DESIGN, EVOLUTION AND IMPACT OF THE AGS/RHIC CONTROL SYSTEM *

J.F. Skelly and J.T. Morris, Brookhaven National Laboratory

Abstract

The AGS/RHIC complex evolved from an accelerator built (1960) before control systems in the modern sense were developed. It has experienced the impact of computer-based control systems from the beginnings of the field, and today employs a distributed control system conforming to the "Standard Model". Whereas present AGS controls labor under a certain "burden of history" as a consequence of extended development, the design of RHIC controls specifically was freed from this constraint. The evolution of this control system is discussed, both its features and deficits. The impact of both upon the operation and flexibility of the accelerators is considered.

1 HISTORICAL PERSPECTIVE

In the beginning, particle accelerators were built without computer control systems. As small computers became affordable, accelerator institutions began to experiment with the new possibilities that computers afforded. The experience at the AGS mirrors that elsewhere, and can be regarded as defining several eras:

- non-centralized computer adjuncts to the accelerator
- centralized computer control, vendor-oriented, with very substantial custom solutions
- distributed networked computer control, standards-oriented, with mostly commercial solutions

These eras are not, however, mutually exclusive; even today systems characteristic of the earliest era can still be found, even at the newest accelerators. Of course, interwoven through these eras is the history of the computer industry itself, from mainframe computers to mini- and midi-computers, to workstations, to personal computers, to commodity single board computers.

Today, accelerator operators and physicists assume that a powerful computer control system should be part of an accelerator; but increasingly these users of control systems challenge controls groups to make the arcane technical aspects of the computers transparent. The success of a controls group in delivering high capability controls solutions that are transparent and user friendly is a major component of what a controls system can contribute to the successful operation of a facility.

2 EARLY COMPUTER INITIATIVES

Attempts to connect computers to the AGS began in 1966, with the introduction of a PDP-8 in the control room. The initial goal was to monitor instrumentation signals in real time, using ADC cards mounted in the computer bus; programming was in assembler using paper tape, output was on a Teletype. A disk and tape were added, and in 1968, a second PDP-8. By March, one could acquire the beam orbit at four times in the acceleration cycle; it took 30 seconds to plot the data.

By June 1968, a steering magnet was under computer control, to minimize beam spill fluctuations in a slow extracted beam. In July, an alphanumeric CRT display was added. In 1970 a custom designed field bus (Datacon-I) was connected to a PDP-8, (replaced within 2 years by a more noise-resistant version, Datacon-II).

2.1 Evaluation

The aspirations of this period clearly were met; the real time achievements of these efforts delivered a capability to the operators well beyond the reach of non-computerized controls. But transparency was nil - these systems were highly arcane. However, it is fair to say that these successes led the way to the pervasive presence of computer controls in today's accelerators.

3 THE MAINFRAME ERA

In April 1971 the AGS took delivery of a PDP-10 mainframe computer, beginning a 15 year era of centralized computer control. A custom hierarchical network was developed, permitting PDP-8s to manage Datacon-II field buses in real time, while reporting their results to application programs that ran in time-sharing mode on the mainframe. A high-speed link (1 Mbit/sec) was developed between the PDP-10 and PDP-8s using custom I/O cards, and a custom PDP-10 driver for the link. A custom monitor was developed for PDP-8s, an early RTOS (Real Time Operating System).

Applications were programmed on the PDP-10 in the Fortran language. Two fast alphanumeric video displays were provided at each of three operator's consoles using commercial display generators (and custom I/O cards and driver). Operator input was provided by a custom panel with knobs, buttons, and a trackball (a custom PDP-10 I/O card and driver). Tektronix terminals provided for graphics displays.

The system eventually provided for 20 field buses, 4 per PDP-8, with 256 addresses per bus. A variety of hardware devices was developed for the field buses, and a custom database system was invented to manage the configuration information for the device inventory. A modular library of tools was developed to support the interface among application programs, the database, and the field bus devices. Today we would describe this library as object oriented, but the term was unknown; the approach just seemed good programming practice.

* Work supported by U.S. Department of Energy

3.1 Evaluation

This period coincided with substantial expansion in the AGS, and a less ambitious control system would have crippled operation. Transparency of the control system was much improved for its users. Although the user interface was primitive by today's standards, it seemed capable at the time, and occasionally even elegant. But by the 1980s, the limitations of this architecture were apparent:

- The hierarchical architecture of the network made redundancy and evolution impractical;
- The extensive custom development became a maintenance burden;
- The commitment to a single vendor made it difficult to take advantage of developments elsewhere in the computer field; as this vendor terminated product lines which the control system employed, the control system became geriatric;
- The commitment to a single field bus, and its limitations, made it difficult to take advantage of the developing single board computer industry.

But there were noteworthy positive aspects:

- Within its appropriate domain, the Datacon-II field bus provided extremely effective service, and is in fact still in use today.. *Though the aging inventory of Datacon devices is now a maintenance burden.*
- The object oriented nature of the software management of the device database and interface was remarkably prescient of later developments in the computer industry, and served as an excellent base for further expansion and improvement.

4 NETWORKS, WORKSTATIONS, AND SINGLE BOARD COMPUTERS

Beginning in the early 1980s, a custom data network was developed for intelligent front end hardware, since no commercial network then available could span the distances required for the accelerator control system. This peer network, called Relway, used broadband cable and employed an Ethernet-similar protocol.

In 1983, an AGS research program using polarized protons commenced, and Relway was used to connect the PDP-10 to several intelligent controllers for pulsed magnet systems required to support acceleration of polarized protons. These controllers employed 16-bit Intel Single Board Computers (SBCs) and Multibus-I.

In 1984 an Apollo workstation was acquired, and over the next several years, a new networked control system was developed in parallel with the mainframe-based system. This effort was the genesis of a control system conforming to the modern "standard model". The peer network connecting the workstations was the vendor's proprietary Domain Token Ring (DTR). The user interface required no custom hardware, but was simply the workstation's GUI (graphic user interface). Relway was used, with improved communications protocols, to link the workstations to FECs (front end computers - 16-bit Intel SBCs); these in turn were linked to intelligent controllers (more 16-bit SBCs) using the General Purpose Interface Bus (GPIB).

Years of experience with applications written in Fortran spaghetti code prompted a switch to the C language. Programmers learned to write GUI interfaces, but the language was the Apollo's proprietary GUI - Dialogue. A library of standardized GUI routines was developed, which provided a common look-and-feel to the application programs. Professional database systems were adopted for configuration information. Apollo-based controls were commissioned in 1986 to support a new transfer line from the Tandem Van de Graff facility to the AGS, permitting the inception of a heavy ion research program at the AGS.

4.1 Evaluation

Early reliability of Relway and the intelligent front end systems was sometimes marginal, leading to impatience in the control room; but the intelligent controllers were crucial to the polarized proton program.

With the introduction of workstations and a GUI, transparency improved for the users. The control room learned to deal with two control systems, a condition that lasted until the PDP-10 was retired in 1995. Improved Relway protocols and the relational database made the control system even more object oriented.

The control system was committed to a distributed networked style. But it remained vendor-oriented not standards-oriented; and Relway was another custom solution that in time became a maintenance burden.

5 OPEN STANDARDS

By 1986, a project was underway to construct a low energy Booster synchrotron injector for the AGS. The Booster was commissioned in 1990, and during this period development in the control system finally began to focus on open standards. Relway and 16-bit Intel FECs were recognized as limitations. Apollo workstations now were employed as front end computers; although their operating system was not real-time, their performance was adequate if they ran no application except the front end code (an administrative restriction easy to enforce). Now console Apollos and front end Apollos on the DTR could communicate directly.

The Apollo FECs were used to enhance the front end capability. As an injector to the AGS, the Booster can run multiple pulses during each AGS cycle, and it becomes attractive to utilize these pulses in different ways - a feature called Pulse-to-Pulse Modulation (PPM). PPM was a front end challenge worthy of the 32-bit Apollos. These Apollo FECs continued to manage controllers (still Intel 16-bit SBCs) over GPIB.

The C++ language was adopted, and controls became object oriented in both language and organization.

The Booster project consumed the controls group until 1990; attention then turned to other issues:

- Ethernet was adopted for all future computers, both at the console and FEC levels.
- The Apollo-proprietary GUI (Dialogue) was abandoned, and the X display system was adopted; old and new GUI tools were wrapped in C++ classes, simplifying use and enhancing a common look-and-feel for applications.
- Network communications among applications and FECs had used Apollo-proprietary protocols, but now were changed to employ Remote Procedure Calls (RPCs) layered over TCP/IP (using UDP).

These changes combined finally to permit vendor independence; Sun workstations were introduced to the network, and software was ported to the Sun platform.

Controls engineers abandoned Multibus, and turned to VMEbus. An FEC was developed for VMEbus, with a commercial RTOS and C++ software. The Datacon-II fieldbuses were connected to VMEbus FECs in 1995, and the PDP-10 and PDP-8s were retired. In 1997 the Intel FECs were replaced, and the Relway was retired. Commercial and custom VMEbus modules also were supported in the VMEbus FEC with modular software.

5.1 Evaluation

During this period, PPM controls were implemented in the Booster and AGS; PPM enhanced the flexibility of accelerator operation and provided new options for accelerator exploitation. Broadly supported open standards were adopted, and the control system achieved a substantial measure of vendor independence.

6 RHIC

A unified controls group assumed responsibility for RHIC controls as well as AGS. Hardware and software design teams were established for RHIC by 1993. Existing AGS controls were considered an inadequate model for RHIC for several reasons:

- AGS controls are designed for pulsed accelerators, whereas RHIC is a state machine;
- AGS controls are heavily influenced by historical development, with dependencies on aging equipment and custom protocols;
- Where new technologies had been introduced, design choices had been constrained by the requirement of backward compatibility.

The RHIC design teams were empowered to build on AGS solutions or develop fresh solutions. The resulting RHIC controls follow the AGS model, with substantial innovation. The standard model is preserved (UNIX workstations, X-window GUI, VME-based FECs, and Ethernet). Much infrastructure is common, including the central network, control room computers, system servers, software development environment, and the accelerator timing system design (event link).

RHIC controls provide additional information links: beam sync links, a permit link, and a real time data link.

Some 150 RHIC FECs use the same RTOS and many of the same VME components and device drivers as AGS VME-based FECs. A new abstraction, the Accelerator Device Object (ADO), provides a software interface to RHIC equipment. The ADO provides more flexibility than AGS device objects (for data delivery protocols and data representation). Most of the ADO configuration information resides as metadata in the FEC rather than in a configuration database as in the AGS. RPCs are layered on TCP instead of UDP.

RHIC and AGS console level applications employ a common object oriented User Interface (UI) toolkit. New UI extensions have benefited both RHIC and AGS controls. Generic AGS software was expanded to support RHIC, including the alarm system, the save/restore facility and monitoring software.

The Parameter Editing Tool (pet) is a key program for RHIC controls, providing a generic tabular style interface whose flexibility matches that of the ADOs.

RHIC applications often employ a manager layer between the application and FECs. This layer separates physics content from the user interface and allows coordination of data from multiple FECs. The manager can present the same ADO interface as FEC software. RHIC physics applications, unlike AGS software, are developed predominantly by machine physicists instead of the controls group, with programming tools provided by the controls group.

6.1 Evaluation

RHIC was commissioned beginning in July and August 1999. The control system performed well and the system design was validated. Effort remaining includes:

- Time correlation of data from multiple FECs
- Perfecting reliability of communication throughput
- Eliminating occasional FEC unreliability
- Smoother recovery from FEC failures

RHIC controls proved flexible during commissioning; the challenge now is to ensure consistent behavior for accelerator operations while maintaining this flexibility. Greater integration of RHIC and AGS controls is also desirable. While tools are largely in place to support this at the console level, the integration path at the FEC level remains a challenge for the future.

7 LESSONS LEARNED

- Repeatedly, the opportunities of one era become the liabilities of a later era.
- Avoiding dead end architectures is an art, but open standards minimize the risk.
- Every custom design eventually becomes a maintenance burden. Commercial products may also, if retained too long.