



Fermi National Accelerator Laboratory

FERMILAB-Conf-91/86

**Integrating UNIX Workstation into Existing
Online Data Acquisition Systems
for Fermilab Experiments**

Gene Oleynik
*Fermi National Accelerator Laboratory
P.O. Box 500
Batavia, Illinois 60510*

March 1991

- * Presented at Computing and High Energy Physics Conference, KEK, Tsukuba, Japan, March 10-15, 1991.



Operated by Universities Research Association Inc. under contract with the United States Department of Energy

Integrating UNIX Workstations into Existing
Online Data Acquisition Systems
For Fermilab Experiments
Online Support Department
Fermi National Accelerator Laboratory
P.O. Box 500
Batavia, Il 60510
Tel: 708-840-3921

Sponsored by DOE contract No. DE-AC02-76CH03000

With the availability of cost effective computing power from multiple vendors of UNIX workstations, experimenters at Fermilab are adding such computers to their VMS based online data acquisition systems.

In anticipation of this trend, we have extended the software products available in our widely used VAXONLINE and PANDA data acquisition software systems, to provide support for integrating these workstations into existing distributed online systems. The software packages we are providing pave the way for the smooth migration of applications from the current Data Acquisition Host and Monitoring computers running the VMS operating system, to UNIX based computers of various flavors.

We report on software for Online Event Distribution from VAXONLINE and PANDA, integration of Message Reporting Facilities, and a framework under UNIX for experimenters to monitor and view the raw event data produced at any level in their DA system. We have developed software that allows host UNIX computers to communicate with intelligent front-end embedded read-out controllers and processor boards running the pSOS operating system. Both RS-232 and Ethernet control paths are supported. This enables calibration and hardware monitoring applications to be migrated to these platforms.

1 INTRODUCTION

Experiments at Fermilab have traditionally based their local and online computing on DIGITAL VAXes and PDP-11s. With the availability of cost effective computing power in UNIX workstations, experimenters at Fermilab are adding such computers to their arsenal. In anticipation of the advent of these machines, Fermilab's Online Support Department has been extending its data acquisition distribution of events, messages, and control to include UNIX platforms. In addition, we have extended our software product support methodology to UNIX.

The Online Support Department supports two major Data Acquisition systems: VAXonline [1] and PAN-DA [2]. VAXonline is a set of VAX and PDP-11 based programs and drivers from which a data acquisition system

can be built. Events can be logged to either 8mm or 9 track tapes, as well as distributed to other VAXes over DECnet or point-to-point parallel links. Figure 1 is a diagram of a typical VAXonline experiment. Typical data rates are on the order of 300-400 Kbytes/sec.

Figure 1. VAXonline

VAXonline has the capability of distributing events and DA system messages among VAXes over DECnet. This has been very successful, and we desired a similar capability for successor data acquisition systems.

PAN-DA is the immediate successor to VAXonline. PAN-DA is VME/FASTBUS based, and is controlled from a VAX/VMS host. It is capable of data throughputs on the order of one Megabyte/sec.

Figure 2. PAN-DA

Figure 2 is a representation of PAN-DA. As is apparent, it consists of a variety of buses, readout controllers, memories, front end processors, and communications media (hence the PAN in PAN-DA). Data buffers can be logged to either 8mm or 9 track tapes, as well as distributed, or "Hoisted", to other computers over TCP/IP, or to the controlling VAX through the VME vertical bus, the Branch Bus. All processing components of PAN-DA may report status messages and send statistics to the controlling VAX for display and logging. Most of the PAN-DA software runs on processors running the pSOS operating system Kernel [3].

It was part of the design goals of PAN-DA to use UNIX compatible paradigms and software so that UNIX systems could be integrated with it in the future. To this end, ancillary event distribution, message distribution, and DA control are TCP/IP based wherever possible. An additional design goal was to integrate PAN-DA event and message distribution into the VAXonline system in order to reuse existing VAXonline and experimenter's analysis software. This goal enables experiments to upgrade from VAXonline to PAN-DA without having to rewrite their existing VAXonline based analysis and monitoring software.

2 EVENT DISTRIBUTION - HOIST

PAN-DA Hoist software permits programs on UNIX or VAX/VMS machines to remotely connect to the data acquisition system and acquire selected classes of data buffers (not restricted to events) at a moderate rate. Hoist uses the client/server model as implemented in the Berkeley UNIX socket interface, and is TCP/IP based. Remote Hoist clients initiate connections to a passive Hoist server. Hoist servers are multi threaded and are therefore capable of handling multiple connections simultaneously.

A Hoist client establishes buffer "requirements" during an initial dialogue. The client can then receive buffers meeting these requirements. Requirements include buffer types (numbers in the range of 0-31), a 128 bit buffer mask, and maximum buffer size. More than one buffer type can be specified per connection, but only one buffer mask (the buffer mask requirement is implemented only for the VMS Hoist server, to match the functionality of the VMS event pool). The data acquisition programs that define a buffer's requirements and the receiving Hoist clients have an implicit understanding of what the requirements mean (e.g. what a buffer of type 1 contains).

Buffers are pushed to Hoist clients with a very simple protocol. Each buffer is preceded by a fixed length header which describes the incoming buffer. This header includes the buffer length, buffer type, buffer mask, and server use statistics. The length is used by the client software to read in the buffer that follows.

The client interfaces to Hoist through a routine library. The library routines can be called from C and FORTRAN from UNIX or VMS platforms. The basic set of routines

- o set up requirements
- o connect to the remote Hoist server
- o set data byte, word, or byte and word swapping
- o receive a buffer
- o close the connection

Users call these routines to receive Hoisted buffers. The C routines are supported on VAX/VMS systems with MultiNet TCP/IP software, UNIX systems or other operating systems that support Berkeley socket extensions. The FORTRAN routines are based on the C routines, and run up against the vagaries of naming entry point symbols among the various UNIX platforms. FORTRAN Hoist routines are supported for VAX/VMS with MultiNet, Sun, Silicon Graphics, IBM Rios, and Ultrix UNIX platforms.

We also supply canned Hoist client and server programs. There is a menu driven VMS Hoist client that places buffers in a VAXonline event pool, a VMS client that dumps events to a screen, and a UNIX client that can dump events to the screen. A pSOS Hoist server serves PAN-DA buffers from VME, and a VMS "DAQ" Hoist server serves buffers from VAXonline event pools. The combination of the VMS Hoist server and client can be used to transfer buffers from one VAXonline pool to another. Figure 3 illustrates the event distribution connectivity provided by Hoist.

Figure 3. Hoist Event Distribution Connectivity

The pSOS Hoist server can serve buffers at a total rate of 180 Kbytes/second. This total throughput is limited by the ENP-10 TCP/IP Ethernet processor, which uses a 68010 processor to handle TCP/IP connections and processing. VAX/VMS microVAX II processors can receive Hoisted buffers at a rate of 150 Kbytes/sec. MicroVAX III and Sun UNIX reception rates were measured at 125 Kbytes/sec.

One Fermilab experiment is already using Hoist to transfer events from PAN-DA to an event checker program. The event checker program was easily modified from reading events from a VAXonline DAQ pool (fed by VMS Hoist client) on a VAX to receiving them on a Sun directly from PAN-DA. Other experiments will use Hoist to get live events to their UNIX platforms during the 1991 fixed target run.

3 MESSAGE REPORTING FACILITY - MRPT AND COURIER

Bringing up and shaking down data acquisition systems consisting of many diverse components such as PAN-DA, or even VAXonline, is never easy. To aid in this process, the Online Support Department incorporates cohesive, distributed diagnostic message reporting in all components of our systems. After a system is up, this facility is used for diagnosing hardware and software failure, reporting media errors, logging system activity, and recording operator comments.

VAXonline includes a distributed message facility called COURIER. Remote programs can send either text or VMS error messages to a COURIER server over DECnet or RS232 serial lines (e.g. from PDP-11s). The COURIER server displays the messages to operator screens and additionally logs the messages to a disk file.

We wished to extend the COURIER capability of VAXonline to PAN-DA in a manner that could be further extended to UNIX systems, while at the same time reusing the VAXonline software. In addition, we wished to retain the functionality of the VMS MESSAGE facility. This facility makes the VMS system error definition and reporting facility available to users. It binds error message text with a unique error message numbers and symbols such that a routine can return a message number to its caller, who can cause the text associated with the number to be displayed by a VMS routine call (or COURIER routine call).

The result of these considerations was the Message RePorTing facility, MRPT. MRPT consists of a TCP/IP based VMS server, which accepts messages sent by routine call from remote clients, and forwards them to COURIER for display. MRPT supports reporting VMS formatted message codes or ASCII strings, and stacks of VMS messages may be reported. Each message is displayed with the node name and process identifier of the sender.

Since UNIX does not provide a user extensible message facility, we use the VMS facility. Most all of our software has a VMS message facility associated with it. We have supplemented the VMS message facility with a set of commands that can take VMS MESSAGE's output and generate error message symbols in C header and FORTRAN include files, so that the errors numbers can be manipulated symbolically in these languages. We also provide a VMS message facility for displaying UNIX error message text. UNIX programs can perform a simple translation of a UNIX error to a VMS error format and report it through MRPT for display and logging.

MRPT is used extensively in PAN-DA to report error, informative, and diagnostic messages from VME, FASTBUS, and VMS and UNIX host systems.

At the current time, we support a VMS MRPT server. MRPT client software is available on VMS, UNIX, and pSOS operating systems. Figure 4 shows MRPT message reporting connectivity.

Figure 4. MRPT Message RePorTing Connectivity

4 UNDERLYING SOFTWARE

Figure 5 is a simplified block diagram of the software layers that supports Hoist and MRPT. The lowest level mechanism of interest is the TCP/IP socket driver. For VMS, the MultiNet product from TGV provides a Berkeley compatible socket implementation. For pSOS, we have developed our own Berkeley compatible (a subset) socket driver based on the CMC ENP-10 intelligent Ethernet controller's TCP/IP implementation. For UNIX we use the native Berkeley socket system (Berkeley extension or system V UNIX) [4].

Figure 5. Layered Software Block Diagram

Hoist software interfaces directly to the socket level. MRPT is based on a Remote Procedure Execution package (RPX) [5] designed specifically for PAN-DA. RPX servers are available for our pSOS based system and VMS. Clients are available for pSOS, VMS and UNIX. Work on a UNIX based RPX server is in progress.

5 UNIX PRODUCT SUPPORT AND DISTRIBUTION

Last but not least, we have applied experiences gained in developing and using software product maintenance methodology tools for our VAX based software products towards similar tools for UNIX. UNIX Product support (UPS) is a product development and release maintenance methodology and database for UNIX. Its most important aspect for the support of data acquisition software at Fermilab experiments is its ability to maintain multiple versions of a software product. Any version can quickly be made the current running version, permitting us to install new product versions at an experiment with the knowledge that if a bug has been introduced, we can quickly roll back to the previous one with little down-time.

UNIX Product Distribution (UPD) permits software products to be distributed to experiments over the network, so new releases can easily be obtained in a timely fashion. This is not an insignificant factor, since Fermilab experiments are geographically far apart.

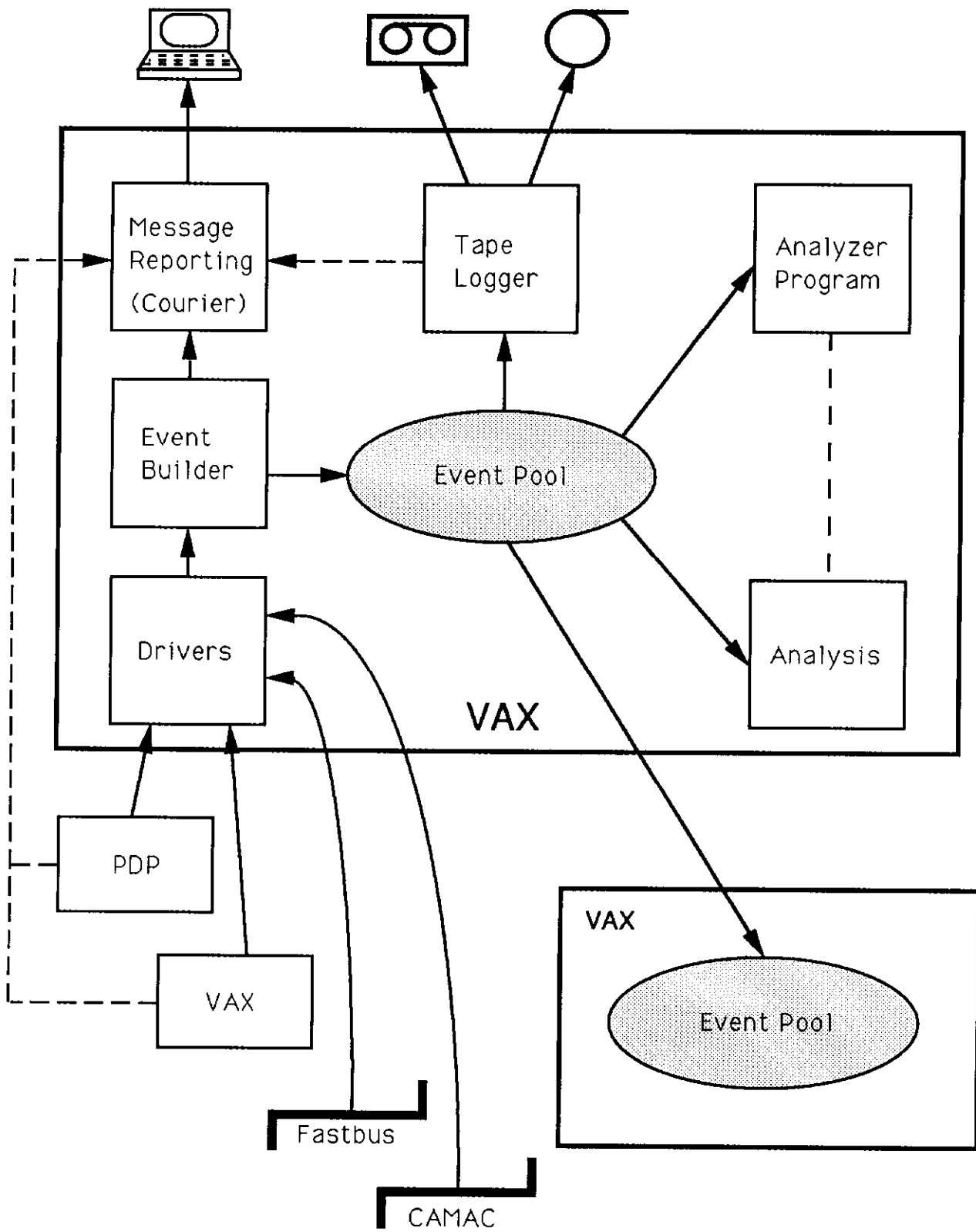
6 CURRENT WORK

In order to make the distributed message reporting facility completely uniform across all platforms, a UNIX MRPT server is needed. When an RPX server becomes available on UNIX, considerable work is still necessary before a UNIX MRPT server can be provided; the VMS error message facility functionality would need to be ported to UNIX, and COURIER like display and logging functionality needs to be provided. These are not urgent concerns to us at the present time.

We are currently in the early design phase of a UNIX project called "commissioner". Commissioner will be a UNIX based program with the ability to receive event data from a variety of sources (only Hoist at first), perform transformations on the data with "event factories", and histogram and plot the derived data with CERN HBOOK and HPLOT. We are planning to use the BuilderXcessory interface builder to make a user friendly X based graphical user interface, and C++ for implementation [6]. We hope to make the data source portion of the resultant product extensible by use of object oriented techniques so that we may easily add other sources such as CAMAC, enabling commissioner to be used as a back end for a "small" data acquisition system.

References

- [1] V.White, et al, "The VAXONLINE Software System at Fermilab" IEEE Transactions on Nuclear Science, Vol.NS-34, No.4, August 1987.
- [2] D.Petravick et al, "The PAN-DA Data Acquisition System", IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.
- [3] D.Berg et al., "A Real Time Integrated Environment for Motorola 680xx-based VME and FASTBUS Modules" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.
- [4] M.Bernett, G.Oleynik, "Uniform Communications Software Using TCP/IP" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.
- [5] E.Berman, D.Petravick, G.Sergey, "Remote Procedure Execution Software for Distributed Systems" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.
- [6] G.Oleynik, "Object Oriented Design and Programming for Experiment Online Applications - Experiences with a Prototype Application", accompanying paper at this conference.



PANDA

