3rd Int. Conf. Accel. Large Exp. Phys. Control Syst. ICALEPCS1991, Tsukuba, Japan JACoW Publishing
ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S03SRD04

Standards and the Design of the Advanced Photon Source Control System*

William P. McDowell, Martin J. Knott, Frank R. Lenkszus, Martin R. Kraimer, Robert T. Daly, Ned D. Arnold, Mark D. Anderson, Janet B. Anderson, Robert C. Zieman, Ben–Chin K. Cha, Fred C. Vong, Gregory J. Nawrocki, Gary R. Gunderson, Nicholas T. Karonis, and John R. Winans
Argonne National Laboratory
Advanced Photon Source
9700 South Cass Avenue
Argonne, Illinois 60439

I. Introduction

The Advanced Photon Source (APS), now under construction at Argonne National Laboratory (ANL), is a 7 GeV positron storage ring dedicated to research facilities using synchrotron radiation. This ring, along with its injection accelerators is to be controlled and monitored with a single, flexible, and expandable control system. In the conceptual stage the control system design group faced the challenges that face all control system designers: (1) to force the machine designers to quantify and codify the system requirements, (2) to protect the investment in hardware and software from rapid obsolescence, and (3) to find methods of quickly incorporating new generations of equipment and replace obsolete equipment without disrupting the existing system. To solve these and related problems, the APS control system group made an early resolution to use standards in the design of the system. This paper will cover the present status of the APS control system as well as discuss the design decisions which led us to use industrial standards and collaborations with other laboratories whenever possible to develop a control system. It will explain the APS control system and illustrate how the use of standards has allowed APS to design a control system whose implementation addresses these issues. The system will use high performance graphic workstations using an X–Windows Graphical User Interface (GUI) at the operator interface level. It connects to VME–based microprocessors at the field level using TCP/IP protocols over high performance networks. This strategy assures the flexibility and expansibility of the control system. A defined interface between the system components will allow the system to evolve with the direct addition of future, improved equipment and new capabilities. Several equipment test stands employing this control system have been built at ANL to test accelerator subsystems and software for the control and monitoring functions.

II. Standards and the APS Control System

The APS control system must be capable of (1) operating the APS storage ring alone and in conjunction with its injector linacs, positron accumulator, and injector synchrotron for filling, and (2) operating both storage ring and injection facilities as machines with separate missions. The control system design is based on the precepts of high–performance workstations as the operator consoles, distributed microprocessors to control equipment interfacing and preprocess data, and an interconnecting network. In a paper presented at the 1985 Particle Accelerator Conference [1] we outlined our initial approach to the APS control system. In this paper we predicted

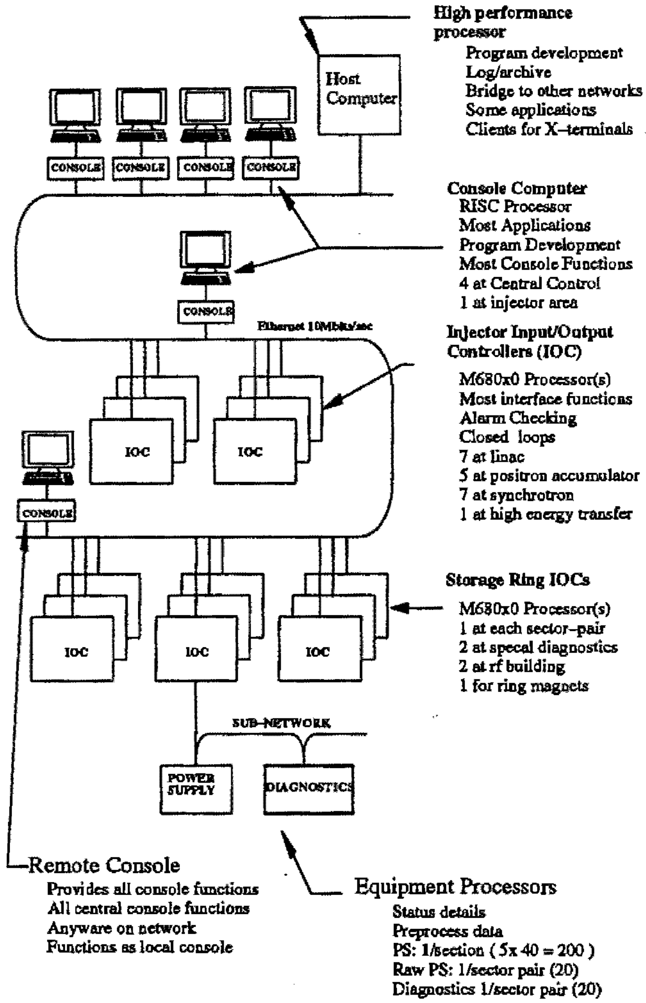

Figure 1. APS Control System

*Work supported by U.S. Department of Energy, Office of Basic Energy Sciences under Contract No. W–31–109–ENG– 38.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S03SRD04

that the control system would use workstations for the operator interface, single–board computers to control the front–end electronics, and a network consisting of either Ethernet or Token–Ring. The APS control system today is remarkably close to the initial design concepts due to rapid performance gains in computing workstations, low cost network connections, both Ethernet and Fiber Distributed Data Interface (FDDI), and availability of real–time operating systems for the front–end computers.

Figure 1 shows in schematic form all major components and their relationships. The current design includes about 45 distributed microprocessors and five console systems, which may consist of one or more workstations. An additional 70 Input–Output Controllers (IOC's) will be used to control the insertion devices, front ends, and beam lines.

The operator interface (OPI) is implemented with a high performance graphic workstation and uses an X–Windows based GUI. Standards play a large role in the selection of the OPI since the hardware, operating system GUI, and network must be compatible. The ideal control system design would be vendor independent in all of these areas. To make the APS control system vendor independent we chose to use standards when selecting these components.

A. Standards

The definition of the word "standard" used for the purposes of this paper is as follows: "Something established by authority, custom or general consent as a model or example" [2]. Past practice at large laboratories has been to invent almost everything that was needed to build a control system. Accelerator control system groups have built computer systems and designed networking schemes. Of course there were good reasons for this – the laboratories were often pushing the leading edge of electronic and computer technology and the required devices and techniques were not available on the open market. This picture has greatly changed. Computer technology has now spread into every corner of our lives. There are literally tens of thousands of companies inventing new uses for computers and pushing the limits of technology. This has had a very positive effect on control system design as the effort required to build a control system can now be redirected towards control and accelerator details rather than details associated with building a computer or computer network. In the Proceedings of the Second International Workshop on Accelerator Control Systems [3] held in October of 1985, no discernible trend can be observed in control system design. This contrasts with the sense one receives from reading the titles of the the papers presented at the 1991 Particle Accelerator Conference. These titles show a ground swell towards what could be called a generic control system. The generic system consists of workstations running UNIX, a network, and front end processors running a real time operating system. We now find standards being followed at all levels of control system design.

B. Operating Systems/Workstations

At the Europhysics Conference on Control Systems for Experimental Physics [4] in October of 1987, discussion panels ran late into the night with the "religious" arguments for the choice of UNIX or VMS as the operating system of choice for control systems. There where convincing arguments presented by advocates of both sides of the discussion. Four years later the argument has been settled, not because either of the opposing sides was won over by a technical argument but because of market forces. The development of the the Reduced Instruction Set Computer (RISC) processor has resulted in UNIX dominating the workstation market.

RISC is a recent innovation in computer architecture (although some people claim that the PDP–8 was a RISC machine). The study of computer instructions and their relative frequency of usage revealed that most of a computer's time is spent in the execution of a small subset of its repertoire. RISC architecture takes advantage of this fact by streamlining the execution of this subset and by implementing the less used and more complex instructions with combinations of the (now fast) small set of instructions. Since there is now a small set of simple instructions, parallel "pipelining" can be used to increase execution speed. In this method more than one instruction can be executed simultaneously by staggering in time the various suboperations. Some processors can even average more than one instruction per clock cycle.

The converse to RISC architecture is Complex Instruction Set Computer (CISC) architecture. Most computer architectures developed prior to 1980 are of the the CISC type, a typical example being the VAX. Today there is a five–to–one advantage in raw MIPS (millions of instructions per second) for RISC devices. This should be discounted to some degree since RISC requires more instructions to perform some types of operations, but an advantage of even two–to–one on reasonable benchmarks is obtainable.

The UNIX operating system itself was originally developed by Bell Telephone Laboratories as a word processing tool, but it was soon modified to support software development tools and finally grew into a full–featured operating system. UNIX was written in the "C" language, also developed at Bell Telephone Laboratories. The keys to UNIX's success are that it is extensible and it is written in a portable language. These attributes allow the user to make enhancements, remove features, and tailor UNIX to specific needs. One indication of this is the fact that the UNIX operating system is available for microprocessors as well as supercomputers. Thus, if a start–up company chose UNIX as its operating system and made the changes necessary to support its chosen computer architecture, any existing software that ran under UNIX could be recompiled to run on their computer. In this way new computer architectures can be introduced with ready–made operating system software and trained users.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S03SRD04

Because of the development of RISC processors and the existence of UNIX, nearly all computer manufacturers are developing and marketing RISC–based computers and workstations which use the UNIX operating system. Competition is driving performance up while keeping prices low and this trend is likely to continue. There is still a market for CISC architecture computers and operating systems such as VAX/VMS, principally due to the installed base of application software and the steady improvements made to the hardware by vendors.

These reasons seem to make it obvious that the operating system of choice for any control system to be delivered in the mid 1990s will be UNIX. The bottom line for APS is the fact that the UNIX operating system provides the control system a large measure of vendor independence. We have the OPI software running on SUN 4 and Digital Equipment DS5000 workstations and expect to port the system to other vendors' workstations.

The GUI "wars" now being fought in the press and on workstations provide a very good reason to conform to standards when writing the OPI software. APS is developing applications using the Open Software Foundation's Motif toolkit. We are extensively testing the software against the two major window managers Motif and Openlook.

C. Front End Systems

The IOC, or front–end electronics, is implemented with single–board computers of the Motorola 680X0 family, packaged along with signal interface cards in VME and VXI form factor crates. Motorola 68020 processors are used in initial configurations with 68040 processors planned for most future configurations. A real–time operating system, VxWorks from Wind River Systems Inc., is used to provide multitasking, high performance front–end software. More than fifteen VME input–output modules are currently supported. These modules include binary input and output, analog input and output, motor drivers, counter timers, and subnet controllers. More modules will be supported as they become available and are required. Most information preprocessing is performed at this level with only engineering units sent to the OPI for display. Signal monitoring can be set up to communicate only on signal change or limit–breaching or at some preselected rate. Local sequential and control–loop operations can also be performed. In this way, maximum benefit is gained from the many IOC processors operating in parallel. This is one area where APS is vulnerable to complications which would arise if the vendor of the real–time software failed. When the posix standard for real–time systems becomes a reality and most real–time vendors conform to the standard, our estimate is that it would take about two months of a very knowledgeable programmer's effort to change real–time kernels.

In addition to local IOC I/O, subnets are utilized to interface remote points where an IOC may not be present to a distant IOC. There are currently three supported subnets in the APS control system: Allen Bradley, GPIB, and BITBUS. The Allen Bradley I/O subnet uses the 1771 series I/O modules to provide basic binary and analog I/O support for the APS control system. Allen Bradley is a inexpensive and rugged standard for industrial control systems. Copper and fiber optic based multidrop subnets are available for this equipment.

Laboratory test and measurement equipment often use the GPIB standard as an interface to an external control system. This multidrop standard presents some serious challenges and potential problems to the system designer. GPIB has a distance limitation of 20m which requires the instruments connected to the bus to be in close proximity to the IOC. In addition, the signals within the GPIB cable are not balanced and thus susceptible to EMI/RFI noise and ground spikes. Signal transfer and isolation techniques are not part of the GPIB standard and although commercial equipment is available to extend the distance of a GPIB interface, it is prohibitively expensive.

BITBUS provides a method of high speed transfer of short control messages over a multidrop network. The BITBUS subnet can be used as a method for remote, single point I/O as well as a gateway for remote GPIB and RS232 signals [5]. A differential, opto–isolated, wired subnet is the BITBUS standard; however, a multidrop fiber optic network has been developed for BITBUS at the APS.

D. Networks and Protocols

Argonne uses Ethernet as the intra–laboratory network. There are backbone cables in the individual buildings with communication between buildings presently done via the Lanmark PBX system. Intra–building FDDI will be available within 6 months. The control system development computers are presently sharing the APS Ethernet backbone with all other APS computing needs (55 Sun Workstations, a VAX Cluster with six members, 18 terminal servers, 40 PC's using Pathworks, and 40 Macintosh systems). Two test stands and six development IOC's are running in this environment without experiencing network–induced problems. A Network General Sniffer is on line at all times should the need arise to diagnose an apparent problem. In the APS facility, however, we plan to use FDDI with Ethernet branches as performance needs dictate.

The central feature of both the OPI and IOC software designs is the protocol for connections between software modules for the purpose of exchanging information. This protocol is called channel access [6] and is built on the TCP/IP Standard. TCP/IP is an integral part of every UNIX–based workstation as well as being built in to VxWorks, the real–time operating system. When an OPI application program needs to connect to a process variable located in an IOC, it issues a broadcast over the network and the IOC in whose database the requested process variable resides provides a response. A socket–to–socket connection is established and thereafter efficient two–way communication takes place. IOC–to–IOC channel access can take place to exchange inter–IOC information. It should be noted that the OPI

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S03SRD04

needs no knowledge of the location of the desired process variable, only its name. Figure 2 shows the relationship of the channel access software in both the OPI and IOC systems. It also illustrates how a mouse and screen "slider" are used to communicate, through channel access software at both ends, with a D–A convertor. Similarly, A–D convertor output finds its way to a screen.

The database which defines all IOC channel connections and properties is distributed over the many IOC's and downloaded at IOC boot time along with the operating system and the particular device driver software modules required by each IOC. The entire database is centrally maintained and configured with a UNIX workstation which, of course, can be any OPI. Figure 2 shows the downloaded location of the IOC database in the overall data flow.

E. X–Windows

In the X–Windows client–server paradigm, an application program is divided into the "client" (which provides the computation and logic of the program) and the "server" (which provides the interaction facilities for the human operator or user). In the APS control system, both client and server are implemented in processors at the OPI level. The client and server need not reside in the same processor so that, for example, a specialized parallel processor may provide client services for a more common workstation server or X–Window terminal. In this way, the OPI's will be able to have windows open to clients operating both locally and on other processors on the network.

F. Application Software

Application software comprise those programs which the operator or physicist invokes to provide a feature or service not provided by the equipment operation level of the control system. An example would be the software required to provide a local bump in the orbit. These programs can be of two general forms. The first is a control panel which is created during a session with a display editor (see Figure 2, upper left). Graphic tools such as buttons, sliders, indicator lights and meters, and graph paper are selected and located on the panel. Static entities which can be used to depict the physical system, such as piping diagrams, are added where appropriate. Connections to IOC channels are specified at this time and the proper drawing list and action code are automatically generated. When complete, the panel is called up for execution, the channel access calls are made, and the control panel is now "live." No actual code is written or compilation made, aside from that originally involved in the tools themselves. The software provides calculation records and allows cascading of physical inputs and outputs with these calculations. This allows very complex operations to be designed. The second form of application program is that of employing classic in–line code generation. In this case standard entry points are provided to the same graphic and channel access tools. Using this approach, an existing code can be adapted to our system by calling the channel access code and displaying the results using either traditional line–by–line or graphical output.

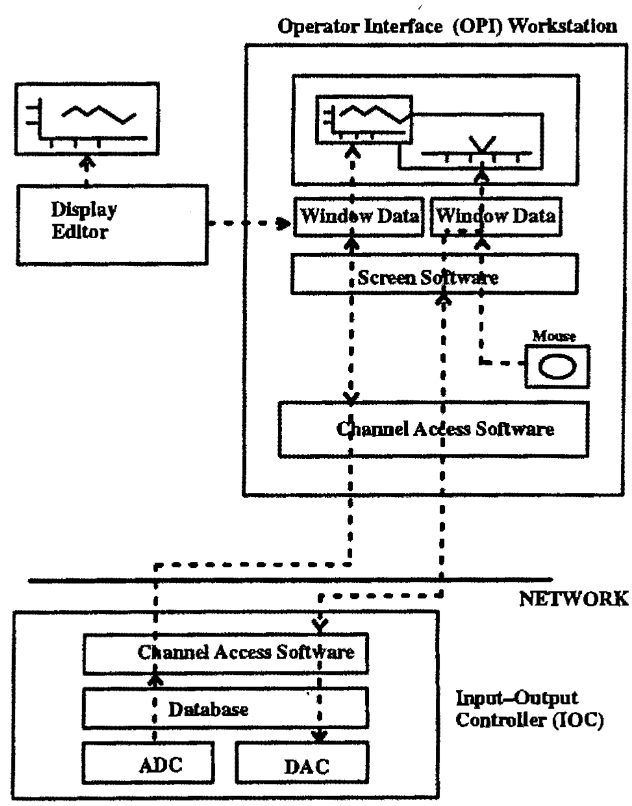

Figure 2. APS Control Environment

All software is being developed under UNIX, including that for the IOC's. In this way, windows can be opened simultaneously at an OPI for software development, actual run–time applications, database configuration, electronic mail, etc. This streamlines software development, database servicing, and system troubleshooting.

III. Interlaboratory Cooperation

At the Accelerator Control Toolkit Workshop [7] in 1988 a group of people responsible for accelerator control systems at laboratories throughout the world spent a week discussing various aspects of control systems. One of the topics discussed was the development of "tools" which could be used at more than one laboratory. Subsequent to this meeting we decided that the APS would pursue the idea of looking at existing control systems with the aim of determining if they could be used at APS. After much discussion we decided to pursue collaboration with Los Alamos National Laboratory (LANL). Discussions were held with the developers from LANL and it was decided that APS would send a representative to LANL who would use the system to develop an application which would be useful to LANL. One of us (Kraimer) spent a summer at LANL developing the software to control a magnet measuring facility. Upon his return he imparted his positive impressions of the system. We then

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S03SRD04

decided to complete an in-depth study of the software. He sequestered himself in his office with the software listings. As he gained understanding of the system he gave tutorials to the APS controls group staff on the internal design of the software. Further group discussions led to the decision to try to form a cooperative development team with LANL. M. Knott (ANL) and M. Thout (LANL) proceeded to develop an agreement that has led to the co-development of the Experimental Physics and Industrial Control System (EPICS). The paper entitled "EPICS Architecture" [8] by L. R. Dalesio, et al. presented at this conference provides a detailed look at the features of EPICS.

IV. Construction Status of APS and the APS Control System

Construction is proceeding rapidly on the physical structure of the APS. As of this date (late October 1991) the linac and injector buildings have steel erected, the concrete for the linac tunnel and positron accumulator ring vault is in place, the control center has reached the first floor level, and the foundations are in for the first section of the storage ring building which will be used as an early assembly area and magnet measuring facility. Barring unforeseen construction delays, the linac and control center are scheduled for occupancy in April of 1992.

The APS control system is now actively supporting two test stands, rf and linac. Work on these test stands started in 1989. In their first implementation they used a predecessor version of the OPI running on a VAXStation under the VMS operating system and a predecessor of EPICS called GTACS (Ground Test Accelerator Control System) for the IOC. As work on a UNIX-based OPI and EPICS progressed, both test stands converted to the UNIX OPI software and EPICS. The APS rf test stand was reported at the Real Time '91 Conference [9]. Two IOC's are being used to implement the linac functions: one for beam diagnostics and the other for control. The test stands have proved to be highly beneficial to both the controls group staff and the linac and rf systems development team members. The controls group has gained experience in using the control system as well as received suggestions for changes and improvements. The test stand staff has been able to concentrate on linac and rf design details without developing their own control system. The only way provided to remotely run the test stands is via the control system.

Progress in the development of EPICS software is continuing. An alarm handler [10] has been developed and is being optimized. We are continuing to add device and driver support for new hardware modules as well as develop new record types such as pulseTrain, pulseDelay, etc. A graphical database link display tool is being developed as a way to document databases and requirements are being developed for a system-wide database and a system-wide error handler to accept and process IOC-generated errors. We are developing low cost IOC's based on single-height VME modules as well as Gespac G64 modules. VXI crates using the standard VME processors and network boards are currently operating. On the OPI side we are running on both the SUN 4 and DEC 5000 platforms and we will soon port the system to Hewlett Packard 700 series workstations.

V. References

[1] M. Knott, G. R. Gunderson, F. R. Lenskzus, W. P. McDowell, "Control System Features of the Argonne 6GeV Synchrotron Light Source," IEEE Transactions on Nuclear Science, Vol. NS-32, No. 5, October, 1985.

[2] Webster's New Collegiate Dictionary, Merriam, 1977.

[3] Accelerator Control Systems, Proceedings of the Second International Workshop on Accelerator Control Systems, Nuclear Instruments and Methods, Vol. A247, No. 1, 1985.

[4] Proceedings, Europhysics Conference on Control Systems for Experimental Physics, CERN Report 90-08, 1990.

[5] N.D. Arnold, G.J. Nawrocki, R.T. Daly, M.R. Kraimer, W. P. McDowell, "I/O Subnets for the APS Control System," to be published in the Proceedings of the 1991 Particle Accelerator Conference, San Francisco, California, May 6-9, 1991.

[6] J. O. Hill, "Channel Access: A Software Bus for the LAACS," in Accelerator and Large Experimental Physics Control Systems, D. P. Gurd and M. Crowley-Milling, Eds., ICLAEPCS, Vancouver, British Columbia, Canada, pp. 288-291, 1989.

[7] Accelerator Automation Application Toolkit Workshop, Sponsored by AT Division and MP Division, Los Alamos National Laboratory and CERN, Geneva, 31 October - 4 November, 1988.

[8] L. R. Dalesio, M. R. Kraimer, A. J. Kozubal, "EPICS Architecture," to be presented at ICALEPCS '91, Tsukuba-shi, Ibaraki, Japan, November, 1991.

[9] R. T. Daly, F. C. Vong, M. R. Kraimer, "The APS Radio Frequency Test Stand Control System," presented at Real Time '91, Computer Applications in Nuclear Particle and Plasma Physics, Julich, Germany, 25-28 June, 1991.

[10] M. R. Kraimer, B. K. Cha, M. D. Anderson, "Alarm Handler for the Advanced Photon Source Control System," to be published in the Proceedings of the 1991 Particle Accelerator Conference, San Francisco, California, May 6-9, 1991.