The Mystery of the Missing Requirements: Optimization Metrics for SPIKE Long Range Planning

Mark E. Giuliano
Space Telescope Science Institute, 3700 San Martin Dr., Baltimore, MD 21217, USA
giuliano@stsci.edu

Abstract

Although the SPIKE system has been used for long range planning Hubble Space Telescope (HST) observations for over 20 years, the core requirements that determine the quality of a plan have never been formally specified. While this requirements deficit has not prevented SPIKE from producing plans that satisfy operations staff, the desire to use different operations concepts for the JWST mission, and the desire to make SPIKE a more agile system have motivated the SPIKE team to extract optimization requirements so that new planning approaches may be explored. This paper first describes the missing requirements and the evolutionary approach to developing SPIKE that lead to them. Newly developed optimization requirements are then described within a multi-objective framework. Finally, ongoing efforts to validate the requirements are presented.

Introduction

"The world is full of obvious things which nobody by any chance ever observes."

Sherlock Holmes -*The Hound of the Baskervilles*

Often software systems evolve in the manner described by the popular bumper sticker that can be paraphrased for polite company as "excrement occurs". This situation results as: Software requirements are specified in terms of solutions not in terms of problems; New software requirements are added which contradict goals from previous requirements; New software modules are added that interact with existing modules in unpredictable way; all of this is occurring within the context of a system that is actively being used for processing changing operational data. As a result, the true goal of a system is often buried deep within software modules and is implicit in the software behavioral requirements. Such systems may be highly successful and meet the needs of end users. This state of affairs is sufficient as long as the system is relatively stable and no major changes are desired to either system output or system algorithms. When changes are desired, the implicit requirements are a problem, as the desired behavior of the system cannot be quantified. New algorithms can only be compared against the output of the current system thus inhibiting the use of new and potentially better algorithms.

This paper describes requirements for HST long range planning and shows how the evolution of the system resulted in optimization requirements being buried in software mechanisms. New requirements are then described within a multi-objective framework. Finally, on-going efforts to validate the requirements are presented.

HST Planning and Scheduling

Launched in 1990, the Hubble Space Telescope is a general purpose space observatory that provides support for near-infrared, visible, and ultraviolet frequencies. HST has multiple science detectors that were designed to be upgraded and replaced during its mission. HST is in a low earth orbit approximately ~600 km above the Earth and orbits the Earth every 96 minutes. The main physical constraint on HST observations is that targets selected by the observer cannot be occulted by the Earth, the Sun, or the Moon. In addition, a user can place other requirements on an observation including the ability to specify time windows for observations (e.g. schedule OBS1 day 330-360), to link observations via precedence or grouping relationships with offsets (e.g OBS1 after OBS2 by 10-15 days), and to link observations via roll constraints (e.g. Same roll OBS1 as OBS2). The time intervals that satisfy all the constraints are called observation *constraint windows*.

Hubble observations are carried out in a repeated yearly cycle. In each cycle, astronomers submit proposals for using the telescope to the Space Telescope Science Institute (STScI). The submitted proposals are ranked by scientific merit by an external Time Allocation Committee. Approved proposals are prepared for execution using the Astronomer's Proposal Tool (APT) and then submitted to STScI. Accepting a proposal represents a commitment by STScI to execute the observations to completion. There are no priorities within the pool of accepted proposals. Each accepted program is assigned a program coordinator at STScI who works with the external astronomer and the internal STScI scheduling staff to see that program gets successfully planned and scheduled. A key action performed by programs coordinators is marking observations as ready for long range planning

HST scheduling is handled in a two-phase process by separate long-range planning and short-term scheduling systems [Giuliano 1998]. In the first phase, long-range planning assigns observations to overlapping least commitment plan windows that are nominally 56 days long. Plan windows are a subset of an observation's constraint windows and represent a best effort commitment to schedule within the window. In the second phase, plan windows are used to create successive short-term schedules for 7-day upload periods. 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. The short-term scheduler provides efficient fine-grained schedules to the telescope.

This paper concentrates on long range planning as performed by the SPIKE planning and scheduling system.

Long Range Planning Use Cases

The SPIKE system is used in three different use cases for HST long range planning:

- Cycle Ingest
  - At the start of each yearly cycle, plan all long range plan ready observations from the current cycle, on top of observations from previous cycles that have not already been executed.
- Cycle Maintenance
  - Maintain the plan in response to:
    - Changes to observation specifications in order to improve science return;
    - New observations becoming long range plan ready;
    - Execution times of observations (i.e. adjusting windows for observations linked to executed observations).
- What if Analysis
  - Perform what if analysis on all available observations to account for major changes in spacecraft constraints and capabilities.

This paper concentrates on the cycle ingest use case as this is where requirements are missing.

Cycle Ingest Planning Requirements

The structure of the input and output requirements for SPIKE long range planning are well defined. At a high level SPIKE operations can be summarized in the following equation:

Plan(observation-data,input-plan,control-parameters) = output-plan

In other words, SPIKE long range planning can be viewed as a function that takes three parameters observation-data, input-plan, and control parameters and produces an output plan. As SPIKE takes a long range plan as input and writes long range plans as output a long range plan will be defined first.

A long range plan consists of a set of records that are stored in a database:

- A record for the plan as a whole:
  - The name of the plan
  - The input plan (if any)
  - The creation date of the plan
  - The user or system producing the plan
  - Comments by the user
  - The version of SPIKE used to create the plan
- A status record for each observation that was loaded into SPIKE at the time the plan was created, including:
  - The observation version number
  - The planning status (unplanned, planned, executed)
  - The lrp-ready status of the observation
- A plan window record for each observation that is assigned a window.
  - A plan window record is a set of intervals that is a legal subset of the observation's constraint windows
  - Not all observations will be assigned a plan window.

        - E.g. observations that are not lrp-ready, or not schedulable within the desired time frame.
- A plan orient record giving the spacecraft roll for each observation that requires a roll assignment.
    - i.e. Observations that are linked via a roll requirement.

SPIKE takes as input:

- SPIKE loads all programs that are ready from the current cycle and any programs from previous cycles that have at least one observation that has not completed.
    - Typically there is a backlog of observations from up to three or four previous cycles.
- Data for each observation
    - Specifications of targets
        - Fixed (i.e. stars, galaxies), solar system, internal
    - Specifications of how the exposures in an observation are laid out in orbits
    - Scheduling requirements
        - Timing links to other observations
        - Phase requirements
        - Desired time windows
        - Desired spacecraft rolls
        - Roll linkages between observations
- An input long range plan
    - Gives plan data for observations from previous cycles
- Control parameters
    - Constraint calculation parameters
        - Sun, moon, and earth avoidance angles and other constraint tolerances
    - Planning control parameters
        - A set of dates within which to plan new observations
        - Tolerances for preferences and control mechanisms

From this input SPIKE determines plan windows and plan orients for all observations from the current cycle that have been marked lrp-ready and that are schedulable in the given time frame.

The Missing Requirements

While SPIKE has strong requirements on the legality of its input and output products, there have been no solid requirements on what makes a good plan versus a bad plan. We know that SPIKE has three high level goals:

- Enabling the efficient execution of HST observing programs;
- Providing stable plan windows to end user astronomers;
- Minimizing the work required by in house users to prepare, plan, and schedule observations.

Unfortunately, there is no direct way to evaluate a long range plan by actually creating short-term schedules from the plan. The short term scheduling process is highly manual and requires knowledge of the position of HST that is only known a few weeks in advance. Although no direct plan evaluation is possible, aspects of what allows efficient execution have been identified since the start of HST operations:

- Balance resources across the planning cycle
    - Preferentially plan observations in windows which can utilize orbits which cross the South Atlantic Anomaly where observations cannot be scheduled (See below for details)
- Create windows that do not complicate the short term scheduling process
    - E.g. The short term scheduler creates week long schedules typically running from Sunday midnight to the next Sunday midnight. Plan windows should not extend into the next short term schedule week by less than 12 hours unless there is no other option.
    - E.g. Plan windows for observations linked by timing requirements (e.g. Ob1 and Ob2 by 10 to 20 days) should not be created such that scheduling the first observation leaves little flexibility for scheduling subsequent observations.
- Create plans that are good for the end astronomer
    - Astronomers want stability in the plan so they know when to plan for data analysis (i.e. know when to hire graduate students).
    - Astronomers want the complete program finished and not to have to wait for the final observations to execute.
    - All things being equal astronomers prefer their observations to be scheduled sooner than later. However, since there are no priorities between accepted proposals we have no requirement for an early criteria.

The space of possible plan windows for an observation is astronomical (pun intended) as it consists all the subsets of an observation's constraint windows that are 56 days (the nominal plan window size) or less. As such early planning software mechanisms concentrated on getting the software to explore the right subset of the search space for an individual observation [Giuliano 2008]::

- Planning criteria were defined that judge the quality of windows for individual observations.
    - South Atlantic Anomaly scheduling, plan window size, and scheduling with respect to resources
- Critics and filters mechanisms were defined that ensure that if certain high quality windows are available then the system will select those windows without ever looking at lower quality windows.
    - If windows that can utilize South Atlantic Anomaly orbits are available, do not consider windows that do not utilize South Atlantic Anomaly orbits.
    - Do not create windows that gratuitously straddle short term schedule boundaries, or that create inflexible link scheduling situations.

SPIKE software mechanisms and requirements were concentrated on getting the system to generate plausible solutions and not on how to evaluate plans as a whole. As such, no requirements or software mechanisms were created that evaluate long-range plans as a whole. Creating plans that are easy to execute in short term scheduling is a hard requirement and necessitates the use of mechanisms like filters and critics that can sculpt windows. In contrast, mechanisms to balance resources by preferentially planning in the South Atlantic Anomaly are heuristic at best. During a cycle ingest, operations staff produce multiple plans by adjusting parameters and weights to these software mechanisms. The operations staff then compare and contrast the plans through displays of resource usage, plan reports, and runs of a resource validation process where SPIKE schedules observations to orbits. The plan selected for execution is chosen by the combined intuitions of the long range planning operations staff. The manual process of creating multiple plans and evaluating the plans takes on the order of 3-4 weeks. The long term goal of writing new requirements is to enable new software tools that will reduce the manual effort required to create cycle ingest plan.

In summary, the solution to the mystery of the missing requirements is that they are located in the tools and procedures used by the operations staff to manually evaluate plans.

The New Requirements

In the fall of 2012 a team of spike developers and users was put together to determine planning requirements for HST long range planning. Although the main bulk of the work focused on optimization requirements, a series of requirement levels were defined that address core legality requirements for plans.

- Level 0
    - Conditions for which observations should get plan windows assigned.
- Level 1
    - Conditions for legal plan windows
        - Subset of constraint window
        - 56 day nominal duration – plan windows are 56 days or less in duration.
        - Includes spacecraft roll for orient linked observations
- Level 2
    - Conditions which are not illegal but we never want to produce windows in these situations
        - Conditions to ease short term scheduling as described above such as link flexibility and avoiding short term schedule boundaries.
- Level 3
    - Preferences for plan windows
        - Preferences for individual programs and observations
        - Resource balancing

The idea is that levels 0-2 express hard constraints that must be true while level 3 contains the missing soft optimization requirements.

Level 3 requirements: Preferences for individual programs

All requirements are minimization criteria.

Proposal packing. It is preferable to plan observations from a single science program as close in time as possible to one another. The system first determines the duration of spread of the observations in each program. The criterion measures the number of programs that have a spread larger than a user specified amount. The measure subtracts out programs that are forced to be further apart due to timing links.

Plan window size. It is preferable to plan observations into larger plan windows. A user parameter specifies the nominal plan window size which gives the size of the longest possible window (a hard constraint). The criterion measures the sum over all planned observations of the difference of the nominal plan window size and the actual plan window size.

Opportunistic Continuous Viewing. Certain target geometries allow a target to be viewed by HST for an entire orbit with no earth occultation. This situation is highly efficient as no target acquisition overhead has to occur for each orbit of the observation. This criterion measures the number of observations that have opportunistic continuous viewing constraint windows but do not have opportunistic continuous viewing in their plan windows.

Orbital Resource Preferences

Balancing orbital resource usage is the most complex part of HST long range planning. There are approximately fifteen 96 minute orbits in a HST day. Based on the distribution of selected targets within a cycle some time periods may be in higher demand than others. Also, about nine out of the fifteen orbits in each day cross the South Atlantic Anomaly (SAA) a region off the coast of South America, as shown in Figure 1, that has unusually high radiation. In each of these orbits, the SAA passage occurs in a slightly different portion of the orbit. These orbits are called SAA impacted orbits. No observations can occur during an SAA passage. However, we can schedule observations in SAA impacted orbits, if the Earth occultation for the observation occurs during the SAA passage. This case is called SAA hiding. Orbits without any SAA crossing are called SAA free orbits. For a recent HST observing cycle 60% of the approved observations can be scheduled only in SAA-free orbits, whereas only 33.3% of the orbits are SAA-free. For any given target, SAA hiding occurs only for a small fraction of a year.

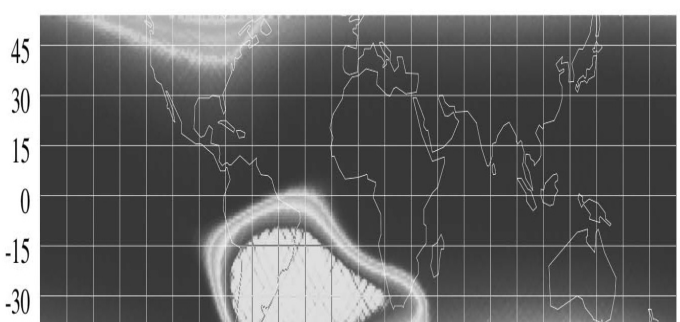

Figure 1. South Atlantic Anomaly (SAA) is shown in lighter shade

SPIKE provides the ability to specify a profile of desired resource usage over the course of the planning cycle. Generally, end users set this profile so that it is higher at the start of a cycle and decreases as the cycle progresses with the goal of filling up the early portions of the plan. The question is how should SPIKE attempt to fill these resources? What is the best plan with respect to these resource levels? Typically the pool of available observations to schedule contains less orbits than the number of orbits under the curve of the resource profile. So a perfect fit of the resource profile is not possible. Should resources be front loaded, distributed evenly, or do we just care that there are no resource overages? The answer the team determined is that resources require a set of three criteria that measure different aspects of resource consumption.

Resource overages. It is preferable to create a plan where no orbit resource is oversubscribed. This criterion measures the square of the resource overage for each day in the planning session. Using the square of the overage makes a single large resource overage on a day worse than have multiple small resource overages spread over multiple days.

Uniform resources. It is preferable to have resources uniformly distributed across the planning session in proportion to the specified resource profile. This criterion measures the sum over all days in the planning session of the square of the difference between the desired and actual resource levels.

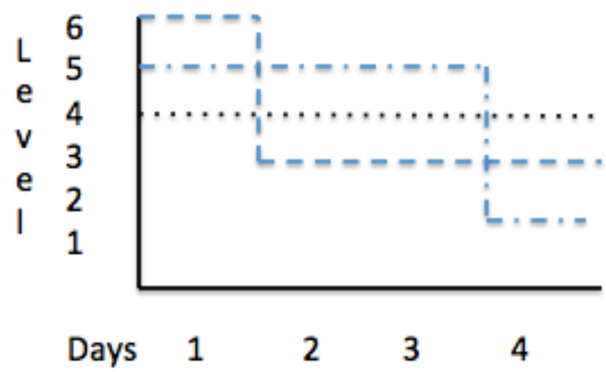

Figure 2. Resource criteria examples

Figure 2 gives examples that distinguish between the overages and uniform distribution resource models. The figure shows resource usage in orbits over a four day interval. The dotted line gives the desired resource limit of four orbits. The other two lines show the profiles of two different plans. Both plans consume 16 orbits of resources. The two plans will have the same score of $12 = (3^2 + 3)$ with respect to the uniform distribution criterion. However, the dashed plan will have a value of 9 for the resource overages criteria where the dash-dot line will have a value of three. Having a separate resource overage criteria models the fact that resource overages are worse than resource under runs.

Hard to fill orbits. It is preferable to plan observations in orbits that are hard to fill. Given the constraint windows for observations within a cycle, SPIKE can build a profile of how many observations fit in each orbit. Typically about 700-1000 observations have satisfiable constraint windows in a given non-SAA impacted orbit. In contrast, an SAA impacted orbit will allow less than 100 observations to schedule. Certain SAA impacted orbits may allow only on the order of 10 observations. This criterion measures the number of hard to fill orbits that are not filled up.

Other Resource Preferences

Data Volume. Each observation requires a specified amount of data volume depending on the instrument being used and the duration of the exposures in the observation. It is preferable to keep the data volume available to dump for a day under a user specified limit. This criterion measures the summed amount of data volume over the limit during the planning cycle.

Large observations. Hubble observations require from 1 to 5 integral HST orbits for scheduling (longer observations are disallowed). It is preferable to plan observations that have more orbits than a user specified limit (currently 3 orbits) planned at different times as having too many of these observations planned for any one day will create a bin packing problem. This criterion measures the number of days where there are more than a user specified number of long observations planned.

Off-nominal. For each observation and schedulable day there is a nominal spacecraft roll that maximizes the positions of the HST solar arrays. It is preferable to schedule as many observations as possible at their nominal angle. However, physical constraints such as guide stars, user constraints such as preferred orient ranges, and assigned roll angles can prevent an observation from scheduling at nominal on any given day. For each day in an observation's plan window we determine how far off-nominal the observation is forced to schedule. This criterion measures the sum of this value over all observations.

Expiring orbits. A long-range plan consists of overlapping plan windows observations. It is preferable that plan windows expire evenly across the planning cycle. Having too many windows expiring in a given week may limit the flexibility of the short-term schedule and/or cause observations to fall out of their plan windows. This criterion measures the number of expiring orbits in each short term schedule interval that exceeds a user specified limit.

A Multi-Objective Approach

A multi-objective component was added to SPIKE as part of work on the MUSE system [Giuliano and Johnston, 2010]. Multi-objective algorithms for planning and scheduling offer many advantages over the more conventional single objective approach. By keeping user objectives separate instead of combined, more information is available to the end user to make trade-offs between competing objectives. Unlike single objective algorithms, which produce a single solution, multi-objective algorithms produce a set of solutions, called a Pareto surface, where no solution is strictly dominated by another solution for all objectives. The idea is to present the user with a set of tools that allow them to explore a surface of solutions to find the one that best meets operational needs [Giuliano and Johnston 2010, Giuliano and Johnston, 2011].

In current operations SPIKE users generate multiple candidate long range plans during the cycle ingest by running SPIKE multiple times with different parameters. The users then use existing reports to decide which plan is better. The long term goal of the work described in this paper is to automate the process of creating candidate plans and to formalize the process of evaluating plans. However, SPIKE is a decision support tool and in the end it will be the users responsibility to select a plan for execution.

An advantage of a multi-objective approach is that the individual criteria do not need to be normalized so that they can be combined. Certainly, users need to understand the scale of each criteria and what each measurement means. However, the costly and time consuming exercise of normalizing criteria values is not needed.

Ongoing Work: Validating the Requirements

All of the above planning criteria have been implemented in SPIKE and are currently being tested. The plan is to have operations staff validate that the criteria match their intuitions obtained through manually examining plans. Operations staff will use existing SPIKE control parameters to generate multiple cycle ingest plans for a scenario constructed using data from the most recent HST planning cycle. The plans will then be compared using both the manual procedure and the new planning criteria. Based on the results of the comparison the criteria will be modified as required.

After validating the criteria the plan is to explore new approaches to long range planning. With the ability to measure the quality of long range plans, the SPIKE team is free to try multiple approaches to planning and to select the best approach. Some possibilities being considered are:

- Using an evolutionary algorithm as described in the MUSE system.
- Build a long range plan from short term schedules created using the SPIKE short term scheduling engine.
- Updating existing plan window generation algorithms.

In addition to creating criteria to evaluate long range plans the working group determined metrics that can be collected over the course of years that directly measure the high level SPIKE goals of efficient telescope execution, stable plans for astronomers, and minimizing work by STScI staff.

Conclusions

Sometimes the true underlying goal of a system is lost in the effort to create operational software that meets user needs. Software requirements and mechanisms are implemented that work towards the perceived operational goal but the real goal of the system is not made clear. This state of affairs is often okay but leads to problems when the system needs to be modified or re-engineered as it is hard to judge whether the new system works as well as the original. This paper described how this scenario applied to optimization requirements for SPIKE long range planning requirements and presents the new requirements that were developed.

'Nothing clears up a case so much as stating it to another person.'

Sherlock Holmes-*Silver Blaze*

Acknowledgements

Thanks to Rob Hawkins, Bill Workman, and Reiko Rager for giving the paper a careful reading. Thanks to the members of the requirements working group Ian Jordan, Bob Boyer, Nazma Ferdous, and Andrew Myers for their work in coming up with a new set of requirements. Finally thanks to the IWPSS reviewers for their relevant suggestions.

References

Giuliano, M.E., and Johnston, M.D. 2010. Visualization Tools For Multi-Objective Algorithms. Demonstration: *International Conference on Automated Planning and Scheduling (ICAPS)*, Toronto, Canada.

Giuliano, M.E., and Johnston, M.D. 2011. Developer Tools for Evaluating Multi-Objective Algorithms. *Interatnional Workshop on Planning and Scheduling for Space (IWPSS).* Darmstadt, Germany

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

Giuliano, M.E., 2008. Handling Oversubscribed Orbital Resources in Hubble Space Telescope Operations. *Workshop on Oversubscribed Planning and Scheduling, at the International Conference on Automated Planning and Scheduling (ICAPS)*, Sydney Australia.