Astronomical Data Analysis Software and Systems IV
ASP Conference Series, Vol. 77, 1995
R. A. Shaw, H. E. Payne, and J. J. E. Hayes, eds.

Electronic Submission of HST Phase I Observing Proposals

H. E. Payne and D. J. Asson

Space Telescope Science Institute, 3700 San Martin Dr., Baltimore, MD 21218

Abstract. The process of proposing for *Hubble Space Telescope* observing time has been revised, to make it simpler for proposers. We describe the simple, flexible, and robust prototype system created to handle proposal submission. Using tools available in the public domain, we created our prototype in a remarkably short period of time.

1. Introduction

Proposing for *Hubble Space Telescope* (*HST*) observing time proceeds in two phases: the scientific merits of the proposal are considered in Phase I; only accepted proposals enter Phase II, where the observations are described in complete detail. The procedure for submitting Phase I proposals has been revised, to simplify it. Our contribution has been to implement a system for electronic proposal submission. Our goals were (1) to provide a single form for producing both the formatted, paper output required by the reviewers and a "database" (flat files, really) used by ST ScI staff to track proposals, (2) to minimize the amount of typing required by the proposer, (3) to provide a robust, secure system for archiving and analyzing proposals, and (4) to provide report generators for tracking the proposals.

The single form turned out to be a LaTeX template that proposers can obtain from a mail server, edit, and e-mail back. An accompanying style file allows proposers to format and print their proposals, since we generally require paper copies, as well (although we did run a successful pilot program for paperless, fully electronic submission). On our end, the e-mailed proposals are carefully archived and sent to a "stripper," which parses them, checks for a variety of errors and omissions, sends an acknowledgment back to the proposer, and sends a report to someone on staff. The `procmail` program allows most of these activities to be automated. Many aspects of our system were inspired by the Kitt Peak proposal handling system.

2. From the Proposer's Perspective

We asked proposers to fill out a heavily commented LaTeX template. All input destined for our electronic database is entered as tagged items, as shown in Figure 1. Proposers can process the same document to obtain formatted hardcopy.

```
%    4. PRINCIPAL INVESTIGATOR
%
%       Identify the PRINCIPAL INVESTIGATOR (PI). If you wish to
%       include your title (Dr., Prof.,...), include it in \PIfirstname,
%       as in
%               \PIfirstname{Dr. Bob}.
%
%       Use \\ to break lines in your address, as in
%
%       \address{Science Hall\\4321 University Avenue\\Anytown, MD 21218}
%
%       but limit yourself to four lines (three \\'s).

\PIfirstname    {Michael}
\PIlastname     {Shara}
\institution    {Space Telescope Science Institute}
\address        {3700 San Martin Drive\\Baltimore, MD USA 21218}
\telephone      {410-516-4543}
\email          {mshara@stsci.edu}
\country        {USA}   % Country of above named institution.
\USstate        {MD}    % 2-letter code required for US proposers only
%\ESAmember     {yes}   % Uncomment this line only if you are an ESA member.
```

Figure 1. A fragment from the observing proposal template.

The layout was designed to give time allocation committees the information they require, with a minimum of distracting clutter.

We (HEP) used the programming capabilities of LaTeX to format the printed output according to the proposer's input. For example, the observing proposal format is different from the archival research (AR) proposal format—AR proposals have a budget and do not have observing time requests. By eliminating the table of observing time requests from the formatted output we spare the reviewers from a possible distraction. Typesetting tagged items as tables was an interesting challenge; column widths and formats are calculated on-the-fly to improve readability, and the Jurriens and Braams `supertabular` style is used to break long tables. Default strings highlight omissions, and the proposer's arithmetic is checked. Although these checks are extremely simple, they did reduce errors.

3. The Processing

Proposal files received at the ST ScI are "stripped" by Perl scripts, extracting the values of the tagged items, and writing them to a file. A thorough check for missing or illegal inputs is performed, and the complete report is sent to a staff member. An abbreviated report is sent back to the submitter, as an acknowledgment. A variety of report generators make use of the information stripped from all of the proposals. In addition, we generate partially completed Phase II proposals, to save proposers from entering information they already gave us for Phase I.

We (DJA) chose an interpreted language to speed development, debugging, and testing, and at the time, Perl was the obvious choice. The entire processing system was assembled in a period of weeks. Its regular expression handling and report generation tools made parsing the LaTeX file quite easy. The report writing tools (`format` and `write`) simplified the task of generating mail messages, analysis reports, time allocation committee reports, and partially completed Phase II templates. Again, the reports vary according to the user's input. Perl's associative arrays and interpreted nature allow actual data values to invoke the routines needed to process them in the parsing or archiving steps, making it a truly data-driven system! Perl regular expressions were used to replace the user specified home institution with a canonical form.

The code is quite generic, and will run on the many platforms that support Perl. The code was adapted to process the abstracts for this meeting, and the core processing software is being used in RPS2, the new Phase II proposal preparation and submission system. In hindsight, the tools might better have been written in Tcl/Tk, to provide an easier path to expanding the system to a graphical interface for generating a Phase I proposal and submitting it.

4. The Glue

The pieces of this system are all held together by the public domain UNIX `procmail` program, a simple utility to trigger various actions via e-mail, depending on the subject line or contents of the message. Users can request proposal templates and they are automatically sent. Proposal submissions are automatically recognized, carefully archived, and sent off for processing. Bounced mail messages and "vacation" messages were automatically set aside. All activity was logged.

5. Status

From the beginning of operations until the Phase I proposal deadline, the system handled a total of 1360 e-mail messages, which included 350 requests for proposal templates, 863 proposals (not counting proposals that were corrected and re-submitted), and 48 PostScript proposals (in lieu of paper copies) in the all-electronic pilot program. Only 2 all-paper (i.e., non-electronic) proposals were received. Proposers and staff agree that Phase I went smoothly. A survey of 125 proposers found most users were favorably impressed with the proposal templates. There are still a few reservations about moving to paperless submission, however, most having to do with PostScript figures and the need for signatures.

Acknowledgments. Our work took place in the context of a committee chaired by Mark Johnston. Many improvements were made at the suggestion of other members of the committee, and it is our pleasure to thank them.