Applying Lessons Learned in Planning and Scheduling the Hubble Space Telescope to the James Webb Space Telescope

Robert E. Hawkins, Ian J.E. Jordan, Mark E. Giuliano
Space Telescope Science Institute
3700 San Martin Drive, Baltimore MD 21218
{rhawkins, jordan, giuliano} @stsci.edu

Abstract

The Hubble Space Telescope (HST) has been performing science observations since its launch in 1990. During this time the scheduling efficiency of the telescope has improved beyond the level postulated prior to launch. The ability to attain these efficiencies is due to many factors including changes in science policy, planning and scheduling software, operational procedures, and the organization of personnel teams. This paper illustrates how the integration of these factors has increased HST efficiency and shows how these lessons learned are being applied to on-going development of the James Webb Space Telescope (JWST).

Introduction

Effective utilization of space-based resources such as astronomical observatories is critical to supporting NASA's mission of advancing and communicating scientific knowledge and understanding of the Earth, the solar system, and the universe. The Hubble Space Telescope (HST) has made significant contributions to our understanding of the nature of the universe. Part of the success of the HST mission has been the creation of systems that allow HST to achieve high science efficiency. Prior to launch, engineering studies predicted that the science efficiency of the telescope, as defined by the Design Reference Mission (DRM), would be approximately 35% (DRM, 1982). This level of efficiency was not achieved until several years post-launch (Miller and Stanley, 1998). HST achieves an average efficiency of around 45% during current gyro-degraded operations, and has achieved average efficiencies up to 50% during normal operations (with peaks into the 55% range) (Adler, 2008). While improvements in planning and scheduling technology were a factor in improving scheduling efficiency, many other factors were critical including changes in science policy, operational procedures and processes, and the organization of personnel teams. To increase science efficiency a system was developed for HST that provides the following capabilities:

- A science selection policy that motivates end users to create efficient observing programs and that selects a mixture of observations which can be scheduled together efficiently.
- Operational procedures and processes that allow expert users control of the planning and scheduling process while automating routine steps in a mixed-initiative fashion.
- A suite of software tools integrated with operational procedures that allow observations to be prepared, planned, scheduled, and analyzed.

These capabilities cannot exist in isolation and are embodied in multi-disciplinary teams consisting of members from science policy, engineering, software development and test, and operations. Most importantly, a mission must be able to manage change in response to changing technologies, mission hardware, and science goals.

The James Webb Space Telescope (JWST) is currently under development for a 2013 launch. The ground systems for selecting, preparing, planning, and scheduling JWST observations are actively being developed. Although the mission has some differences from HST, it shares many of the same problems and issues. This paper provides a historical overview of the factors that allowed the HST mission to achieve higher than predicted efficiency, and then shows how the lessons learned in the HST mission are being applied to the development of JWST systems.

The remainder of the paper is organized as follows. The first section gives background on the HST and JWST missions. The following section outlines HST lessons learned in process flow, software, and teaming and describes how the lessons learned are being applied to JWST development. Finally, challenges in managing change are discussed.

Background

HST Mission

HST is a general-purpose space observatory that provides support for near-infrared, visible, and ultraviolet frequencies. HST has multiple science instruments that were designed to be upgraded and replaced during its mission. HST's Low Earth Orbit (LEO) dominates the scheduling of observations. Aside from the LEO, the main

physical constraint on HST observations is that the target selected by the observer must not be scheduled within a minimum angular separation from the sun or moon. In addition, a user can place other requirements on an observation including the time windows for observations (e.g. schedule OBS1 day 330-360), links for observations via precedence or grouping relationships with offsets (e.g OBS1 after OBS2 by 10-15 days), links for observations via spacecraft roll constraints (e.g. OBS1 same roll as OBS2).

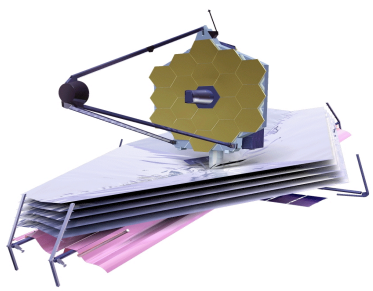

Figure 1: The James Webb Space Telescope

JWST Mission

The James Webb Space Telescope (JWST) will be a large, infrared-optimized space telescope, designed to find and study the first galaxies that formed after the Big Bang. JWST will have infrared sensitive detectors and a 6.5-meter segmented primary mirror that allows it to also look through interstellar dust clouds to see and study the formation of stars and planets. The telescope will have a lifetime of 5 to 10 years and will be placed about 1.5 million km from Earth in an orbit around one of the semi-stable Lagrange points in the Earth-Sun system (L2). Observer specified constraints (both physical and proposer-defined) on JWST are similar to those on HST.

Lessons Learned

The process for implementing observing programs at the Space Telescope Science Institute (STScI) has evolved substantially since its inception. The changes in processes, software and the organizations that support them have been instrumental in delivering improved efficiency. This section describes those changes, couched in terms of the flow of a proposal through the observation implementation process at the Institute. The current process that has evolved for implementing programs on HST, and how that model is shared with JWST is described, as well as some places where they differ. Improvements that have been implemented in each step to enhance HST observing and how they are being applied to JWST are then discussed.

The Observation Implementation Process

STScI is charged by NASA with operating the HST and JWST observatories. STScI is responsible for all phases of science implementation including:

- Proposal preparation and approval.
- Program definition – The detailed definition of accepted observing programs from the astronomical community.
- Program planning and scheduling of science observations and engineering activities.
- Program archive and data analysis.

Treatment of the long-term archival and data analyses processes are outside the scope of this paper. Instead, the focus will be on the improvements in the processing of programs in the yearly "Cycle" of observing on HST.

In practice, observations are carried out in a repeated yearly Cycle where each Cycle consists of 3 phases: proposing, submission, and implementation. However, there is not always a clean step-by-step break between these processes, and there is seldom a clean break from Cycle to Cycle.

In the first phase, astronomers submit proposals for using the telescope to STScI. An external Time Allocation Committee (TAC) ranks the submitted proposals by scientific merit. During the second phase, programs (accepted proposals) are then filled out in detail, and submitted to the institute and checked for consistency, efficiency, and finally "ingested" into the program implementation systems. In the final phase, a Long Range Plan (LRP) is created from the body of submitted programs (and those remaining from previous cycles). These programs are then scheduled and uploaded to the spacecraft at regular intervals.

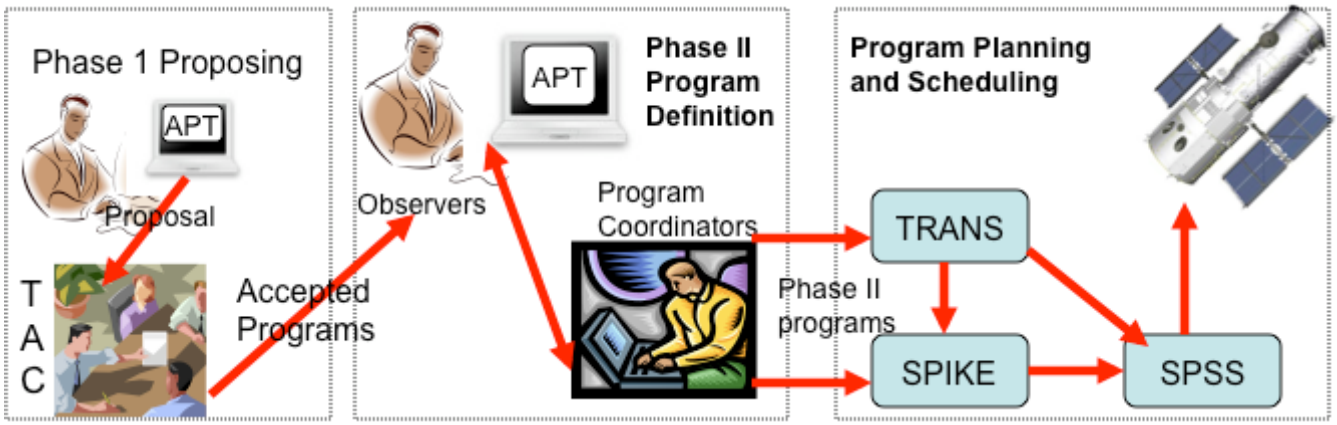

Figure 2:HST proposal process.

Figure 2 depicts the proposal process and software systems used for HST. After a call for proposals is issued to begin a new observing Cycle, proposers use the Astronomer's Proposal Toolkit (APT) to create and submit proposals. The TAC reviews and accepts a subset of the submitted proposals. Observers continue to use APT to define their proposals, iterating with program coordinators at STScI. TRANS (Gerb, 1991) is used to create the execution structure. Planners utilize SPIKE (Johnston and Miller, 1994) to create the LRP. Finally, the Science Planning and Scheduling System (SPSS) creates short-term schedules.

JWST will use the same coarse-grained process as HST, but the cadence will differ. For example, HST's cadence has weekly uploads detailing a week's worth of observations filling the 15 ~95 minute-orbits per day. JWST will have 7-10 day uploads organized around a 21-day station-keeping cycle. Additionally, JWST's orbit enables a completely different execution style, allowing for more onboard autonomy than HST has.

HST is serviceable, with the 5th manned servicing mission planned for May of this year. JWST is unlikely to be serviced, due to its remote station, unless robotic servicing occurs. However, JWST will have more advanced storage and computing power than HST.

TAC/LRP Planning Feedback

The prime concern of the TAC is the scientific merit of the proposals, regardless of schedulability. However, HST experience has demonstrated that the TAC needs insight into the scheduling properties of the body of proposals that are under consideration, and the resultant resource usage over the cycle.

In HST's Cycle 12, a confluence of large programs targeted in one part of the sky or having overlapping timing constraint windows caused the LRP to become overloaded with observations during one season. This was a significant concern from policy perspectives as well as from observer and observatory planning points of view. Without prompt feedback to the TAC on the potential impact of such program combinations, guidelines to observers, conditions on approval, or modification in the choice of programs approved could not be made. This resulted in program submissions that were incompatible with timely program completion and sustained telescope efficiency. In the case of Cycle 12, modifications to more than one of the large programs were required after submission, resulting in significant rework and program changes by the observers.

Subsequent to this, HST Project and STScI management undertook changes with the ground system's planning branch to extend existing tools (SPIKE) to address this problem for future Cycles. The changes allowed summary information from large proposals to be used after Phase I submission and up through the time of the TAC meeting to forecast relative demand on HST's scheduling windows. Target special requirements in proposals exceeding 100 orbits were required in Phase 1 to ensure STScI had sufficient information to forecast inter- programs conflicts.

To extend this insight to JWST, the following actions are being taken:

- STScI management plans to encourage proposers to submit more fully defined proposals during Phase 1 to enable the same type of tool-based analysis described above.
- Experimental tools are being developed that allow resource-balanced plans to be created based on Phase 1 proposals. This capability would allow acceptance of a set of infeasible proposals to be avoided by informing the TAC of potential trade-offs in scheduling efficiency. By exposing schedule trade-offs in the pool of possible proposals, the TAC can achieve the desired balance between accepting high priority science and executing more science programs.

Program Preparation Software

During the 20+ years of software development for HST, the tools used by observers to prepare programs have evolved. For the first 5 cycles, the observer was faced with text-based forms, e-mail submission and a multi-step process to understand and enable the schedulability of their program. A submitted program would be processed by several software systems at the institute to determine whether it fit into orbits appropriately, whether it was schedulable, whether special requirements could be met, etc. This could be an arduous and complicated process that would sometimes require weeks to return feedback to the observer.

Although the initial program preparation process was highly manual, there were several tools available at STScI such as SPIKE and TRANS that could provide useful feedback on the schedulability and feasibility of scheduling. The first step in automating the process was implemented in the RPS2 system that provided observers with a downloadable tool, which ran and displayed the output of these existing systems (Koratkar, 1999). Although RPS2 presented results graphically, the system was mainly a controller that ran syntax checking, SPIKE and TRANS jobs in batch, and then statically displayed the output. Even though RPS2 was not truly interactive, it reduced the amount of time required to provide

schedulability and feasibility feedback to observers from days to on the order of tens of minutes.

Though RPS2 was successful, it had many shortcomings. The process for creating legal programs was cumbersome since the RPS2 editor was missing many interactive features. Its performance was slow as all the underlying software components were run in batch. In addition, observers often had to iterate with the system to fill up their orbits. Since each iteration could take tens of minutes, this could make program preparation painful for observers who were most often constructing their programs on the day of the submission deadline.

Initially RPS2 was conceived as a stopgap measure until an automated system could be developed. RPS2 was used for program preparation and submission from 1994-2002. During this time, several proof of concept studies were performed, evaluating how a program preparation tool should best be structured. In addition, the TRANS system was updated during this time frame (Curtis et. al., 1999). The existing TRANS code was becoming increasingly hard to maintain and was not structured to handle science instruments planned for installation on upcoming HST servicing missions. Although not a primary goal of the TRANS reengineering, one side effect of the effort was that the system could be easily integrated with an on-line GUI to display HST orbit structures.

In 2002, the current generation of program definition and submission software, APT, was released. This software had the advantage of being a truly GUI based system that interacts with observers and the underlying schedulability and feasibility engines in an on-line fashion. APT solved many of the run-time problems and provided many other useful features:

- Programs are constructed using a specialized editor that ensures only legal values are selected. For example, after choosing a specific HST instrument, only parameters associated with the instrument are available.
- APT provides interfaces with existing astronomical resources such as target catalogs.
- APT provides tools allowing users to effectively pack their allocated orbits. Users can specify what exposures go into which orbits and have options to automatically fill orbits.

The evolution of orbit packing support in APT shows the value of mixed initiative planning. A major complaint with RPS2 was that users had to perform multiple iterations to get exposures into the right orbits. Increasing or decreasing exposure durations might cause the observation to require more or less orbits than desired. The problem was that RPS2 made the decision on how many orbits an observation should require. In APT, the decision of which exposures go in which orbits was placed in the hands of the user. Instead, APT provides tools that increase or decrease exposure duration so that the exposures assigned to an orbit fit in the orbit. This approach gave astronomers the control they needed to determine what exposures go in what orbits while automating the process of filling up the orbit.

The JWST mission plans to reuse the APT platform, building on its interfaces and capabilities. Several lessons learned from HST are being applied to JWST. First, with HST the APT system and the underlying SPIKE and TRANS scheduling engines are distributed to observers as part of the downloaded application. Distributing the schedulability engines complicates development and delivery since these systems are specialized for each supported platform. For JWST, schedulability engines will run on servers at STScI. Only platform-neutral JAVA code for the core APT interface will be downloaded to the observer's desktop. Secondly, JWST observers will fill out templates in APT to specify observations. These templates encapsulate the desired instrument usage rules and shield users from having to understand some activity related details of the instruments when preparing programs. Finally, the JWST APT will provide advanced visualizations such as displaying how mosaic observations lay out on the sky.

In addition to these improvements, APT is faced with the following challenges:

- Supporting the HST and JWST missions simultaneously in a manner where both missions can leverage the improvements made for the other.
- Moving from a downloaded application to a system that runs seamlessly from a browser or on other computational devices.
- The JWST development plan requires that APT be constructed early in the process to support ground system and instrument testing. This presents a challenge in maintaining a state of the art interface to the observer as new technologies become available.

Time Allocation: Using the Right Units

Initially, STScI allotted HST time in a way that a number of other observatories currently do: in conventional civil time units (seconds/minutes/hours) (Mutchler et. al. 1994). Observers would structure their observations optimally from their perspective and submit them to STScI. However, the natural units of observing elements

(exposures) were not inherently sized to fit in with the periodic HST visibility unit: the orbit. Target visibility is highly variable depending upon the geometry of the sun, moon, earth, and HST orbit plane. A typical target may be above the earth-limb avoidance zones anywhere from 45 minutes to several days continuously (very rare), averaging in the neighborhood of 55-minutes. However, observers might construct a series of sequential exposures that would only allow partial filling of available visibility (say, for example, three twenty minute exposures, where scheduling two in an orbit would leave 10-15 minutes of the orbit unusable because the subsequent exposure could not fit in the remainder of the orbit). This led to inefficient use of the observatory.

The original process and software matched this presumption of allocating time vs. orbits, thus the observer had no insight into how to make their observations fit into orbits, and no motivation to do so. Policy and software changes to address this resulted in the TAC allocating HST time in multiples of the natural HST time-unit: orbits. Observers would request HST time based upon nominal-length orbits, and it would be their responsibility to pack these orbits. The release of RPS2 to replace RPSS was necessary for the switch from allocating HST in spacecraft time to integral HST orbits. The integration of TRANS into the proposal preparation tools allowed observers the desired insight into how exposures could be executed in HST orbits.

JWST will not be in low-earth orbit, so observers will once again receive their allocation in spacecraft time. However, as with HST, efficiently scheduling JWST will start with encouraging the observer to craft efficient programs. First of all, a program that can achieve good science efficiently will be more likely to win approval. However, at submission time, efficient programs are also rewarded with more on-target exposure time, since programs that require large slews will have their observing time "taxed" by a constant quantity based on the average time required for each slew. (Henry, 2008) Interactive tools will continue to be developed to enable observers to easily specify efficient proposals. Development is underway on interfaces for the more complex JWST instruments.

As described above, the JWST program preparation tool is incorporating the concept of "templates" to ease the task of creating scientifically useful observations for JWST instruments. In addition, templates allow for increased ground system flexibility and efficiency. For example, consider an observation that needs to take a series of exposures at different wavelengths at several adjacent target pointings. This observation could be executed with an outer loop that slews to each pointing and an inner loop that iterates over the desired wavelengths. Alternatively, the observation could be executed by an outer loop that iterates the wavelengths and an inner loop that iterates over the target pointings. The choice of implementation will impact the efficiency of the observation, depending on the relative overheads for wavelength changes versus target slews. Templates allow observers to specify the pointings and wavelengths so that they can get the correct science for their observation. However, templates do not allow observers to specify which implementation to select. The choice of implementation can be made late in the scheduling process based on efficiency and engineering concerns.

Organizational Teaming

Over the course of the years, the management and organization of people into teams and divisions (groupings of teams) has had significant impact on how work is done and on inter-team communication. The original organization of the Institute was such that a program went through 3 divisions (Miller and Johnston, 1991) before reaching operators who were able to determine whether it was schedulable or not, then had to be bounced back to the observer (through those same 3 divisions) to get changes made. HST observation implementation was originally organized by function: one group was responsible for user support, another for observation preparation, and others for long-range planning and short-term scheduling. This led to a fragmentation of responsibility and many hand-offs. No single point of contact existed for an observer to solve problems with their observations. As each group had different tools, technologies, and terminology, it was costly to get work completed. Not only were the user groups compartmentalized but also the software developers supporting the users were spread out across multiple divisions, which were different from the operations divisions. Even if someone had a good idea on how to change the software or improve a process it was hard to figure out which division was responsible for approving and implementing the change.

In 1994, the Institute was reorganized to remove some of these organizational impediments (Miller and Stanley, 1999). The Project to Re-engineer Space Telescope Observing (PRESTO) resulted in a new division (O'Dea et. al. 1999) responsible for the front-end activities associated with HST observing, from science program selection to command load generation. An important part of this change was creation of a team of operations staff called Program Coordinators (experts from those 3 divisions) who had direct interaction with the observers, and were assigned programs and tasked with the responsibility of shepherding them through the implementation process. Operations improved by having one group responsible for successful implementation of

observations, which involved several different software systems. In addition, PRESTO integrated users with their supporting software developers and facilitated creating innovative solutions to problems. As a result of this re-organization, the quality of user support increased while the cost of operations decreased.

The organizational structure of STScI is constantly in flux as the mission of the Institute evolves and the leadership changes. Some organizations have worked well by forming effective teams while others have introduced overhead and reduced team cohesion. It is our hope that the organization of STScI during the JWST era will support effective teaming.

Planning and Scheduling

The initial concept for HST was to build week-long flight calendars 2 months in advance, with little or no advanced planning. A single operator would transform form-based program submissions into executable specifications by hand and schedule them on a weekly basis using only the short term scheduling software. TRANS, in addition to other systems, had to be developed pre-launch to replace this manual process, as it would not have succeeded. This process allows for efficient management of the subset of observations selected to execute onboard the observatory. However, a significant problem with this approach was a disconnect between the yearly program selection process and the weekly calendar creation process. Since weekly schedules were created in a greedy fashion, this approach would have yielded even more inefficient HST usage by not planning around larger scale scheduling issues.

In the years immediately following launch, this procedure evolved into having programs planned over the full Cycle by assigning the individual observations to particular 1-week calendars throughout the year. This was an attempt to create a long-range schedule that balanced resources globally while giving short-term schedules flexibility. However, this approach did not work optimally because it was not robust to change, and required long range schedules to be heavily oversubscribed due to temporal constraint modeling mismatches. The long-range schedule had to be oversubscribed by 100% in order for the short-term scheduler to create effective schedules for HST. This resulted in volatility in the plan, which was undesirable to operators and HST observers who needed to plan for their data reduction activities.

A less rigidly defined yearly planning method was needed, so a new concept was implemented where individual observations would be assigned up to 8-week duration least-commitment *plan windows*, anywhere within which the observation might be validly scheduled. This allowed flight calendar schedule changes to have less impact on the science plan since individual observations could 'slide' around within the plan windows and not be as easily subject to rippling delays caused by changes to other observations and calendars close to flight.

Currently, planning and scheduling at STScI is a two-phase process. In the first phase, an LRP is crafted which assigns each observation plan windows. Plan windows are a subset of an observation's schedulable windows and represent a best effort commitment to schedule within the window (Giuliano, 1996). In the second phase, successive short-term schedules (STS) are created. The short-term scheduler uses plan windows to drive the creation of efficient telescope schedules.

This two phase process allows a separation of concerns in the scheduling process: Plan windows globally balance resources, are stable with respect to schedule changes, and provide observers with a time window so they can plan their data reduction activities. Short-term schedules provide the detailed, time-sequenced activities that HST needs in order to execute the observations.

Command-loads uplinked to HST are derived directly from a flight calendar (nominally of 7-days duration) building process that starts well in advance. In the late 1980s (before HST launch) and early 1990s, the build procedure for a given flight calendar began 8 weeks prior to first on-orbit execution. The long lead-time was required due to computing-power limitations, lead-time requirements for Tracking and Data Relay Satellite System scheduling, padding to send the schedule to Goddard for uploading, and procedures that had not yet been optimized. This long lead-time had impacts throughout the flight-preparation system. One impact was that relatively early flight-ready dates were required for the final versions of the programs. Additionally, changes in calendars close to flight to accommodate observation changes, additions and failures, resulted in frequent rework to subsequent calendars.

As allowed by increases in compute power, improved tools, and more efficient procedures, the horizon time was found to be shrinkable. In the late 1990s and early 2000s, the scheduling horizon was progressively shrunk to 21-days, 14-days, and then 11-days (where it now remains). This allowed much of the previous calendar rework to go away.

In the JWST domain, STScI plans to reuse the two-phase LRP/STS concept. However, there are several changes to short-term scheduling that will help make the system robust and flexible in order to handle change. Uploads to HST specify a list of activities to be executed, each with a time-stamp indicating the absolute execution time. In

contrast, the Observation Plan (OP) uploaded to JWST contains an ordered list of observations, each with an earliest start and a latest start time. JWST's execution agent examines the observations in the OP one at a time. For each observation, the following cases apply:

- The current time is prior to the early-start, in which case the execution agent waits until that time.
- The current time is later than the late start, in which case the execution agent skips the observation.
- The current time falls within the window between early and late-start, in which case the execution agent starts the execution of the observation. The execution agent moves on to the next observation either if this observation finishes successfully or fails due to some anomaly.

The mission design and the advanced hardware of JWST enables this method, as there are coarse-grained scheduling windows rather than the shorter duration orbital windows of HST.

The advantages of this approach are several. In time-stamped observing such as HST uses, all time stamps are conservatively padded in order to avoid anomalies due to boundary conditions and ephemeris uncertainties. This will not be necessary for JWST. This approach reduces the pressure on the ground system to accurately estimate overheads in order to maintain efficiency. In the case of an observation anomaly such as failure to obtain guide-stars, HST remains idle, while JWST's execution agent is enabled to move on to further observations. This system also makes it easier to handle short-notice changes in the schedule, such as Targets of Opportunity.

JWST has a 6-month quasi-orbit, a planned nominal 21-day station-keeping activity cycle, a planned 7-10 day upload schedule, and a requirement to respond to anomalies and targets of opportunity in 3 days. One of the challenges for JWST short term scheduling is that it is not yet clearly understood how the cadences will work together. Another challenge is that a short-term schedule is not only an assignment of observations to precise times, but also a determination of the early start and late start. Although the upload contains only earliest start and latest start windows, the short-term scheduler still needs to assign observations to a precise time to propagate changes to the rest of the ground system. Scheduling windows implied by the earliest start and latest start paradigm are akin to plan window creation.

Handling Efficiency Drivers

Designing the software around the domain to handle the big efficiency drivers is another way that STScI has been able to increase efficiency on HST. Handling domain-specific tall poles has been a big factor in achieving high efficiency for HST. Although the JWST mission will have different constraints, it is expected to also have a tall pole resource that limits efficiency. These are discussed in turn.

HST Efficiency – The South Atlantic Anomaly

The South Atlantic Anomaly (SAA), a region of high radiation off the coast of South America, is the main driver of HST efficiency. Depending on the scientific instrument being used, approximately nine of a day's fifteen orbits pass through the SAA for 25-30 minutes of the orbit. In each of these SAA impacted orbits, the SAA passage occurs in a slightly different portion of the orbit.

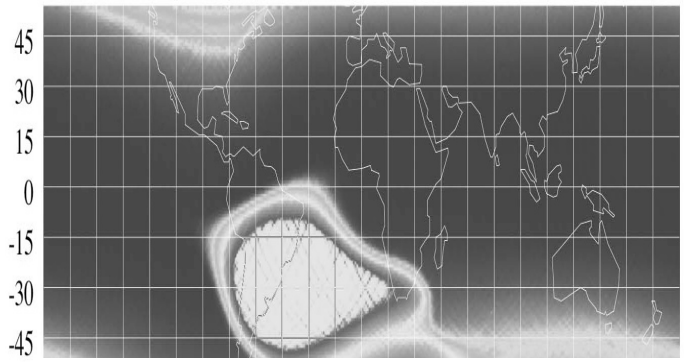

Figure 3: The South Atlantic Anomaly is shown in the lighter shade

No observations can occur during an SAA passage. However, STScI can schedule observations in the remainder of an SAA impacted orbit. If the Earth occultation for the target occurs during the SAA passage, this case is called SAA hiding. Orbits without any SAA crossing are called SAA free orbits, which are more scarce and in greater demand. For any given target, however, SAA hiding only occurs for a small fraction of a year.

There are three points at which SAA scheduling has been addressed in the HST scheduling process: the time allocation process, long range planning, and short term scheduling.

For the most part the TAC does not consider schedulability when approving programs. However, during recent observing Cycles, insufficient programs were available to fill HST orbits, so supplemental observations were required. All-sky scheduling information and SAA-hiding sky locations were provided to the director and TAC to help them decide which programs to select.

Since resource management is handled in the LRP, it is a major factor in maximizing SAA hiding efficiency. The following constructs have been implemented in long range planning in order to achieve this:

- The resource model needs to distinguish SAA-impacted and SAA-free orbits. Over and above this, the system needs to track each of the 9 SAA-impacted orbits in a day as a separate resource.
- Understanding the resource model uncertainties and their levels, along with tracking the expected level of unplanned observations was important for setting subscription levels properly to prevent oversubscription and 'calendar falloffs'. Some examples of unplanned observations are calibrations, target-of-opportunities, and director's discretionary time.
- A guess and repair search strategy that uses iterative deepening to find solutions.
- Software critics, criteria, and filters that ensure that the high value portion of the search space is efficiently explored.
- A simulation based scheduling tool that validates plans.

The tools are designed to work in a mixed initiative mode where they support expert users in creating and diagnosing plans (Zimmerman, 2002). The purpose of the tools is to highlight resource conflicts, and to ensure the end user that any resource conflicts found in schedules are inherent in the input data and not an artifact of the scheduling software. (Giuliano 2008b) (Ferdous 2006).

Even if the long-range plan balances SAA resources this does not ensure SAA efficiency unless the short term scheduling software knows how to exploit the opportunity. SPSS uses information from the LRP and its own reports to schedule observations in SAA-impacted orbits when possible.

JWST Efficiency – Momentum Management

JWST scheduling requires the ability to minimize momentum build-up during scheduling. JWST utilizes a sun shield about the size of a tennis court to keep the observatory cold. Solar radiation pressure on the sunshield causes angular momentum to accumulate in the spacecraft's reaction wheel assemblies. The wheels have a limited capacity to store momentum, and stored momentum must be dumped using spacecraft thrusters. The resulting use of non-renewable fuel makes momentum management a potential limiting factor in the lifetime of the mission. As the momentum accumulated by an observation varies over time, momentum management is expected to be a major constraint driving the efficiency of JWST scheduling. The JWST momentum resource constraint has several interesting features:

- The model is intrinsically three dimensional.
- Resource consumption for an observation varies over time in a non-linear manner.
- Resource consumption is vector additive in nature — scheduling an observation at a particular time can either add to or subtract from the overall accumulation.
- Momentum provides both a hard constraint due to a limited capacity, and a preference to consume as little resource as possible.

These features are different from the types of resources covered in the planning and scheduling literature (Laborie 2003) where activities consume and release a constant capacity. In particular, the non-linearity of the domain prevents us from employing techniques commonly used to handle resource constraints.

There have been internal JWST studies on momentum management describing the constraint (Petro 2005) and potential approaches for achieving scheduling efficiency (Giuliano 2008a, Kinzel 2006). Current approaches to achieve momentum-based efficiency concentrate on treating momentum build up as a resource constraint. We expect the techniques and tools developed for handling HST resource constraints will be applicable for JWST momentum management.

Modeling Mismatches

The software systems for HST evolved by adding components as they were needed and not by a coordinated effort. As a result, the spacecraft models for the subsystems differed. For example, each system has a slightly different model of the observations to be executed onboard the telescope. The user enters programs in terms of exposures and observations. The TRANS system creates a hierarchy of activities from the exposures in an observation based on how the exposures fit into orbits coupled with engineering constraints. The short-term scheduler, SPSS, does not know about exposures or observations but only the intermediate structures. SPIKE has minimal knowledge of the orbit structure. As such, it is hard to communicate between the scheduling systems. In addition to differences in modeling observations, each system has its own routines for computing temporal constraints. The SPSS model of the physical telescope constraints is more detailed than the model in SPIKE. However, these constraint calculation routines are not encapsulated in a way that can be used by other systems. The SPIKE model of constraints is well encapsulated but does not model all of the constraint details. Mismatches

in constraint calculations at best lead to wasted time for operations staff or at worst, missed opportunities for HST science. Model mismatches are an on-going concern for HST operations and software development.

For JWST, STScI is evolving those same systems (from the proposal tools, through long and short range planning and scheduling) to use the same computational models. A central data repository provides common schedulability results to be shared across all proposal planning systems. The JWST proposal planning system design incorporates a new component - outside of the LRP, STS and APT software, but shared by all - called the Proposal Constraint Generator (PCG). The PCG will provide APT with the scheduling windows for observers to visualize. This will be provided through a web service hosted at STScI, communicating details to the proposer's desktop. These constraints will also be computed and stored in a local STScI database by PCG for the LRP and STS software to use. This design will provide both a single mathematical model for schedulability to avoid mismatches and a central repository to store the data, removing the need for file-based storage and duplication.

Handling Change

Ultimately, the clearest lesson from developing software for space-based telescopes is to do anything and everything that can be done to anticipate and handle change. In particular, we've noted that there are two primary components of change that regularly occur in space-based missions: changing science requirements and changes in hardware (both expected and unexpected).

The science programs that actually execute on board in later 'Cycles' are going to change once experience is gained about what the observatory is **really** capable of doing (either for good or bad). Scientists will become more innovative once they have an observatory to work with and this will change what science programs are proposed. Furthermore, interests of astronomers will change, during longer missions like the Great Observatories. For example, when HST was launched there was little thought given to using transit and other methods to detect extra-solar planets.

Systems need to be designed to accommodate for degraded modes and hardware failures as the observatory ages. HST has encountered many unexpected changes:

- Due to the delay of Servicing Mission 4 following the Columbia disaster, Hubble observing has been operating in a degraded pointing mode, using 2 gyros instead of the planned 3 gyros. This mode required significant change in the activity management model.
- The motors for solar arrays had problems upon launch, which limited the roll of the spacecraft. This problem required changes to the ground systems software, as the model had originally been hard-coded.
- After the STIS instrument was installed on HST, the MAMA detectors had to become SAA-free only demanders, due to hardware problems.

The latter of these changes was actually handled with minimal effort by the planning and scheduling systems. This shows that some of the ground system architecture choices were good, however it is hard to predict in advance where to build in software flexiblity.

Design for Change

Governmental funding requirements dictate the need for deployment years in advance so that systems can be validated and verified. However, the ultimate requirements are often not known until after commissioning.

There was flexibility built into the HST systems, but many of the examples shown required significant coding to enable the ground system to handle them. When possible, systems architects should apply lessons learned from previous experience to build systems that can flexibly handle potential changes. Some questions to consider during ground system software design are:

- What features of the software can be table or data driven? For example, JWST APT will incorporate XML-based models specifying the legal syntax of instruments that can be changed as needed.
- What features can be implemented using an object-oriented approach so that creation of new behavior is possible via implementing protocol methods? An example of this would be adding a new type of temporal constraint.
- What types of features would require too much effort to accommodate a-priori?

The creation and empowerment of multi-discipline teams is an important factor in managing change. By empowering teams, they will be able to respond to challenges proactively. By having members with different expertise in the ground system, teams will be able to understand challenges and to brainstorm process and software solutions. In particular, mixing software developers with expert software users helps to create mixed initiative tools that give the ultimate decision making power to the users while automating routine tasks.

Conclusions

In order to enable the change process, there needs to be institutional commitment to support cost-effective change for the duration of the mission.

From (Johnston and Miller, 1994):

> "Our recommendation to developers of future systems with requirements similar to HST would be to build in the expectation of change from the outset and to carefully examine factors in the design which are sensitive to a high rate of change."

References

Adler, D.. et. al. 2008. *Maximizing scientific return for the Hubble Space Telescope in a post-SM4 world*. Observatory Operations: Strategies, Processes, and Systems.

Curtis et. al., 1999. *TransVERSE: An Architecture for Configuring Astronomical Observations*. Astronomical Data Analysis Software and Systems.

Design Reference Mission SE-01, Lockheed Missiles and Space Company, 1982.

Ferdous, N. and Giuliano, M., 2006. *Validating Resource Usage in Least Commitment Scheduling*. International Workshop on Planning and Scheduling for Space Baltimore, MD, Space Telescope Science Institute.

Gerb, A. 1991 *Transformation Reborn: A New Generation Expert System for Planning HST Operations*, Goddard Conference on Space Applications of Artificial Intelligence, NASA CP 3110, 283-295.

Giuliano, M. 1998. *Achieving Stable Observing Schedules in an Unstable World*. In Astronomical Data Analysis Software and Systems VII. 271-274.

Giuliano, M.E. and Johnston, M. 2008a *Multi-objective Evolutionary Algorithms for Scheduling the James Webb Space Telescope*. Proceedings of the International Conference on Automated Planning and Scheduling(ICAPS), Sydney, Australia.

Giuliano M.E., 2008b Handling *Oversubscribed Orbital Resources in Hubble Space Telescope Operations* in the Proceedings of the Workshop on Oversubscribed Planning and Scheduling ICAPS, Sydney Australia.

Johnston, M. and Miller, G., 1994. *Spike: Intelligent Scheduling of Hubble Space Telescope Observations*. Intelligent Scheduling, ed. M. Fox and M. Zweben, Morgan-Kaufmann.

Kinzel W., 2006. *Managing Angular Momentum Accumulation by Visit Sequencing and Roll Section*. JWST mission internal report 713.

Koratkar A. et al., 1999. *APT: HST proposal preparation environment for the second decade*. STScI internal technical memorandum ITM-1999-06.

Laborie, P. 2003. *Algorithms for Propagating Resource Constraints in AI Planning and Scheduling: Existing Approaches and New Results*. Artif. Intell. 143(2):

Miller, G. and Johnston, M. 1991. *A Case Study of Hubble Space Telescope Proposal Processing, Planning and Long-Range Scheduling*. Computing in Aerospace 8, American Institute of Aeronautics and Astronautics.

Miller, G and Stanley, P. 1998. *Applying the Lessons Learned from HST Operations to New Missions*. Observatory Operations to Optimize Scientific Return. SPIE Proceedings Vol 3349, pp. 20-29 (1998).

Mutchler, et al., 1994. *Observing with HST I: A New Phase I Proposal Process*, AAS.

Henry, R. 2008. *JWST SOC Proposal Planning Subsystem Requirements Document*. JWST-RQMT-002940.

O'Dea, Christopher, . *User support: lessons learned from HST*. Proc. SPIE Vol. 3349, p. 55-62, Observatory Operations to Optimize Scientific Return, Peter J. Quinn.

Petro L. et al, 2005. *The angular momentum of JWST in a schedule of the Science Operations–Design Reference Mission*. JWST mission internal report 556.

Zimmerman, L., Asson, D., 2002. *SPIKE: A dynamic interactive component in a human-computer long-range planning system*. 3rd International Workshop on Planning and Scheduling for Space, Houston, Texas.