60 hours. One request can generate multiple scheduled tracks, and S^3 keeps track of the relationship between tracks and requests so that it can continuously check whether the tracks satisfy the request specifications.

The major phases of the S^3 scheduling concept (Figure 1) are as follows:

1) **Create scheduling requests.** The overall workflow starts with service users defining their scheduling requests in their own workspaces. Users can invoke the scheduling engine component of S^3 to expand requests into tracks, iterating on the request specification details until they confirm the intended characteristics of the resulting tracks. At any point, users can push their requests and corresponding tracks to the master schedule. In this timeframe, it is expected that potential contentious areas will become apparent.
2) **(Re-)generate schedules.** By some months ahead of execution, requests associated with the master schedule go through an automatic regeneration phase. A system process will regularly pull a portion of the master schedule (requests and tracks) into a workspace schedule and will then invoke the scheduling engine to lay out tracks attempting to meet the request specifications and simultaneously to resolve conflicts to the maximum extent possible. The resulting schedule is pushed back to the master, for inspection by users.
3) **Create workspace for conflict resolution.** Following the automatic generation phase, users can create workspaces to work on specific areas of contention. For example, it may be that some combination of scheduling requests leads to an oversubscription of DSN resources in some time period, which cannot be resolved within the stated flexibilities described in the scheduling requests. If a problem can be resolved by a unilateral change that an authorized user is willing to make, they can simply make the change and push the workspace to baseline the change to the master schedule. If multilateral agreements must be reached, then the workspace may be shared among an arbitrary set of other users to work on a mutually acceptable resolution.
4) **Edit shared workspace.** This is the heart of the collaborative environment provided by S^3. In this phase, users can make use of a range of S^3 features that enable and encourage rapid communication, reaching and documenting decisions, constructing what-if scenarios, analyzing alternatives, and finally the arrival at a mutually acceptable solution. These features include: online indicators of user status (available, busy, etc.); single and multi-user chat sessions; sending links to workspaces or to ancillary files via chat; capturing chat transcripts to wiki pages which can be subsequently edited and commented upon; notification of shared workspaces; copying workspaces to investigate alternatives; editing scheduling requests and tracks; and the use of the S^3 scheduling engine to automatically search for conflict resolutions. These features are described in Section III.
5) **Create change proposal.** Once a mutually acceptable solution has been reached, S^3 provides an automated workflow for a "change proposal" to obtain signoff from affected users that they agree to a specific update to the schedule. A change proposal is a special state of a workspace schedule that permits no changes and collects concurrences (or rejections) from authorized service users. S^3 provides notifications to all affected

users that a change proposal is awaiting their action, and to the originator as signoffs take place. Any number of change proposals may be active in the system, involving any set of users, for any timeframe.

6) **Approve change proposal.** Authorized service users can accept or reject change proposals. If they wish, they can create counter proposals as copies of the original change proposal workspace state, then modify them to reflect any desired changes. S^3 keeps continuous track of the approval status of all change proposals. The originator can withdraw obsolete proposals.
7) **Push approved changes to master schedule.** Once all approvals have been recorded, S^3 will either automatically push the changes to the master if the change proposal was configured as "auto push", or will permit the originator to push the changes on behalf of all the affected users (all of whom must have concurred to the changes).

III. S^3 Architecture and Technology

S^3 is architected utilizing a Model-View-Controller (MVC) design pattern (Figure 2). The MVC design pattern allows S^3 to cleanly separate business logic (model), data access code (model – resource layer), the presentation layer (view), and the actions binding all layers (controller). By utilizing the MVC design pattern, S^3 loosely couples Java objects allowing S^3 developers to easily extend the application by taking advantage of object reuse. This also makes S^3 easier to maintain, as there are fewer interdependencies among components.

A. S^3 Technologies Stack

The following is a summary of the core technologies used to build the S^3 web-based solution:

- **Databases:** S^3 utilizes Sun One LDAP for authentication, Oracle 10G for its central repository, and the Atlassian Confluence wiki.
- **Adobe BlazeDS:** BlazeDS (data services) provides server-based Java remoting and web messaging technology that enables developers to easily connect to back-end distributed data sources, and push data in real-time to Adobe Flex and Adobe AIR applications for more responsive user interactions. S^3 utilizes BlazeDS to distribute data to its Flex schedule visualization component.
- **Java:** Building on portability, scalability, and open standards, S^3 was developed using Java while applying J2EE Best Practices.
- **Dojo**: The Dojo Toolkit is a modular, Open Source JavaScript toolkit designed to aid in the development of cross-browser web applications. The toolkit provides abstraction layers for AJAX communication and for graphical user interface (GUI) components.
- **Flash Player:** The Macromedia Flash Player is a multimedia and application player used as a web browser plugin. Flash Player executes an SWF (a vector graphics file format) for schedule visualization and direct editing of DSN tracks. The custom-developed Gantt viewer was created using Adobe Flex technology.
- **XML:** XML provides the ability to describe data in an easy-to-process industry standard format. XML is the primary language for S^3 internal and external interfaces.
- **XMPP**: XMPP (Extensible Messaging and Presence Protocol) is an instant messaging protocol based on open standards. S^3 is using XMPP for peer-to-peer and group chat functionality. OpenFire was chosen as the XMPP server for S^3.

B. Communications Protocols Used in S^3

S^3 utilizes existing standards to interface the different elements of the system. Each layer of the MVC pattern (Figure 2) communicates using different mechanisms and protocols. These interfaces include:

- **HTTP Request and Response**: Within a browser-based application, the basis for communication between the user (browser) and the server is through iterations of HTTP requests from browser to server and HTTP responses from server to browser.
- **XML HTTP Request and Response**: The XML HTTP request and response is analogous to the standard HTTP request and response. The difference between the two is that JavaScript invokes the XML HTTP Request in the background of the browser to perform asynchronous communication with the server. This technique is known as AJAX.

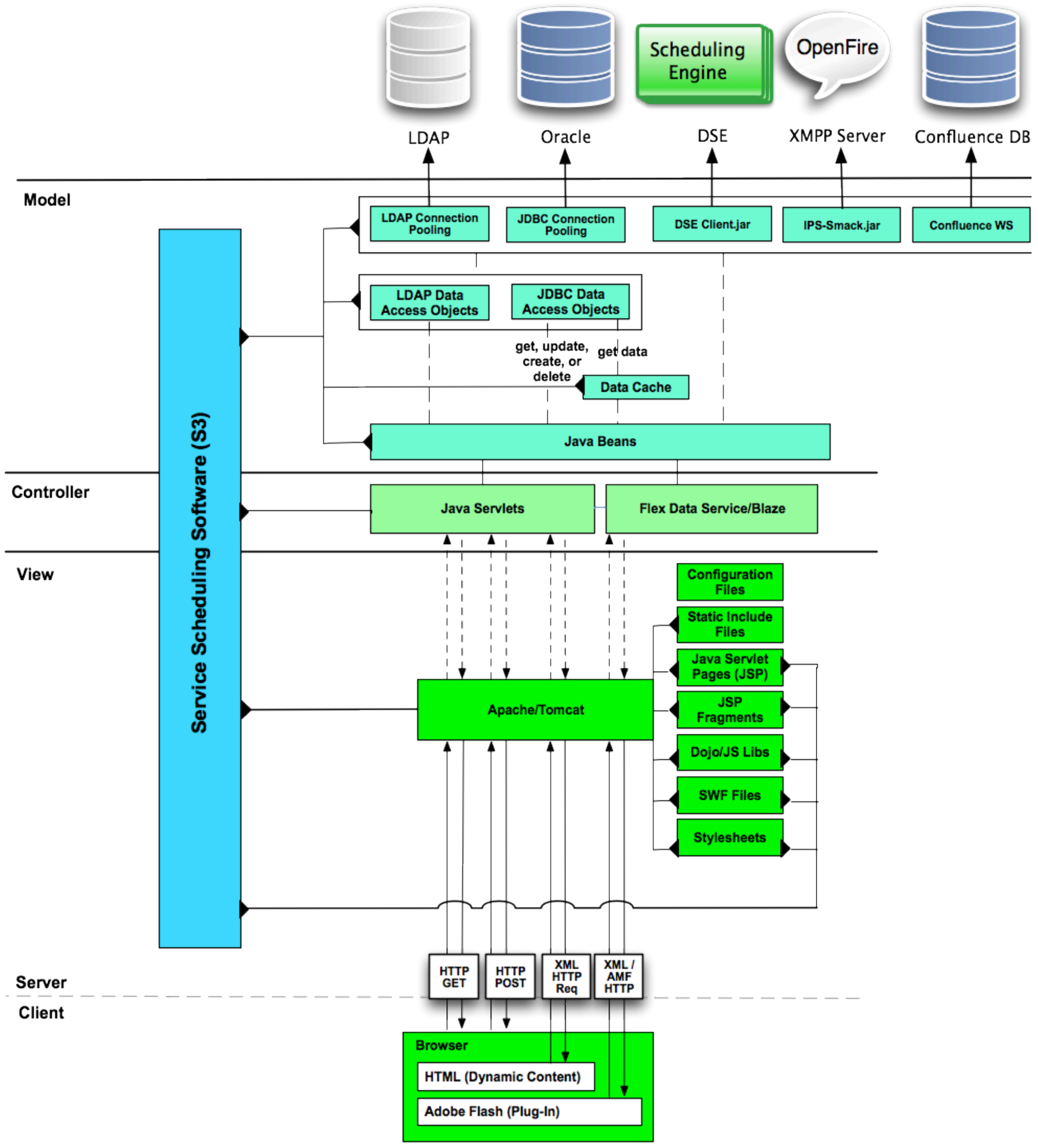


Figure 2. S^3 architectural layer view.

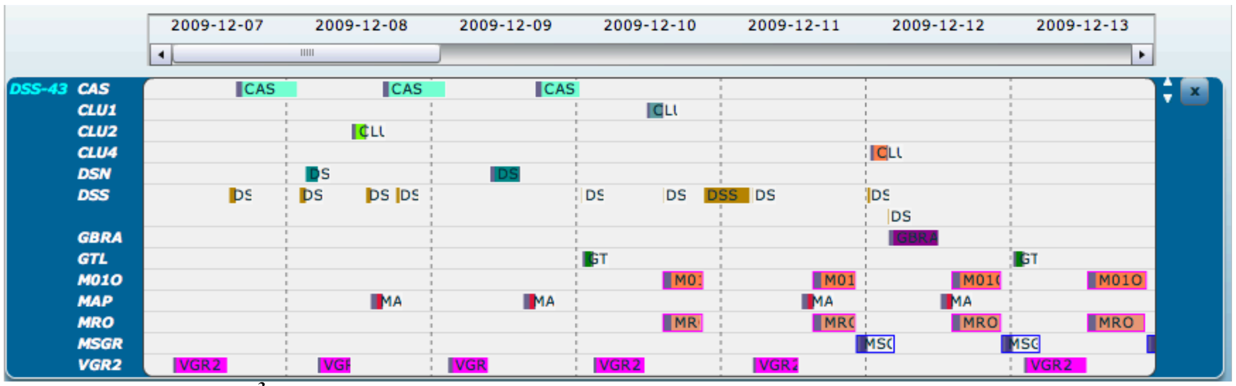


Figure 3. Example of S^3 schedule visualization via Flash in a web browser.

- **Java Method Invocation**: The S^3 system incorporates a number of Java class libraries that communicate and pass data from class to class.
- **Java Database Connectivity (JDBC)**: JDBC is an API (application programming interface) for Java that specifies the methods for access to a database. S^3 uses the vendor-supplied Oracle JDBC driver to create connections and communicate to the Oracle database over TCP/IP.
- **Comet:** Comet is a web application model that allows a web server to push data to a browser without the browser specifically requesting it. Comet encompasses all techniques that allow this type of client-server interaction. Two Comet techniques used by S^3 are HTTP Polling and BOSH (Bidirectional streams Over Synchronous HTTP).
- **HTTP Polling:** HTTP polling is a Comet technique based on the client querying the server at regular intervals regardless of whether the client has information to transmit to the server. Data made available on the server is queued until the client polls the server. When polled, the server will send all queued data to the client. This technique creates a trade-off between bandwidth used and responsiveness. A short polling interval increases bandwidth consumed, but provides greater responsiveness. A long polling interval decreases bandwidth consumed, but the client is not as responsive.
- **BOSH:** BOSH is another Comet technique that emulates a bidirectional data stream using the stateless HTTP protocol. By definition, HTTP requests and responses are not long-lived. In the BOSH paradigm, a connection from the client to the server always exists. When the server has data to send to the client, that data may be transmitted immediately. On receipt of data, the client immediately opens a new HTTP connection to the server to be ready for the next server transmission. Not only is responsiveness high due to the server being able to send data immediately, but also network bandwidth is conserved because the client creates new HTTP connections only when necessary.
- **AMF:** Adobe ActionScript Messaging is a binary message format used to efficiently serialize ActionScript objects. The AMF channel allows the Flash Player to asynchronously send and receive AMF-formatted messages over HTTP to and from BlazeDS.

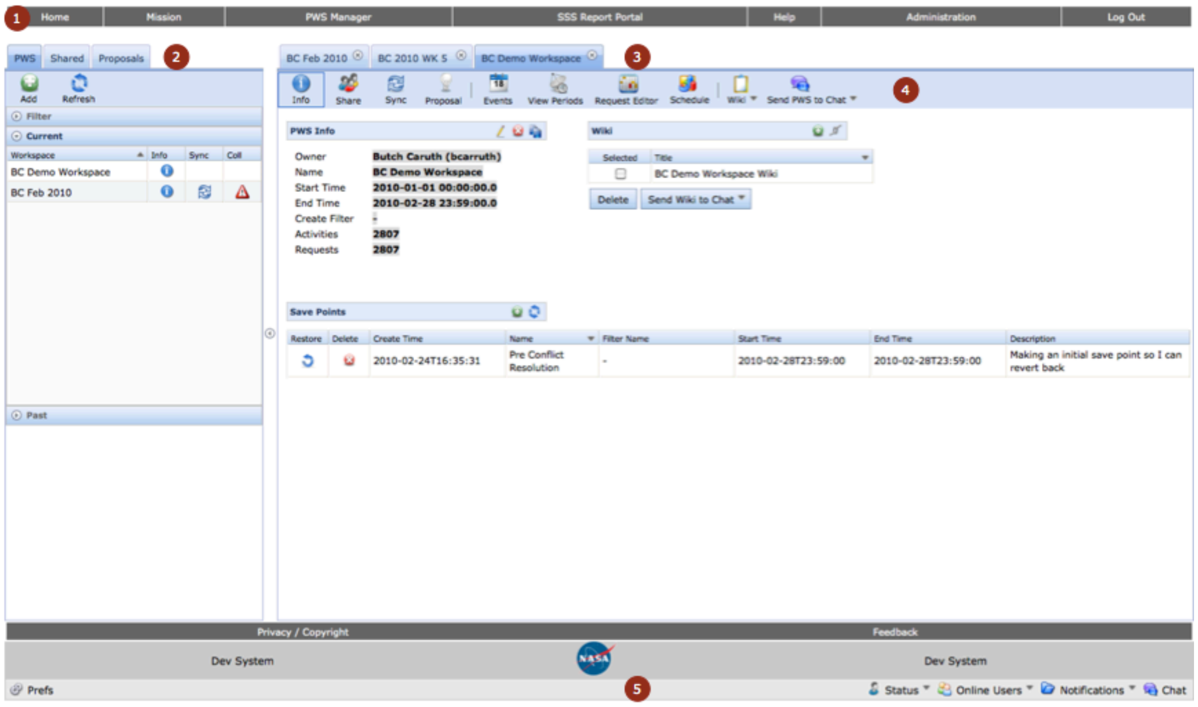
C. Implementing the User Experience

The display and authorized manipulation of scheduling data using a web browser interface can be problematic, but technology exists today that makes it possible to provide not only a secure data transfer, but also a satisfying user experience. HTML has come a long way and been enhanced with JavaScript, CSS, XML processing, and asynchronous communications (AJAX) to address many user-experienced difficulties. Powerful web-based applications are being written with this combination of technologies. The Dojo Toolkit supplements standard HTML, CSS, and JavaScript with a wide array of graphical elements and layouts that allow developers to create more sophisticated user interfaces. Dojo also provides high level interfaces for event handling and AJAX communication. In spite of the capabilities provided by Dojo, there are still situations when a bitmapped graphical environment is necessary for display or for the level of control needed in a scheduling application. Consequently, a satisfactory web-based scheduling interface must rely on plug-ins to the browser. The Flash plug-in provides both secure communication and an extensive graphical library of functions, and is widely available for most browsers and platforms. Dojo and Flash work well together in S^3: Dojo handles application data input and tabular display, while Flash is used for schedule data manipulation and visualization, all within the same user context. See Figure 3 for an example of S^3 schedule visualization.

IV. S^3 Collaborative Environment

A. S^3 Workspace Manager

S^3 cleanly encapsulates workspace management, schedule visualization and editing, scheduling engine event and request management, wiki integration, peer-to-peer and group chatting, file sharing, and tight integration with a high performance scheduling engine, all within a single rich internet application. Figure 4 illustrates the workspace manager user interface, which includes the following features:



Workspace	Info	Sync	Coll
BC Demo Workspace	●		
BC Feb 2010	●	●	⚠

Owner	Butch Caruth (bcarruth)
Name	BC Demo Workspace
Start Time	2010-01-01 00:00:00.0
End Time	2010-02-28 23:59:00.0
Create Filter	-
Activities	2807
Requests	2807

Selected	Title
☐	BC Demo Workspace Wiki

Restore	Delete	Create Time	Name	Filter Name	Start Time	End Time	Description
●	●	2010-02-24T16:35:31	Pre Conflict Resolution	-	2010-02-28T23:59:00	2010-02-28T23:59:00	Making an initial save point so I can revert back

Figure 4. Workspace Manager User Interface

1) **Primary Menu**. The primary menu allows authorized users to navigate to all areas of S^3 including their home page, mission preferences, Workspace Manager, the reporting portal, help, and administration.
2) **Workspace Explorer**. The Workspace Explorer organizes all workspaces associated with the user. Associations may include owned workspaces, workspaces that are shared with the user, and proposals affecting the user. The Workspace Explorer menu bar allows users to add new workspaces or refresh the listing. Users may filter the workspace list and sort on any columns in the listing. The workspace listing also includes visual indicators to alert the user in the event the workspace is out of sync (different from the master baseline schedule) or has activity or request collisions (i.e. divergent changes) with the master schedule. Each tab (PWS, Shared, and Proposals) offers context sensitive columns. For instance, the Proposal tab displays and allows sorting based on proposal due dates. The separator bar between the Workspace Explorer and Workspace Manager allows the user to shrink or fully collapse the area to maximize limited browser view area.
3) **Workspace Manager**. The Workspace Manager organizes all aspects of a workspace. Users may open multiple workspaces concurrently. A tabbed interface provides quick access to all open workspaces.
4) **Workspace Toolbar**. The toolbar is responsible for organizing the functions that may be performed on a workspace. The Workspace Toolbar is organized into three sections: workspace, scheduling engine, and collaboration. The toolbar contains buttons for the following functions:
   a. **Workspace Related Functions**
      i. **Info**. This information area provides statistics for the workspace, shows all wiki associations, and allows the user to manage save points.
      ii. **Share**. The share feature allows users to share their workspaces with other users or groups. The owner may grant three permission levels to other users: read, write, or publish.
      iii. **Sync**. The synchronization tool analyzes and displays differences between the workspace and the master schedule. This tool also takes into account user authorizations, and it groups activities or requests in order to show which data the current user is authorized to push to the master schedule.
      iv. **Proposal.** The proposal tool converts a workspace into a change proposal. Proposals are used to gain acceptance of a proposed schedule change affecting multiple missions, where the owner does not have the authority to push all affected activities or requests to the master schedule (see Section V for an example change proposal scenario).
   b. **Scheduling Engine Data**
      i. **Events.** This provides the capability to view and manage the various constraining events (collections of time intervals) used in scheduling request definitions.

ii. **View Periods.** This provides visualization of available view periods and utilities to convert view periods to scheduling engine events.

iii. **Request.** This allows the user to define and manage scheduling engine requests and requirements.

iv. **Schedule.** This provides visualization and direct editing of the workspace activities (schedule) via a custom Gantt chart viewer developed using Adobe Flex.

v. **Strategies.** This toolbar button allows users to execute user-defined scheduling engine strategies against the workspace to help automatically resolve schedule violations and conflicts.

c. **Collaboration**

i. **Wiki.** This allows users to share a wiki page associated with the workspace with an active peer-to-peer or group chat.

ii. **Send Workspace to Chat.** This allows users to share the workspace with an active peer-to-peer or group chat.

5) **S^3 Toolbar.** The S3 toolbar is docked to the bottom of every page and provides easy access to user preferences, a button to set current status, a list of other online users and their statuses, recent notifications with a link to all notifications, and a button to launch the S^3 chat application.

B. The Workspace

S^3 is designed around the concept of a single master (baseline) schedule and an arbitrary number of workspace schedules covering user-specified time ranges. Workspace schedules can be used for "what if" scheduling, developing alternative contingency schedules (e.g. for different launch windows), and for resolving conflicts in the master schedule. S^3 users can define workspaces either as empty, or can immediately pull in the current scheduled activities and requests from the master. Optionally, a user-defined filter may be applied to select a subset of the full set of missions. S^3 will monitor differences between the workspace and the master, and provides tools such as the synchronization tool mentioned above for reporting differences. Visual alerts in the S^3 Workspace Explorer provide a quick indication if a workspace is out of synchronization with the master schedule, allowing the workspace owner to take the appropriate action. The synchronize actions are defined as *push* and *pull*:

- **Push** allows users to publish modified (new, changed, deleted) activities and requests to the master schedule based on their authorizations on behalf of one or more missions. The synchronization tool organizes changes (additions, updates, and deletions) to the workspace into two groups, authorized and not authorized, and only allows the user to push their authorized changes to the master (see discussion of permissions below).
- **Pull** brings updated activities and requests from the master schedule into the workspace.

The synchronization process ensures that users are working on the latest master schedule data prior to allowing a push event from the workspace into the master. This is accomplished by maintaining a reference to the revision of the activity or request the workspace pulled from the master. If the master revision reference is earlier in the workspace than is currently contained in the master schedule, a collision is reported within the workspace, both with visual indicators in the Workspace Explorer and within the synchronization tool. In the case of collisions, users determine whether the appropriate action is to *override and push* to update the master schedule, or to *override and pull* to update the workspace with the current data from the master schedule.

S^3 allows the creation of workspace *save points* at any time. Save points capture the state of the workspace including its current definition (time boundaries and applied filter) and the current revisions of the schedule data. A workspace might be in a desired state, but a user may want to explore other schedule possibilities by changing certain scheduling request parameters or executing different strategies against the workspace. If a save point is created, the user may freely edit the scheduling requests, look at the results, and then decide if the new results are better than before. If the older results prove to be better, all changes can be reverted by restoring the save point. User-created save points are easily created, restored, or deleted. In addition, S^3 will automatically create save points at certain critical events, such as during the conversion to a change proposal. These system-generated save points cannot be deleted.

S^3 supports a full range of permissions, so only authorized users can make changes to their own requests or activities, and can share their workspace schedules with other users for simultaneous display and modification. S^3 leverages existing authorization tables managed by the DSN Service Preparation System (SPS). These permissions determine which mission's activities and requests may be pushed to the master schedule by a particular user. One important aspect of S^3 is that a user may change *any* other mission's data within their workspace in order to find a suggested scheduling problem solution. However, S^3 will prevent a user from pushing any changes into the master schedule for missions for which they do not have change authority. S^3 provides a change proposal feature in the event the desired solution involves multiple missions where the workspace owner does not have permission to push all of the changes to the master schedule. This feature will be covered in more detail below.

S^3 allows for distributed permissions as well. Any workspace owner can grant read, write, and/or publish permission to other users or user-defined groups of users. This allows for full collaboration on any workspace. By granting read permission, the owner is allowing other users to view the workspace. Read permission allows for viewing all workspace statistics, save point history, all associated activities and requests, and for visualization of the workspace. Write permission includes read functionality and adds the capability to make changes to the workspace data as well as executing functions such as creating new save points. Write permission does not authorize other users to change the workspace definition, such as time boundaries or applied filters. Publish permission allows the workspace owner to distribute the ability to push the workspace to the master schedule. In the event the workspace owner is authorized for missions A, B, and C and the user to which publish permission was granted is authorized for missions D and E, then the user who was granted publish permission is now authorized to push for the union of the missions or A, B, C, D, and E.

All changes throughout S^3 are versioned for auditing and recovery.

C. S^3 Collaboration

One of the most important uses of workspace schedules is for peer-to-peer schedule conflict resolution: when resources are oversubscribed in some time frame, affected users can work together to come up with a compromise workspace schedule to resolve the problem, then submit the result as a change proposal for electronic concurrence. When all affected users have concurred, changes are automatically merged into the master schedule. Of course, concurrence is not required, and S^3 fully supports alternative scenarios including proposal rejection as well as multiple levels of counter proposals.

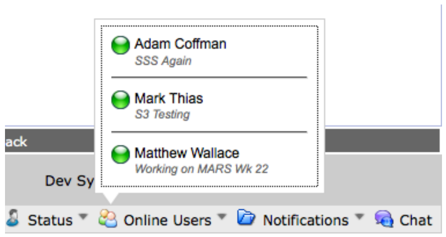
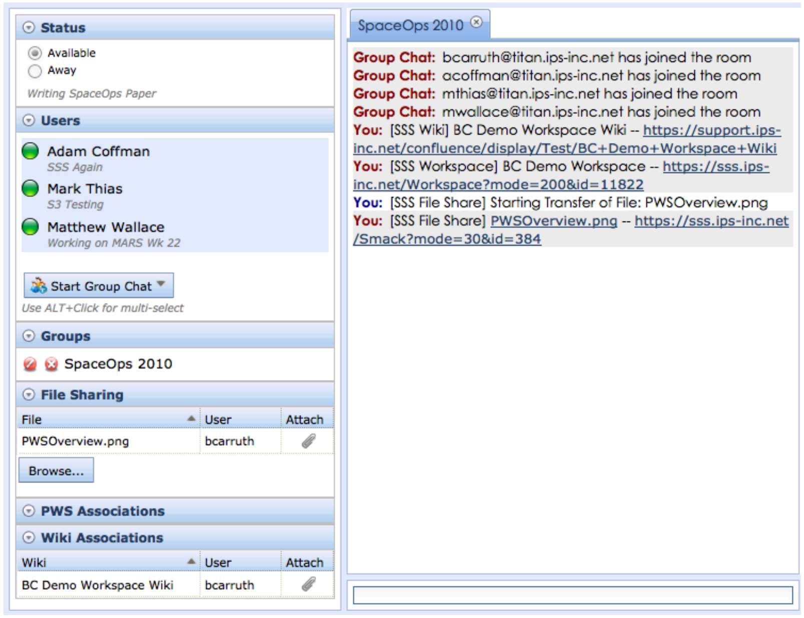


Figure 5. Example of online status.

S^3 offers a well-integrated mash up of social networking, wiki, notification, and chat features to facilitate a collaborative negotiation process. Upon login to S^3 a user may set their current status, making it available to all users via the S^3 toolbar and the chat feature. The status may be used to inform others as to which workspace schedule they are currently working on, or simply to say, "I'm not available right now." See Figure 5 for an example of current user status display.

A chat interface was specifically designed for S^3 on top of the XMPP/Jabber protocol. The chat feature is launched via the S^3 toolbar where it pops out in a separate browser window. Users are free to create peer-to-peer or group chats in order to collaborate on their workspaces, share files, or just catch up on current events. Chat features unique to S^3 include the ability for users to share their workspaces and wiki associations directly from the Workspace Manager. If a workspace is shared in a chat session, all participants are automatically granted read access to the workspace. This saves the owner from having to manually grant the read privilege. The workspaces are listed in a PWS (Workspace) Association area where participants can click the link and the workspace will be opened in their Workspace Manager as a new tab. Other special features are enabled upon workspace wiki association with a chat session. Once the wiki association has been made, chat participants may click an icon to automatically capture the chat transcript to the Wiki or another icon to attach any shared files to the wiki. In the event a negotiation process has taken place in an S^3 chat session, capturing the chat transcript in the wiki for future reference is

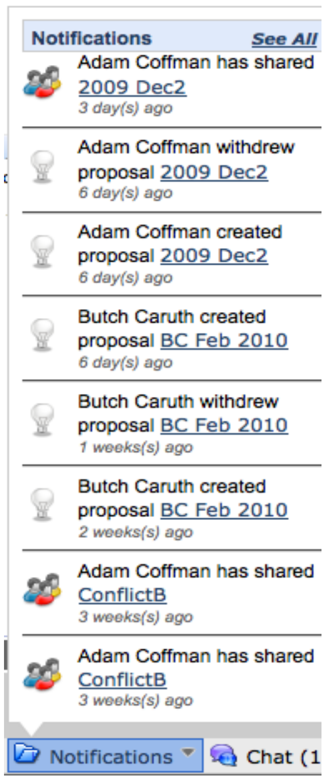


File	User	Attach
PWSOverview.png	bcarruth	

Wiki	User	Attach
BC Demo Workspace Wiki	bcarruth	

```
Group Chat:  bcarruth@titan.ips-inc.net has joined the room
Group Chat:  acoffman@titan.ips-inc.net has joined the room
Group Chat:  mthias@titan.ips-inc.net has joined the room
Group Chat:  mwallace@titan.ips-inc.net has joined the room
You:  [SSS Wiki] BC Demo Workspace Wiki -- https://support.ips-
inc.net/confluence/display/Test/BC+Demo+Workspace+Wiki
You:  [SSS Workspace] BC Demo Workspace -- https://sss.ips-
inc.net/Workspace?mode=200&id=11822
You:  [SSS File Share] Starting Transfer of File: PWSOverview.png
You:  [SSS File Share] PWSOverview.png -- https://sss.ips-inc.net
/Smack?mode=30&id=384
```

Figure 6: Multi-user chat illustrating workspace sharing.



Adam Coffman has shared 2009 Dec2
3 day(s) ago

Adam Coffman withdrew proposal 2009 Dec2
6 day(s) ago

Adam Coffman created proposal 2009 Dec2
6 day(s) ago

Butch Caruth created proposal BC Feb 2010
6 day(s) ago

Butch Caruth withdrew proposal BC Feb 2010
1 weeks(s) ago

Butch Caruth created proposal BC Feb 2010
2 weeks(s) ago

Adam Coffman has shared ConflictB
3 weeks(s) ago

Adam Coffman has shared ConflictB
3 weeks(s) ago

Figure 7. S3 Notification

only a single step. See Figure 6 for an example illustrating the S^3 chat interface.

As described above, a workspace owner may share their workspace with other users by manually granting permissions to users or groups, or by sharing the workspace in a chat session. In either case, the S^3 notification framework will notify all users with whom the workspace was shared. The notification feature has three levels: first, a "toaster" notification appears in the bottom right corner of the browser. Toaster notifications are messages which pop-up then gradually fade away after five to ten seconds. The second method for alerting is to increase the notification count in the S^3 toolbar. If the user is not around or watching their browser when the toaster notification appears, they can click the notifications icon on the S^3 toolbar to view their eight most recent notifications. Finally the "See All" link may be clicked to view all past S^3 generated notifications. The two types of notifications visible in Figure 7 are workspace sharing and change proposal notifications.

Consider a scenario in which a user has created a workspace, modified scheduling requests which generated new activities, executed scheduling engine strategies to check for violations and resolve conflicts, documented the solution in the wiki, shared the workspace with their group, discussed the workspace via chat, and captured that transcript into the associated wiki page. All users who collaborated on the workspace now agree on the acceptable scheduling solution. The owner needs to push the results to the master schedule. However, if they are not authorized to push all affected missions, then the workspace needs to be converted into a *change proposal*. The owner defines the due date and description, and optionally decides if the activities and requests should be pushed to the master schedule automatically upon concurrence by all affected missions. The workspace is now frozen so no additional changes may be made. Authorized schedulers for the affected missions will be notified that they have a new pending proposal. These schedulers may view the schedule and related data and decide if the change is acceptable. If so, the scheduler would click the "approve" icon. Notification is again delivered to all affected schedulers that a particular mission has approved the proposal. The updated activities and requests may be pushed to the master schedule, either automatically or manually, once all missions have concurred. The master is now up-to-date and the process may begin again for the next schedule or conflict. In the event a user rejects the proposal, notification is delivered to all participating users. The proposal could be withdrawn and the workspace could continue to be used to find an acceptable solution. S^3 also allows users with read permission on the workspace to create a copy of the workspace in order to work the solution concurrently and optionally turn their workspace into a counter proposal.

D. DSN Scheduling Engine

As noted above, S^3 users define scheduling requests to convey their DSN service requirements, and it is the DSN Scheduling Engine (DSE) component of S^3 that is responsible for interpreting these requests, expanding them into tracks, checking for satisfaction of the request specifications, and checking for conflicts in the schedule. The DSE also incorporates algorithms, available to users as strategies, that can repair violations, resolve conflicts, optimize the schedule, and provide details about constraint time intervals. The DSE is tightly integrated into the S^3 architecture, e.g. changes to requests or tracks are dynamically checked for violation or conflict and the results are recorded in the S^3 database. The DSE is described in more detail in another paper at this conference[4].

V. Schedule Change Proposal as an Example of S^3 Collaboration

Figure 8 illustrates one possible example of how S^3 features could be utilized for conflict resolution. In this example consider a DSN resource such as a 70m antenna as oversubscribed, with conflict between one of the Mars missions and the Cassini orbiter at Saturn. The workflow would be as follows:

1) **Set SSS Preferences**. S^3 allows for users to customize multiple preferences for use throughout the system. Two examples include Mission Filters and User Groups. Users may create named filters consisting of one or more missions (DSN service users). A DSN scheduler may be responsible for scheduling Mars related missions and thus may create a 'Mars' filter including Mars Reconnaissance Orbiter, the two Mars Exploration Rovers, Mars Odyssey, and Mars Express. This 'Mars' filter could be used when defining a new workspace. The same scheduler may collaborate with a common set of other schedulers on a regular basis. For this reason, an S^3 user may create named groups of other S^3 users. These groups may be used when sharing workspaces with other users with whom they frequently communicate, or for initiating multi-user chat sessions.
2) **Create a Workspace.** An S^3 user creates a workspace to resolve the conflict. The workspace time boundaries would encompass the conflict and include any time necessary to work out an acceptable solution. In this scenario the Mars scheduler defines a new workspace for a one-week period and populates the workspace (pull from master) with the current DSN tracks and associated scheduling requests from the master schedule.
3) **Create or Associate a Wiki (optional).** The user would like to document the conflict to facilitate the upcoming collaborative efforts, to capture any negotiations, and for archival purposes. S^3 provides an "add Wiki" button that creates a new Wiki page based on a negotiation template, and then associates the Wiki with the workspace. The user edits the Wiki page, describing the conflict directly in S^3. A user may also associate existing Wiki pages with the workspace, e.g., common procedures or notes, for easy reference.
4) **Share the Workspace (optional).** The user now has the workspace created and a Wiki page describing the conflict and would like to share it with their team or other users. In this scenario the user shares the workspace with the custom group created in Step 1, including the Cassini schedulers. All users in the group are notified and the workspace appears in their Shared tab in the Workspace Explorer.
5) **Collaborate with other Schedulers.** In this step the conflicted missions and other users work together to jointly develop a solution to the problem. Users can exploit all the S^3 collaborative features and even external tools such as e-mail or teleconferences to discuss the changes with any affected missions. The negotiation process may be captured in the Wiki page created in Step 3. If the user takes advantage of the S^3 chat feature, any negotiation decisions captured in chat may be added to the Wiki page as a comment with a single click of an icon. The workspace eventually reaches a state where the conflicts are resolved, but the workspace owner

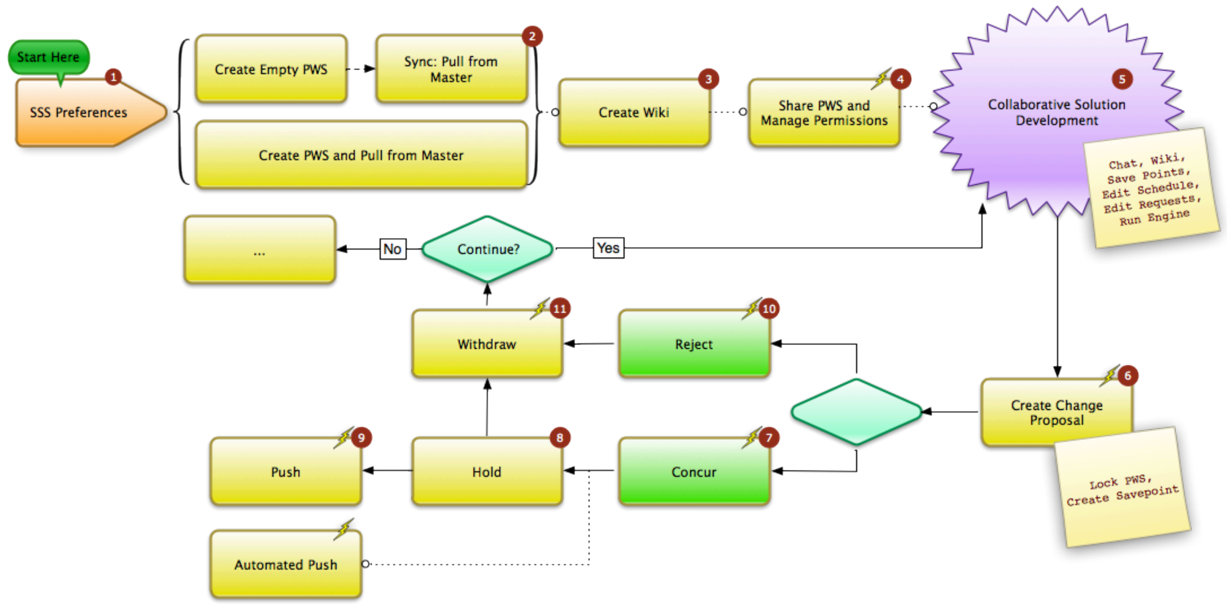

Start Here
1
SSS Preferences
Create Empty PWS
Sync: Pull from Master
2
Create PWS and Pull from Master
Create Wiki
3
Share PWS and Manage Permissions
4
Collaborative Solution Development
5
Chat, Wiki, Save Points, Edit Schedule, Edit Requests, Run Engine
...
No
Continue?
Yes
Withdraw
11
Reject
10
Create Change Proposal
6
Lock PWS, Create Savepoint
Push
9
Hold
8
Concur
7
Automated Push

Figure 8. A change proposal scenario.

only has the authority to push tracks and requests for Mars missions to the master schedule, and we suppose that the resolution involved a change to a Cassini track.

6) **Create a Change Proposal.** The owning user converts the workspace into a change proposal. S^3 also captures a snapshot of the current state of the workspace as a save point. In this scenario the workspace owner can approve any changes on behalf of the Mars missions, but not those for Cassini. The Cassini schedulers are notified and the workspace is automatically shared with them, if it hadn't been already. The Cassini schedulers may review and either accept or reject the proposal.
7) **Nominal Workflow – Concur.** If the users took advantage of the S^3 collaboration features and/or used existing processes to notify the Cassini mission schedulers about the proposed solution, then the nominal workflow would be for the Cassini mission to approve the proposal, simply by clicking an icon on the Workspace Proposal tab. Once all missions have concurred, the workspace is ready to be pushed to the master schedule.
8) **Hold and Decide.** When the proposal is created the owner may choose to automatically push the workspace to the master schedule once all missions have concurred, or to hold and manually push the workspace. One reason for manually pushing the workspace to the master would be for contingency schedules: the contingency could be ready and preapproved and then only pushed to the master if the contingency is activated (e.g. a launch slip).
9) **Push Proposal to Master.** The workspace activities and request are now merged into the master schedule based on either the auto-push selection or the user's specific action of pushing the changes to the master schedule.
10) **Workflow – Reject.** In the event an affected mission disagrees with the proposal they may reject the proposal simply by clicking an icon and optionally providing a reason for the rejection. All missions included in the proposal are notified immediately.
11) **Withdraw or Continue?** In the case of a rejected proposal, the workspace owner has the choice of a) doing nothing, b) withdrawing the proposal and continuing to use the workspace, or c) withdrawing the proposal and returning to Step 5, collaborating with other schedulers to ultimately find a resolution of the conflict. This process may be iterated as many times as necessary until an acceptable solution is found. S^3 tracks all previous proposals for a workspace. It also includes logic such that if multiple proposals are created from a particular workspace, and a mission has previously approved a proposal, then that mission no longer has to approve future proposals as long as their scheduled activities do not change.

VI. Conclusions

S^3 represents a major change in the DSN scheduling software suite and is in the process of deployment during 2010. It embodies a new concept for DSN scheduling – that of peer-to-peer collaborative scheduling in an environment developed specifically for that purpose. S^3 draws on new and powerful web technologies to provide a rich client experience in a web browser, and provides a compelling demonstration of the potential of these technologies for applications of this kind.

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.

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]"Space Communication Cross Support—Service Management--Service Specification. Blue Book. Issue 1." CCSDS, 2009.

[4]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.