RADARMAP: A MULTI-PLATFORMED RANGE DISPLAY UTILITY

Juan J. Rivero and Michael J. Rarick

ABSTRACT

RadarMap is designed to take PCM-encoded radar data and process it to display the trace of radar targets on a map of the Eglin Test Range. Written in C, X Window, and OSF/Motif, RadarMap runs on a DECStation 5000/240 and utilizes the Loral Data Gathers C functions library to directly access PCM parameters from a Loral System 500 telemetry rack. X Window (a hardware-independent bitmapped graphics display system), OSF/Motif, and the Data Gathers libraries allow portability to other operating systems that support a C compiler and these libraries.

KEYWORDS

X Window, Motif, Radar, Aircraft Control

INTRODUCTION

Until recently, real-time graphics applications were only run on mainframes using specialized graphics terminals. The evolution of computer hardware and software now allows these applications to be run on much smaller, more economical workstations. In particular, RadarMap enables a user to operate in a virtually standalone environment with only a small telemetry rack and a general purpose workstation. This configuration provides both increased security and decreased hardware costs. Its platform portability reduces system life cycle costs by eliminating software rewrites for processor and display upgrades. X Window and Motif are examples of reusable software that will carry telemetry systems into the 21st century. This paper discusses the platform independence of RadarMap, its software development and reusable design, and future plans for additional capabilities.

APPLICATION INTRODUCTION

Users of the 96 CCSG/SCW special access required (SAR) room are isolated from Central Control Facility display rooms and other visual real-time representations of objects in the Eglin Test Range. The telemetry system in use is a Loral 500 telemetry rack and DECStation 5000/240 UNIX workstation shown in Figure 1. The system provides an X Window interface for the operation of telemetry software, but the display functions of this software are not sophisticated enough to provide a graphical display of the Eglin Test Range.

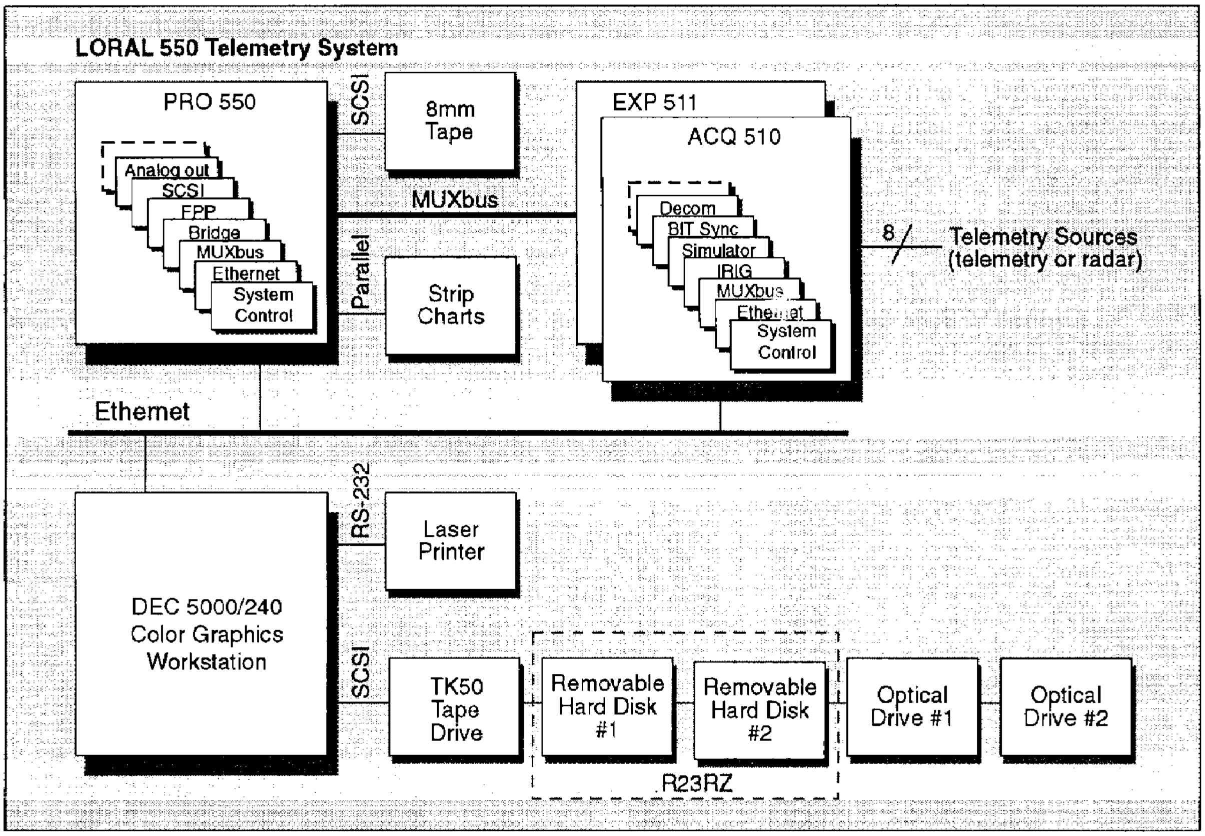


Figure 1- Telemetry System and Workstation Connectivity

In an effort to provide additional functionality to these customers, a two-person, two week task was assigned to develop a demonstration program that would use PCM-encoded radar information to display, in real-time, an object moving around the Eglin Test Range. The application was to be written using pre-existing resources on the DECStation 5000/240: a C compiler, the X Window (Xlib) object libraries, and the Loral Data Gathers C functions library. The prototype was also required to operate independent of the existing telemetry software.

A prototype was created that fulfilled all of these requirements in the allocated time. This program created a resizable standalone window, drew a color-differentiated map of the Eglin Test Range, and moved an aircraft icon (rotated appropriately to indicate bearing) over the map using real-time radar data or previously recorded PCM data played back through the telemetry rack. After the prototype was demonstrated, a follow-on task was assigned to develop additional features that would improve the display control capabilities of the RadarMap program. Later, that task was expanded with the intent that the new program could eventually be used as an aircraft control application. Work currently continues on implementing all of the necessary aircraft control functions in RadarMap.

APPLICATION DESIGN

The RadarMap software is built around a two-process design shown in Figure 2. One process makes the Data Gathers calls and processes parameter data before sending engineering unit (EU) information (latitude, longitude, altitude, etc.) to the second process via interprocess communications (pipes and message queues). The second process contains all of the X Window and Motif code responsible for opening a window, drawing the Eglin Test Range map, and drawing the aircraft icon in the appropriate location based on the data passed to it from the first process. The second process is also responsible for the user interface that provides display and data processing controls.

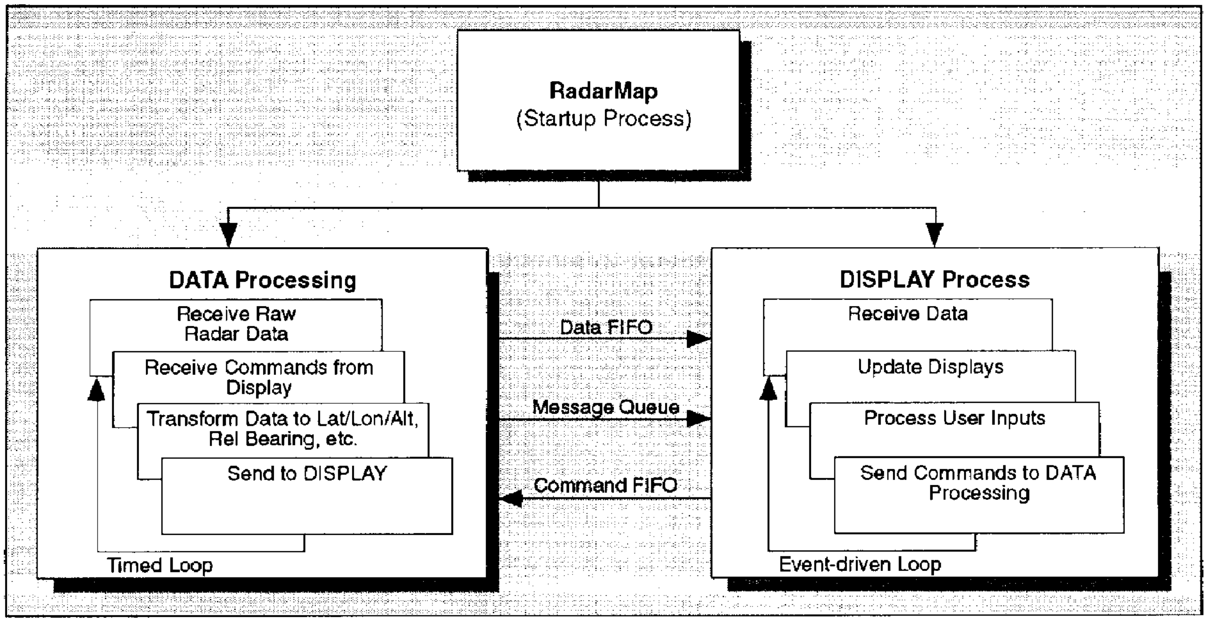
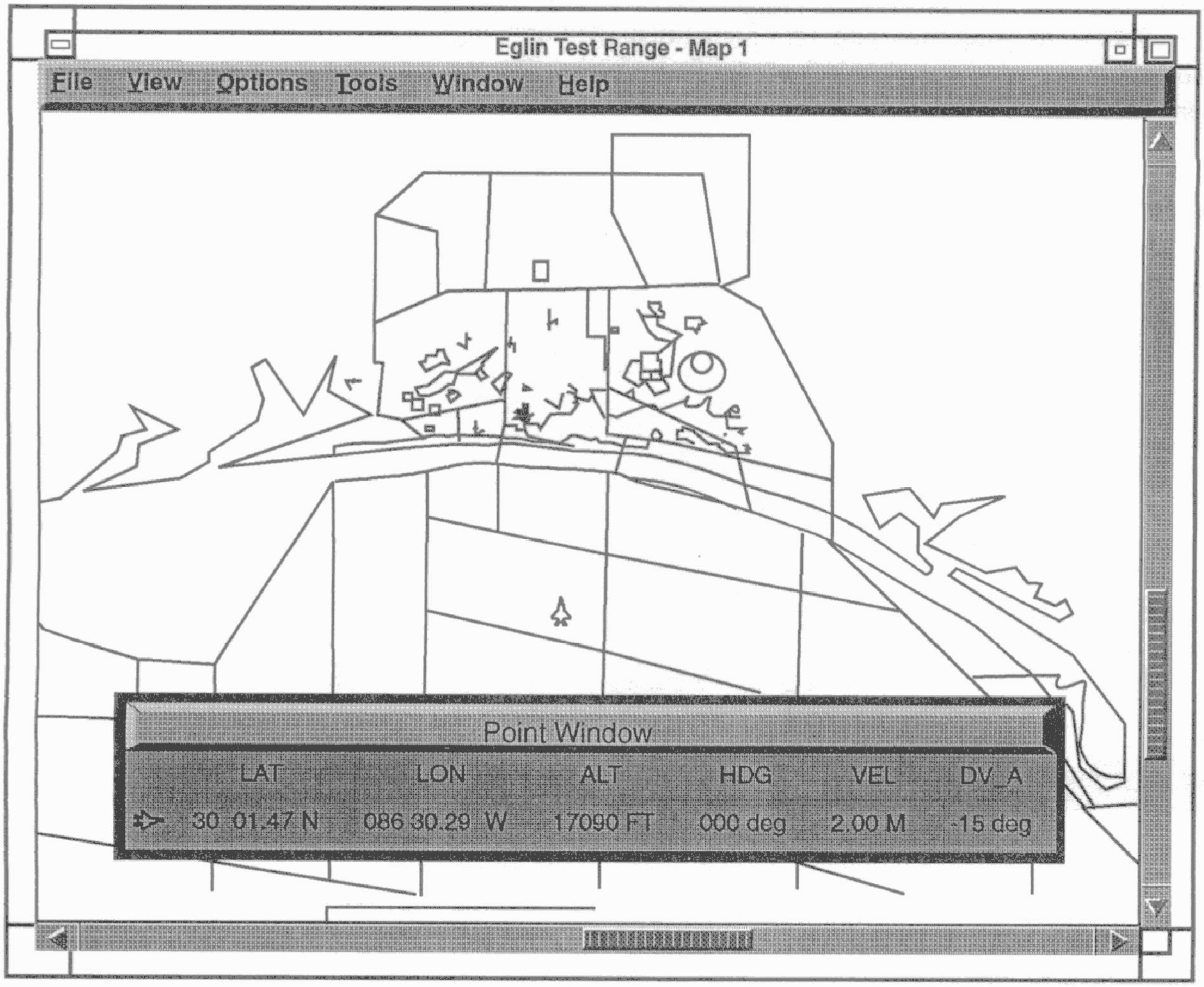


Figure 2 - RadarMap Process Structure

The Data Processing process sets up the Data Gathers calls and waits for parameter data to arrive from the telemetry rack. When data arrives, the Data Processing process applies smoothing algorithms, calculates variables, and passes a structure of processed data to the Display process. It is also responsible for monitoring the message queue coming from the Display process for control messages that effect processing procedures or Data Gathers settings (e.g., sample rate). Errors or state changes are reported back to the Display process to be acknowledged by the user. Algorithmic processing is based on a large collection of time tested routines, ported from FORTRAN to C, that have been in use at Eglin AFB for fifteen to twenty years.

The Display process contains X Window and Motif code that primarily displays shooter and target icons on top of a scrollable, zoomable map of the Eglin Test Range (Figure 3). The user controls the display functions with a combination of a menu bar, drop-down menus, and dialogs. Additional text and plot windows may be opened to supplement the information displayed in the map window. Dialogs also allow the user to control the processing of telemetry data and message reporting from the Data Processing process. Although the telemetry hardware currently restricts support to a maximum of eight streams, RadarMap is capable of supporting many more streams, if needed.



	LAT	LON	ALT	HDG	VEL	DV_A
➪	30 01.47 N	086 30.29 W	17090 FT	000 deg	2.00 M	-15 deg

Figure 3 - RadarMap Main Window

APPLICATION FEATURES

Development on RadarMap continues with current efforts involving the addition of functionality derived from the Eglin Aircraft Control Program. RadarMap, as with any other Motif application, can be run from within the Loral telemetry software or separately on any X Window terminal. The RadarMap display can also be reduced to a single icon so that the rest of the screen is visible, reducing the need for additional computer terminals.

RadarMap is controlled through the combined use of a three-button mouse and a keyboard. Either the mouse or keyboard can be used to access the menu bar for commands present in pulldown menus and dialog windows. The mouse alone can perform operations when used in the RadarMap Main Window. The first mouse button is used to activate and manipulate all of the Motif user interface elements such as push buttons, toggle buttons, scroll bars and menus. The second and third buttons can be used to zoom and display the latitude/longitude value at the pointer's current location. The Eglin Test Range map image is displayed in the RadarMap Main Window, surrounded by scroll bars to pan across the entire map image and a menu bar with pulldown menus. Auxiliary windows containing text and graphics displays can be manually opened and closed from the pulldown menus or from dialogs used to customize the appearance of each type of auxiliary window. Text windows display current shooter/target and vector pairing information. Graphic windows display tail warning and ground attack plots.

Extensive viewing commands are provided to the user to display targets and the Eglin Test Range. Keyboard short-cuts allow for quick, gradual zooming in and out of the map image, along with a dialog to rapidly move from one magnification to another. The user can center on a target or, if several targets are turned on, the center of a group of targets. Also, a pulldown menu provides a quick method for switching between up to six different map views that can contain different locations and magnification levels. Numerous controls allow the user to control the presentation of the map image itself. Each Individual map object (e.g., range boundaries, towns, roads) can be activated and have its own color assigned. Additional icons and figures can be drawn on top of the map image to denote locations or boundaries of interest such as fixed targets and shot boxes.

Each radar object has a collection of display settings associated with it, the most noticeable being the icon and color used to represent it on the map image. Two dozen icons are available for each of the eight objects that RadarMap can display, along with two trail types and a character label displayed next to the object icon. The icon can be manually sized or automatically scaled to match the map magnification level. Each object has a trail of previous locations that can denoted by dots or lines in the same color as the icon. A separate dialog controls the processing of radar information by the Data Processing process. Data sample rates are specified, along with data smoothing algorithm settings and data processing message controls.

The primary data processing of each stream of radar data involves the conversion of variables extracted from a radar stream to latitude, longitude, and altitude. RadarMap uses a table of radar site locations to translate azimuth, elevation and range from a site to calculate latitude, longitude, and altitude of the tracked item. If available, weather and boreshot data is used to adjust the radar variables for atmospheric and environmental variations. The calculated data is passed through an end-point polynomial smoothing algorithm to minimize wide variances in heading and position changes caused by inherent radar inaccuracies. Up to four target-shooter pairs can be assigned by the Display process. The Data Processing process calculates 14 different parameters between these pairs such as slant range, relative bearing, and delta altitude for aircraft control vectoring. This process also handles messages generated from the Display process to communicate with the Loral telemetry hardware via the Data Gathers library functions.

FUTURE DEVELOPMENT

Portability and reusability issues continue to influence the design of RadarMap. Future plans include adding the capability to read and process telemetry PCM words out of TCP/IP (Transmission Control Protocol/Internet Protocol) packets provided by the Eglin Standard Data System (SDS). Rather than using Data Gathers calls, RadarMap will send a TCP/IP message to the SDS to request any of the different types of radar data (in raw PCM form) that the SDS can provide. The SDS will return packets containing radar data that is then processed by RadarMap into the variables that are currently received from the Loral telemetry system. This means the RadarMap program could receive and process data from a variety of sources besides the current telemetry rack configuration.

CONCLUSION

Continued support and development of RadarMap and the subsequent Aircraft Control Program will benefit by the platform portability built into the software' s design and required resources. The use of a hardware-independent graphic display system, of a standard programming language and function/system libraries, and separation of the application from directly-connected telemetry hardware will reduce system life cycle costs by eliminating or minimizing costly software rewrites. X Window, OSF/Motif, C, and TCP/IP support is currently available on platforms ranging from desktop PCS to workstations and mainframes, making current RadarMap development work applicable now and well into the future.

REFERENCES

Heller, Dan, Motif Programming Manual, for OSF/Motif Version 1.1, Volume 6, O'Reilly & Associates, Inc., Sebastopol, CA, 1991.

Nye, Adrian, Xlib Programming Manual, for Version 11, Volume 1, O'Reilly & Associates, Inc., Sebastopol, CA, 1992.

User's Manual: Data Gather Library, Loral Instrumentation, San Diego, CA, 1992.