GA–A24070

# HARDWARE AND SOFTWARE UPGRADES TO DIII–D MAIN COMPUTER CONTROL SYSTEM

by
D.A. PIGLOWSKI, B.G. PENAFLOR, B.B. McHARG, Jr.,
K.L. GREENE, R.M. COON, and J.C. PHILLIPS

SEPTEMBER 2002

# DISCLAIMER

GA–A24070

# HARDWARE AND SOFTWARE UPGRADES TO DIII–D MAIN COMPUTER CONTROL SYSTEM

by
D.A. PIGLOWSKI, B.G. PENAFLOR, B.B. McHARG, Jr.,
K.L. GREENE, R.M. COON, and J.C. PHILLIPS

This is a preprint of a paper presented at the 22nd Symposium on Fusion Technology, September 9–13, 2002, in Helsinki, Finland, and to be published in *Fusion Engineering and Design.*

Work supported by
the U.S. Department of Energy
under Contract DE-AC03-99ER54463

GA PROJECT 30033
SEPTEMBER 2002

# ABSTRACT

The complexities of monitoring and controlling the various DIII–D tokamak systems have always required the aid of high-speed computer resources. Because of recent improvements in computing technology, DIII–D has upgraded both hardware and software for the central DIII–D control system. This system is responsible for coordination of all main DIII–D subsystems during a plasma discharge. The replacement of antiquated older hardware has increased reliability and reduced costs both in the initial procurement and eventual maintenance of the system. As expected, upgrading the corresponding computer software has become the more time consuming and expensive part of this upgrade. During this redesign, the main issues focused on making the most of existing in-house codes, speed with which the new system could be brought on-line, the ability to add new features/enhancements, ease of integration with all DIII–D systems and future portability/upgrades. The resulting system has become a template by which other DIII–D systems can follow during similar upgrade paths; in particular DIII–D's main data acquisition system and neutral beam injection (NBI).

# 1. INTRODUCTION

The DIII–D National Fusion Facility provides an experimental environment for high temperature plasma and tokamak physics. The facility, located in San Diego, California, on the campus of General Atomics (GA), is in the continual process of upgrading it's computing resources in order to remain current with state of the art technologies. This upgrade also continues the process of creating an "open" software system which began at DIII–D back in the mid 1990s [1]. These upgrades allow the facility to remain productive and meet an ever increasing list of research demands.

At DIII–D, main sub-systems, i.e. neutral beam injection (NBI), plasma control and main data acquisition, are controlled by individual computer systems. Coordination is accomplished via several methods but overall control is handled by a central system. This central system has been the target of a major upgrade over the last year. The existing system has been used successfully for over half a decade and was part of a multi-system replacement started back in 1993 [1]. In this age of almost instantaneous obsolescence, this system would be considered by today's standards to be antiquated and past the end of it's useful lifetime. The hardware is VME based; difficult and very expensive to maintain. Additionally, the operating system (REALIX) is proprietary and is no longer supported by the vendor. Third party control system software, which is critical to the functionality of the system, is also no longer supported by the original vendor as well. The initial phases of this upgrade have targeted this system for all the above reasons as well as to expand capabilities and increase the reliability.

This paper will discuss the work to date on the upgrade of DIII–D's central operational computer system; both hardware and software. Motivation for the upgrade, some broad goals and conceptual ideas for the how/why the upgrade has been implemented are also presented.

## 2. MOTIVATION, CONCEPTS AND GOALS OF THE UPGRADE

The upgrade of the main operational control system has focused on providing the DIII–D operators with an efficient and reliable computer interface to the many components and parameters required to produce a plasma discharge. It is expected that this new system will have a lifetime well past the next few years. However, emphasis has also been given to future upgradability such that future upgrades would not be such a drain on both manpower and cost.

Learning from the pitfalls of the previous system, the goal of the upgraded system was not merely to replace antiquated systems but instead to correct many of the shortcomings in the process [2]. A key motivation was to remove or miminize the dependence upon third party and proprietary software. Such dependencies had been inefficient and costly in the past. What turned out initially to be a time saver eventually grew into a set of separate problems; i.e. non-portability, failures of the software to perform as advertised, no control over needed bug fixes and alterations in the code. These problems worsened as the old system aged and was pushed harder. The lesson learned was that unless a product was widely used in other public applications or the source remained open (available to all at no cost), the products should be avoided.

Another key motivation was to move to a hardware platform which is widely used such that parts could be easily obtained at a reasonable cost. Eventually should this system turn out to be viable, it could serve as a template for other DIII–D sub-systems, mainly data acquisition and/or NBI. With this in mind, a major concept for software design was that it be modular, flexible and somewhat simplistic. Software development should take full advantage of existing in-house codes as well as any openly shared software. It was foreseen that this would help bring down the manpower needs for both development and on-going maintenance of the over all system.

## 3. HARDWARE CONFIGURATION AND OPERATING SYSTEMS

Initial considerations were given to the choice of computing hardware and corresponding operating system (OS). These two choices were obviously tied together since one could have a limiting impact upon the other. Substantial improvements in computer technology, both in overall performance and reliability over the last few years have made replacement of the aging VME system very attractive rather than utilizing the existing cpu, hard disks, chassis, etc. The performance of the old VME machine, although adequate at the time of purchase, pales by comparison to present day machines. Adding to the decision to replace was the cost of much of this new hardware, regardless of the platform, would be but a fraction of the original. Of the possible operating systems, a Linux system appeared to be the most promising candidate. This was because of a familiarity with other UNIX environments and because Linux exists inside of a global atmosphere of open source and code sharing; both crucial concepts that were being adhered to. Linux is also becoming increasingly popular and has, over the last few years become more user friendly and robust. With Linux decided upon for the OS, the possible hardware configurations were narrowed and the eventual system was mapped out to be single computer with a standard PCI bus; in our case a single Intel Pentium IV. This hardware and OS combination is identical to configurations being implemented with the plasma control system (PCS) upgrade, undertaken at the same time as this upgrade [3]. All the components in the final configuration, with the exception of a single I/O interface board are mass produced off-the-shelf equipment.

DIII–D has a heavy investment in CAMAC equipment. Individual CAMAC modules have been used here for decades and have been relied upon for control, single point input/output, timing, clocks and waveform data acquisition [4]. Replacement of these modules, in whole or in part, was not the scope of this upgrade. Other peripheral hardware which the main system needs to communicate with can be done through ethernet or RS-232 connections. However CAMAC equipment and the interface to them was a limiting factor in the selection of possible hardware. The chosen hardware interface was the Kinetic Systems PCI serial highway driver, model 2115. The choice to utilize this hardware was enhanced by the fact that other DIII–D sub-systems were already using this specific model providing in-house experience with the hardware and reducing the need

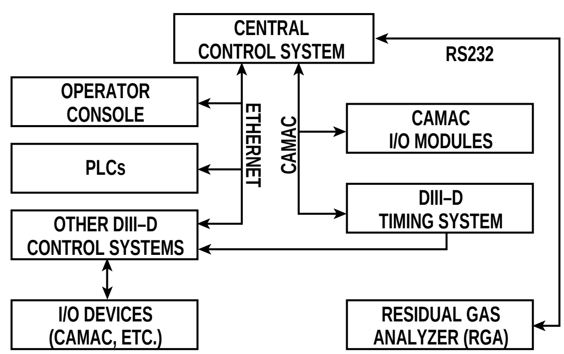


*Fig. 1. Layout of main DIII–D control system.*

of additional spares. The resulting configuration of the DIII-D's main control system is shown in Fig. 1.

# 4. SOFTWARE DESIGN

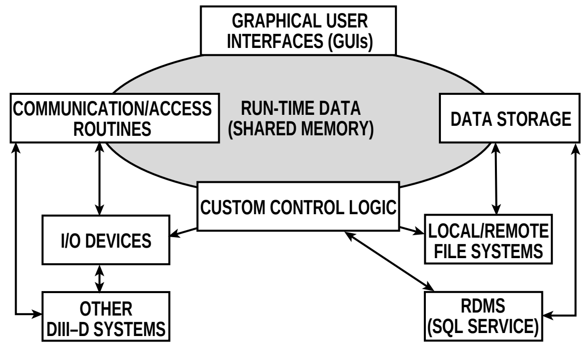


*Fig. 2. Diagram of major software components.*

After analyzing the needs of the system and the short comings of the previous system, software was grouped into four critical components: run-time data access, hardware interface, graphic user interfaces (GUIs) and control logic (Fig. 2). The approach to software design has centered upon the creation of a software solution which would be simpler and more modular than before. With a base set of these four components, other DIII–D sub-systems will be encouraged to implement a similar control system thereby spawning near identical control nodes which are easier to bring on-line and eventually easier to maintain or enhance. As specialized needs come about, they can be addressed at that time and/or be folded into the basic control components.

In order to expedite the development, what could be salvaged from the old VME base system was retained. Data schemes, control logic, and certain stand alone codes were ported almost directly. This was accomplished easily since most of the previously designed software was written in standard C. The structure of run-time data points, i.e. digital input/output (I/O), analog I/O, etc. were also retained in order to give the end users and programmers a sense of familiarity with the new system and also facilitate the transfer of data structures between the old and new systems. The largest software needs were access logic to the run-time data, a software driver for the Kinetic Systems CAMAC interface, and GUIs.

Run-time data access is accomplished via a structured shared memory region and centralized set of access routines. These routines interface identically as older routines did so that existing software, i.e. control software, can remain as before or need minimal changes. This method for data sharing has proved to be extremely reliable and simpler than the previous centralized data server. Another important software need which was filled with a customized in-house piece of coding was the software driver for the Kinetic Systems PCI serial highway driver. In this case, the choice of the Linux OS proved to be very helpful. On-line examples and templates were easily obtained via the Internet. Customized coding was minimal and the turn around time from creation to full implementation was surprisingly quick.

The graphical user interface component of the system has grown to be the largest part of this upgrade. Individual windows and displays can be very complicated and intricate to program. End users also expressed a desire to be able to alter GUIs without the need of a software developer.

Therefore, to aid in the creation of the many windows and displays, an off the shelf software package has been used, Borland's Kylix GUIs generator. Although this goes against one of the major goals of this upgrade, i.e. dependence on third party software, the Kylix product is widely used and support of the product line is expected to be ongoing. Also having separated out the GUI component does afford the luxury of replacing GUIs easily. Almost any user interface can then be easily substituted for the Kylix GUI because the existing interface to the remaining components can be used.

# 5. IMPLEMENTATION

The initial phase of this upgrade was begun in the latter half of 2001. At that time, development of the software driver for the Kinetic Systems highway driver was completed. During the same time, first iterations of run-time databases and general library routines were also completed. The second phase took place during the first quarter of 2002. For this phase a mere subset of the original control system was separated out and functionally implemented on the new system. This phase encompassed the basic use of all four software components on the actual finished hardware. Portions of the original control system were carefully selected as to not adversely affect the operation of DIII–D. As these components proved the viability of the new system beyond a proof of principle, additional parts of the old control system were migrated and used in full production mode. These included the residual gas analyzer (RGA) and thermocouple (TC) access. Long term data storage, for the control system, is accomplished by using Microsoft's (trademark) SQLserver. Bridge software, to communicate to this RDMS was easily available on the Internet free of charge. As expected, performance was much better than the previous VME system for these components. For example, data I/O access has averaged less than 1 s rather than the three plus second average of the old system. In the summer of 2002 final implementation of the control system was put into place so that it would be ready for the pending DIII–D operational cycle starting in the winter of 2002–2003. No major problems have been encountered and work has progressed nicely.

## 6. CONCLUSION

Bringing on-line the new computer upgrade for the central DIII–D control system has been a success. Although some work and testing still remain, performance is far better than the previous system and the potential hardware maintenance costs have been reduced with the new set of hardware. The speed in which this upgrade took place was largely due to a strong base of existing in-house software, openly available software/operating system, using Kylix for GUIs and simplifying the necessary modules to generic basic components. This greatly reduced the amount of overall software which needed to be created from scratch. All goals have not yet been achieved; mainly the total removal of all third party proprietary hardware and software. However, the upgrade has been a substantial step in the correct direction. Future enhancements will be easy to implement and reliability also appears to be good.

Several DIII–D sub-systems may need to be supported in the future with computer control systems and the PC processor/Linux combination coupled with a near complete in-house set of software offers the best foreseeable future.

## REFERENCES

[1] P.A. Henline, "Use of Open Systems for Control, Analysis, and Data Acquisition of the DIII–D Tokamak," Proc. of the 15th IEEE/NPSS Syposium on Fusion Engineering, (Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey,1993), p. 127.

[2] D. Piglowski, B.G. Penaflor, J.C. Phillips, "Use of the Accessware Interface/Database Software in the Neutral Beam Control Systems Used by the DIII–D Tokamak" Proc. of the 17th IEEE/NPSS Symposium on Fusion Engineering, San Diego, California, Vol. 2 (Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey,1998), p. 819.

[3] B.G. Penaflor, J.R. Ferron, M.L. Walker, D.A. Piglowski, R.D. Johnson, Fusion Eng. and Design **56-57**, (2001) 739.

[4] G. Bramson, DIII–D Timing System, IEEE Trans. Nucl. Sci. NS-34 no.4 (1987).

## ACKNOWLEDGMENTS

We would like to thank the personnel at the DIII–D facility who aided in the preparation of this paper. Work supported by U.S. Department of Energy under Contract No. DE-AC03-99ER54463.