Generating a Long Range Plan for a New Class of Astronomical Observatories

Laurence A. Kramer

Space Telescope Science Institute (STScI)
3700 San Martin Drive
Baltimore, MD 21218
kramer@stsci.edu

Abstract

We present a long range planning (LRP) system, the Spike Plan Window Scheduler, which has been in use for observations on the Hubble Space Telescope (HST) for the past four years and which is being adapted for the Space Infrared Telescope Facility (SIRTF) and Next Generation Space Telescope (NGST) orbiting astronomical observatories. Due to the relatively underconstrained nature of this domain, generating a long range plan is not handled in the traditional AI planning sense of generating operators to achieve goals. Rather, producing an LRP is treated as a type of scheduling problem where what is being scheduled are not the scientific observations themselves, but "plan windows" for the scientific observations. This paper investigates planning subproblems which arise in this type of domain. In particular, we discuss the SIRTF Long Range Plan which requires planning of "instrument campaigns" in conjunction with observation plan window scheduling.

Introduction

Automated planning systems for planetary missions are faced with a problem involving months of relative inactivity followed by a few hours of intense activity that must be planned with great precision. In contrast, orbiting astronomical observatories have a much more constant rate of activity, but the bulk of the scientific activities are relatively underconstrained as to when they can schedule and how they should schedule with respect to each other. Scientific observations for the new observatories SIRTF and NGST will be even less constrained than for HST in that these observatories will not be placed in a low Earth orbit and will thus have less restricted observing windows. For these observatories, though, there will continue to be a premium on achieving high observatory efficiency while addressing planning and scheduling problems peculiar to each observatory.

In this paper we describe the Spike Plan Window Scheduler [Giuliano, 1997] which is used to generate a Long Range Plan for the Hubble Space Telescope. We have confronted a number of interesting challenges in modifying the Plan Window Scheduler for the SIRTF observatory. Addressing these challenges has shown that while long range planning can be treated as a scheduling problem, there are still planning issues to tackle during the process of scheduling.

The Domain

The advent of orbiting astronomical observatories has generated a need for software systems to automate planning and scheduling of activities on those observatories. For the purposes of this paper we mainly will consider activities that are of interest to a long range planner. These are the actual scientific observations, which can be thought of as the "pictures" that the telescope takes (in fact, these "pictures" could be spectrograph readings or images in non-visible light spectra).

For the most part, these observations are of targets that are fixed with respect to our solar system -- galaxies, stars, even visibly blank areas of the sky that may prove interesting. Now, if an astronomer wants to take a picture of the Eagle Nebula (M16), it doesn't matter too much whether she takes the picture today, tomorrow, or next week. The target's not going anywhere. If the observatory, like HST, is in a relatively close in orbit of the Earth, it is not accurate to say that she can take the picture of the Eagle Nebula at *any* time. Most instruments on the telescope are sensitive to too much light, so there are certain observing times when the Eagle Nebula is too close to the Sun, or the Moon, or the bright Earth itself to take an observation. In addition, there are many other valid scientific reasons why an observation can't be performed during certain times with certain instruments. For fixed targets, though, the conjunction of all these constraints on observing leaves one with a (possibly discontinuous) observation window which is still on the order of days or weeks in duration. We refer to these feasible observation windows as **constraint windows**.

Often, two or more observations have constraints placed upon them linking their relative execution times. For instance we might specify several observations of a Cephid variable star to be taken one week after each other. Even these types of constraints do not generally alter the duration very much of the *initial* constraint windows for the individual observations. We emphasize *initial*, since as observations are fixed to absolute times on a short term schedule, their linked successor observations can see their constraint windows collapse radically through constraint propagation.

This issue is treated in great depth in [Kramer and Giuliano, 1997] and we will investigate some consequences for long range planning later in this paper.

There is another significant, though numerically much smaller, class of observations for HST and other observatories. This is the class of "moving" targets. Generally, these are objects in our solar system like planets or moons and objects that enter the solar system on a periodic basis (e.g. comets). These observations may have a number of small, choppy constraint windows throughout the year, or a singular small window for even rarer events such as a view of Saturn's rings edge-on. In addition, a small percentage of viewing time for orbiting observatories is allocated to events that cannot be predicted in advance. These "targets of opportunity" include such phenomena as supernova.

Several newer observatories such as SIRTF and NGST are being designed to orbit farther away from the Earth. This will have the effect of increasing the duration of baseline constraint windows for observations from the days to weeks range experienced with HST to a range of weeks to months. And while it will certainly be the case that many *short term* activities for these observatories will be much easier to schedule, there will still be a number of the same issues -- moving targets, targets of opportunity, and contingent observations -- which complicates long range planning.

Although the bulk of observations for HST and other orbiting observatories are of the relatively underconstrained variety, there are a number of reasons why constraint windows, while of reasonable duration, do not remain completely stable over the course of a long range planning period (typically, a year to year-and-a-half). That is, the constraint windows will remain fairly stable in size, but may shift in time. There are a number of reasons for this:

- The ephemeris (position over time) of the observatory is not known accurately more than a few months in advance, causing constraint calculations to change over time.

- Performance of components on the satellite may change or degrade over time, requiring new operating scenarios.

- Observers refine their observing programs over time. [Giuliano, 1997]

Success Criteria for a Long Range Planner for Orbiting Astronomical Observatories

Efficiency. Orbiting astronomical observatories are an expensive investment with a limited life span. It is important that the long range planner be designed so as to maximize the amount and quality of scientific observations performed by adequately informing a short term scheduler when and what to schedule. Also, the long range planner should help optimize the use of finite resources such as propellants and coolants so as to prolong the life of the observatory.

Stability. It is usually the case that principal investigators (PIs) responsible for an observing program use a long range plan's "best promise" of when their observations will schedule to plan their future work: hiring of graduate students, data analysis, and planning of coordinated observa-

tions. It is important that a long range planner produce a plan that remains as stable as possible over an extended time span.

Mutability. As pointed out, there are a number of causes which might affect where an individual observation or set of observations might feasibly be planned. A long range planner should be able to revise the parts of the plan which need to change without adversely affecting the stability or goodness of the long range plan as a whole.

Long Range Planning for HST

Since launch in 1990 the Spike system [Johnston and Miller, 1994] has been used to generate a long range plan for the Hubble Space Telescope. For the first five years of operations the Spike Long Range Planner produced plans which were not successful in meeting any of three criteria listed above. The plans led to low efficiency schedules, were unstable, and were very resistant to incremental change. Why was this? An important reason was that the original planner planned observations to week long bins for input to a short term scheduler. By design these bins had to be well oversubscribed so that there would be an adequate mix for the scheduler. Unfortunately, it is hard to tell what a good mix will be without access to most of the knowledge of the short term scheduler. Months ahead of time this task is virtually impossible. As a result, many weeks' schedules based on input from the Spike LRP were well undersubscribed and some well oversubscribed. Observations that couldn't be scheduled in their target week would need to be replanned, and possibly put off an entire year if highly constrained.

To address these problems the long range planning system for HST was redesigned. Central to this redesign was a rethinking of the HST planning and scheduling operations concept. Based on this new operations concept and maintaining the very sound underpinnings of the Spike system -- the constraint propagator and astronomical objects model, a new long range planner was substituted for the old one.

The Plan Window Operations Concept

The long range planner creates approximate 4-8 week **plan windows** for observations. A plan window is a subset of an observation's constraint windows, and represents a best effort commitment to schedule in the window. Plan windows for different observations can be overlapping. In addition the windows for a single observation can be non-contiguous. [Giuliano, 1997]

By design now the long range plan is slightly undersubscribed, but when the short term schedule creates a time line for a given week it has a large pool of observations from which to choose. A good and efficient scheduling mix can be generated. Many observations in the pool produced by the planner will *not* be chosen for a given week's schedule, but typically these observations will be able to be scheduled in a later week of their plan window. Observations which are in the last week of their plan window are given priority for scheduling on the next short term schedule.

The SPIKE Plan Window Scheduler

The new long range planner which implements the plan windows operations concept is the Spike Plan Window Scheduler. It can be described as

> ...a function which maps a set of input observing programs, search criteria, and a previous long range plan into a new long range plan. On execution of a long range plan the observations are partitioned into those which have [plan] windows assigned in the input LRP and those which do not have input windows. In general the scheduler will assign new windows to those observations which are not scheduled in the input plan and will pass through windows for observations which are scheduled in the input plan. The system assigns windows for observations which are not scheduled in the input plan in two steps. First, the system uses user defined criteria to greedily find the best window for each observation. In the second step a stochastic search is used to adjust the resource levels. [Giuliano, 1997]

Basically the Plan Window Scheduler is treating the planning process as a problem of constrained optimization where it tries to find the best subinterval of an observation's constraint windows to assign as a plan window. The objective function which is evaluated considers a number of criteria specific to an observation as well as criteria which will produce a good plan globally. Each individual active criterion will return a score as a real number between 0 and 1 inclusive. The criteria are weighted individually to express their relative importance to the overall scoring function. New planning criteria can be added to the planner as needed.

An example of a criterion that affects a small group of observations would be: "Attempt to schedule plan windows for observations in the same observing program as closely together as possible." Thus, if OB1 and OB2 are in the same observing program, but are planned a year apart, this criterion will return a score close to 0. If the plan windows for the two observations are identical, the score will be 1. An example of a planning criterion that is global in nature is one which balances resources across the extent of the planning period.

Plan Window Scheduler Architecture

The Spike Plan Window Scheduler is implemented as a CLOS program in Allegro Common Lisp, currently running on Sun/Solaris. There are separate modules for the Planning Criteria, the Resource Model, and the Scheduler. The Scheduler is comprised of one or more Scheduling Steps, which can be independent of each other, and thus added to, deleted, and specialized as necessary. For HST we initially implemented two scheduling steps: FIND-BEST and REPAIR. The FIND-BEST step finds the best plan window for each observation mostly independent of other observations. The REPAIR step is implemented as a specialization of FIND-BEST to level resource imbalances such as expected observation duration per day. REPAIR uses the same code as FIND-BEST, but adds two additional criteria to the scoring function: a randomizing criterion which adds a stochastic element to the search and a resource criterion.

The Plan Window Scheduler is run daily to generate a new Long Range Plan for HST. Each new plan is basically the previous plan plus additions for new observations and alterations for observations that have been changed or fixed on a short-term schedule. Small changes in an observation's constraint windows do not cause replanning of that observation. Its assigned plan windows are intersected with the new constraint windows to produce possibly somewhat smaller plan windows. In some cases a change in constraint windows necessitates no change in plan windows.

A recent modification illustrates the modularity of the Plan Window Scheduler design. As noted above plan windows are adjusted for constraint windows that are slightly smaller or that have shifted away from the plan windows somewhat. One scenario that wasn't being automatically handled was generating new plan windows for an observation whose constraint windows have "relaxed" around an existing tightly restricted plan window. In order to increase scheduling flexibility it was desirable to automatically expand existing plan windows as constraint windows grow. This problem was solved without modifying existing code, but by adding a new EXPAND-PWS step to the scheduler. It reuses code from FIND-BEST, but applies different planning criteria which maintain an existing plan window in place (it may shift in time by a small amount) while expanding it.

As well as being a good model for software maintainability and reuse, the Spike Plan Window Scheduler meets the three success criteria for a long range planner listed above. It generates a plan which leads to **efficient** schedules by assigning plan windows which are optimized for individual observations. The long range plan is **stable** by retaining existing plan windows or modifying them only slightly from plan to plan, and by only replanning those whose constraint windows have altered radically. The long range plan is **mutable** as new observations may be planned or existing ones replanned without affecting the bulk of the plan.

The SIRTF Mission

The SIRTF mission has much in common with that of HST. Like the Hubble, it will be a an orbiting "space telescope" that will mainly concentrate on fixed astronomical targets. It too will allocate a smaller percentage of time to viewing moving targets and targets of opportunity. As does HST, SIRTF operations will routinely give their observers an indication of approximately when their observations should execute while allowing them to modify their observation specifications until fairly shortly before they are actually scheduled to execute. For both missions it is assumed that all planned observations *will* execute (as opposed to ground-based observatories where many observations are simply dropped due to bad weather), so all observations are of basically equal priority. One of the primary goals of the SIRTF mission is science efficiency. The combination of these attributes points to SIRTF as a good mission to reuse the Spike Plan Window Scheduler in producing a stable yet flexible LRP.

The major difference between HST and SIRTF also makes it a good candidate for the Plan Window Scheduler. SIRTF will be placed in an "Earth trailing" (orbiting the Sun, lagging behind the Earth) orbit as opposed to Hubble's

"Low Earth" orbit. Thus, typical observing windows for SIRTF targets will be much less affected by Earth light and Earth occultations and will be much longer in duration. Some SIRTF targets will be observable year-round, with most others observable for months or weeks at a time. Of course, there will be additional constraints placed on many observations that will narrow the observing windows somewhat, but in general baseline windows for SIRTF will be significantly larger than for HST. This underconstrained problem argues for fairly large plan windows to guide a short term scheduler.

There is one major difference between HST and SIRTF, though, that seriously complicates the generation of a long range plan using the plan windows concept. HST has several scientific instruments, most of which can be run in parallel, so that for instance a wide-field image of a target can be executed at the same time as a spectrograph of the same target. This is not the case for SIRTF which will have three science instruments -- IRS, IRAC, MIPS -- which must be run sequentially. In addition, all of its instruments are cryo-cooled and it will cost a reasonable amount of time and some of the fixed supply of cryogen every time there is a change over from one instrument to another. These requirements argue that SIRTF observations be planned to "instrument campaigns" where only observations using a certain instrument be scheduled for days or even weeks at a time. In fact, this is a design requirement for the SIRTF mission. But on the face of it seems to be at odds with the plan windows concept which allows and encourages plan windows for observations to overlap, only optimizing the extent and number that overlap to resource limits.

We have been able to address this concern by introducing the concept of **instrument windows** to implement instrument campaigns. Instrument windows are defined to be discrete, but adjoining, subintervals of the nominal planning interval.The Spike Plan Window Scheduler has been extended to schedule plan windows and instrument windows in concert.

The Spike SIRTF Plan Window/Instrument Window Scheduler

A prototype implementation of the Spike SIRTF Plan Window/Instrument Window Scheduler has been completed and work is in progress to refine the prototype into an operational system before the planned SIRTF launch in December, 2001. A major design goal of this long range planning system is to reuse much of the existing Spike code and class hierarchy. To this date no existing Spike code has been rewritten. All enhancements necessary for SIRTF have been achieved by specializing Spike base classes and methods as well as writing some new code. Objects have been added to model SIRTF's resources and new planning criteria have been added to supplement some which have been reused from HST. New scheduling steps have been added to the Scheduling module, which we will describe in some detail shortly.

The main technical hurdle to be overcome in the design of the SIRTF long range planner was the issue of concurrent scheduling of plan windows and instrument windows. The former are by nature non-exclusive, while the latter must be exclusive. We can schedule multiple, overlapping plan windows "within" an instrument window as long as those plan windows are only for observations which require the same scientific instrument. In fact, we allow plan windows for an observation to schedule in multiple instrument windows as long as those instrument windows are designated for the same instrument.

For example, suppose we have an observation which will use the Infrared Array Camera (IRAC) and which has relatively unrestricted constraint windows. The nominal duration of plan windows for SIRTF will be 40 days to cover the natural viewing cycle for most observations (the majority will be observable for 40 days every six months). Even if there are two-week instrument campaigns for MIPS and IRS intervening between IRAC campaigns, a 40-day plan window for an IRAC observation could easily intersect two IRAC campaigns, while "skipping" the four-week interval between.

We considered two approaches to achieving this scheduling problem:

Approach I, plan windows first.

1. Generate plan windows for all observation first.

2. Assign instrument windows to areas where groupings of plan windows for like instrument windows occur.

3. Repair plan window assignments to segregate assignments by instrument window and to balance resources.

Approach II, instrument windows first.

1. Generate instrument windows to some idealized model based on projected resource usage and other criteria.

2. Assign plan windows to the best instrument window(s) based on scoring of planning criteria.

3. Repair plan window assignments for resource usage.

Given SIRTF observations' generally large constraint windows, we quickly rejected Approach I. There was nothing to drive plan window assignment, and thus just about anything could go anywhere. Segregating by instrument window then becomes a huge problem.

Approach II was chosen, with some refinements. Before discussing these, certain other attributes of SIRTF scheduling should be noted. First of all, it is anticipated that a significant fraction -- maybe 10% -- of observations may be classified as needing **absolute time** scheduling. We consider those observations that must schedule in no more than a one-day window to be an absolute time event. Secondly, there is a preferred ordering of instrument campaigns for SIRTF instruments in order to best utilize resources. The existence of absolute time observations helps enable our refined Approach IIa; instrument campaign ordering makes it more interesting.

Approach IIa, absolute time observations first.

1. Schedule plan windows for all absolute time observations. These plan windows will by definition be small, since the constraint windows are small. As each absolute time

observation is planned, create an instrument window "around it" such that this instrument window is somewhat larger than the plan window, but must not intersect unlike instrument windows. If an instrument window already exists where an observation needs to go, either it will be a window matching the observation's instrument or it will not. If it matches the plan window can be scheduled in that window. If it doesn't match, a conflict is noted to be handled later.

2. Generate instrument windows to fill in gaps between the instrument windows generated in (1) based on projected resource usage, desired instrument window size minima and maxima, and the preferred window ordering criterion.

3. For all non-absolute time observations, sorted by most highly constrained first, assign plan windows to the best instrument window(s) generated in (2) based on scoring of planning criteria. If a like instrument window does not exist where a highly constrained observation needs to be planned, a new instrument window may be generated by shrinking an existing one or an existing instrument window may be extended to accommodate the observation. In any case, instrument windows may not be modified so as to "orphan" an interval that is needed by an absolute time observation. Other, **relative time** observations may be replanned in a different instrument window if necessary. During this process instrument window size and ordering are maintained as much as possible.

4. Repair plan window and instrument window assignments for resource usage and other imbalances not handled in (3).

Approach IIa has been implemented by reusing much code from the HST Plan Window Scheduler and by defining four new Scheduling Steps, corresponding to (1-4) above: ABSOLUTE-TIME, INSTRUMENT-CAMPAIGN, RELATIVE-TIME, and REPAIR. Let's consider the INSTRUMENT-CAMPAIGN in some more detail.

Planning Within a Scheduling Problem

In the INSTRUMENT-CAMPAIGN step we are faced with the following initial conditions: A set of assigned instrument windows with gaps in between them. We would like to achieve a state where all the gaps are filled with new (empty, in the sense that no plan windows are assigned to them yet) instrument windows that are of some optimal size and ordering. Basically we have a problem that is amenable to classical AI planning approaches. The domain is highly constrained and there are clear goals that can be achieved by applying a sequence of operators to a defined world state.

There are a number of approaches that could be considered for implementing the Instrument Campaign Planner, however, most of them are overkill. Consider the gap-filling problem. Do we generate N instrument windows of length M to fill the gap, or X windows of length Y, and so on? For large gaps there are many possible assignments that satisfy the instrument window ordering problem. On the other hand, we are in general trying to generate as large instru-

ment windows as possible within some bounds. This cuts the search space a good deal.

A more important consideration is that at this point in our overall planning and scheduling problem (instrument windows + plan windows for all observations) it isn't important to try to get an optimal but computationally expensive solution, as this solution will likely be undone in the RELATIVE-TIME and REPAIR steps. We need a planner that gives us a quick solution which is *good enough*.

The solution we provide is somewhat similar to that employed in *case-based* planners [Weld, 1994] in that we use preexisting plans to synthesize a new plan. We differ from case-based planning in that our domain is very well defined and constrained so we don't have the problem of matching and possibly modifying plans to use. Instead, we take a brute-force approach of selecting all plans that apply to the preconditions, scoring each one, and choosing the best.

We call these "pre-packaged" plans **window optimization scripts**. Window optimization scripts come in three types: standalone, auxiliary, and compound. A standalone script is one that is atomic. It performs one action that can be scored as a solution to forwarding plan goals. An auxiliary script is atomic as well, but is always combined with other scripts to produce a script which can be scored. A compound script refers to a list of sub-scripts -- standalone, auxiliary, and compound -- which it calls to do its work.

An example of a gap-filling windows optimization script is CREATE-WINDOWS-TO-RIGHT, a compound script whose sub-scripts are CREATE-WINDOW-TO-RIGHT, MAKE-NEXT-WINDOW-CURRENT, CREATE-WINDOWS-TO-RIGHT. During the INSTRUMENT-CAMPAIGN scheduling step, CREATE-WINDOWS-TO-RIGHT is one of a number of scripts which is tested and scored as a plan to fill the gaps between existing instrument windows. At all times one instrument window is designated as the *current* window. CREATE-WINDOWS-TO-RIGHT calls CREATE-WINDOW-TO-RIGHT to generate a new window (to the right of the current). If this action is successful, MAKE-NEXT-WINDOW-CURRENT is called to advance the current window pointer to the newly created window. Next, CREATE-WINDOWS-TO-RIGHT calls itself recursively to continue the process of generating new windows to fill the gap. This plan is scored and if better than the current best is saved as best plan. Its effects are undone and the next applicable plan on the list is tested.

Another example of a windows optimization script is EXTEND-BOTH-TO-MAX-LEFT-FIRST. It fills in a gap by calling EXTEND-WINDOW-TO-MAX-ON-RIGHT, MAKE-NEXT-WINDOW-CURRENT, and EXTEND-WINDOW-TO-MAX-ON-LEFT. This and several other scripts are sufficient to do a good job of filling gaps between instrument windows, as our choices of operators is relatively small. As testing has revealed a few cases that are not solved by existing scripts, several new ones have been added.

Issues and Future Work

While the Plan Window Scheduler approach has proved to be quite successful in generating long range plans for several observatories, some issues keep it from being a fully automated solution. Consider the issue of "Plan Window Conflicts" which is being addressed for the HST LRP.

Inherent in the Plan Window Concept is the notion that plan windows for a number of different observations will overlap a given time period. Typically, plan windows for any given observation will be fairly large (more than one week in duration), and the overlaps tend to be somewhat large as well. It is usually the case, then, that a feasible ordering of observations can be found once a short term schedule is constructed. Many observations can be placed on one week's short term schedule, and those that can't can be placed on a succeeding week's.

When some observations are scheduled, though, occasionally conflicts occur. That is, two or more observations claim the same time slot on a short term schedule and the conflict is irresolvable without manual work including modification of the observation specifications. How do these conflicts occur?

1. Sometimes the conflicts are due to observations which have very tight constraint windows (a few hours) which intersect. While this type of conflict derives from the nature of the observations themselves, it is also the easiest to detect beforehand. After detection, the conflict can be reported so that one or more observations can be altered.

2. Some conflicts occur when observations have plan windows which appear to have a good deal of scheduling flexibility built in, but which in practice do not. The cause of this is that the short term scheduler has knowledge of some constraints which are either not known by the long range planner, or which are only computed by the LRP in a statistical sense. A possible solution to this problem is to provide better communication between the short term scheduler and the long range planner. In theory the LRP could detect those observations that are "prone" to this problem and call the short term scheduler to test schedule them for possible conflicts.

3. The third type of conflict is even more difficult to detect. It occurs when a number of observations have long plan windows, but all of them terminate a short time into a given week. Now suppose each of these observations is not scheduled during the first few weeks of their plan windows. This is a very common occurrence as there will be higher priority observations to schedule whose plan windows do end during a given week and thus *must* be scheduled. If enough of the observations with long plan windows are "ignored" on earlier weeks, they may all end up being "must go" for the present week. But, if they all need to schedule in the first part of the week, there may not be enough time to schedule all of them. One possible solution to this problem is to detect groups of observations whose constraint windows all end at the same time and stagger plan window end times.

The Plan Window Conflict problem is one that affects a small percentage of observations, but it presents some interesting problems which do not seem insoluble within the plan window scheduling paradigm. Given the nature of the observatory domain these problems might be avoided with a solution of just short term scheduling on demand, but the costs of this approach would far outweigh the benefits.

Open questions also remain as to how easy it will be to maintain a stable Instrument Campaign schedule for SIRTF in a operational environment with observations changing over time, new observations being planned, and unforeseen events occurring. More work remains to be done on the REPAIR phase of the planner. We will get a better idea of how the Plan Window/Instrument Window Scheduler will perform as we conduct more realistic simulations, however some modifications will be inevitable during the life of the mission. In any case, the Plan Window concept has been shown to provide a good deal of flexibility -- both for scheduling observations, and for handling different planning challenges.

Summary

We have looked at a domain -- long range planning for orbiting observatories -- where planning may be best handled by methods other than the classical goals/operators paradigm of AI. This is still a planning problem in a more generic sense of creating a template so that activities can later be sequenced efficiently and resources managed effectively over time.

The plan window scheduling methodology we describe will continue to have applicability for observatory long range planning, and should be considered for other domains where similar attributes are found: most activities are loosely constrained and coupled, need to be roughly planned well in advance, but may not be completely well defined until they are ready to schedule.

References

Giuliano, Mark 1997. *Achieving Stable Observing Schedules in an Unstable World.* ADASS '97, Sonthofen, Germany.

Johnston M.; Miller G. 1994. *Spike: Intelligent Scheduling of Hubble Space Telescope Observations.* in Zweben M., and Fox M. eds. *Intelligent Scheduling* (San Francisco: Morgan-Kaufmann), ISBN 1-55860-260-7 (1994), pp 391-422.

Kramer L.; Giuliano M. 1997. *Reasoning About and Scheduling Linked HST Observations with Spike.* In the Workshop Notes of the International Workshop on Planning and Scheduling for Space Exploration and Science. Oxnard, CA.

Weld, Daniel 1994. An Introduction to Least Commitment Planning. *AI Magazine* 15(4):27-61.