User Interface for the Control of the Gemini Telescopes

S. S. Smith and K. Gillies

Gemini 8-M Telescopes Project, Box 26732
Tucson, Arizona 85726-6732 USA

ABSTRACT

A discussion of the interactive operator user interface developed for the Gemini 8-m Telescopes is presented. Topics include the use of a layered synthesized view of the area of interest on the sky, a data driven approach to the control of the subsystems, and an adaptive view on the health of those subsystems.

The synthesized view utilizes information from pre-existing databases; guide, wavefront and scientific detector image data; as well as operating and performance limits. This information is presented to the user in layers, with each layer containing an observer, subsystem or other logically oriented view. The ability to control which layers are presented, as well as which parameters are directly modifiable is vested in both the user and the configuration software. Implementing this above a data driven control interface encourages the use of observing templates. Exhaustive parameter control with parallel realization in the lower level mechanisms results in fast, fine grained and repeatable control.

Combining the major control interfaces, each with a different view of the desired behavior, error in behavior, and possible corrections allows the operator to spend more time optimizing observations, rather than setting up equipment. Maximizing time with quality light falling on the science detectors is a primary goal.

Keywords: Gemini Telescope Control User Interface Tcl/Tk EPICS

1. INTRODUCTION

The Gemini Telescopes operator interface is designed to provide quick access to the tools needed for normal interactive operation of the telescope and its subsystems. In addition, lower level tools are needed to provide for engineering, debugging, and the development of future capabilities. The overall design of the Gemini software system requires each principle system work package - Telescope Control System (TCS), Data Handling System, and Instruments - to provide at least an engineering interface. This allows the Gemini Observatory Control System (OCS) group to concentrate on providing a higher level of control that integrates the operation of the TCS with the other principal systems. This paper covers the current version of the system support associate (operator) interface.

A second major consideration is that the Gemini design anticipates the use of planned observing, which has implications on the type and amount of information provided to the operator. A traditional observing session consists of an astronomer providing the operator with the right ascension(RA), declination(DEC) target position, and sometimes little else. In this scenario, the observation process is basically *ad hoc*. In contrast, planned observation requires comprehensive preparation, including predetermination of guide stars, instrument parameters and telescope position. In addition, any standardized configuration information found to contribute to the efficient operation of the telescope must be provided. This information is checked for errors and completeness before the operator is allowed to proceed with the observation. Such comprehensive preparation should result in smaller effort on the part of the operator to setup the basic observation, allowing more time for fine tuning, and diagnostics. It is anticipated that both types of observation will be used in equal amounts at the Gemini telescopes, requiring the control software to cover the requirements of both.

Applications are being built using Tcl/Tk[1] and are implemented to follow the ESO Graphical User Interface Common Conventions.[2] Although several graphical user interface (GUI) builders were investigated, a decision was

Other author information: (Send correspondence to S.S.S.)
S.S.S.: Email: sss@gemini.edu
K.G.: Email: kgillies@antaeus.gemini.edu
Full color versions of the figures included in this paper are available, please contact the author for more information.

made to use direct [incr tcl]/[incr widgets][3–5] coding. Since the OCS group is charged with producing widgets that correspond to logical higher level TCS system functions, the new widgets would need to be added to the existing GUI builders. Even though most GUI builders for Tcl/Tk do not allow this capability, with the [incr tcl] improvements, it is entirely feasible to code the GUI directly by gluing together standard and custom widgets. Having taken pains to develop the widgets in a manner consistent with the ESO Skycat,[6,7] rtd, and paneledit* technologies, integration with a future version of paneledit should be possible. This should allow the use of a GUI builder during later development without delaying the progress of current TCS work.

Towards these ends, three operator interface consoles (application windows) are being developed. The acquisition console provides a simulated view of the sky including information from star catalogs, and images from the digitized image servers. The TCS console provides a display of the current position of the telescope and access to other major telescope components. The health, alarm and status (HAS) console is used to provide the operator with up to date status information on the telescope and its subsystems. Each of these top-level consoles uses pop-up windows, dialogs, forms, and integrated help, as well as providing shortcuts to the other GUI components.

An important facet of the overall Gemini software implementation is the use of the EPICS[8,9] control system in the real-time components of the TCS. This has resulted in a database type interface to all command and status items. The basic control loop will modify a number of named attributes, notify the affected systems to apply the changes and perform the actions, monitor other attributes to determine completion, and repeat as needed. The real-time systems provide control at a useful level. For example, the TCS continuously tracks a new RA and DEC simply by changing the target RA and DEC attributes and telling the TCS to apply the changes. Of course, many other named attributes affect the actual tracking of the telescope, but they are changed only as needed. This has the advantage of making most interactive commands a simple matter of setting database values; therefore, there is no need to write a script that sequences and synchronizes actions. In some instances, for example creating a mosaic of CCD images spread across an area of the sky, more control over the system sequencing is needed. These higher level tasks are accomplished using a separate observing tool[10] or by manual operation.

2. PRIMARY USER INTERFACES

2.1. TCS Console

The console that contains the traditional telescope information, the TCS console, is shown in figure 1. This console contains several widgets specialized for telescope control. The top area displays the current position of the telescope and the "Preview" or next position, which can be setup while still observing the current object. To the right of this area are several pushbuttons that bring up other important screens including the handset (figure 2), the TCS acquisition console (figure 4), and a relative chop setup form (figure 3).

The widget that controls which instrument port gets the light beam is in the center area. Most of the effort on this console has been concentrated on the TCS and its subsystems. This area will have more detail added as the instrument and adaptive optics systems are integrated. To the right of the beam control area are two sets of LED-type widgets that indicate the current status of the TCS subsystems in two areas: busy (actions are ongoing) and following (alive and ready to accept commands). Either of these two widgets may be moved to the HAS console, depending on integration and operational experiences.

Toward the bottom of the console another widget shows a cartoon view of the dome. Both the enclosure shutter (at zenith) and telescope position (45 degrees south) are projected from above on an altitude/azimuth plot. This area shows at a glance any misalignment between the telescope and the dome as well as any approaching limits. Planned enhancements include plotting object tracks of the next few scheduled observations, thus allowing the operator to use time-to-limit information in managing observations. Additionally, areas with degraded tracking can easily be indicated on this widget. As with most of the other non-text based widgets, the dome view can be resized, up to a full screen view. While such a large display might only be useful for presentation purposes, having a 10cm square display with 6-12 objects traced in RA, DEC, and time (past, present or future) has obvious value to the operator. The two wrap-state widgets, to the right of the dome view widget, show the current position of the cable wraps, anticipated time-to-limit information, and allow control of forced unwraps. Cable wrap status and control are required as the

*Paneledit is the ESO GUI builder tool.

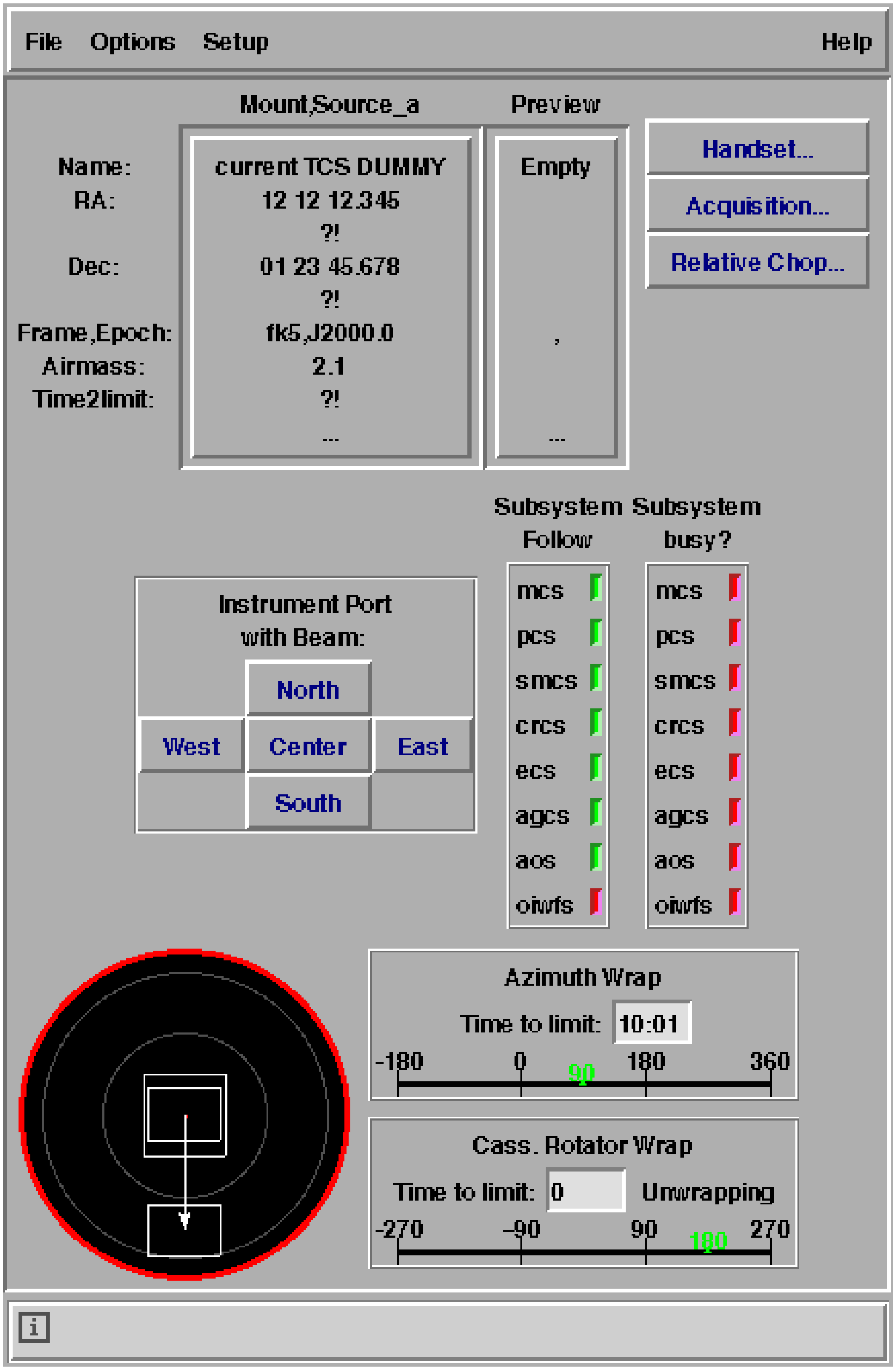

Figure 1. TCS Console

TCS will choose the shortest motion to place the telescope at a given position, even if future events cause a wrap limit to be reached. Higher level management of the wrap state can be envisioned, but is not appropriate at this level of control.

At the bottom of the TCS console is a help display widget. This widget displays one line of text with embedded bitmaps and even [incr tcl] scripts to help the user understand the widget over which the cursor passes. This is based on a similar feature used in the ESO Skycat[7] program. For instance, while the pointer is being moved over the dome view widget, the help area displays such information as the current telescope position, dome position, and limit values. The user of these tools will be a Gemini employee with significant experience, but this type of help system is expected to ease the effect that altitude can have on mental abilities. In addition, the text of these short help lines can be used to look up more detailed information from a searchable HTML documentation server with the results displayed on a standard HTML browser. This two-level linked approach to online help is expected to provide a clean and intuitive solution, given the creation of sufficient help documentation.

2.1.1. Handset

Since the electronic design of the TCS control system does not include a hardware handset for direct control of the mount, a software based handset (as depicted in figure 2) was created. This approach offers some advantages. This handset can be used to move not only the mount but also any of the wave front sensor probes, endpoints of chopping observations, fields cameras, or groupings of the above. The "Connected to:" area shows what the handset is currently driving. Source_A is the primary science position. The "..." button pops up a small form that allows selection of any or all of the individual targets or any one of the standard groupings (currently chopping, simple imaging or customized). The middle section of the handset screen consists of the arrow buttons, and two buttons to allow the offsets that have been generated by the handset to be removed. Clicking on an arrow button moves the connected targets by the current increment value. The keyboard arrow keys are bound to these actions and should prove to be useful since they allow auto-repeat. The "Absorb" button zeros out the offset but adds it back into the base position of the affected targets, resulting in no change of the actual position of the target. The "Clear" button simply zeros the offsets, which will move the target back to its original position. Below the arrow keys is a spin widget with small up and down arrows to the right side of a numeric entry field. Using the up and down arrows cycles through a list of movement increments defined by the operator. Typing directly in the entry field allows arbitrary movements when the arrow keys are activated. Planned improvements include adding a direction rosette to allow moving along a slit (or other frame of reference) and allowing for other increment units in addition to arcseconds.

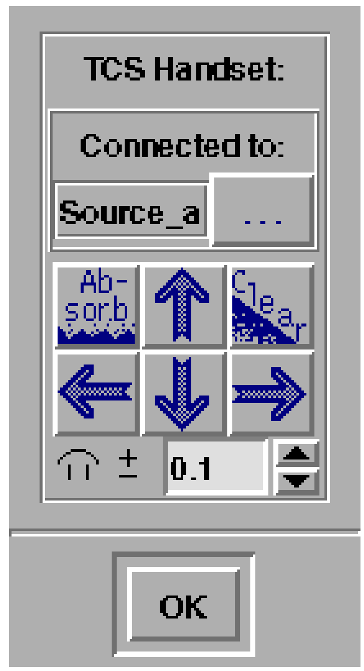

Figure 2. TCS Handset Widget

2.1.2. Relative Chop Form

As an example of the standard text based entry methods used, the relative chop form is presented in figure 3. In addition to the pointer tracking short help as described above, this form shows several other features worth discussing. Each line is a widget with a label, entry field, and a unit selector. The entry field on each line is setup to verify input that is typed in. A regular expression (i.e. to allow only numbers) and/or callback routines that can perform range checking, or other arbitrary verification can be used. For an RA or DEC value this might include checking to see if the location is currently observable. Since typing in only a portion of a value can result in a value that fails the higher level check, most instances of this widget are setup to use the regular expression check until an "Enter" key is pressed. Once a change is accepted, the background of the entry changes color to indicate that the modification has been accepted but has not yet been acted upon. After all modifications are made, the apply button at the bottom of the form will send the new attribute values to their respective systems, and the entry field background color will cycle to a third color until the motion is complete. The background then returns to the normal idle color until the next interaction occurs. The use of three colors - one for idle status, one for active modification

Figure 3. Relative Chop Input Form

and one for movement - allows the widget to display more information in a smaller area. There are over 500 attributes in the TCS system alone; conservation of screen real estate is a priority. Of course, status-only displays are created for the most referenced attributes in the HAS console.

The option menu button, which in the chop form displays units, has a label and small tic mark on the right hand side indicating a pop-up menu containing the additional units for each attribute. Callback routines are used to implement unit conversions so non-standard restrictions can be accommodated. Standard conversion routines are available for the simple one variable conversions. More complex routines are needed for converting between units which need to take into account more than one attribute, i.e. from RA/DEC to azimuth/elevation coordinates. Support for this level of unit conversion takes more programmer effort but results in an interface that can be adapted to many operational uses.

2.2. Acquisition Console

The acquisition console, shown in figure 4, is based very closely on the ESO Skycat program.[6,7] A subclass of the Skycat widget was created, and the functionality needed to control the telescope targets was added. Most of the existing functionality of Skycat was preserved, with the exception of the user's ability to draw arbitrary items on top of the image being observed. Since an indication of the field of view, probe locations, and other details was desired, the graphics overlay has been reserved for use by the software. Skycat allows the results of star catalog searches to be displayed graphically, overlayed on any FITS formated image. In particular, an area of an image can be displayed with the guide stars indicated by circles. Images can be obtained from an image server or other FITS source. This allows a display of the anticipated field of view of the telescope and easy selection of wave front sensor probe positions. Similarly, chopping requires several targets to be specified; use of a familiar drag-and-drop interface makes this relatively painless. If a selected target is a known guide star, a "snap to location" feature eliminates any error due to operator selection of the target, leaving only the pointing error of the telescope and the error in the guide star positions. By overlaying the actual wavefront sensor or guide camera images, even these sources of error can be significantly reduced.

Other features of Skycat that are useful include the builtin world coordinate system (WCS) calculations. Having a position on the screen, in an underlying FITS file, or in sky coordinates and being able to simply access it in another coordinate system allows for easy implementation of much of the functionality desired in an acquisition control GUI. Since the guide star and image access used by Skycat is based on a standard HTTP server protocol,[11]

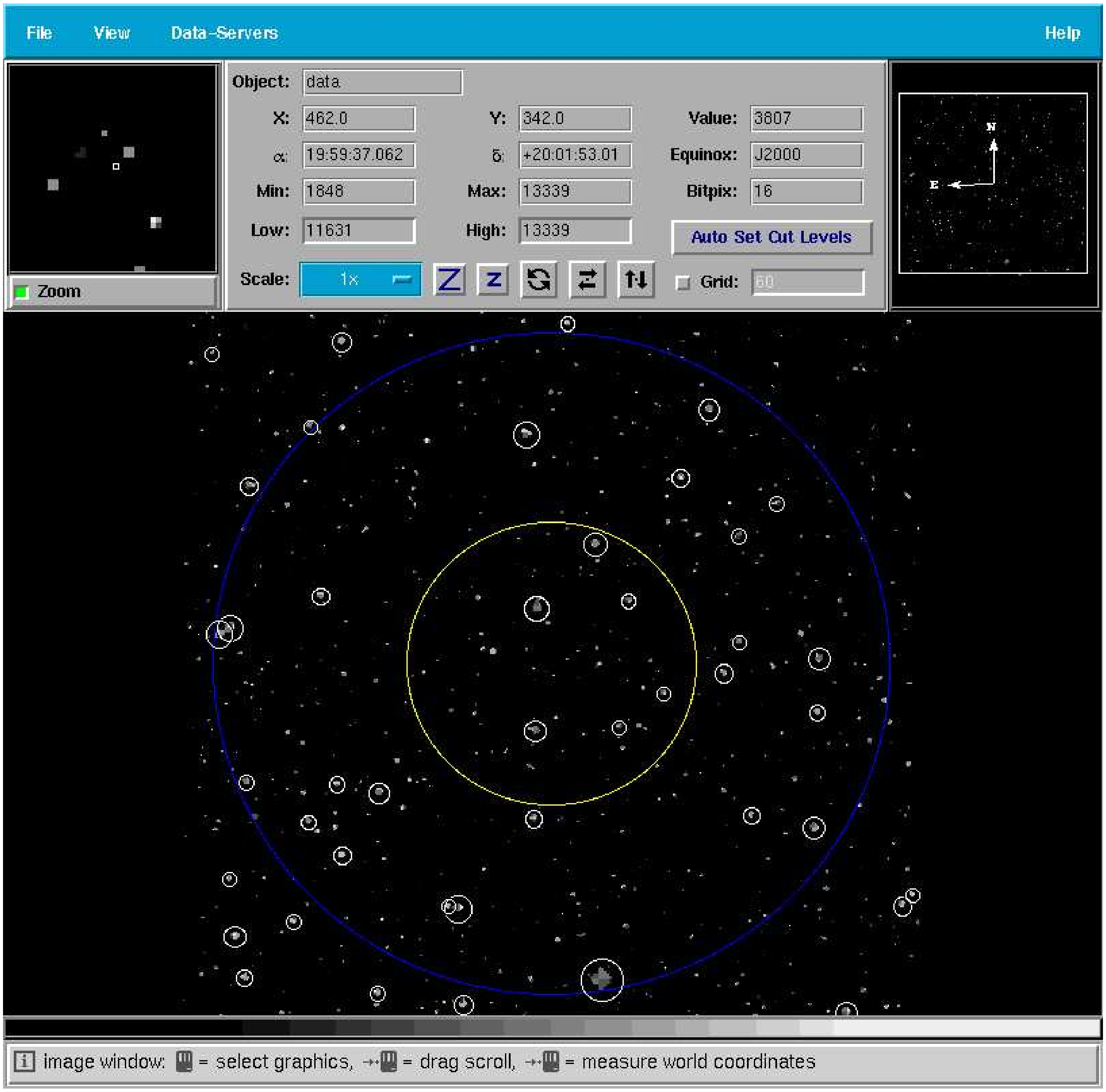

Figure 4. TCS Acquisition Console (enhanced ESO Skycat)

adding new catalogs, for example a more accurate DSS, will improve the accuracy and coverage without requiring any code changes.

2.3. Health, Alarm, and Status Console

While the user interfaces presented above are capable of controlling the telescope during normal operation, the success of implementing such a complex system requires a well informed operator. Insuring that this information is available requires presenting an appropriate level of detail; the control of that level of detail has to be available both to the observer and to the system software. As experience is gained with the system, the software needs to be able to learn or be taught more appropriate behaviors. Satisfying these goals is the primary duty of the HAS console. By presenting a view of the principle systems, their subsystems, and the individual health related attributes of the subsystems in a hierarchical display, browsing of all portions of the system is possible. By hiding subtrees of the hierarchy, it is possible to leave out the parts of the system that are working well while displaying only those elements that are failing or near their limits. This functionality is under development now.

3. CONCLUSION

One of the most important tasks facing the OCS group is to integrate the engineering GUI's provided with the delivered hardware into a seamless environment for nightly operation. For example, there are currently 32 screens provided with the alpha TCS release. An example of an EPICS screen is shown in figure 5. This form allows the setting of the science target, source_A. As a group, the engineering screens are able to control most of the planned functionality of the TCS but in a very inefficient way. Each form takes up the majority of the display screen, and most tasks require accessing several of the forms. Add to this the interactions between the various attributes, and it requires a practiced expert to control the telescope. While the delivered engineering screens have been very useful in debugging and testing the individual components, they are inappropriate for use in everyday operations.

The design and development of the Gemini Telescopes operator interface has been the result of much planning. As the subsystems are integrated and the final version of the TCS becomes available, many cosmetic changes are expected to occur in the operator applications. The basic functionality discussed here will be provided but quick rearrangements of the screen is straighforward due to the use of functional widgets that contain knowledge of their domain. This versatility will allow us to take advantage of the knowledge gained while working towards first light.

ACKNOWLEDGEMENTS

The Gemini 8m Telescopes Project is managed by the Association of Universities for Research in Astronomy, for the National Science Foundation under an international partnership agreement.

REFERENCES

1. J. K. Ousterhout, *Tcl and the Tk Toolkit*, Professional Computing Series, Addison-Wesley, 1994.
2. European Southern Observatory, *ESO Graphical User Interface Common Conventions*, 05 October 1994. Doc. No. GEN-SPE-ESO-00000-0266, Issue 1.0.
3. M. J. McLennan, "[incr tcl]: Object-oriented programming in tcl," in *Proceedings of the Tcl/Tk Workshop*, USENIX, June 1993.
4. M. J. McLennan, "The new [incr tcl]: Objects, mega-widgets, namespaces and more," in *Proceedings of the Tcl/Tk Workshop*, pp. 151–159, USENIX, July 1995.
5. "[incr tcl] - object-oriented programming in tcl/tk." URL. http://www.tcltk.com/itcl/.
6. T. Herlin, A. Brighton, and P. Biereichel, "The vlt real time display," *ADASS'95, Astronomy Society of the Pacific Conference* **101**, 1996.
7. M. Albrecht, A. Brighton, T. Herlin, D. Durand, and P. Biereichel, "Access to data sources and the eso skycat tool," *ADASS'96, Astronomy Society of the Pacific Conference* **102**, 1997.
8. "Epics documentation." URL. http://epics.aps.anl.gov/asd/controls/epics/EpicsDocumentation/WWWPages /EpicsDoc.html.
9. "Epics: A control system software co-development success story." URL. http://www.aps.anl.gov/asd/controls /epics/EpicsDocumentation/EpicsGeneral/epics_success.html.

10. S. Wampler, K. Gillies, P. Puxley, and S. Walker, "Science planning for the gemini 8m telescopes," in *Telescope Control Systems II*, H. Lewis, ed., *Proc. SPIE* **3112**, 1997.
11. M. Albrecht, M. Barylak, D. Durand, P. Fernique, A. Micol, F. Ochsenbein, F. Pasian, B. Pirenne, D. Ponz, and M. Wenger, "Astronomical server url." URL. http://vizier.u-strasbg.fr/doc/asu.html.

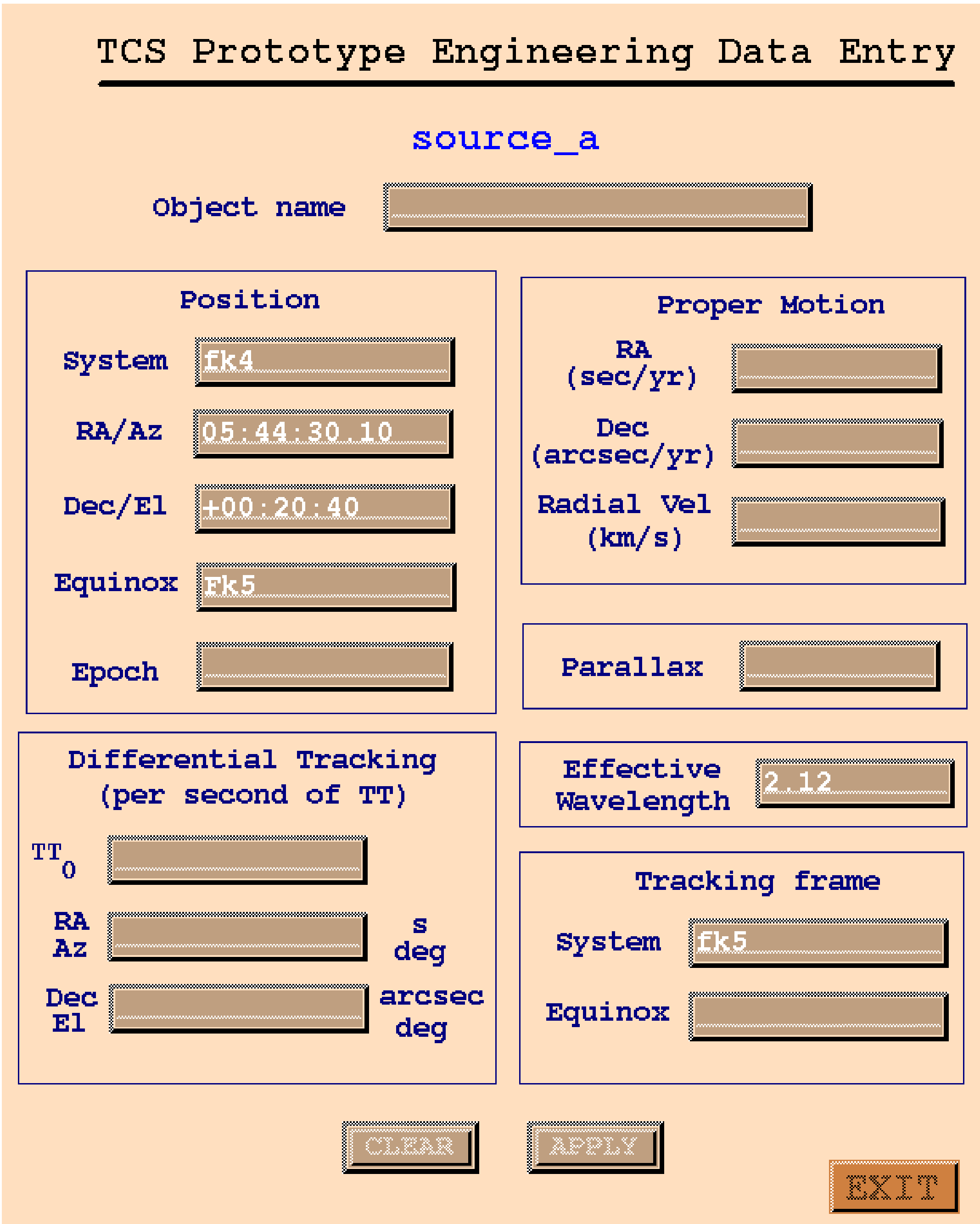

Figure 5. EPICS TCS Engineering Screen