Fermilab

FERMILAB-Conf-03/131 June 2003

Web Tools To Monitor And Debug DAQ Hardware

Eugene Desavouret, *Member, IEEE*; Jerzy M. Nogiec

***Abstract*-- A web-based toolkit to monitor and diagnose data acquisition hardware has been developed. It allows for remote testing, monitoring, and control of VxWorks data acquisition computers and associated instrumentation using the HTTP protocol and a web browser. This solution provides concurrent and platform independent access, supplementary to the standard single-user *rlogin* mechanism. The toolkit is based on a specialized web server, and allows remote access and execution of select system commands and tasks, execution of test procedures, and provides remote monitoring of computer system resources and connected hardware. Various DAQ components such as multiplexers, digital I/O boards, analog to digital converters, or current sources can be accessed and diagnosed remotely in a uniform and well-organized manner. Additionally, the toolkit application supports user authentication and is able to enforce specified access restrictions.**

I. Introduction

THERE has always been a need for a viable debugging and monitoring system in any research and development environment. Typical solutions that are found when working with data acquisition (DAQ) equipment range from a locally run test programs, to fully automated test suites.

At the Fermilab's Magnet Test Facility (MTF), which contains multiple distributed DAQ systems in development and production stages [1]-[3], the same debugging problems exist. The simplest approach to debugging and monitoring has involved connecting a terminal to a serial port of a target DAQ computer and allowing a single user to interact with tasks via the computer operating systems' shell, which automatically excluded other interactive access through remote logins. A better method of testing was the remote login (rlogin) session. Although rlogin was limited to one user, it allowed for a remote access. Another improvement was the development and use of a remote shell daemon, which allowed for multiple concurrent sessions with the target system.

Manuscript received May 13, 2003. This work was supported by Fermilab, operated by the Universities Research Association under contract with the U.S. Department of Energy contract NO. DE-AC02-76CH03000.

Eugene Desavouret is with the Systems Development and Support Group, Technical Division Fermilab, Batavia, IL 60510 USA (telephone: 630-840-4402, e-mail: desavouret@fnal.gov).

Jerzy M. Nogiec is with the Systems Development and Support Group, Technical Division, Fermilab, Batavia, IL 60510 USA (telephone: 630-840-3081, e-mail: nogiec@fnal.gov).

The above-mentioned methods are commonly found at other facilities, too. Still, they lack an easy to use interface and are frequently not uniform in their approach, test detail level and expected sequence of actions. Sometimes, they also require a significant, and detailed knowledge of the device and operating system as well as privileges to run test programs.

The correct function of DAQ systems requires continuous monitoring to ensure their correct behavior and permit immediate recovery and corrective actions. The tools used to monitor DAQ systems vary widely from ad hoc scripts to SNMP-based tools and applications with built-in monitoring subsystems.

Engineers and technicians need a simple method for debugging and analyzing hardware installations for the many different data acquisition configurations that are in use. One can argue that a better method of solving these problems is one that allows a uniform approach to monitoring and debugging via a widely accessible GUI that is easily maintainable, and allows remote access for more than one user, from almost any platform. This would make it possible to debug or monitor operations from almost any location with access to the network, including off-site computers.

In response to these requirements, a web-based toolkit has been developed, which allows for remote monitoring of installations and computers and debugging or testing of DAQ hardware. It has an easy to use GUI working in an intuitive fashion with web browsers that allow remote and multi-user access to the single board DAQ computers. The system has little impact on the host and client operating systems, and is easily maintainable and extensible.

Fig. 1. Initial system, target, and test selection.

II. TOOLKIT

The web toolkit consists of a set of HTML pages displayed in a web browser, specialized web servers, test handlers, and support utilities. The system comprises a central web server used for selecting a test target, and specialized target web servers running on tested or monitored computers.

The user selects a system, then a computer to test, and finally a device or monitoring program (Fig. 1). The tested computer is selected either by its name or its function. The Computer Configuration System, an integral part of the Distributed Monitoring and Control System [1], provides the necessary mapping between computers and their functions inside systems.

The user-specified requests are sent to an appropriate specialized web server on the chosen target computer. The web server executes a proper test handler and sends a reply page back. Multi-frame page arrangements allow for concurrent viewing of results and selecting new tests.

The toolkit approach allows the software developers to independently develop and maintain portions of the system. Individual test module sets are independent of one another, and may be developed and included or excluded as required for differing hardware configurations. The toolkit conforms to an architecture that is parallel and independent to application architecture and therefore offers a parallel and independent way to monitor prototype and working applications, and debug components selectively, since the toolkit doesn't always use the same software configuration. It allows for easy integration with any existing system.

With short target task execution times, and with the small amounts of data handled, the impact on the rest of the system is minimal.

III. SPECIALIZED WEB SERVERS

The heart of the toolkit is a portable, specialized web server that is developed in two configurations and runs in two different environments.

The first of these web servers runs under a host operating system and is responsible for configuring the web browser frames and selecting the target web server. The second specialized web server is deployed on each of the target DAQ computers, and handles requests from the host browser by spawning a test or monitoring task and returning HTML pages generated by that task to the host browser.

The specialized servers have a limited set of allowed operations, and serve a limited set of pages from limited origins. Therefore, they are more secure than general-purpose web servers. The scope of their use is to handle the setup and communication between the host and client, spawning tasks on the client, translating parameters, and presenting data and setup options to the user.

The simplicity and limitations of the specialized web servers are an advantage because of very light processor loads and low maintenance when compared with the more complex general-purpose web servers.

IV. GRAPHICAL USER INTERFACES

A number of user interfaces have been developed. They are initialized using HTML and embedded JavaScript. Initial pages allow the user to select its target and select between monitoring and debugging.

Following are some of the GUIs and short descriptions that will allow for better understanding of the simplicity, extensibility, and ease of use of this system.

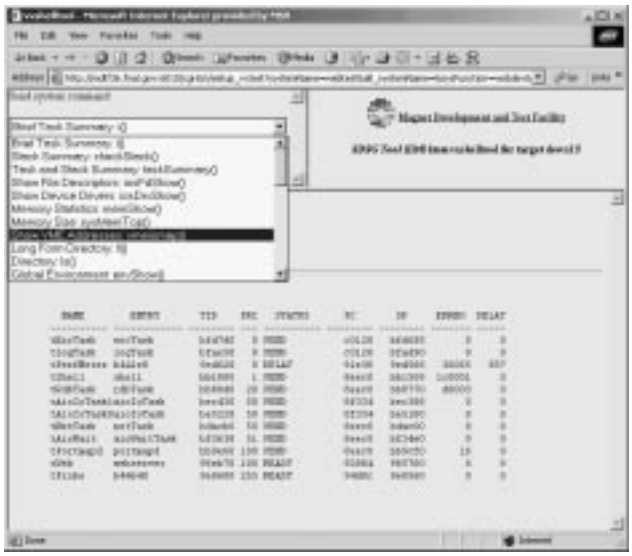

Fig. 2. The vxshelltool menu items in the upper left, with target DAQ computer output in the lower frame.

The shell tool (vxshelltool) (Fig. 2) executes VxWorks system commands on the target computer. Most of the commands are standard functions that are provided by Wind River Software. Others commands may be created and included. The vxshelltool is useful for casual as well as expert users. The tool receives the output of the target system commands and sends them to a frame in a browser.

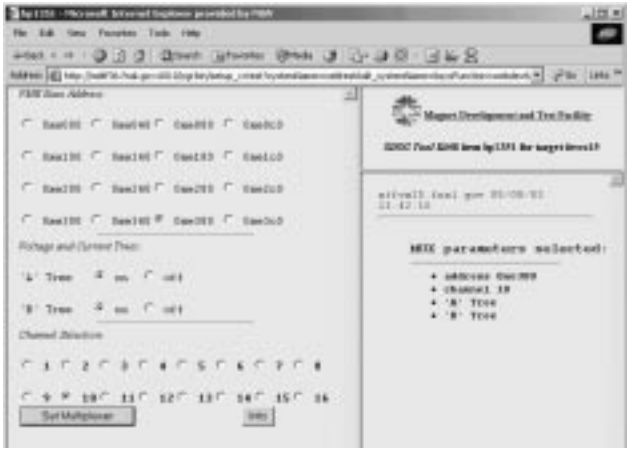

Fig. 3. The hp1351 multiplexer tool, with control in the left frame, and feedback in the lower right.

An example of a hardware test component is the hp1351 multiplexer tool (Fig. 3). The multiplexer tool was developed

to allow technicians to set channels and configurations in a Hewlett-Packard 1351 multiplexer so that the ever changing wiring configurations of measurement test stands could easily be traced and verified. Base address, channel, and channel configurations are selected, applied, and the selection parameters are returned by the target computer and displayed.

V. Debugging and Monitoring

Multiple target DAQ computers are accessible from a single host browser interface, and multiple devices may be observed on the targets by reselecting toolkit items from the start page (Fig. 1). No special logins, hardware connections, or other constraints, such as secure shells and other access services are required of the user.

Users may set up multiple monitoring operations and views by configuring several browser windows to view separate targets. The application is used to help the operator by performing spot monitoring of the target operating system state (tasks, CPU, memory, etc.), resources, device configurations and states, network configurations, and providing the user with statistical data.

VI. Extending the Tool Kit

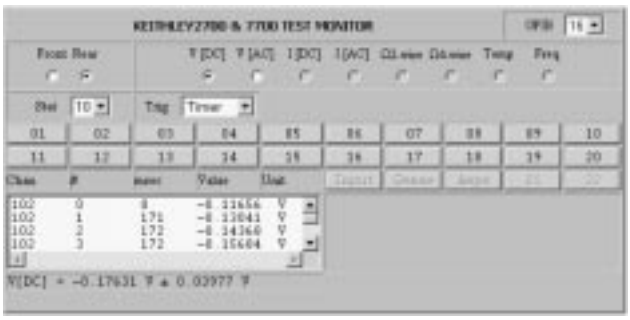

Fig. 4. The Keithley GPIB tool is used to configure and read channels from a model 2700 or 7700 multimeter.

As a part of the standard support package for a new DAQ module, not only a device driver is developed, but also a new toolkit item consisting of a toolkit test handler and associated web pages. An example of this is the latest addition to the toolkit (Fig. 4), an interface for the Keithley multimeter and multiplexer, which allows for interactive selection of channels and single or multiple readouts.

Writing new VxWorks components for new hardware tests easily extends the toolkit. The developer needs to know the API for the device, and have a rudimentary knowledge of HTML.

VII. Security

Security considerations dictate that users must be authenticated, host and client machines must be monitored and authorized, and web services offered should be limited.

Software that can control hardware (motion control, power supply activation, etc.) should be very well monitored and controlled. Remote access to systems is restricted to read-only wherever possible. Networks are restricted by the use of firewalls with specified access lists which allow only identified and trusted computers to access the DAQ network.

VIII. Summary

The toolkit provides an integrated, unified monitoring and debugging interface that is independent of application architecture. The use of the toolkit by operators and technicians obviates the need for them to have expertise with VxWorks setup and operations, and is thus a time saver. Additionally, multi-user access is possible without the need for special access permissions, and available on target operating systems that normally permit only single-user access. Remote access is enabled without the need for login services.

The advantage of using a uniform debugging and monitoring architecture is that the technical personnel do not have to learn different ways of operating the systems in order to observe or debug hardware. The host machine may use a SunOS, Solaris, Linux, or PC operating system, and any web browser. The client may run under any of the operating systems that support networking and allow services that will support the toolkit components. Presently, the toolkit has been developed for use with Unix and VxWorks target computers.

Those who write the VxWorks support programs are not required to know the inner workings of the web servers. Allowing the users to access the boards without having to use services such as rlogind, telnetd or rshd enhances the security of single board computers in the network, which is another benefit of the toolkit.

IX. Acknowledgment

Eugene Desavouret wishes to thank Jerzy M. Nogiec, Peter O. Mazur, John C. Tompkins, and Michael J. Lamm of Fermilab's Technical Division/Development and Test Department for supporting this project and allowing him the freedom to experiment and develop this and other solutions in computing.

Jerzy M. Nogiec and Eugene Desavouret also thank our colleague, Sergey Kotelnikov, who developed the Keithley GPIB Test and Monitor interface.

X. References

[1] J. M. Nogiec, E. Desavouret, D. Orris, J. Pachnik, S. Sharonov, J. Sim, J. C. Tompkins, and K. Trombly-Freytag, "A Distributed Monitoring and Control System," International Particle Accelerator Conference PAC'97, Vancouver, 1997

[2] J. M. Nogiec, E. Desavouret, J. Pachnik, S. Sharonov, and J. Sim, "An Open Distributed Monitoring and Control System," Proceedings of the International Conference on Computing in High Energy Physics CHEP'97, Berlin, 1997

[3] J.M. Nogiec et al, "A Flexible and Configurable System to Test Accelerator Magnets", PAC'01, Chicago, 2001