Space Applications of AI. NASA CP 3009, pp. 91-106, reprinted in Telematics and Informatics, 5, 197-212.


Electronic Submission and Preparation Tools - Experience with remote preparation and submission has been positive. Virtually all proposals are submitted electronically. Distribution of proposal preparation software allows astronomers to uncover questions and problems before submission. This allows the proposer to have greater control over the proposal and saves on the cost of proposal corrections after submission. We plan to expand this to provide more user-friendly software, more powerful tools and make the system available on more types of computers (see also reference Sup2(1)).

Proposal Changes - Proposal processing was designed with the assumption that proposals would change very little after submission and that nearly all proposals would be available at the stated submission deadline. These have not been good assumptions. Not only has there been a desire on the part of the proposers to modify their plans, but external factors (e.g. launch delays, the mirror problem) have forced wholesale revisions of the proposal pool. Perhaps the lesson to be learned is: don't expect change - rely on it.

Interleaved scheduling - Ground-based astronomical observatories have traditionally been operated in a “block” mode - an observer is given a block of time to gather data. With the HST, observations from different proposals are interleaved in order to maximize efficiency. This goal is being achieved and has allowed scientists to propose observations which would be very difficult to accomodate with traditional block scheduling.

Development Methodology - Recognition that requirements were not well defined was an explicit feature of the development approach. Users and developers worked very closely in the development of procedures and software systems.

Acknowledgements: Many dedicated people contributed to the development and operations of the systems described here, but space does not allow us to thank them individually. We gratefully acknowledge the comments of Hans-Martin Adorf, Kirk Borne, Bruce Gillespie, Tony Krueger and Jeff Sponsler.

REFERENCES

Had such extensive data not been available, the creation of a substantial body of test data would have been essential.

Key to the success of these systems is the staff at the STScI. Both user and developer staff are dedicated to the success of HST’s mission. Most users have a scientific background and welcomed the support that software could provide. Users contribute regularly to requirements analysis and project planning. In addition to computer scientists, the development staff includes several astronomers who have a good understanding of the system requirements. Developers and users were usually located in the same building which provided an important degree of contact which is not possible through telephone conferences or formal reviews.

It is interesting to note that some of the most useful software tools were "reactionary", that is, a user would informally ask for some software tool to handle a problem which was previously unknown or thought to be of low priority and a developer would provide it in a very short time. This tool then rapidly grew in importance to encompass a significant part of operations. (Perhaps this is just rapid prototyping at its finest!) The important lessons here are that developers and users must maintain good communication and developers must have the flexibility to work on unscheduled activities. Of course, the converse is true too in that substantial effort was devoted to some tools which were never of much use operationally.

The PEP system is written mostly in C (which, in 1984, was a somewhat radical step since most of the HST ground system was in Fortran or assembly language). PEP also includes a substantial amount of database commands (in IQL) and some Lisp and OPS5. There are approximately 100,000 lines of code.

Transformation, Spike and parts of PEP were developed using “artificial intelligence” technology, including Lisp, object-oriented programming, expert systems, and artificial neural networks. Lisp has proved to be an excellent language for implementing these systems (the use of Lisp in 1987 for Spike was even more radical than the use of C for PEP in 1984). This language fully supports both symbolic and numeric processing and the rich Lisp development environment provides many software development tools (including interpreted and compiled modes, symbolic debuggers and window-oriented data inspectors). Lisp supports a rapid prototyping approach so that ideas can be quickly implemented and modified on the basis of experience. Spike and Transformation consist of about 65,000 lines of code each. There are ~350 rules in Transformation.

Lisp was standardized as Common Lisp in 1984, and object-oriented features were incorporated into the standard in 1989. (Prior to this, Lisp could be justly criticized for the proliferation of incompatible dialects.) Common Lisp is available on a large number of delivery platforms and has proven to permit straightforward porting of source code. Transformation and substantial portions of Spike are currently supported on VAXes, Sun Sparcstations and Texas Instruments Explorers with no source code changes. Another important trend is that development of operational Lisp applications no longer requires expensive, special-purpose workstations (“Lisp machines” such as the TI Explorer) - performance on general-purpose workstations (e.g. Sparcstation) equals that of a Lisp machine. Consequently, we are transitioning both Spike and Transformation to such workstations. Software development tools on a Lisp machine are much better than those on the general-purpose workstations, but the gap is narrowing and is not a decisive factor.

The most important area requiring standardization (in Lisp and other languages) is user interface. High resolution, graphical, mouse and menu displays are of tremendous benefit to users. X-windows provides a low-level standard, but there is currently no high-level standard for a wide range of platforms. A number of different groups are working on such systems and there is reason to hope that a suitable standard will soon emerge.

9.DISCUSSION

In this paper we have presented the operations and evolution of the proposal processing, planning and scheduling systems for HST science operations. Many aspects of the HST experience can be applied to other astronomical missions and to aerospace missions in general.

Key Concepts and How They Fared

Insulate users from unnecessary details - The HST and its ground systems are quite complicated. In order to make the HST a resource available to the entire astronomical community (not just a few experts), the STScI developed extensive software (PEP, Simple, Spike and Transformation) and organized service groups (SPSO, USB, SPB and others) to help proposers. This goal has been accomplished to a large degree. Although any mission should be designed with ease of use as an important goal, as instrumentation becomes increasingly sophisticated, the need to provide human and software support will increase.
by NASA. SPB used Spike to verify this timeline and to schedule GTO observations during weeks when time was available.

Scheduling of the GTO proposals in Cycle 0 used Spike in an interactive mode: Planners would display individual proposals on timeline displays and choose times of high suitability for observations. Various automatic scheduling tools were sometimes used in conjunction with manual commitments. Scheduling of Cycle 1 proposals has also been largely interactive with little use of the high-level, automated schedulers. This has primarily been due to a larger than anticipated rate of change of proposals which results in a small pool of proposals which are ready for execution.

**Proposal Changes and Status Tracking**

Much of the systems and procedure development was based on the concept that proposals would be frozen after submission (an upper limit of 10% of the proposals changing after submission was often used for planning purposes). This has proved to be a very poor assumption in that many proposals have changed after submission (e.g. the mirror flaw caused all proposals to be re-examined for feasibility). The assumption of proposal stability greatly simplified the design of the software and was probably the only viable choice given the pressure of launch. Most people would not have believed a prediction of the actual proposal change rate.

Proposal changes have had a large impact on proposal status tracking. Between submission and archiving of data, a proposal is processed by many systems and it is important to be able to report on the status of a proposal across all systems. Two factors complicate 1. HST status tracking: Proposals are generally executed piecewise over many weeks or months in order to optimize individual observations and overall operations. 2. The augmentation software systems were developed in parallel by different groups with the result that there is currently no integrated status tracking mechanism. We are currently enhancing Spike's status tracking mechanism to provide a global view of proposal status and are examining ways for PEP to incrementally process proposal changes.

**8. SOFTWARE DEVELOPMENT AND MAINTENANCE EXPERIENCE**

In this Section we discuss some of the experience from the software development and maintenance process.

A common feature of PEP, Trans and Spike development was that each system had to be developed in a short time (about 6 months for the initial system, with substantial extensions continuing over several years) and with a small staff (2-3 people initially). It was also impossible to specify in advance a complete set of requirements for these systems since many important factors were unknown. These considerations led to the use of a rapid prototyping software development methodology instead of a more classic style (requirements definition, design, implementation and test). In rapid prototyping, initial versions of the system are delivered to users for evaluation. Their experience with the prototype defines further requirements and changes are incorporated into the system. A tool-oriented approach was also encouraged, i.e. the development of general software routines which could be used for other applications later in development.

The most significant advantage of rapid prototyping to the HST was that it allowed PEP, Trans and Spike to be implemented in time to support testing and operations - in an environment with changing requirements there is no choice but to use an adaptive software development methodology. Perhaps the most serious complication of this approach is that once the prototype is used operationally, it becomes increasingly difficult to make large changes to it. (In an ideal rapid prototyping situation, a prototype can be discarded after evaluation). Once operational, it is necessary to ensure that each version of the system is upwardly compatible with the previous version. A corollary to this is that pressure on the users to do operational work can prevent them from participating in the software development process, e.g. critiquing initial requirements and evaluating prototype software. This can lead to a divergence between the needs of the users and the products of the developers and must be avoided.

While following a non-classical development methodology, PEP, Spike and Transformation nonetheless paid careful attention to such classic factors as configuration control and testing. Developers use code management tools (i.e. Unix rcs and DEC CMS) to prevent uncoordinated changes. Building, testing and installing the software is automated with software tools to the largest extent possible to detect problems before delivery to the users. There are several releases of each software system so that developers, independent testers and users can test software before installation into operations.

The incorporation of realistic test data proved to be quite important. Due to delays in the launch of HST, we had several hundred GTO and GO proposals which were constantly used to test prototype systems and to make development decisions.
amounts of software (written in C) were necessary to drive the vendor software to closely mimic the HST proposal forms on a VT-100 terminal. Since the original implementation, the vendor has dropped support for a number of critical routines in this product. The adoption of SQL, not IQL, as the ANSI standard query language has made obsolete the queries, reports and database commands used in PEP. (Database evolution is also discussed in Sup2(8).) We are currently revamping PEP to loosen the coupling to a database machine, use a graphical user interface, and to support distributed processing on workstations. Given the success of remote proposal submission, another important goal is to provide the astronomical community with more proposal preparation tools.

**Experience with and Evolution of Transformation**

When first proposed in 1984, the concept of an automated transformation of scientific proposals to implementation parameters was quite novel. Since that time, the Transformation system has demonstrated the capacity to routinely perform this task and allows STScI staff to focus more attention on innovative or difficult observations. Additionally, as improved implementation strategies are devised, Transformation is modified and it is possible to quickly re-transform large numbers of proposals to benefit from improvements.

The Transformation expert system was originally implemented in the production rule expert system language OPS5. This language was well suited to the initial transformation process in that it was possible to capture the knowledge of the experts (Operations Division staff who use SPSS) in a set of largely independent “if-then” rules. From its development in late 1985 through 1989, OPS5 Transformation was successfully used to support several test exercises of SPSS (though most transformation was done manually). Toward the end of this time, Transformation software had become very hard to extend and maintain. The transformation algorithms were now procedural and iterative, with very little opportunistic components: the execution of one rule tended to depend strongly on the execution of other rules. Iteration and procedural execution are difficult to express cleanly in OPS5. The rulebase had grown from ~300 to ~900 rules which were highly coupled. The result was that apparently small changes to Transformation would require a large amount of development effort and extensive debugging.

Transformation was rewritten in Lisp and changed from an opportunistic rule system to one where the rules are executed in a specific order Sup2(4). Four developers rewrote Transformation in five months. Lisp-based Transformation has been very successful: the bulk of proposal data currently in SPSS is the result of the Transformation expert system. As experience has been gained in operating the HST, a large number of changes to the transformation process have been identified (and will continue to be identified for the foreseeable future). Over the past year, Transformation has incorporated over 300 modifications (ranging from small changes to major new capabilities). Requirements specification and testing, not code development, are usually the rate-determining steps for adding new features to the system. Operations Division staff manually modify the SPSS database to accommodate changes which are in the Transformation analysis-develop-test pipeline.

When used on actual GTO and GO proposals, Transformation (and also Spike) reported large numbers of diagnostic messages. Often these problems were due to an incomplete Transformation rulebase (since it is still under development). The frequency of problems allowed us to effectively prioritize work in determining requirements and implementing the code. A significant number of problems uncovered by Transformation and Rulebase were due to an inadvertent specification by the proposer of inconsistent requirements in the PEP proposal. Although the PEP system performs syntactic checking on proposal information, Spike and Transformation are the first systems that can detect problems related to planning and scheduling. (In particular, accurate instrument overhead times and orbital viewing conditions are calculated by Transformation and Spike and these can reveal problems with the proposal.) We are currently investigating how to incorporate such checks in PEP and RPSS. Not only will this provide proposers with immediate feedback of certain classes of problems, but it will lessen the number of delays in the scheduling process due to proposal changes.

**Experience with and Evolution of Spike**

HST science operations is divided into cycles in which proposals are solicited from the astronomical community, selected, scheduled and executed. In the long term, cycles will consist of about 1 year of HST observing. Early HST operations consist of two special phases: "Orbital Verification" (OV), which assessed the basic capabilities of the telescope and instruments and "Cycle 0" observations, which contain a mix of Science Verification (SV) and GTO observations. OV ended in November 1990, and Cycle 0 ended in June 1991.

The Spike system was first used to support HST scheduling for Cycle 0. The timeline for SV observations was established
Experience with and Evolution of Proposal Processing

Development of the PEP system began in September 1984 and the initial entry system was on-line in April 1985 for the entry of 430 GTO proposals (in anticipation of a June 1986 launch of the HST). At this time all proposals were submitted on paper forms.

Development of PEP continued and produced evaluation tools for proposals (e.g. the duplication checker and resource usage estimator), TACOS tools, Transformation and RPSS. Initially, electronic submission of proposals was optional. As the benefits of the system became apparent, RPSS submission became mandatory for all proposers with access to a suitable network. Electronic submission is now used in more than 90% of the proposals. RPSS began a major evolution of the system away from paper forms and to providing proposers with software tools. The STECF has also developed advanced proposal submission tools.

With the experience gained in GTO proposal submission and delays in the launch of HST, in April 1987 the Phase I/Phase II concept of proposal processing was introduced (along with the Simple system). The motivation was to reduce the amount of work required to submit and review proposals and require detailed specification only for accepted proposals. Prior to this, proposers were expected to submit all information for selection. 556 GO proposals were received (using paper forms only) in October 1988. The TAC was convened in April 1989 and selected approximately 160 proposals for execution.

Unlike most ground-based observatories which grant blocks of time (e.g. a few nights) to a proposal, HST proposals are interleaved to achieve high operational efficiency. As a consequence, an HST proposal can specify repeated or linked observations over a longer time range than is possible for block scheduling. Astronomers have in fact taken advantage of this flexibility: in the current HST proposal pool about 25% of the exposures are linked to other exposures with a separation greater than a few days and over 50% of the exposures are linked to another exposure.

PEP Database

A key aspect in the design of the PEP system is the use of a relational database computer to store all proposal and status data. All access to the database (data entry, editing, reporting, etc.) is via software running on a host computer. SOGS uses VAX computers as the host, Britton-Lee (now Sharebase) IDM500 as the database machine and Omnibase as the interface software. Compatibility with SOGS and the desire to gain experience with the VAX-Omnibase-IDM architecture prior to SOGS delivery lead to this choice of database configuration for PEP as well.

Important advantages of the relational approach are the ability to quickly develop a system which stored proposal data and the capacity to formulate queries and reports using a non-procedural “fourth generation” query language (IQL). In fact, users quickly learned the query and report generation languages and heavily rely upon them.

There are however, several disadvantages to this architecture. Storing all proposal data in a database lead to performance and database contention problems. Not all proposal information decomposes nicely into third-normal-form relations.
Software Systems

The Spike system was developed at the STScI and uses artificial intelligence (AI) technology to perform long-range scheduling using Transformation results as input. Important features of Spike include:

- A constraint representation and propagation mechanism which includes the ability to express human value judgements as well as strict constraints that can never be violated
- Proposal evaluation tools that allow planners to display observations and constraints on a high-resolution graphics workstation and hardcopy devices
- Automated and manual scheduling tools
- Automated tools to track the status of the scheduling process, including information sent to and received from SPSS.

Spike provides several high-level automated scheduling tools including a scheduler based on constraint satisfaction problem techniques which can schedule several thousand scheduling units on a multi-year timeline in a few tens of minutes. (Other techniques have been prototyped or fielded: neural networks, rule-based expert systems, genetic algorithms, and a rescheduling system.) Spike also provides a graphical window-based interface to workstation users for controlling the scheduling process and for displaying scheduling data.

The HST was designed so that two instruments can operate in parallel, thus increasing the scientific output of the observatory. Spike matches candidate parallel observations with suitable primary observations.

Organization

SPB is responsible for long-range scheduling, while short-term scheduling is the responsibility of the SPSS Branch of the Operations Division. Several months prior to execution, the list of scheduling units for a week is transmitted by SPB to SPSS. After short-term scheduling, SPSS return the scheduling status of each scheduling unit, i.e., whether or not it was scheduled, the start time of the observation, and the spacecraft roll angle. After execution of the observations by the HST, data is processed by the Post Observation Data Processing System (PODPS) which provides Spike with the definitive list of observations that were executed. Spike compares this to SPSS-supplied schedule and notes to the user any differences.

7. OPERATIONAL EXPERIENCE AND EVOLUTION OF THE SYSTEMS

The previous sections discussed the proposal processing, planning and long-range scheduling systems. In this section we examine the experience gained in using these systems over several years, describe the evolution of the systems and attempt to draw "lessons learned" from this experience. Table 1 is a timeline of events relevant to proposal processing and scheduling.

The STScI was established in 1981 and soon began developing a detailed concept for science operations. The Science Operations Ground System (SOGS, which includes SPSS) was developed by TRW for NASA and was originally envisioned to perform many of the tasks of proposal entry, evaluation and long-range scheduling. By 1984 it had become apparent to STScI staff that SOGS would not be able to adequately handle proposal entry and that the process of transformation was far too complex for non-experts. As modifications to SOGS were technically and managerially difficult, the concept of “augmentation” software systems was developed: these were independent systems which pre- or post-processed information for SOGS. PEP and Transformation were the first such systems, followed later by Spike, Simple and others. This approach has been successful and has allowed the phased introduction of new capabilities with minimal interruption to operations and development.

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1981 Dec</td>
<td>STScI founded</td>
</tr>
<tr>
<td></td>
<td>SOGS development begins</td>
</tr>
<tr>
<td>1984 Sep</td>
<td>PEP development begins</td>
</tr>
<tr>
<td>1985 Apr</td>
<td>Entry of GTO proposals in PEP</td>
</tr>
<tr>
<td>Dec</td>
<td>Transformation used for testing</td>
</tr>
</tbody>
</table>
readers and difficult to maintain. The volume of information in a proposal would be unwieldy: since it is dealing at a lower level of detail, the SPSS representation is roughly 50 times larger than the PEP representation. In addition, for most observations there are several legal ways to transform from the PEP to SPSS representations so it would be difficult to ensure a uniformity of transformation between proposers (which is an important operational consideration).

**Software Systems**

The Transformation expert system converts the proposal Exposure Logsheets and Target Lists into scheduling units. This is not simply a matter of reformatting information since the proposal description is at a higher level than scheduling units. Transformation must determine the order to execute observations (when not specified by the proposer), group observations to minimize overhead operations and pick specific implementation scenarios, etc. Transformation is the first system which examines proposals for problems related to planning and scheduling, e.g. circular references in precedence requirements, conflicting timing requirements. (PEP's checking is mostly a syntactic analysis.)

The diagnostic and report files generated by Transformation are used by STScI staff to understand how a proposal will be implemented and to track down problems.

The input to Transformation is a file generated from the PEP system. Transformation produces output files which are input to Spike and SPSS.

**Organization**

Upon the hand-off of a proposal from USB, SPB transforms it and checks the output for diagnostic information. A working group of SPB, Operations Division and other STScI staff review the transformation output for each proposal prior to scheduling. The Transformation rulebase is still under active development with major contributions from the Operations Division and SPB.

### 6. LONG-RANGE SCHEDULING

Scheduling the HST is a challenging problem for several reasons:

- **size** - a year's observing pool consists of tens of thousands of exposures for a few thousand astronomical targets
- **constraints** - there are a large number of interacting constraints on the observations. Some constraints couple observations separated by long time intervals (months to years).
- **uncertainty** - scheduling is far in advance of execution, and many constraints can not be predicted in detail in advance
- **multiple evaluation criteria** - there is no one overriding factor which determines the schedule and a trade-off between competing factors is necessary
- **replanning** - continuous modification of the schedule is necessary as observations are executed and proposals are changed

It is widely recognized that scheduling problems are computationally difficult and belong to the class of problems called "NP-complete". This means that the best known algorithms for finding solutions have exponential complexity, i.e. a small increase in the number of objects to be scheduled can result in a huge increase in the time to find a solution.

A two-level, hierarchical approach has been used for HST science scheduling by dividing the problem into long- and short-term scheduling. Long-term scheduling covers a multi-year interval and allocates scheduling units to particular weeks, while short-term scheduling covers a one-week period and creates a detailed timeline of activities.

As mentioned earlier, HST operations are largely pre-planned, with little real-time interaction. In order to provide flexibility, proposers are allowed to identify conditional exposures. These are exposures (or groups of exposures) which may be executed depending on certain conditions (e.g. on the analysis of other exposures in the proposal). Long-range planning tracks the status of conditional exposures. The proposer notifies the STScI as conditions are fulfilled and the long-range plan is modified accordingly.
(e.g. NGC4486, WFC, 5 minutes). The proposer then runs the “validate” program which performs a syntax check of the proposal file and can detect a wide range of errors including typographical errors (e.g. a misspelled filter name), range checking on values (e.g. a target declination exceeding 90 degrees), and missing information (e.g. an exposure referencing an undefined target).

When the proposal file is complete and passes this validation step on the local computer, the proposer logs into STScI’s RPSS VAX computer via a network (Internet or SPAN) and submits the proposal. Validation is invoked once more as a final check, and if successful, the proposer is assigned a remote submission identification number. This proposer writes this number on a Proposal Submission Form and mails this to the STScI (paper copies of the rest of the proposal are not required). The proposal file is copied by RPSS to a secure data area.

The ability to locally validate and electronically submit a proposal is an extremely valuable tool for both proposers and the STScI. Proposers can detect and correct a large class of errors and are assured that typographical errors are not introduced by data entry personnel. The STScI can process proposals more rapidly and avoid costs of manual entry.

To our knowledge, RPSS was the first system of its kind for a major scientific installation (it has been in use since February 1986).

RPSS software currently runs only on VAX VMS computers. Proposers without access to VMS can use the RPS computer at the STScI over a network. Proposers without any computer network access may submit proposals on paper forms (only a very few proposals are received on paper).

Proposals are examined to determine if the exposures duplicate exposures in another proposal or in the data archives. This duplication checking is necessary to prevent waste of HST observing time and to ensure that GOs do not duplicate observations which are reserved for the GTOs. PEP includes a duplication checking expert system [Sup2] (L(6,10)) which provides the ability to identify exposures which have similar (but not necessarily identical) target positions, instrument configurations and exposure times. The duplication checker produces a list of potentially similar exposures and a confidence factor (high, medium or low chance of duplication). All such candidates are examined by astronomers to determine if duplication has actually occurred.

Organization

Initial processing of Phase II proposals is performed by the STScI User Support Branch (USB). Upon receipt of a (paper) Proposal Submission Form, the appropriate RPSS file is retrieved and entered into the PEP system. USB staff carefully read the proposal to look for potential problems and to be sure that the proposal conforms to any limitations set by the TAC. Other STScI staff may be asked to review the proposal as well. The proposer is informed of any relevant problems or proposal modifications. Upon completion of this process, the proposal is delivered to the Science Planning Branch (SPB) for transformation and long-range scheduling.

5. PROPOSAL PLANNING AND TRANSFORMATION

An HST proposal specifies an observing program in terms familiar to an astronomer. As described in the previous Section, a goal of the proposal process was to allow astronomers to use the HST without having to become experts in detailed aspects of the telescope, instrument and ground systems.

Detailed, short-term scheduling is performed with the Science Planning and Scheduling System (SPSS), developed by TRW. The fundamental input data structure to SPSS is the scheduling unit. The structure of a scheduling unit reflects the design of the SPSS software as well as spacecraft and orbital considerations (such as small adjustments in the pointing of the telescope, availability of guide stars, communications contacts and instrument configurations). The process of converting from the PEP to the SPSS representation is called transformation.

At first it might seem that the need for transformation was artificially induced by the proposal forms - if proposers submitted the information in SPSS syntax, no transformation would be necessary. However this would put an unwarranted burden on the proposer for it would be necessary to understand details of how SPSS and the HST spacecraft work to describe even simple observations. The instruction manual would quite literally be thousands of pages in length and therefore ignored by
Software Systems

The TAC Operations Support (TACOS) system was developed to assist in tracking information during proposal review. TACOS provides a simple yet powerful natural language interface to a “spreadsheet” of proposal information. Items recorded for each proposal include measures of resources consumed (e.g. exposure time), proposal properties (e.g. scientific instruments used) and rank assigned by the reviewers. In TACOS it is straightforward to define queries and reports, either in real time during a meeting or pre-defined for later use. Another tool provided by TACOS is used to assign proposals to referees, avoiding potential conflicts of interest (e.g. one cannot referee one's own proposal).

The Simple system provides proposal reports for the peer review, management of referee comments and generates the TACOS input.

Organization

SPSO establishes proposal selection policies (under the direction of advisory groups, NASA and the Director) and manages the selection process. USB provides technical support for selection. Panel and TAC members are drawn from the astronomical community.

4. PHASE II PROPOSAL PROCESSING

The Phase I proposal provided information needed for proposal selection. Proposers who are awarded HST observing time must submit a Phase II proposal which specifies the additional information needed to implement the observations. The STScI designed the Phase II proposal with the following goals:

- Oriented towards the astronomical community - easy to understand, and concise and logical in the amount and sequence of data requested
- Able to accommodate simple and sophisticated observations; novice and experienced HST users
- Allow the proposer to specify what data should be collected without becoming needlessly encumbered by instrument, telescope and ground system minutiae.
- In the case of non-electronic submission, allow data entry by non-technical personnel with limited training.

A Phase II proposal is organized into a number of sections or forms:

- **Target Lists** - description of target position and brightness.
- **Exposure Logsheet** - list of individual exposures, constraints, and special instrumental parameters
- **Ancillary forms** - for specialized observations, additional forms may be required
- **Cover Page and General Form** - identical to the Phase I proposal

Some examples will illustrate the additional information required in a Phase II proposal: Target coordinates must be specified to sufficient precision to be located in the instrument aperture. Time separations between exposures must be specified with allowable tolerances (e.g., repeat every 6.5 ± 0.5 days for 4 times, or group a set of exposures to be executed within 35 minutes. Non-standard instrument parameters can be specified (e.g., a readout time for the detector).

Another feature of the Phase II forms is the ability to define a general set of observations (called a *sequence*) and to invoke that pattern for any number of different targets, exposure times, etc. This not only can simplify the preparation of the proposal, but it can help communicate the proposer's plan of observation to STScI staff. Sequences are analogous to subroutines or macro expansion in a computer programming language.

Software Systems

The Phase II proposal is prepared using the Remote Proposal Submission System (RPSS). RPSS includes software which is distributed to the proposers (via STEIS).

Using a local computer, the proposer prepares the RPSS proposal file, which is a text file in keyword-value format. The keywords correspond to items on a paper form (e.g. target name, instrument, exposure time) and the proposer supplies the values
• Observation Summary Form (OSF) - a summary of the observations, including target data and instrument usage
• Budget Forms (optional)

As discussed below, some of this information is submitted electronically, via electronic mail, to the STScI.

Software Systems

To submit a proposal, the proposer first accesses STEIS via Internet and downloads a proposal template file and formatting software via anonymous File Transfer Protocol (FTP). Next the proposer edits this template file to provide the information needed for the Cover Page, OSF, and, optionally, the General Form. The formatter program uses this file to produce formatted copies of the Cover Page, General Form and OSF. The formatter also performs some syntax checking of the file and reports any problems to the user. In addition, the formatter program uses the information on the OSF to estimate the requested amount of exposure time, spacecraft time, and the efficiency of the proposal and enters this information on the Cover Page. The proposer then mails this file to the STScI (via Internet, Bitnet or SPAN). Finally, the proposer submits multiple paper copies of all the forms via regular mail. If the proposer certifies that access to electronic mail is impossible (as in some countries), then the STScI will accept a proposal on paper copies of forms published in the Call for Proposals.

The Phase I proposal file has three sections, corresponding to the Cover Page, General Form (optional) and OSF. Each section has a different format, reflecting the hybrid nature of the system (see Section 8). The Cover Page section is Phase II "RPSS" format, which is essentially pairs of keywords and values (see Section 4). The General Form section is simply plain text to be used by humans and not computers. Information in the OSF section must be in a specific tabular format which resembles the paper form. As mentioned above, a software tool reads the OSF section to estimate the total amount of exposure time used in the proposal.

When the proposal file is received at the STScI, information from the Cover Page and OSF is automatically extracted by software and entered into proposal processing systems (Simple and PEP). This has resulted in a substantial savings in data entry personnel and equipment, and elimination of typographical errors during entry. (Even though proposers are still required to submit multiple copies of the proposal on paper for peer review, the above advantages benefit the proposer as well.) The Cover Page section is processed by the Proposal Entry Processor (PEP, see Section 4), which assigns a unique proposal identification number and stores the cover page data in a relational database. The OSF data is read by the Simple system to provide resource estimation, duplication checks, and reports on the proposal pool.

Organization

The STScI offers a variety of services to assist proposers. There are manuals which describe the telescope and instruments. The STScI publishes a Newsletter with updated information on the observatory. The Space Telescope Electronic Information Service (STEIS) provides the community with the most up-to-date information on the observatory, along with access to documentation, computer programs and sample HST data. The USB has primary responsibility for assisting astronomers in proposal preparation.

European astronomers are also assisted by the Space Telescope European Coordinating Facility, which provides services including consultation, a newsletter and proposal preparation tools Sup2(1).

3. PROPOSAL SELECTION

The evaluation of the proposals is accomplished via a two-stage, peer-review process. In the first stage, proposals are considered by Panels of astronomers on the basis of the astronomical field (e.g. solar system, stellar astrophysics, galaxies, quasars, etc.). The result is a ranked list of proposals within each field and recommended telescope time allocations for each proposal. In the second stage, the Telescope Allocation Committee (TAC) combines the panel recommendations into a balanced overall program for the HST. The TAC consists of the chairpersons of the review panels plus a number of astronomers who did not participate in the first review. The TAC makes recommendations to the Director of the STScI who makes the final allocation of observing time.

The goal of selection is to choose a set of proposals which maximize the scientific achievements of the HST while not exceeding the available amount of observation time and other limited resources.
NASA has awarded a portion of the HST observing time during the initial years of operations to astronomers involved in the development of the telescope and instruments. These are the Guaranteed Time Observers (GTOs). Other scientists applying to HST time are called General Observers (GOs).

**Overview of Software Systems**

A number of software systems play an important role in supporting the HST mission. The Simple system is used to process Phase I proposals. (The name "Simple" derives from its origin in simplifying the process of proposal preparation prior to selection as discussed in Section 7.) SPSO develops and maintains Simple. The Proposal Entry Processor (PEP) is used to support both Phase I and Phase II proposal processing and includes the Remote Proposal Submission System (RPSS). The Transformation and Spike Systems respectively support transformation and long-range scheduling. The Advance Planning Systems Branch develops and maintains PEP, Transformation and Spike. These three systems make use of artificial intelligence techniques (see Section 8). Short-term scheduling is performed with the Science Planning and Scheduling System (SPSS), which was developed by TRW.

2. **PHASE I PROPOSAL PREPARATION AND SUBMISSION**

To compete for HST observing time, an astronomer must submit a “Phase I” observing proposal. The purpose of the Phase I proposal is to supply the information needed for the selection of HST observing programs. Successful proposers are required to submit a more detailed “Phase II” proposal (Section 4). In many respects an HST Phase I proposal is similar to proposals for other astronomical observatories and to proposals in other areas of science. The proposal must describe a problem, its importance, the data to be collected, the analysis to be performed and how this will solve the problem. Due to the large oversubscription factor (and the necessary consequence that most proposals will not be granted observing time), the Phase I process was designed to place the minimal possible burden on the proposers and reviewers.

Information in the Phase I proposal is grouped according to “forms”:
- **Proposal Submission Form** - a checklist for the other forms
- **Cover Page** - summary information such as title, name of investigators, abstract, total spacecraft time requested, etc.
- **General Form** - the scientific justification and related information
A year's scheduling pool of about 300 proposals comprises tens of thousands of exposures on a few thousand targets. This presents a formidable scheduling problem. A hierarchical approach to HST scheduling has been developed. A long-term plan covers several months or years. From the long-term plan, week long segments are extracted for short-term scheduling.

Short-term scheduling takes a list of scheduling units from the long-term plan and generates a week-long sequence of activities (called a calendar or timeline). Next, high-level spacecraft instructions are attached to the activities on the timeline. The output of the process is a Science Mission Specification (SMS) and can be thought of as the "assembly language" which drives the HST.

The STScI delivers the SMS to the Payload Operations Control Center (POCC) at NASA's Goddard Spaceflight Center, where it is checked for errors and constraint violations which would affect the health or safety of the HST and instruments. From the SMS, the POCC prepares the binary command loads for the two onboard computers which control the observatory. The POCC also takes requests for Tracking and Data Relay Satellite (TDRS) links from the SMS and passes them on to the TDRS Network Control Center. Some requests will not be granted due to higher priority users (e.g. Shuttle or other satellites). The POCC notifies the STScI of this and the timeline is modified by using an onboard tape recorder to hold data for later playback or rescheduling the observation.

Execution of the observations is monitored by both the STScI and POCC (by monitoring real-time transmissions or by playback of recorded science and engineering telemetry). The STScI analyzes data to ensure that scheduled observations were completed successfully and delivers the data to the proposer. (Failed observations may be rescheduled.) Since HST observations are an important astronomical resource, an archive of all HST data is maintained. The proposer is normally granted exclusive access to the data for a proprietary period (usually 1 year), after which the data is available to the scientific community.

Outline of Paper

We have given an overview of HST science operations, from proposal preparation through data collection and archiving. The purpose of this paper is to present a case study of the “front-end” systems: proposal preparation through long-range scheduling. The next several sections discuss these steps in some detail: Phase I proposal preparation (Section 2), proposal selection (Section 3), Phase II proposal submission and processing (Section 4), transformation (Section 5), and long-range scheduling (Section 6). We discuss not only the software systems, but the operational and organizational structure of these processes. Operational experience and the evolution of the systems from their initial design to current implementation are discussed in Section 7. Section 8 relates experience from the software development and maintenance efforts. The final section presents the conclusions and a discussion of future directions for the proposal processing and scheduling systems.

STScI Organization

A brief description of the STScI organization (Figure 2) will help the reader to follow the discussion. The User Support Branch (USB), provides primary support to HST observers. The Science Program Selection Office (SPSO), conducts the proposal selection. The Science Planning Branch (SPB) transforms proposals and develops the long range plan. The Operations Division performs short-term scheduling, observation monitoring and routine data processing.
scribed below), which are used by the long- and short-term scheduling systems.

Figure 1 - Overview of HST Science Operations. Boxes show major stages in processing and supporting software systems are given in parenthesis.

In addition to the constraints imposed by the proposer's scientific program, there are a large number of other constraints which must be considered. Many orbital factors exert a strong influence on scheduling: targets are occulted (blocked) by the Earth for up to 40s(m) each 95s(m) orbit. Observations cannot be taken during HST's passage through the South Atlantic Anomaly (SAA), which may last more than 20s(m). The telescope cannot point too closely to the Sun, Moon or bright Earth limb. The roll orientation of the spacecraft is constrained in order to maintain correct power and thermal balance. Communications with HST is via the Tracking and Data Relay Satellites (TDRS) and links will be available for only part of an orbit (this also limits the amount of real time interactions with the HST and instruments). Slews are relatively slow (90s(o) in ~15s(m)), so efficient ordering of telescope pointings is important. Available electrical power limits the number of instruments that can be in “standby” or “operate” modes, and cycling between instruments can take several hours.

As a consequence of these and other factors, the operation of HST is almost entirely pre-planned. Detailed weekly schedules are established about two months in advance. Real-time control of the telescope is normally limited to a choice between predetermined options (e.g. an instrument filter) or small adjustments in the telescope pointing for target acquisition. Changes to the schedules caused by unexpected astronomical events (e.g., a supernova), instrument anomalies, changes in TDRS schedules, etc. must be accommodated and factored into the long-term plan and to related exposures.
ABSTRACT

Launched in April 1990, NASA's Hubble Space Telescope is actively pursuing a program of scientific observations. The Space Telescope Science Institute is responsible for conducting the scientific operations of the HST and has developed several integrated software systems which support the preparation, selection, planning and scheduling of the HST science program. In several areas, these systems have provided significant advancement in the state-of-the-art of spacecraft and observatory operations. This paper presents a discussion of these systems and lessons learned from operational experience. Much of this is applicable to other aerospace missions.

1. INTRODUCTION

NASA's Hubble Space Telescope (HST) provides important new capabilities for astronomical observation. Orbiting above the distorting effects of the Earth's atmosphere, the HST provides an unequalled combination of sensitivity, spectral coverage and angular resolution, despite a well-publicized manufacturing flaw in the primary mirror. Initial observations with the HST have returned interesting data on many objects including planets, gravitational lenses, galaxies, and supernovae. The current configuration of scientific instruments includes two cameras, two spectrographs, a high-speed photometer and astrometric sensors. A Shuttle-based servicing mission scheduled for 1993 calls for the replacement of the Wide Field/Planetary Camera with a second-generation camera and the introduction of optics to compensate for the defect in the primary mirror.

An astronomer wishing to observe with the HST submits an observing proposal which describes the scientific problem to be studied and how HST observations will be used to solve the problem, including a list of astronomical targets, instruments to be used and the duration of the exposures (see Figure 1). The proposal is normally submitted via electronic mail.

Based on the recommendations of a peer review committee, the Director of the STScI selects the proposals to be awarded observing time. In selecting proposals, the reviewers and Director must produce a science program for the HST which is balanced among different sub-fields of astronomy. While scientific merit is the most important selection criterion, the selection process must also take into account various resources which are in limited supply, such as unocculted viewing time, electrical power, and communications. In other words, a mixture of proposals which can actually be implemented must be chosen. Proposals must also be reviewed for duplication with existing HST proposals so as not to spend telescope time on redundant observations. As with other observatories, competition for HST time is keen. In the first proposal solicitation, the STScI received proposals asking for about 10 times more observing time than was available.

If a proposal is selected, the astronomer submits a more detailed observing proposal, called a Phase II proposal (to distinguish it from the less detailed Phase I proposal used for selection). This phased approach reduces the amount of work for peer reviewers and proposers who do not receive observing time.

Phase II proposals are also normally submitted electronically, and the STScI provides software to assist in proposal preparation. The Phase II proposal gives specific exposures, instrument configurations and constraints on exposures. There are a variety of scientific reasons why an astronomer might place additional constraints on exposures and between exposures. For example, exposures may be designated as acquisition or calibration exposures. Some exposures must be executed at particular times, specific orientations on the sky or when the spacecraft is in the Earth's shadow. Exposures may be constrained to be executed in a certain order or within a designated time interval. In the case of time-variable phenomena (e.g. binary stars, Cepheid variable stars) the proposer may require repeated observations at specific time intervals.

A major design goal of the Phase II proposal was to allow an astronomer to describe what is important from a scientific point of view without necessarily knowing how observations will be implemented in detail (e.g. a single exposure from an astronomer's point of view may be implemented as several sub-exposures due to instrumental or orbital considerations). The process of converting a proposer's specification into a form suitable for scheduling is called transformation. Transformation involves several tasks including determining the ordering of the observations, grouping to minimize telescope movement, instrument reconfiguration and other overheads, providing extra observations and instrument activities necessary to obtain the requested data (e.g. data readouts), etc. A requisite part of this task is allocating observations to scheduling units (de-
A CASE STUDY OF HUBBLE SPACE TELESCOPE

PROPOSAL PROCESSING, PLANNING AND LONG-RANGE SCHEDULING

Glenn E. Miller and Mark D. Johnston
Space Telescope Science Institute
3700 San Martin Drive
Baltimore, MD 21218