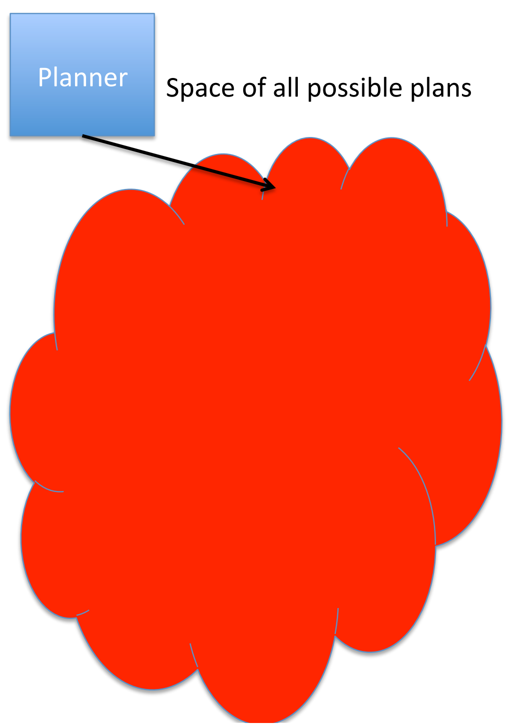
The Mystery of the Missing Requirements: Optimization Metrics for SPIKE Long Range Planning

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

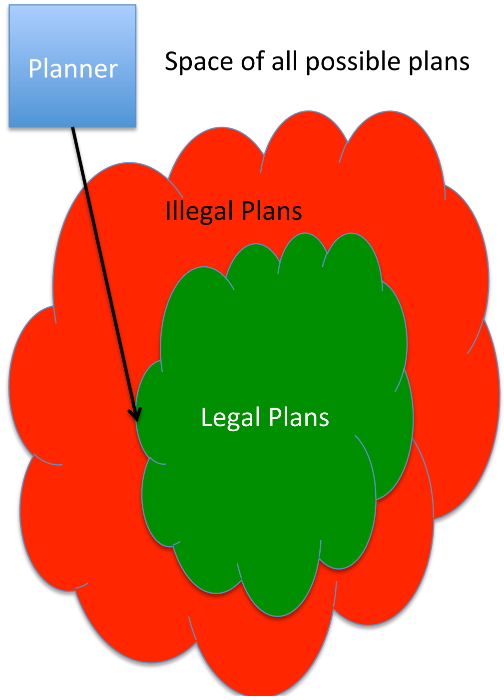
A Common Scenario?

- Have a planner that can produce plans in a domain



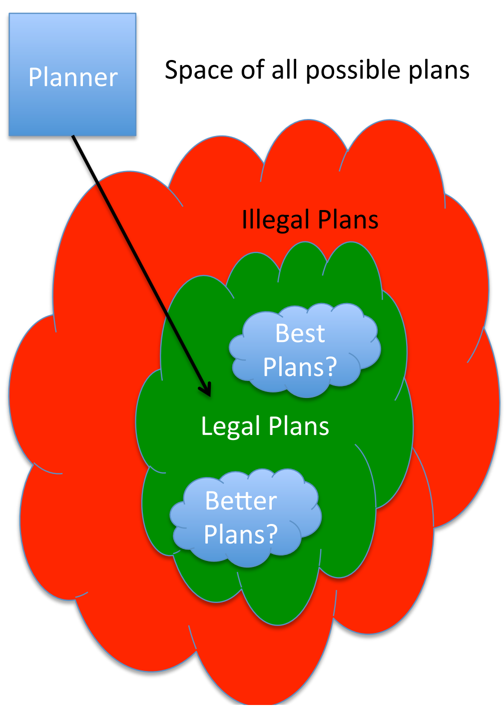
A Common Scenario?

- Planner has requirements on what makes a plan legal
  - Only produces legal plans



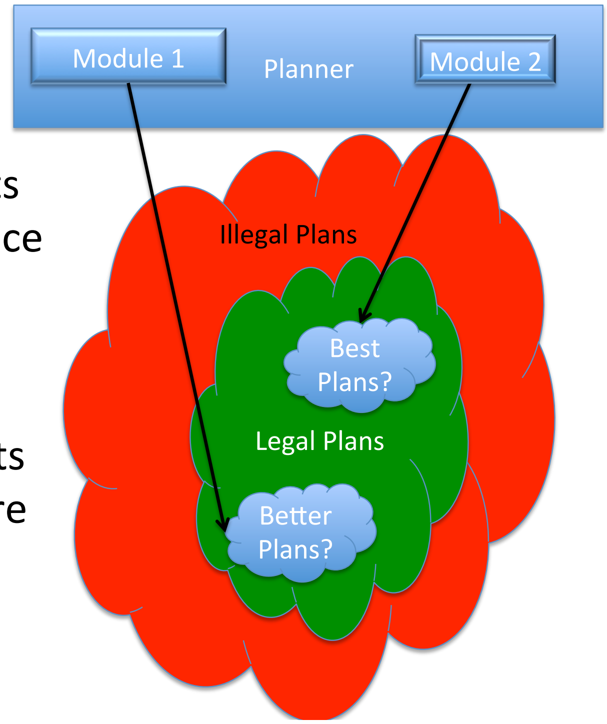
A Common Scenario?

- The space of legal plans is large
- Only have heuristics as to what makes one legal plan better than another



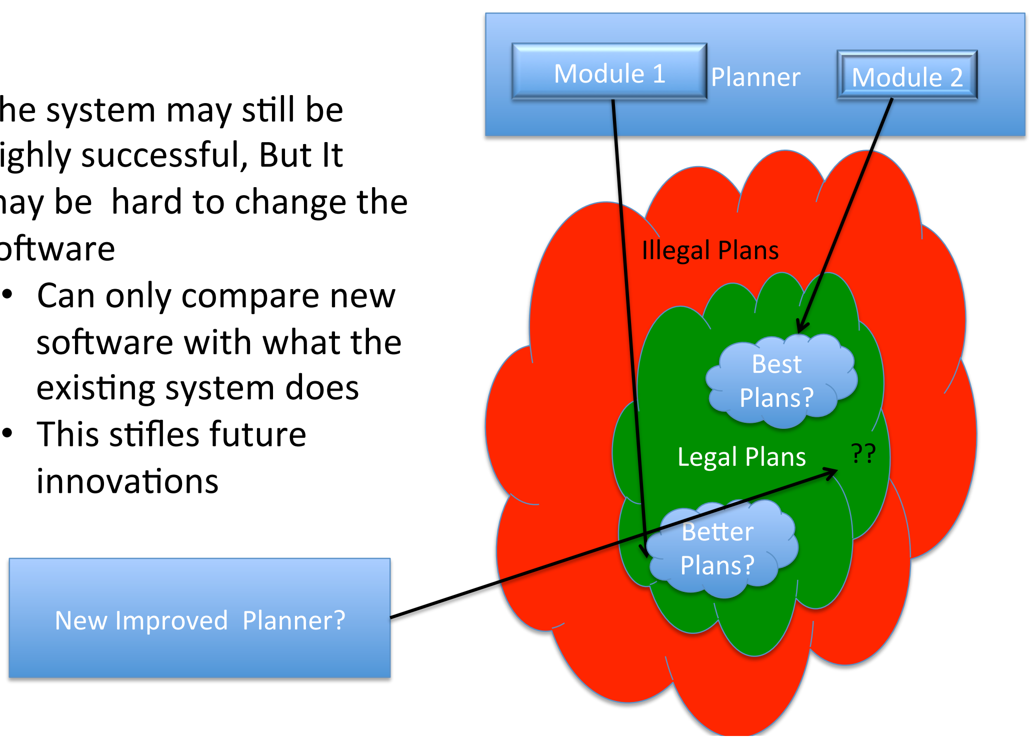
A Common Scenario?

- Procedural heuristics are coded in modules to:
  - Explore the good parts of the legal search space
  - Avoid bad parts of the legal search space
- The actual planning optimization requirements are implicit in the software mechanisms



A Common Scenario?

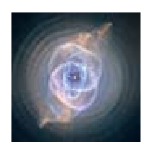
- The system may still be highly successful, But It may be hard to change the software
  - Can only compare new software with what the existing system does
  - This stifles future innovations



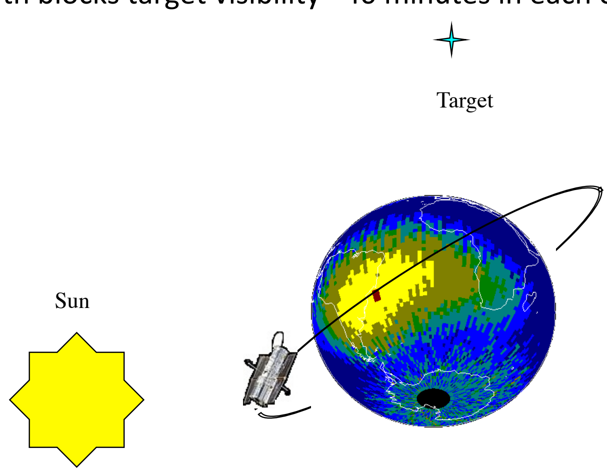
Agenda for Today's Presentation

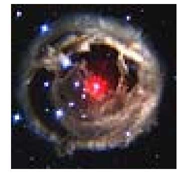
- **How the requirements became missing**
  - **HST Mission**
  - **Planning and scheduling HST**
  - **Missing Requirements**
- New Requirements for SPIKE ingest
- The plan forward and Lessons Learned

HST Mission



- HST is a general purpose space observatory
  - Near-infrared, visible, and ultraviolet observing
- HST orbits the earth every 96 minutes = 15 orbits per day
  - The earth blocks target visibility ~40 minutes in each orbit





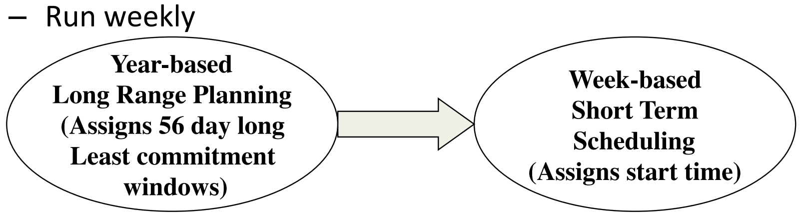
HST Observing Cycle

- HST observations are executed in a yearly cycle
  - Astronomers submit proposals to STScI to use HST
  - Time Allocation Committee approves time to proposals in integral orbits based on scientific merit
  - Astronomers prepare detailed observation program
  - In house staff plan and schedule observations
    - *Ingest all new proposals for the cycle in a Long range Plan*
      - Maintain the long range plan
    - Create short term schedules from the long range plan
  - Astronomers analyze data and publish results

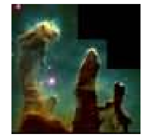
HST Planning and Scheduling

Two Phased Approach

- **Long range planning**
  - Assigns observations for a cycle to 56 day long least commitment plan windows.
  - Concerned with ***resource balancing***, plan stability
  - Baselined once a year maintained daily
- Short term scheduling
  - Creates week long second-by-second schedules using plan windows as input.
  - Concerned with schedule efficiency
  - Run weekly

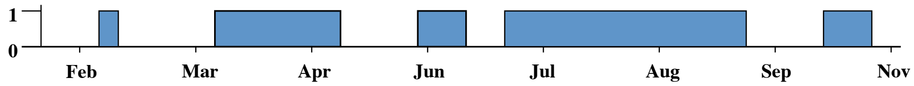


Long Range Planning



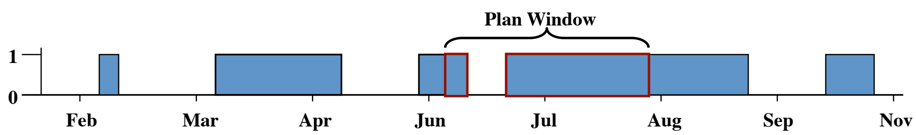
1. **Calculating Constraint Window**

   Observation constraint windows are calculated from all physical and observer specified constraints, and denote the timeline of when the observation can be scheduled.



2. **Generating Plan Windows (PW)**

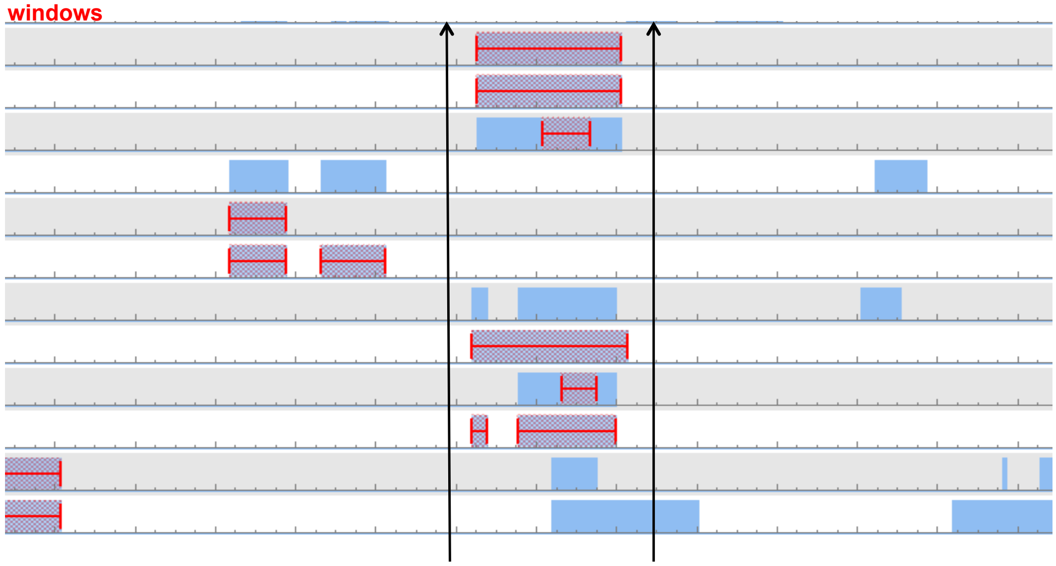
   Using least commitment scheduler, SPIKE, observations are assigned plan windows, which are the preferred window for scheduling.



   Plan windows are a subset of the constraint windows and are nominally 56 days long.

Plan Windows

The red bars give plan windows

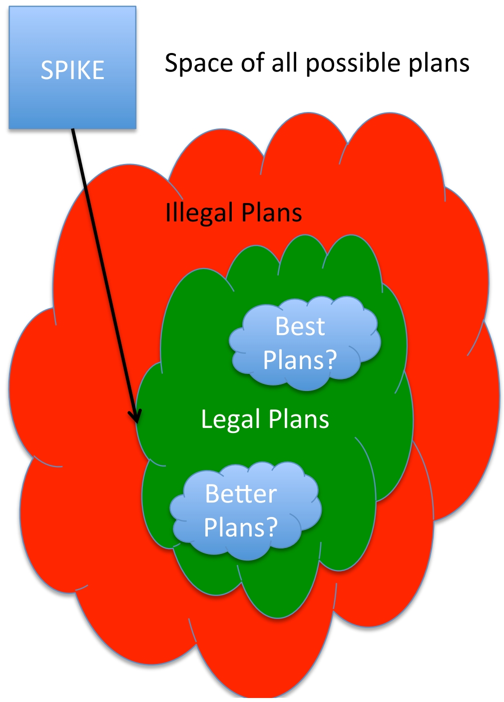


The short term scheduler uses unexecuted observations with open plan windows to create its candidate list for a weekly schedule

Missing Optimization Requirements

- SPIKE has strong requirements on what makes a legal plan
  - Input/output
  - Constraint checking
- Even though SPIKE has been creating long range plans for close to 20 years there are no formal metrics for plan quality

  Much knowledge about quality is in the heads of users



We Have High Level Goals

- The overall LRP process should:
  - **Enable efficient execution of the telescope**
    - **Allow the short term scheduler to create efficient second by second schedules**
  - Be fair to end user astronomers:
    - Provide stable plan windows so they know when to hire graduate students for data reduction
    - Group proposal observations together
  - Minimize the effort for in house teams to prepare, plan, and schedule proposals.

Cannot Directly Measure LRP Quality

- Ideally we could measure LRP quality by simulating the LRP short term scheduling process
  - Create multiple LRPs for a cycle
  - For each LRP create successive short term schedules
  - Measure
    - The spacecraft efficiency of the schedules
    - The stability of the produced plan windows
- In practice this is not possible:
  - Short term scheduling is a highly manual process
  - Cannot produce meaningful short term schedules in advance as we do not know the space craft ephemeris

Deconstructing a Search

- Search algorithms can be reduced to the formula
  - *Algorithm = Generate * Test*
  - Where the generation and test can be combined in many ways
- The two aspects of the formula correspond to the following questions:
  - Generation: How do you explore the space of legal plans in order to get good plans?
  - Test: How do you determine if one plan is better than another?

Why the requirements went missing

- Initial software development concentrated on creating software mechanisms for generating solutions and not testing solutions
  - Search space for plan windows is astronomically large
    - Every possible subset of constraint windows spanning a 56 day interval
  - Needed to have software mechanisms to generate good plan windows
    - Cannot just test those randomly generated
- In 1995 at the start of plan window operations did not have the operational intuitions to know what makes a good versus a bad LRP

Agenda for Today's Presentation

- How the requirements became missing
- **New Requirements for SPIKE ingest**
  - **Motivation**
  - **Requirement levels**
  - **Preference requirements**
  - **Multi-Objective Perspective**
- The plan forward and Lessons Learned

Motivation

- The long term motivation of this work is to
  - Incorporate manual work by LRP operations into SPIKE
    - As part of the cycle ingest the LRP team manually creates and evaluate multiple plans
    - Work will
      - Allow the coding of new schedulers that automatically create multiple plans
      - Provide plan evaluations
  - Increase the quality of long range plans
    - Better understanding of what quality means

Requirement Levels

- Divide requirements into different levels
  - Level 0 = Conditions for which observations should get plan windows assigned
  - Level 1 = Conditions for legal plan windows
  - Level 2 = Conditions which are not illegal but we never want to produce windows in these situations
  - Level 3 = Preferences for plan windows
- Levels 0-2 are hard constraints while
- Level 3 represents soft preferences
  - This is what we are trying to capture

Hard Constraints - Examples

- Level 0 = Conditions for which observations should get plan windows assigned
  - E.g. assign windows for observations which are marked ready in a data base
- Level 1 = Conditions for legal plan windows
  - E.g. must be subset of constraint windows
- Level 2 = Conditions which are not illegal but we never want to produce windows in these situations
  - E.g. Unless it cannot be avoided don't create plan windows which extend into a subsequent short term schedule for just a few hours

Level 3 Plan Criteria

- These criteria evaluate a plan as a whole with respect to some feature
- Current mechanism supports minimization criteria
  - i.e. criteria where we prefer a small measure
  - E.g. prefer to minimize unplanned orbits
  - Shows what we ***prefer*** and how we ***measure*** the preference

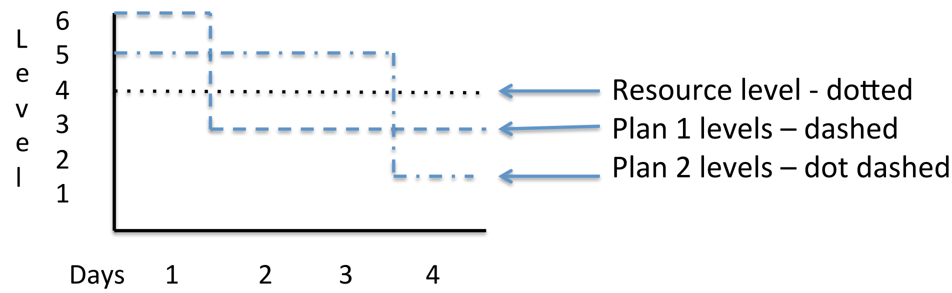
Plan Criteria Summary

- Implemented as plan criteria:
  - *Proposal packing, Uniform orbit resource distribution, Avoiding resource violations*, Data Volume, Off nominal, Window Size, Large observations, Expiring orbits, Great Attractor, SAA hiding, Link set flexibility, minimize orbit adjustments
- Implemented as hard constraints:
  - SAA window sculpting, Slivers and SMS boundaries, North point SAA
- Not implemented:
  - Opportunistic CVZ, Constraint tolerances
- A few example criteria will now be given

Preference: Proposal Packing

- **Prefer** to schedule observations in a proposal close to each other in time
  - Completing a program makes the astronomer happy
  - User defined parameter specifies a proposal plan window spread threshold
- **Measure** the number of proposals with observations planned apart greater than the spread threshold
    - Don't include proposals that are forced farther apart due to timing links

Resource level Criteria - Example



- Example above show how two plans consume a resource
- Both plans consume 16 orbits
- How should our criteria distinguish between these two plans?
- Defined two separate criteria

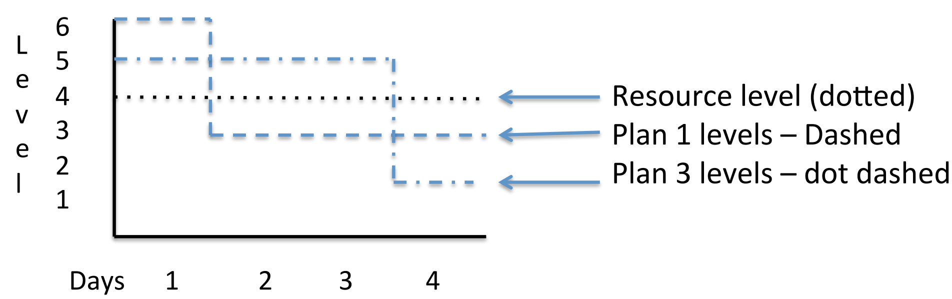
Uniform Orbit Resource Distribution

- **Prefer** plans with a uniform distribution of resources
- SPIKE tracks resource usage for all orbits in a cycle
  - Each resource has a user specified desired resource level
- Departure from the desired resource level is bad either for over subscription or under subscription
- **Measure** the deviation from the expected level
  - For a user specified set of resources
  - Use the square of the deviation

Avoid Resource Violations

- **Prefer** plans without resource oversubscription
- **Measure** the amount of of oversubscription from the user specified desired levels for a user specified set of resources
  - Sum the square of the oversubscription

Resource level Criteria - Example



- Both plans have a score of 12 for uniform orbit distribution = ($3^2 + 3$)
- Dashed plan has value 9 for resource overages while the other plan has value 3

Multi-Objective Approach

- Track criteria values separately during planning
  - Do not combine
- Build a Pareto surface of potential schedules
  - No member of the surface is strictly worse for all criteria than any other member
- Provide end users with tools to select a solution for execution
  - **Decision as to what schedule to use still lies with end user**
  - Tools provide decision support for current manual tasks
- An advantage is that normalizing criteria values is not critical
  - Users need to understand the values but the software does not need to formalize the comparison

Agenda for Today's Presentation

- How the requirements became missing
- New Requirements for SPIKE ingest
- **The plan forward and Lessons Learned**
  - **Evaluating the requirements**
  - **Next steps**
  - **Lessons Learned**

Next Steps: Evaluating the Plan Criteria

- All of the implemented criteria were delivered to operations 2/1/2013
- During the winter and spring of 2013 the LRP operations team will evaluate the plan criteria to validate the correlation between criteria scores and LRP quality
  - Generate multiple LRPs using existing switches in SPIKE data (e.g. changing criteria weights)
  - Compare criteria scores with validation results and raw intuitions of LRP quality
- Feedback to development team on tweaks to plan criteria and new plan criteria

Next Steps: Future Cycles

- Investigate new scheduling approaches
  - The core of a genetic algorithm scheduler for SPIKE has been constructed
    - Cache all interesting assignments for each link set in a preprocessing step
    - Use a genetic algorithm to explore the space of assignments
- Automate manual steps performed by operations
  - Generating multiple-candidate long range plans
  - Evaluating long range plans to select the best plan for operations
- Feedback from JWST
  - A JWST end-to-end scheduling effort will evaluate LRP quality.
    - This process will feedback into HST

Lessons Learned

- I do not believe it would have been possible to have defined plan criteria at the start of plan window operations in 1995
  - No experience using the operations concept
  - Need to be able to generate good candidate solutions
- But I think we could have defined these 5-10 years ago

IWPSS Song?

- Here is a link to a song about sequestration
- https://www.youtube.com/watch?v=oMzF852y70c