Astronomical Data Analysis Software and Systems VI
ASP Conference Series, Vol. 125, 1997
Gareth Hunt and H. E. Payne, eds.

Planning and Scheduling Software for the Hobby•Eberly Telescope

Niall I. Gaffney

Hobby•Eberly Telescope, RLM 15.308, University of Texas at Austin, Austin, TX 78712 E-mail: niall@astro.as.utexas.edu

Mark E. Cornell

McDonald Observatory, RLM 15.308, University of Texas at Austin, Austin, TX 78712 E-mail: cornell@puck.as.utexas.edu

Abstract. We present the initial design of the planning and scheduling software for the 9.2 meter Hobby•Eberly Telescope (HET). These tools are to be used by the PIs to prepare proposals for telescope time and to construct observing plans to be executed by the HET's observing queue, and by the astronomers running the telescope to evaluate and schedule the proposed queued observations. We also outline our model for the operation mode of the HET in queued observing mode.

1. Introduction

The HET[1] (Sebring & Ramsey 1994) is a 9.2 meter telescope located at McDonald Observatory near Ft. Davis, Texas. To reduce construction costs, the telescope is fixed in altitude at 55°. By changing the azimuth of the telescope, different declinations can be reached. Objects can then be tracked with an Arecibo-style tracker. Because of this geometry, the HET can access only a limited range of hour angle at any given declination and can track an object for roughly an hour at a time. Thus, efficient use of the telescope will rely strongly on software tools to know when and how an object can be observed and how best to sequence observations over the course of a night. Once completely operational, 85% of the nights are expected to be used in a queued mode where the night is dynamically scheduled as observing conditions change. In this mode, data are acquired by resident astronomers on behalf of, potentially, many different PIs for many different observational projects each night.

2. Operations

The operations of the HET (Kelton & Cornell 1994; Kelton, Cornell, & Adams 1996) is four phased: a proposal phase, a planning phase, an observing phase,

[1] The HET is operated by McDonald Observatory on behalf of the University of Texas at Austin, the Pennsylvania State University, Stanford University, Ludwig-Maximilians Universitat München, and Georg-August Universitat Göttingen.

and a verification phase. The first two phases take place three times per year, when the database driving the queue for the telescope is constructed, while the last two are executed every night data are taken.

The proposal phase consists of the traditional PI-TAC interaction, where the PI, using our planning tools, creates a telescope proposal which is reviewed and granted time by the TAC. In this phase, the PI needs to be able to predict approximately the ability of the HET to observe a typical object under nominal conditions.

In the planning phase, the TAC informs the HET operations team of the time allocated to the proposal. The HET operations team then gives the PI an account on the operations database server. The PI composes a plan, which is submitted to the database server via e-mail. The plan is checked via a procmail-style e-mail system. In this phase, the PI requires tools to determine how to make observations under both nominal and degraded conditions. Further, the PI must be able to determine what constraints need to be set for the observations, and what effect these may have.

The observing phase begins with the operations team compiling a master database of observing projects. This database is then shipped to the telescope where resident astronomers compile a plan for each night based on the current and forecasted conditions. They then execute this plan, make real time changes to it as conditions change, and gather data. Data are then shipped back to Austin via a KPNO "Save The Bits" style queued ftp transfer protocol (Seaman & Bohannan 1996). For this phase, a manual or automatic scheduling tool is needed to determine what to do tonight, based on the current and forecasted conditions, time allocation shares remaining, TAC rankings, and other constraints.

The verification phase is one in which the PI determines whether the data are as intended. The data are transfered to Austin, where they are bundled for each PI. The PIs are informed by e-mail of new data, which they then retrieve via ftp. Alternately, the PI may use a ftp mirroring package to retrieve new data automatically. The PI then examines the data and informs the operations team of any needed changes in the plan. This phase requires only that PIs be able to retrieve data via ftp and use their own data analysis tools to examine the data.

3. Software Design

Our system contains both planning and scheduling tools. Planning tools are used to determine the feasibility of observations to be made with the HET. Scheduling tools are used to sequence observations during a night and over the course of an observing tri-semester. These tools are being used by both PIs and the operations team to schedule observations on the telescope during all four phases of operations. We have developed tools under SunOS and Solaris operating systems, using freely available packages, to allow simple porting to other UNIX systems.

We have embraced two fundamental concepts in our software development: (*i*) the tools that will be used by the telescope operators/schedulers are the same as those used by the PIs, and (*ii*) that the tools should be easily modifiable as the optimal method of operating this new system becomes apparent. Thus, we have written simple tools which can be linked together to predict and schedule the telescope for any observing plan. Further, we have written Tcl/Tk (Welch

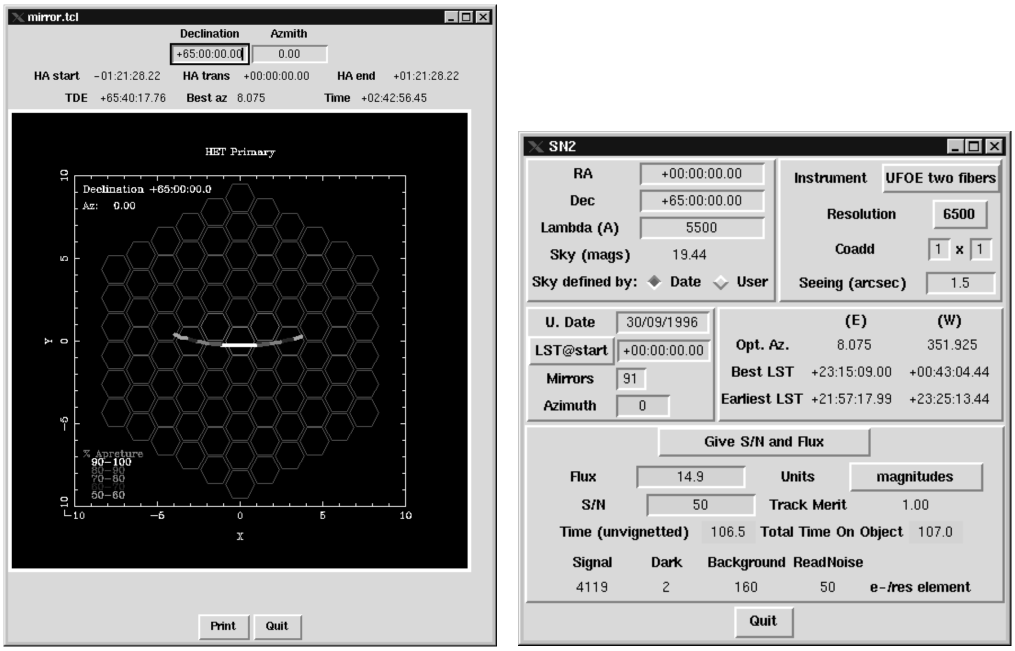

Figure 1. Two HET tools, the mirror tracking tool, which shows a potential track of the tracker across the primary mirror (left) and a Signal-to-Noise estimator (right). The tracking tools shows the TkSteal extension capturing a PGPLOT window and incorporating it into a single window environment while the Signal-to-Noise tool exemplifies the functionality of a TCL/TK GUI running over our base tools.

1994) graphical overlays to assist the user initially in creating plans. The main benefit of this system is that a user, once familiar with the system, can then create scripts in any scripting language which has access to system calls, to make plans for an entire observing program, thus reducing the time required to process a great number of objects in a GUI environment.

We used the TkSteal extension (Delmas 1996) to Tk 4.0 to minimize the number of windows used by each tool. This extension allows the Tcl/Tk script to "steal" any X-window spawned external to the Tcl/Tk script, and embed it in the Tk window as if it were an independent widget. By "stealing" windows spawned by PGPLOT (Pearson 1996), rather than creating a Tcl/Tk widget that interfaces with the PGPLOT or other graphics library, we can create a tool that is not tied to the X-windowing environment, minimize the number of windows, and allow the program that spawned the window to retain interactions with that window without any additional Tcl/Tk code. This allows us to present the user with a complete GUI in a single window, with graphics that are created by a program external to the Tcl/Tk script.

4. Project Progress Monitoring

We have also implemented a scheme to allow PIs to retrieve and examine data, and to propose future observations, in a secure environment. Each PI is given a password-protected WWW/ftp account from which data and status information can be accessed. Thus, PIs can actively monitor their projects via the WWW. Further, they can access public statistics for previous nights, such as partner share, which projects got data, weather/seeing conditions, and instrument availability. However, the PI's password is required to access all sensitive information, such as object names, positions, and setups for observations. A more primitive method of monitoring involves a modified GNUfinger Perl script, which checks to see if the PI has new data, much as the standard finger script checks for new mail.

PIs may also passively monitor the progress of their project, since an e-mail notification will be sent to the PI of each project that acquired data in the previous night. Thus, PIs need not worry about their project until data are acquired. Finally, PIs will be allowed to set up a ftp mirroring (McLoughlin 1994) script that will automatically retrieve any data that had not been previously retrieved. This script may be run as frequently as daily, or as infrequently as once a week.

Acknowledgments. We are thankful to the rest of the HET operations team members, Mark Adams, Tom Barnes, Ed Duthcover, Earl Green, George Grubb, and Phil Kelton. Many thanks go out to Frank Ray for providing us with the initial work on tracking objects with the HET. Further, we would like to thank the Project Scientist, Larry Ramsey, and the myriad of other astronomers who have tested our tools and provided us with useful suggestions.

References

Delmas, S. 1996, TkSteal[2]

Kelton, P. W., & Cornell, M. E. 1994, in A. S. P. Conf. Ser., Vol. 79, Robotic Telescopes, ed. G. W. Henry & M. Drummon (San Francisco: ASP), 136

Kelton, P. W., Cornell, M. E., & Adams, M. T. 1996, in A. S. P. Conf. Ser., Vol. 87, New Observing Modes for the Next Century, ed. T. Boroson, J. Davies, & I. Robson (San Francisco: ASP), 33

McLoughlin, L. 1994, ftp mirroring software (mirror.pl), details and code are available at ftp://src.doc.ic.ac.uk/computing/archiving/mirror/

Pearson, T. J. 1996, PGPLOT User Manual[3]

Seaman, R., & Bohannan, B. 1996, in ASP Conf. Ser., Vol. 101, Astronomical Data Analysis Software and Systems V, ed. G. H. Jacoby & J. Barnes (San Francisco: ASP), 432

Sebring, T. A., & Ramsey, L. W. 1994, in Advanced Technology Optical Telescopes V, SPIE Tech. Conf. 2199

Welch, B. 1994, Practical Programming in Tcl and Tk (New York: Prentice Hall)

---

[2] http://panther.cimetrix.com/sven/tksteal.html

[3] http://astro.caltech.edu/~tjp/pgplot/