Modification of an Existing Planning and Scheduling System to a New Mission

Carl A. Johnson

Space Telescope Science Institute
3700 San Martin Drive
Baltimore, Maryland 21218
Cjohnson@stsci.edu

Abstract

The Space Telescope Science Institute (STScI) is currently participating in the development of the planning and scheduling system for the Far Ultraviolet Spectroscopic Explorer (FUSE) Mission at Johns Hopkins University (JHU). FUSE, one of the first of the Explorer class missions, appears to be a model for how future Explorer missions will rely on organizations with experience in developing planning and scheduling systems to offset their limited budgets, resources and development time. The FUSE project is an excellent example in demonstrating the process by which portions of an existing planning and scheduling system are modified to meet the different operational requirements and interfaces of a new spacecraft and instrument. This paper examines the experience gained from participating in such a process. It highlights some of the elements of the science mission planning process, addresses some lessons learned to date and describes areas where planning and scheduling software needs to be flexible.

Background

The Far Ultraviolet Spectroscopic Explorer is a PI-class NASA astronomy mission that will explore the Universe through high-resolution (lambda/delta lambda = 24,000-30,000) spectroscopy at far ultraviolet wavelengths (905-1195Å. FUSE is scheduled as a three-year mission within the NASA Origins program.

The FUSE satellite is composed of the spacecraft and the scientific instrument. The instrument consists of four co-aligned telescope mirrors (~ 39 cm x 35 cm clear aperture). The light from the four optical channels is dispersed by four spherical, aberration-corrected holographic diffraction gratings, and recorded by two delay-line micro-channel plate detectors. Two channels with SiC coatings cover the range 905-1100Å and two channels with LiF coatings cover the range 1000-1195Å. A Delta II vehicle will launch FUSE into an 800 km, 25-degree inclination orbit, from Cape Canaveral in the September of 1998.

The mission has both a Principal Investigator (PI) and Guest Investigator program with a PI observing time allocation of 60%, 40% and 25% respectively for the first three years. The GI submission process is a two- phase approach. In phase one, GIs provide preliminary targets and scientific justification suitable for an allocation committee to select programs. In phase two, GIs provide the targets and observation information needed by the science planning team for scheduling of the observations. The PI submission process only requires the above mentioned phase two portion with the PI teams handling their own target allocations. A representative science program developed from the science objectives contains approximately 1,400 observations of some 1,000 objects with cumulative on-target exposure times ranging from several minutes to more than 50 hours. Based on its duration, an observation may be broken into smaller observations to provide scheduling flexibility.

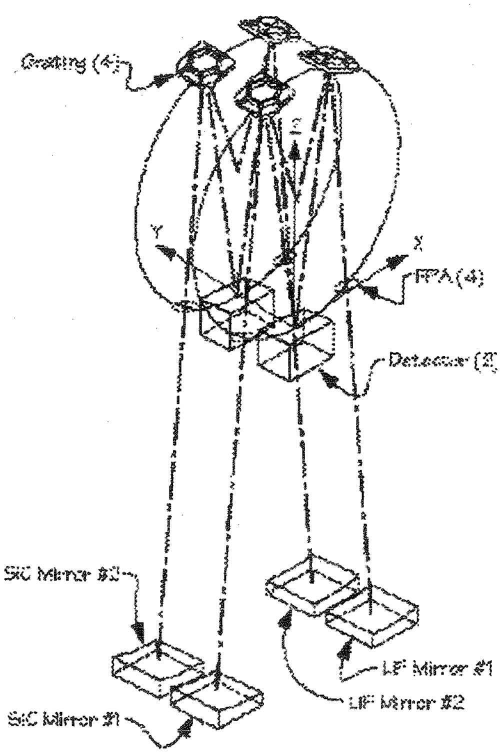

Figure 1. FUSE Instrument Diagram

Among other elements in its contract, STScI was selected to provide system engineering support and planning and scheduling software for FUSE. The system engineering responsibilities include the definition of any new scheduling requirements, refinement of the operations plan, and support of other science operations requirements definition and development tasks. The planning and scheduling software to be delivered from STScI is a version of Spike that takes into account FUSE mission scheduling requirements.

Spike is a general system for planning and scheduling developed at STScI under contract with NASA. Spike has a full set of features to support planning and scheduling that includes a powerful yet efficient method (suitability functions) of representing a wide variety of strict and preferential constraints, absolute constraints, relative constraints, resource constraints, and a Constraint Satisfaction Problem (CSP) based scheduler.

Objective

The design and development of a science mission planning system is a non-trivial task. The objective of this paper is to provide some lessons that the development of the FUSE science operations system has revealed during its continued progress. Some of the lessons learned are an expression of the reality of the FUSE projects and other past experiences that may also pertain to your projects. The hope is that by remembering some of the information presented here, your development may be more successful.

FUSE Science Operations Concept

The FUSE science mission planning concept is similar to the one used by HST and other observer based satellites. As mentioned above, it uses a two-phase proposal approach. The first phase of the proposal is the submission of requests to receive observation time on the instrument. Potential observers submit the ASCII text version of a LaTeX proposal form electronically to the Guest Investigator program at Goddard Space Flight Center (GSFC). At the time of receipt, a message of acknowledgment is sent back to the submitter assigning the proposal a unique identifier. Should a problem be found later, an additional e-mail message will be sent with more details. Proposals can be re-submitted by placing their unique identifier in the subject line. A received proposal is saved to a file and a backup copy is sent to the FUSE project to provide feasibility analysis for the Target Allocation Committee (TAC). The TAC meets and determines proposals or portions of proposals that are to be allocated time. Observers are notified on the results of the TAC. Accepted programs are now required to submit a completed phase 2 form.

Phase 2 of the operations concept is the submission and scheduling of the accepted programs and begins with the receipt of the completed phase 2 forms from observers. The phase 2 form is an ASCII file, submitted electronically, that uses a set of keywords to define targets and observations. This form is parsed and its information is loaded into a relational database. The science planning team evaluates each observation and assigns them a particular observation type. This value and the information loaded into the database from the phase 2 form is used to expand the observation into the actual data elements used by Spike for long range planning and short term scheduling. The science mission planning team uses Spike as a long range planning tool to organize observations into one week scheduling bins. The completed long range plan is baselined, putting information on when a particular observation is to be scheduled back into the database for use in determining observations' field and guide stars. The science mission planning team uses Spike and a long-range plan to generate short-term schedules for portions, nominally one day, of a long range bin. These schedules identify the sequence of activities needed to complete an observation along with any activities required to move from one observation to the next. Short term schedules produce an ASCII Mission Planning Schedule(MPS) file that details the activities, events, and tables in the format needed by the Spacecraft Control Center(SCC) to generate spacecraft command loads. The MPS files are baselined to assure that they have been generated using the approved long range plan for their time frame and to pass more detailed information about the observations into the database for use by science data processing.

The above science mission planning concept has several areas worthy of highlighting. The first is the decision of the science mission planning teams to review each observation and identify it as a particular observation type. This decision was made to simplify the amount of information required by an observer in their phase 2 form and to give the team more flexibility in determining how an observation was to be scheduled. The science mission planning team felt that this method would minimize the need for observers to understand the details of how the spacecraft, instrument, and scheduling worked and allow them to concentrate on the type of science needed. Also of note is that only electronic submission of phase 1 and phase 2 forms are accepted.

A highlight of the FUSE design is an application called the Exploder. In HST Operations, proposals are broken down to exposure level data before being stored in a planning and scheduling database. The initial granularity for generating database inputs was at the proposal level making it difficult to regenerate data for an individual observation without re-running the entire proposal. FUSE operations learned from this and elected to load the proposal into the database before any action to manipulate the data was taken. The Exploder is responsible for converting observation information from the phase 2 form, the observation type supplied by the science mission planning team, observation type related rules, observation type related algorithms, and observation level overrides to generate the data needed by Spike for its long rang planning and short term scheduling. Using a technique learned from OPUS, the Exploder stores its rules and overrides directly in the database. This allows changes to how the Exploder will expand information to be modified without the need of a build of the software. The algorithms, essentially rules that are too complex to be implemented using the rules method, are defined in the database. Elements of that definition are the order in which the algorithm should be called relative to other algorithms for a particular observation type and a list of parameter inputs to be read into the algorithms. Though not as flexible as the rules, existing algorithms can be added for a particular observation type via the database and defined input parameters can have their values changed without a software build. This feature, especially in the rules definitions, will allow FUSE to add and change rules as they become known during early operations. Another feature of the Exploder is that its granularity is at the observation level. The program allows the user to specify program, observation, list of observations, or entire database for the Exploder to create Spike input data, however, the program processes these an observation at a time. The benefit to this feature is that it allows science operations to regenerate Spike input data for an observation should its scheduling information change without affecting other observations in that observation's program.

Another highlight in the FUSE design is that Spike has the responsibility for breaking observations into its lower level exposures. In the HST scenario, all observations and exposures were defined before planning and scheduling software developed an actual schedule. In the FUSE Spike, the breaking of observations into exposures is pushed pack to where all the orbital information and non-observing overheads are known. This allows Spike to optimize the observation for any given orbit relative to the overall exposure duration of the observation. In addition, Spike has the ability to increase or reduce this overall exposure duration to improve efficiency. The end result is a schedule where the observation and its exposure duration take advantage of every second of available orbit time. To improve efficiency even more, most activities are considered time relative. This allows the spacecraft to start an activity earlier if previous activities have taken less time than expected. This works well for activities, like acquisitions and peakups, whose duration may be difficult to estimate at the Spike planning and scheduling phase.

Lessons Learned

In any project, hindsight is a wonderful thing. When looking back, there are often a number of things that you would do differently if given the opportunity to go back. Below are some of the things I would do differently or at least be prepared when they occur.

Staffing Profiles

Lesson 1. Consider the maturity of the subsystem or software to be developed against the overall maturity of the project when identifying your expected staffing profiles and delivery dates.

When costing out the FUSE proposal, we allocated staff in a uniform and smooth profile. This was done to keep the progress of work on a nice steady pace and to prevent the shuffling of developers between projects. The hope was that enough requirements would flow from the FUSE project to keep our staff busy. Two realities quickly set in. First, since the Spike software was a mature product, the level of information that our system needed was initially higher than what the FUSE project could provide. The project had not developed to the stage that it could give answers. An example of this was ground station contacts, we wanted to know how ground station contact information and its formats was to be passed to Spike while FUSE was still trying to figure out where and what type of ground station they were going to purchase. Second, the limited number of staff members forced each of them to carry a number of different responsibilities. This created a situation where work on any given task tended to come in bursts and where some activities needed to be pushed back due to staffing constraints. This put us in a situation where not only was the project not prepared to field our questions, they didn't always have the staff to track the answers down.

For the case where your subsystem is more mature than the overall system, you may need to adjust your staffing profiles to fit the growth pattern of the project. Recognize in advance that much of the information that you need will come in at the last possible moment and adjust you staffing profile to handle this situation. It may seem that you can use the information needs of your system to drive development. This may not work and may only serve to alienate. Nevertheless, not pushing at all can make it difficult for you to make your deadlines. The solution may lie in the middle ground, using your system to help keep focus on the information that need to be gathered and the design decisions that need to be made.

Operations Concept Definition

Lesson 2. Develop your science operations concept to a level of detail that allows you to validate your design and implementation decisions.

The high-level definition of a science operations concept is generally a data product presented in the Preliminary Design Review (PDR). In the case of an orbiting observatory, this concept presents the steps that a particular observation would flow through the system to get scheduled on the spacecraft. For the PDR, staying high level is a result of the overall amount of information covered at the review and the available time of the reviewers. A pitfall is that the concept does not get evaluated and reviewed at a more detailed level. This causes problems later on in the mission when holes are found or when it is discovered that several people had different impressions of what the concept really is.

Sample Observation Scenarios

Lesson 3. Define a list of the types of observations required for your mission and continue to expand their detail throughout your design and development phases.

An excellent tool for validating your science operations concept and design is the development of science observation scenarios. These scenarios begin as descriptions of observations that will be required of the spacecraft and then become more refined as the operations software develops. Initially, visualizing how a particular observation will flow through the system to the spacecraft will uncover potential flaws in the system. This is especially true for the rarer types of observations (i.e. moving target, bright target requiring acquisition followed by an offset) that may require additional instructions to be passed to the spacecraft than that required for a nominal observation.

As your development continues, the scenarios will be expanded to include the actual inputs needed for each piece of your science operations system. For example, as the definition of the phase one and phase two formats are defined, putting the scenarios in the selected format will validate the structure and identify any missing keywords. Later these can be used to actually test your phase one and phase two parsing software.

Re-Validating Existing Algorithms

Lesson 4. Review the algorithms in the existing code and validate them against the new operations concept, instrument and spacecraft.

A new mission means a new operations concept, a new instrument, and a new spacecraft. Any of these may invalidate an algorithm in an existing software product. The HST version of Spike assumed that its orbit was circular and therefore could use simpler orbit propagation models. In the HST operations concept, this is acceptable because a later software application (SPSS) not used by FUSE has more detailed ephemeris information. For FUSE, Spike needs to be able to handle the situation where the orbit may not be circular and thus needs to have more refined orbit interpolation algorithms. The validation of Spike against the operations concept allowed this problem to be found and resolved with minimal impact. Spacecraft slewing is example where a different spacecraft may invalidate an existing algorithm. The FUSE spacecraft uses eigenaxis slewing while HST uses great circle slewing. By catching these mismatches early in the development phase, FUSE was able to modify its version of Spike. These two examples serve to highlight that no matter how robust an existing application is, early and thorough review of its algorithms and design against the characteristics of the new mission will save headaches down the road.

Software Flexibility

The schedule of the FUSE project is such that detailed information on how the spacecraft and the instrument works will not be available until after much of the software has already been developed. Planning and scheduling software must be developed to minimize the impact of this situation. One technique used is to move operational information into table loads that can be stored in the database. This allows software to be developed using a standard method independent of the actual data that will follow. During actual operations, this technique allows the software to respond to changes without the need to recompile and redeliver code. Using the database to store the information allows a measure of configuration control. Some examples where this technique is used for FUSE are: The software that reads and loads the completed phase 2 forms into the database; The rule definitions used by the Exploder; The Algorithm order and input parameters used by the Exploder; and the definition of MPS activity formats generated by Spike.

The FUSE science planning system uses existing software, for example Spike, that already has a set of defined input and output data formats. Whenever possible, these formats were adopted or an effort was made to minimize the changes to an existing format. For the cases where formats existed for both sides of an interface, one method was chosen over the other. For example, though Spike currently has a format for reading in ground station contacts, the format from the LEO-T ground station software was selected. The reasons for this selection were that FUSE had less control over the LEO-T format and the likelihood of Spike using the LEO-T format on a later project. In developing Spike, acceptable old formats were not thrown away as other formats were added. The idea is that as Spike incorporates the common formats of these products it will become more robust.

Summary

The FUSE project has demonstrated that it is cost effective to adapt and apply existing planning and scheduling software applications to a new mission. The benefits are that it allows a new project a cheaper, faster and better mechanism for the development of its operational software.

Two features that resulted from the incorporation of STScI's existing planning and scheduling software into the FUSE science operations concept are:

- The incorporation of features implemented by OPUS into a new application called the Exploder that creates the Spike input products via phase 2 program data, observation types, rules, and algorithms.
- The technique of putting information into the database rather than directly into an application, thereby allowing modification without software builds.
- Pushing the breaking of observation into exposure into Spike to allow further optimization of orbit time.

- Making activities time relative to further allow improvement in efficiency of scheduling after it is loaded on board the spacecraft.

The incorporation of existing planning and scheduling software into the FUSE science operations concept to date has identified some lessons that can be applied to other missions. The are:

1. Consider the maturity of the subsystem or software to be developed against the overall maturity of the project when identifying your expected staffing profiles and delivery dates.
2. Develop your science operations concept to a level of detail that allows you to validate your design and implementation decisions.
3. Define a list of the types of observations required for your mission and continue to expand their detail throughout your design and development phases.
4. Review the algorithms in the existing code and validate them against the new operations concept, instrument and spacecraft.

For further project that incorporate existing software products, it is important to remember that improving the flexibility of your software now may make it easier later. Two areas where flexibility improvements can be realized are database usage and standardize inputs/outputs. For the database, creating mechanisms to pull in program information allows quicker turn around times for mission specific changes. Further, if these relations are designed with other projects in mind, they make it only necessary to modify data in the database to bring an application on-line. This has been demonstrated by OPUS. Standardizing inputs and outputs to match up with the software products that applications will most likely interface with will improve the plug-in capabilities of that application. For example, it is very likely that Explorer missions will predominantly use ground stations so by incorporating the ground station contact file format from LEO-T, Spike will support this system without any modifications.

References

Web Related Sites

The Restructured FUSE Mission, Science and Technical Forum: Revision 1, February 1996
fuse.pha.jhu.edu/pubinfo/docs_pub/scitech/scitech.html

Spike: Planning and Scheduling,
presto.stsci.edu/spike/

FUSE Mission Planning Database Design Document,
www.pst.stsci.edu/~cjohnson/fuse_db_design.html

FUSE Exploder Design Requirements,
www.pst.stsci.edu/~cjohnson/exploder_design.html

OPUS: An Operational Pipeline Unified System
samvax.stsci.edu/rose/opus/index.html