Design of a Telescope Control System Interface

Timothy Hudson
Sam Houston State University
Computer Science Department
Huntsville, TX 77341 USA
stdtgh13@shsu.edu

Graciela González
Sam Houston State University
Computer Science Department
Huntsville, TX 77341 USA
gonzalez@shsu.edu

Abstract

A telescope control system interface provides a means of communication between a telescope and its operator. This communication link has historically been unfriendly to the average user, often relying upon very basic graphics, text-based screens, or command-line interfaces.

In this paper, we present a design for the telescope control system interface using graphical technology. Specifically, we intend to code an application in C++ that employs the OpenGL application programming interface. By taking graphical tools often used for video games and applying them to our user interface, we are able to present information in an aesthetically pleasing manner that is intuitive, concise, and flexible. When fully implemented, our interface will allow operators to control a telescope precisely by using visual cues.

Keywords

3D, Interaction Design, Interdisciplinary Design, User-Centered Design / Human-Centered Design, User Experience, User Interface Design, Visualization

Project/problem statement

Our project aims to create a replacement graphical user interface (GUI) for a telescope control system (TCS) at McDonald Observatory in the Davis Mountains of West Texas, USA.

The Hobby-Eberly Telescope at McDonald Observatory.

With an effective aperture of 9.2 meters, the Hobby-Eberly Telescope (HET) is the third largest optical telescope in the world. Its control room accommodates a resident astronomer and a telescope operator (TO), who operate the telescope jointly. While the astronomer decides what objects to observe and records the resulting data, the TO operates the telescope via the TCS interface. To be more specific, the TO physically controls telescope movement, conveys information to the resident astronomer when needed, monitors the TCS for error conditions, and records the state of many variables if and when errors occur.

Unfortunately, the current TCS interface at the HET is not as efficient as it could be. Part of the need for improvement derives from the fact that the current TCS interface was borrowed from that of another telescope and was not designed with the HET in mind.

Dr. Jim Fowler, Lead Systems Analyst for our project, asked us to concentrate on the design aspect of the TCS and produce more efficient ways of representing data available to the telescope operator. He believes the current TCS interface (**Figure 1**) is antiquated and produces too much engineering data to the screen via text. Of this data, only about 20% is actually needed by the TO. In addition, telescope operators currently need many different windows open to view pieces of information from multiple servers. Instead, he would prefer a graphical interface that neatly combines only the most necessary information.

To summarize the goals of the new interface design, we believe we can enhance the user experience for the telescope operator by:

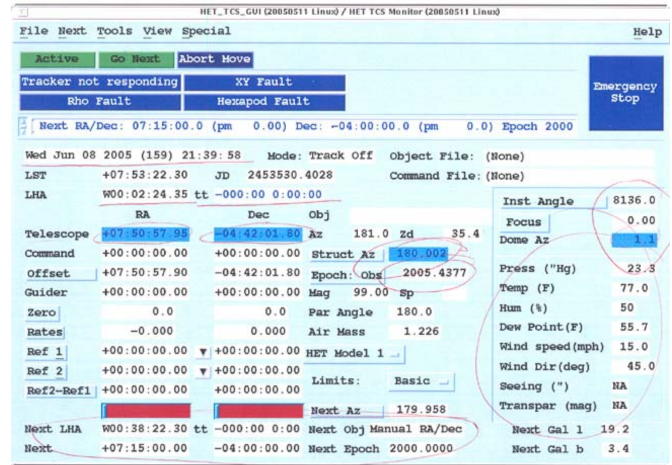

Figure 1. Current TCS interface. Only the circled information is usable for most operations. Important information on weather conditions is currently obtained by running a separate program.

- Using intuitive, truly graphical elements that are easy to use and accurately reflect information from the telescope.
- Merging into a single interface the many data streams necessary to present only and all pertinent information.
- Notifying the TO when errors occur (rather than forcing the TO continually search for error conditions).
- Keeping strict records for the TO that can be easily accessed.

Another major goal of ours was to make a telescope control system interface that could easily be changed to suit multiple users and even different observatories.

We wanted not only to design a practical user interface, but also to provide an underlying architecture with technology choices that allow for maximum system functionality. We feel that the best method to provide such an interface is by using OpenGL and the advanced graphical techniques it provides within a full programming platform (C++).

Background

Timothy Hudson is a Computer Science undergraduate student from Sam Houston State University (SHSU). He was accepted by a National Science Foundation (NSF) sponsored Research Experiences for Undergraduates (REU) program at McDonald Observatory. His main task was the design and implementation of the new interface. Dr. Graciela González, Assistant Professor of Computer Science at SHSU, served as an advisor to Mr. Hudson on issues related to user interface design and adequate documentation of results. The 10-week REU program started June 6, 2005, and endured through August 14, 2005. During that time, Mr. Hudson was under the direct supervision of his REU Advisor, Dr. Jim Fowler, Lead System Analyst.

Challenge

Given an REU program length of only 10 weeks, time constraints were tight. Our intention was to have, by our deadline, a working “mock-up” system in place rather than a paper-based design.

We chose to employ OpenGL because it is the graphics industry’s most widely used and supported application programming interface (API), and has the highest degree of platform-independence [1]. The tools identified as the best choices available for the system at the site (OpenGL, X Windows) were not ones the principal author was familiar with, thus a steep learning curve was induced. In addition, the transition entailed setting up a dual-boot system with Linux, sorting out various hardware compatibility issues, and finding the proper integrated development environment in which to program.

To complicate things further on the system side, the TCS server currently runs under Solaris, but is scheduled for a shift to Linux within the next few months. Therefore, we attempted to make our code operable on both platforms.

Familiarizing ourselves with all of the astronomical terms and measurements used in the current TCS interface proved another major challenge. Even more complexity is added by the unique design of the HET, which does not share the traditional telescope design from which the current interface was borrowed. Instead, it uses a fixed altitude system with a movable secondary instrument. This instrument is suspended over the primary mirror by a complicated tracker system that can move with six degrees of freedom.

Solution

Our solution involves determining the best design possible through end-user input, and using advanced graphical elements to implement that design.

A. Process:

We expected a visual interface to be perfect for a TCS due to the amount of information that needs to be monitored by the user. We eagerly took the opportunity to dive into the world of three dimensional

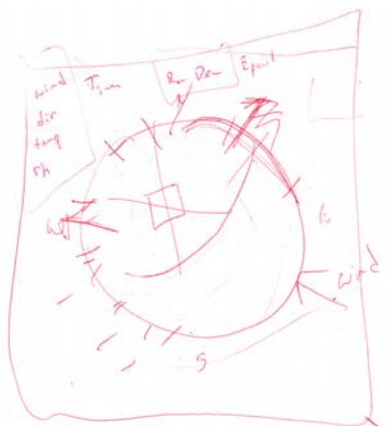

Figure 2. Sketch of a suggested interface design, drawn by Dr. Jim Fowler. He proposed that a circular design would correspond well with the degree of motion of the telescope.

programming, and provide just such a visual interface. After a lengthy discussion of the various possibilities with Dr. Fowler, he concluded that a circular design would correspond well with the degree of motion of the telescope (**Figure 2**). On that circle we could inscribe information such as viewing angle, dome angle, and wind direction.

Dr. Fowler also suggested the possibility of an image in the background (within the circle) of the stars being viewed. Unfortunately it would be a very lengthy process to get an image from one of the guide cameras on the telescope. For this project, he suggested that we might obtain images from the Digital Sky Survey instead. We also identified the need to represent weather status on-screen, and information such as the current epoch, status of the louvers, and the location of the next object to be observed. Finally, we considered drawing the trajectory of the observed object directly onto the star field.

In order to ensure that the end-user would be satisfied with the design of our new interface, the principal author observed and conducted interviews with each of the four telescope operators at McDonald Observatory. This input proved to be very beneficial to our design process. To see the current interface in use allowed us to visualize our own interface. For instance, in addition to the main TCS window, there is a “hand paddle” window for changing specific telescope orientation attributes and a “tracker XY plot” for visualizing the trajectory of an object. These extra windows tended to obfuscate the interface, and we felt that they should be integrated into the main window. Currently, the control system for the dome is operated by a completely different piece of software, which adds one more window to the TO’s screen. In the new interface, we intend for it also to be integrated.

All of the telescope operators at the HET assisted us by explaining various terms and features associated with the current interface, as well as contributing their own ideas and perspectives. Telescope operator Chevo Terrazas said that certain values (such as the most recently used focus) are lost from session to session, and it would be convenient to recall such values. We have considered adding a recall button next to each such value. Chevo also provided us with the HET warning limits and closure limits for different weather elements. Telescope operator Vicki Riley proposed a more thorough history feature – one that could display multiple history lines in a scrollable fashion. She also proposed a more automated dome control system. Telescope operator Sergey Rostopchin mentioned that all of the TOs have to keep a written log of movements made, and all of the status conditions at the moment when an error occurs. He suggested an automated logging system for errors and certain variables, as well as the possibility of capturing screenshots. Telescope operator Martin Villareal Jr. happens to be completely colorblind. We continually asked him to check the visual and contrast settings of our TCS interface. He also requested we change the way information is transferred via network from the resident astronomer’s machine to the TCS interface.

Through the interview process, we realized that each TO has a strong preference and/or need with regards to the look of the TCS. Sergey prefers his font to be a specific size, and Vicki could not read small text even if she wanted to. Martin only sees shades of contrast. Thus we needed to make the interface easily customizable. We considered per-user settings, but Dr. Fowler prefers a more standardized interface. We have decided to make such capabilities easily toggled so as not to limit the usability of the software.

B. Solution details:

In order to implement our GUI, we decided to employ Dr. Fowler's circular design (which imitates the rotation of the HET) surrounded by various other information. We decided to utilize the following graphical features, each of which is readily available in OpenGL.

OpenGL has both 2D and 3D applications. The latter, however, is arguably the prime focus of the API. By allowing the programmer to easily construct geometric primitives in memory, OpenGL facilitates the assembly of entire 3D environments. OpenGL then handles the math associated with transforming the 3D image onto a 2D screen. By representing text with geometry, OpenGL can scale and rotate text at very precise increments. In addition, OpenGL can scale images, allowing the user to decide just how large any piece of information should be. We intend to take advantage of these and other features to make the interface highly customizable.

Pop-ups and mouse-overs, although not specific to OpenGL, play an important role in the efficiency of any GUI. By hiding text information until it is needed, the overall size of the GUI can be reduced. Pop-ups are utilized to warn the telescope operator of errors or other immediately pertinent information. This negates the need for a dedicated error message space. Pop-up menus provide pertinent program options without the need to browse through other menus or remember complex keyboard shortcuts.

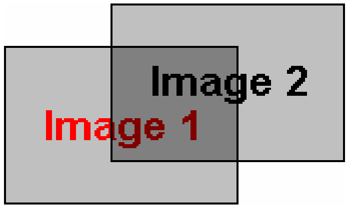

Figure 3. A semi-opaque window achieved through alpha-channel blending.

For every visual element of an OpenGL program, color information is stored in four discrete channels. The first three channels determine the overall color of the element while the alpha channel determines the opacity of the element. Alpha blending allows objects to be drawn on the screen with partial opacity, revealing any objects drawn below it as demonstrated in **Figure 3**. This feature can be applied to messages or windows so that information on-screen can be overlapped without being fully obscured, in a fashion similar to that of "augmented reality" [2].

Though not a current requirement, we would like to make a telescope control system interface that is completely configurable. By containing each piece of on-screen information in a "module," and allowing the configuration of each such module, we can produce an extremely flexible GUI. In one module, for example, we might put temperature information. The text size and colors within that module could be arranged in any fashion; then, the module itself could be dragged to any desired position on the screen. In this manner, any number of modules could be added or removed from the TCS, making it easy to upgrade in the future. For our telescope control system interface, we decided to add a weather module, a text module, a history log module, a dome status module, a radar map module, and a tracker module. We decided against modules for calibration, scheduling, guidance, data information, and maintenance. All of these functions are already managed by other programs at the HET. After arranging modules in a suitable manner, the arrangement can be saved, so as to personalize the TCS in any number of ways.

There are many ideas for the improvement of the layout of the TCS interface that we have come up with over time. Our initial design is shown in **Figure 4**.

In our design, we chose to use bars for weather sensors which get longer with magnitude and change color in a gradient fashion. When a warning condition is reached, a "light" below the bar starts blinking (after a pop-up is acknowledged). Statistics for each bar are hidden and shown using a mouse-over. The weather map utilizes scaling, allowing it to be clicked on to display an expanded view, and then re-minimized to its position on the interface. Another expandable image contains a 3D model of the dome, showing its orientation and the status of the shutter. All instruments can be scaled to utilize the desired amount of space.

All of these design aspects are combined to minimize the size and clutter of our interface. Our design is constrained only by its perceived usability, so we have done our best to make things as intuitive as possible. If our interface saves a telescope operator any amount of time, then we have potentially saved the McDonald Observatory a significant amount of money due to the high cost of telescope observation time.

C. Results:

When showing the initial mockup to Dr. Fowler, we went over each element of the TCS interface in detail, and he seemed pleased with the results. The telescope operators also expressed approval of the ideas we incorporated.

Our project implementation is still at an early stage. We have concentrated mostly on the visual aspects of the TCS interface, trying to make them as user-friendly as possible. We have partially implemented a system to add information modules to the interface with little to no program alteration. Eventually, we intend to implement the commands necessary to facilitate communication with the TCS server. A screenshot of our current progress can be seen in **Figure 5**.

We believe our design to be fairly solid given the reactions we have received so far. A broad goal of ours, however, is to examine TCS interfaces from other telescopes not here at McDonald Observatory so that we have input from external sources. We project and hope that our interface will prove useful to the HET staff for years to come.

References

[1] OpenGL Overview
www.opengl.org/about/overview.html.

[2] Introduction to Augmented Reality
http://www.se.rit.edu/~jrv/research/ar/introduction.html.

Acknowledgements

Special thanks to Dr. Jim Fowler and Dr. Matthew Shetrone for providing guidance within the REU program. Thanks to telescope operators Sergey Rostopchin, Martin Villareal Jr., Vicki Riley, and Chevo Terrazas for their time and input. Thanks to Dr. Brian Oetiker, Dr. C. Renée James, and Gigi Nevils for their support.

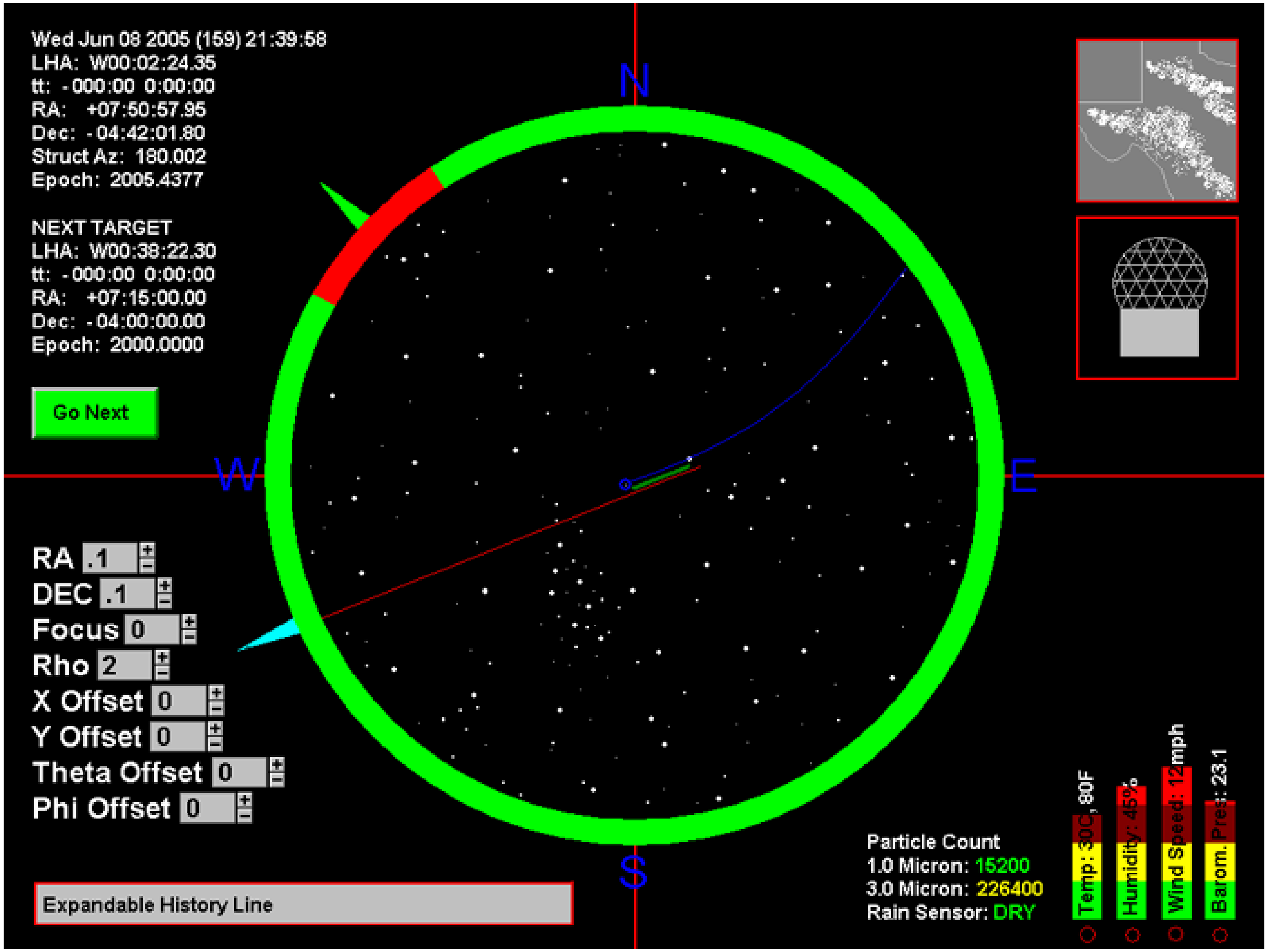

Figure 4. Proposed telescope control system interface. The central sphere indicates telescope orientation, wind direction, and dome position. Inside the sphere are the current objects in the field of view, on which tracker data is superimposed. On the right side of the sphere (from top to bottom) are an expandable weather map, a dome control system, and other weather statistics. On the left of the sphere (from top to bottom) are target information, "hand paddle" controls, and an expandable history line.

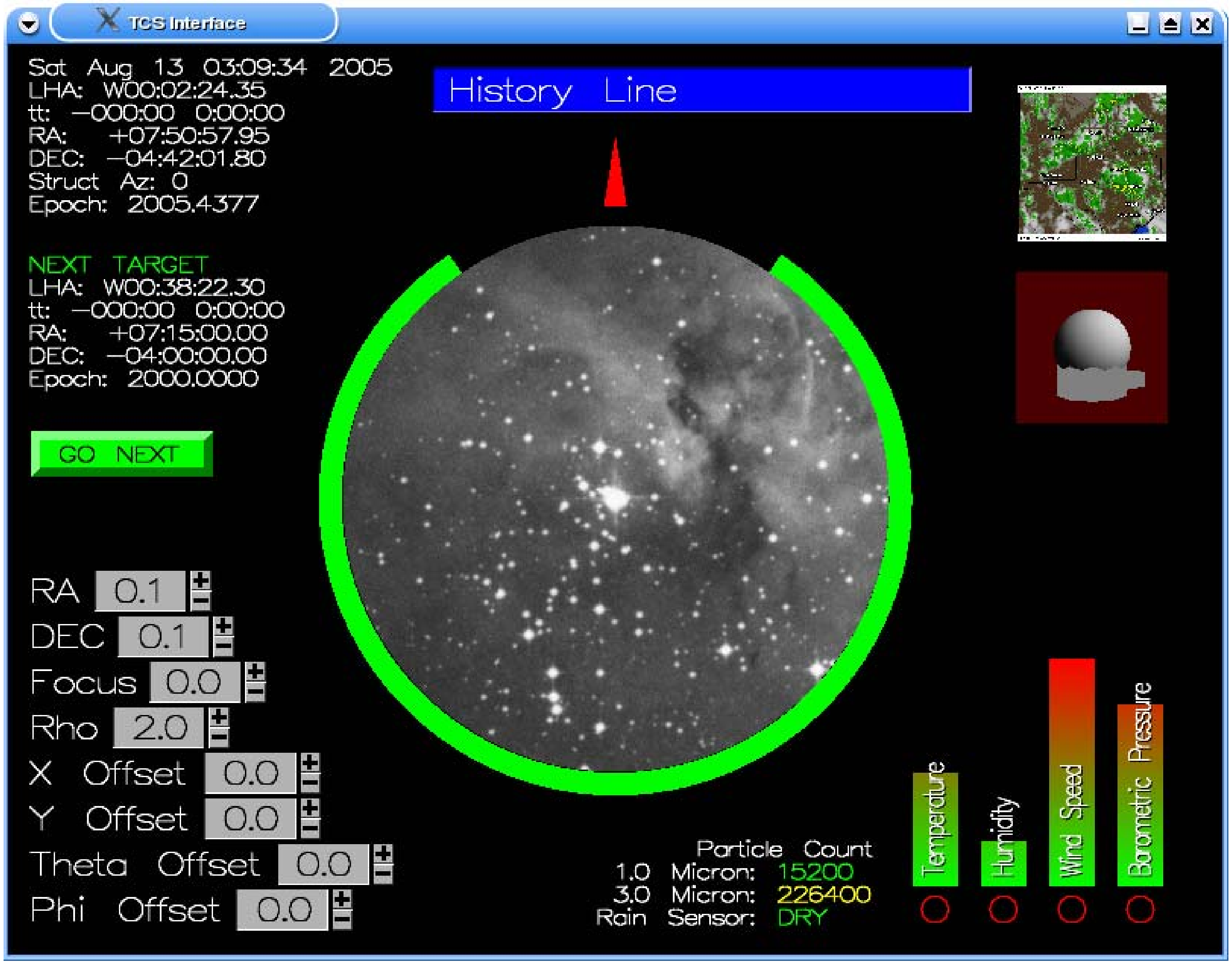

Figure 5. Screenshot of the current level of interface implementation. The interface currently incorporates text rendered via geometry, clickable buttons, pop-ups, and pop-up menus. For the bulk of the interface, flat geometry is orthogonally rendered on a single plane, giving it a 2D appearance. The central image was obtained from the Digitized Sky Survey.