

FERMILAB ACCELERATOR APPLICATIONS MIGRATION PROJECT

S. L. Lackey and F. X. Zhang, Fermi National Accelerator Laboratory*, Batavia, IL 60510, USA,

Abstract

The present accelerator control system at Fermilab includes over 500 specialized programs and 146 instances of the parameter page application with specific configurations. These applications were originally written in either Fortran or the C Programming Language under VMS. As the central tier of the control system has been replaced with Java™ processes running under Unix and Windows and a framework has been developed for applications programs in Java™, the large project of replacing these programs has begun. This is a report of the process and progress made with examples of improvements and consolidation of functionality.

BACKGROUND

The present Accelerator Controls Network, ACNET, was developed in the early 1980s at Fermi National Accelerator Laboratory to control the Tevatron and its injector machines. Ten years later, the PDP-11 computer consoles were replaced with Digital Corporation MicroVAX computers utilizing the X-windows protocol. [1] The use of X-windows has resulted in console instances running on desktops as well as in the control room.

The new century brought another console upgrade to personal computers running Microsoft Windows 2000 and displaying VMS applications via X-windows. These newer style consoles allow both old and new applications to be displayed simultaneously on one machine.

MAJOR OBSTACLES

Number of Applications and Libraries

In addition to the more than 500 applications and numerous console library functions, there are 37 user libraries each with dozens of library functions that are available to developers in the existing system. Libraries contain code to interface with various systems such as refrigerators, beam position monitors, programmable logic controllers, oscilloscopes, analyzers and other specialized equipment.

The task of replacing this vast number of lines of code certainly seems daunting.

Limited Resources Available

The Accelerator Controls Department has a group of ten full time applications programmers who often are consumed with maintenance of existing programs. Operators and Physicists have traditionally often contributed specialized applications but the number of people outside of the department who are actively programming applications has fallen over the last decade.

*Operated by Universities Research Association, Inc. under contract with the U.S. Department of Energy.

Artifacts of Ancient History

Since many of the applications were written originally to run on machines that had very little memory, often the code was heavily overlaid. This creates complexity in the code that is no longer needed and makes reading and understanding the code more difficult. Since it is also highly likely that the original author is no longer available to explain why he or she may have done things a certain way, often code has been left untouched that is no longer useful or even functional.

Operating During The Upgrade

The fact that we are in the middle of a very important collider run is both an obstacle and an advantage. It may be difficult to fully test the setting functionality of some programs since shutdowns are infrequent. Conversely, readings can be easily compared with those displayed by the old applications. Operators can easily compare the information available from both old and new applications thus increasing confidence in the accuracy of the new applications. Conversely, old habits die hard and as long as the old program is readily available, some will shun the new.

Collider and fixed target operations also result in more maintenance responsibilities for our applications programming group. Since the old applications are still being utilized daily, they must function properly and keep pace with any hardware changes in the field. If at all practical, an application is replaced rather than modified.

ATTACKING THE OBSTACLES

Consolidation of Applications

The first step has been to go through the existing applications to categorize each program. This process has helped to identify duplication of functionality and programs that are closely related and therefore could be consolidated into one more general program.

More difficult is the identification of obsolete or unnecessary programs. Just because a program has not been used recently doesn't mean that it is no longer needed. It may serve a very important yet infrequent function. Such programs will most likely be the last to be replaced. Recently, a working group has been organized to gather requirements from other Beams Division departments to clarify which programs are indeed needed for the future and which may not need to be ported to the new applications framework.

Some of the functionality of libraries routines are no longer necessary as the Java Language and Java Foundation Classes replace many of the library routines for the graphical user interface programming. Java also has Structured Query Language functionality included in the language.

Recruit and Train Personnel

We have successfully recruited help from almost all of the Beams Division Departments as well as from the Computing Division here at Fermilab. Tutorial presentations were made and users guides have been made available to prospective developers. In addition, summer students have been able to contribute in the short time they are here since Java™ is emphasized in the classroom. A summer student needs to work closely with an experienced accelerator programmer to be effective.

It has been somewhat more difficult to entice veteran C and C++ programmers to learn Java™ but those who have rapidly become productive contributors. The transition is not difficult for C programmers as the syntax is very similar. The biggest adjustment is going from a procedural language to an object oriented one.

Use of Open Source Libraries

Various open source projects are available and several have been used in our new applications. While we have attempted to use widely accepted libraries in order to minimize risk. We have had some experiences with packages that have bugs introduced in new releases which may not be quickly resolved if they involve the less popular features of a package. More stable products are less likely to require major changes in coding with a new release but there are no guarantees. Often the reporter of a bug is encouraged to contribute the fix to the project.

Presently, we are using JfreeChart [2] and Plotmy [3] plot packages. Use of these packages has saved many hours of programming effort.

Request and Implement User Feedback

In some cases, pop-up messages with links to the new program have been added to the VMS versions to encourage use of a replacement program. This has proven to be very helpful and has resulted in many useful suggestions from users. Also, a logbook style message board has been implemented to announce changes or improvements to applications and for users to report bugs or suggestions for enhancements.

People can become very excited about using the new application when they have been asked what they don't like about the present version and see their suggestions actually implemented in the new version.

EXAMPLES

Quench Protection Applications

There are in total twelve console applications that have to do with quench protection in the Tevatron. In the past, some quench protection applications contained information from both the Tevatron main bus magnets and the low beta magnet circuits and others were specialized. After consulting with operators, it was decided to completely separate the two types of quench protection monitor programs as the operators generally do not work with these systems simultaneously. The twelve

applications are being replaced by two applications, one for the Tevatron main bus quench protection monitors and one for the low beta quench protection monitors. At this writing, the low beta application has been deployed and the Tevatron version is still under development.

The basic layouts of the various VMS applications have been preserved when practical to help users make the transition to using the newer programs. There is a main program from which the user can launch frames that replace the functionality of several applications. This grouping also simplifies access to all applications useful for dealing with quench protection issues. With the new system all applications could be running simultaneously on one console. Previously, only three primary applications and three secondary applications could be running concurrently on one console instance. In the example below one can see the menu bar and tool bar which is part of the accelerator application framework [4] and therefore a part of every application.

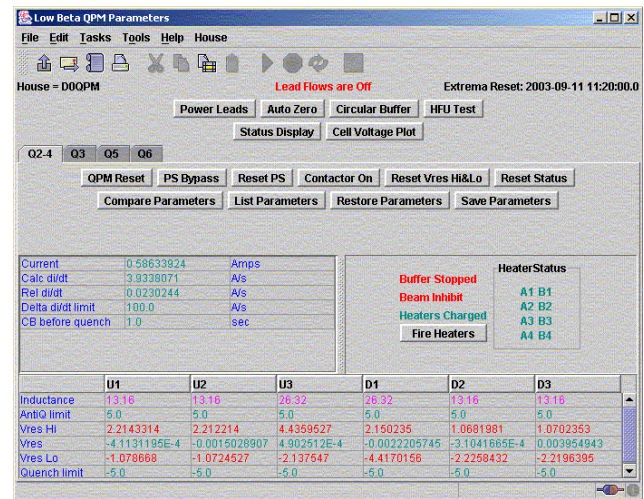
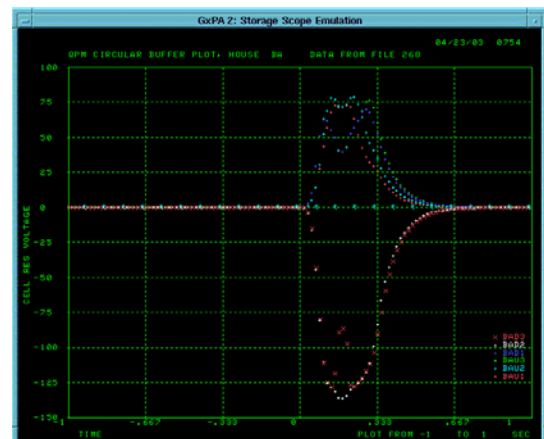


Figure 1: The Low Beta QPM Program User Interface

The following two plots are examples of circular buffer plots from the old and new low beta quench protection applications. The data is similar though not identical.



The new plot is using the Ptolemy plot package from the University of California Berkeley and has mouse zooming capability in addition to its more modern look. The user may launch as many circular buffer plots as needed with the new framework. The VMS application limited the user to one circular buffer plot at a time.

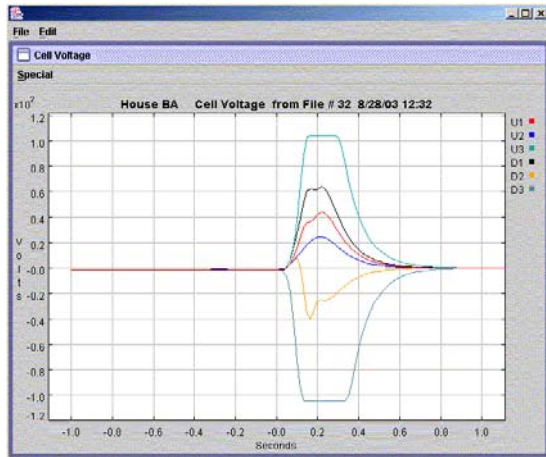


Figure 3: Similar Plot Using Java Application

The Parameter Page

The parameter page application continues to be the most widely used application in the Fermilab accelerator controls system. It was one of the first applications to be written for the new applications framework. Each instance of the parameter page was permanently assigned to a particular slot of an index page and could be configured to display a specific set of parameters. These configurations have been preserved in the new system and the familiar index page identifiers have been retained.

The parameter page has been replaced with two applications: the Parameter Page and the Parameter Page Lite. The Parameter Page is useable only from within the Accelerator Controls network. The Parameter Page Lite is a servlet available to users off-site.

CONCLUSIONS

As of this writing, there are 96 applications deployed in the new framework. The initial emphasis has been placed on writing new applications rather than replacing existing

applications. Of the 96 applications written so far, 68 of these have completely new functionality.

When replacing an existing application, rather than study the existing code, which may be very old, poorly commented Fortran code, it is more effective to interview users to see how they use the program, what other programs or information are often used in conjunction with the program and perhaps should be incorporated in the new version. The actual operation of the accelerator complex may have changed substantially since the functionality of the original program was conceived.

Veteran users often request that the interface resemble as much as possible the old familiar user interface. Operators with less experience prefer the more modern look and feel of the new framework environment. A delicate balance needs to be achieved in order to encourage use of the new applications by all users. As soon as practical, the old version should be removed from the controls system otherwise maintenance will be doubled.

ACKNOWLEDGEMENTS

Many people are contributing to the effort. Especially important has been the work of Andrey Petrov who has developed the accelerator controls framework which is the basis for all of the new applications. Timofei Bolshakov, the first developer of new applications for this project, has developed a majority of the applications used for the collider Shot Data Analysis, SDA [5], project. Timofei and Andrey have collaborated on the Synoptic Display application[6]. Also, Carl Schumann has written examples and guidelines for new developers.

REFERENCES

- [1] P. Lucas, et. al., Modern Operators' Consoles for Accelerator Control at Fermilab, PAC 1991, pp. 2523-2525.
- [2] <http://www.jfree.org/jfreechart/index.htm>
- [3] <http://ptolemy.eecs.berkeley.edu>
- [4] <http://www-bd.fnal.gov/appix/af-guide.pdf>
- [5] <http://www-bd.fnal.gov/sda>
- [6] T. Bolshakov, et. al., Synoptic Display—A client-server system for graphical data representation, ICALEPCS 2003, Gyeongju, October 2003.