THE ADVANCED PHOTON SOURCE CONTROL SYSTEM*

Martin J. Knott, William P. McDowell, Frank R. Lenkszus, Martin R. Kraimer, Ned D. Arnold, Robert T. Daly,
Gary R. Gunderson, Ben–Chin K. Cha, and Mark D. Anderson
Argonne National Laboratory
Advanced Photon Source
9700 South Cass Avenue
Argonne, IL 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. The control system must be capable of operating the APS storage ring alone, and in conjunction with its injector linacs, positron accumulator and injector synchrotron for filling, as well as operating both storage ring and injection facilities as machines with separate missions. The control system design is based on the (now classic) precepts of high–performance workstations as operators consoles, distributed microprocessors to control equipment interfacing and preprocess data, and an interconnecting network. 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.

II. HARDWARE ORGANIZATION

A. Operator Interface

The operator interface, called an OPI, is implemented with a high performance graphic workstation. The units being evaluated are based on the RISC architecture and the UNIX operating system. Several factors auger for using the highest performance units affordable: the uncertainty of accelerator control system design specifications (and usually the complete lack thereof); the ever–increasing processing load presented by improving operating systems; and the desire to have excess performance to meet future, unexpected requirements as the accelerator complex is commissioned and its unique properties exposed. In addition, our decision to employ the X–windows paradigm will place additional overhead on the OPI implementation.

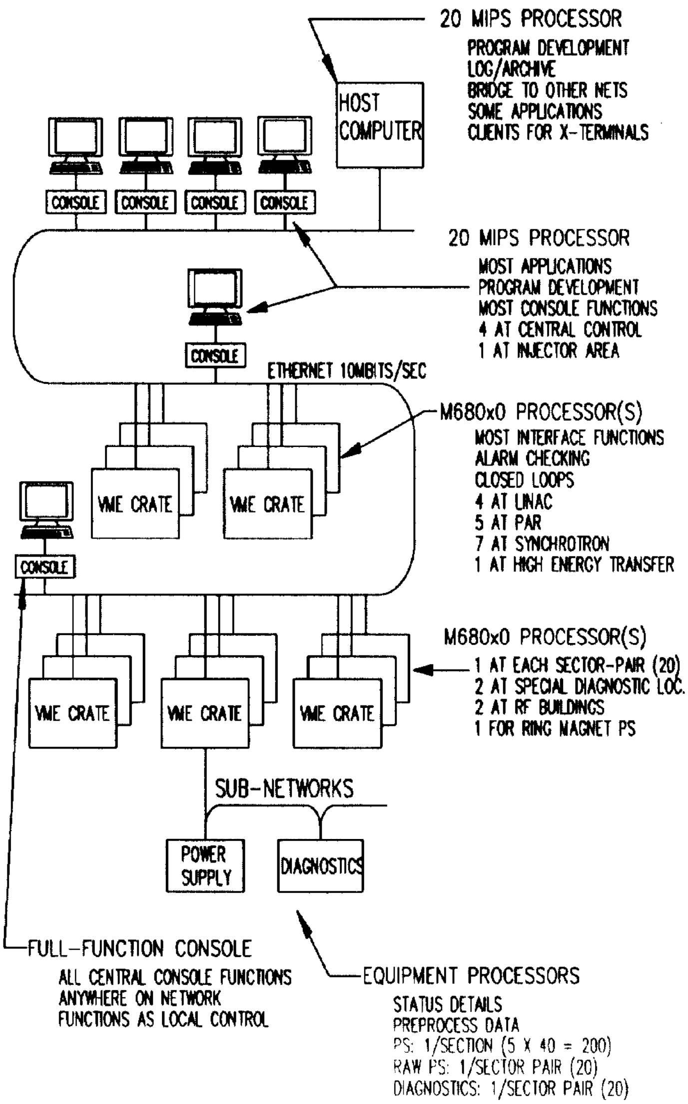

Figure 1. APS EPICS Control System

The use of workstations and X–windows provides several opportunities for control system engineers when implementing an accelerator control system. The most obvious benefit is the

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

ability to custom design the "soft" control console (actually the screens) to present only those application windows and data presentations needed for each of the wide variety of tasks performed by operators, engineers, and machine physicists. More subtle is the option of including live video from beam viewers and access–control systems as windows on the same screen. Even audio input–output is now accommodated. Since most electronic instrumentation is now easily interfaced, it becomes optional to use the actual instruments in the console. They can be left near their input signals and output information sent to the workstation for display.

B. Equipment Interface

The input–output controller, called the IOC, is implemented with single–board computers of the Motorola 680X0 family, packaged along with signal interface cards in VME form factor crates. The APS control system will use about 45 such crates. Motorola 68020 processors are utilized 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 maximum, predictable, real–time performance. More than twenty VME input–output module types are currently supported with device–specific software and more planned as they become available and desirable.

Most information preprocessing is performed at this level with only engineering units sent over the network. Signal monitoring can be set up to communicate over the network only on signal change or limit–breaching. Sequential and control–loop operations can also be performed. In this way, maximum benefit is gained from the many IOC processors operating in parallel.

C. Networks and Subnets

The communication network is planned to be baseband Ethernet, with 4 to 5 field segments repeater–connected to a backbone segment. Subnets can be driven from the IOC with GPIB, RS–232, and Bitbus presently supported. These subnets allow various instruments and low–cost, few–point interfaces to be connected and controlled. The Bitbus can be used to make remote, multidrop connections to GBIP and other interface subsystems. [1]

III. SOFTWARE ORGANIZATION

The software system being developed for the APS, the Experimental Physics and Industrial Control System (EPICS), is the result of a collaborative effort between the Argonne APS and the Los Alamos National Laboratory (LANL) GTA control system groups. The software is structured in layers, somewhat reflecting the distributed, layered nature of the hardware.

A. Channel Access

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. When an OPI application program needs to connect to an IOC, it issues a broadcast over the network and the appropriate IOC responds. A connection is established and thereafter, efficient two–way communication takes place. IOC–to–IOC channel access can take place to exchange inter–IOC information. 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, an A–D convertor output finds its way to a screen graph.

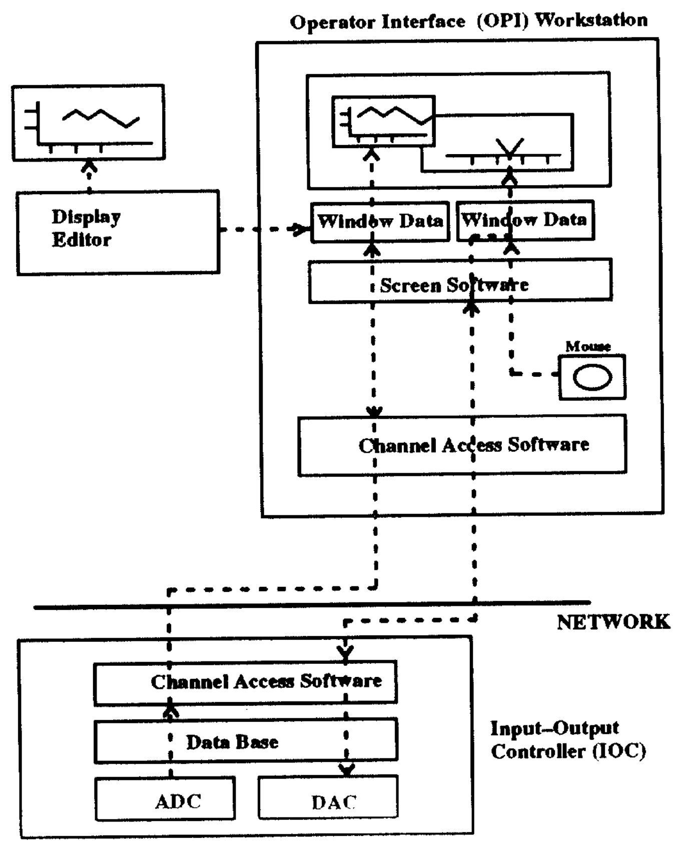

Figure 2. APS Control Environment

B. Distributed Database

The database which defines all IOC channel connections and properties is distributed over the many IOCs and downloaded at boot time. Also downloaded at boot time are the operating system and the particular device specific 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.

C. 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 our 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. In this way, the OPIs will be able to have windows open to clients operating locally as well as in other processors on the network.

In what is known as an X terminal, the server software is implemented in ROM (or downloaded into RAM) with various amounts of RAM screen memory depending on the implementation. This comparatively low–cost device can fully implement all features of an OPI with the client services provided by a dedicated host computer or one of the fully configured workstations. Since a thinwire Ethernet is all that is needed to provide all control and monitoring functions anywhere in the APS complex, such X terminals, which are even available in lap–top configurations, are being considered as local control devices.

D. Panels and Programs

Application "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). 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 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 second form of application program is that employing classic in–line code generation. In this case hooks are provided to the same graphic and channel access tools. Using this approach, an existing code can be adapted to our system by adding hooks to the tools required.

E. Software Development

All software is being developed under UNIX, including that for the IOCs, which uses VxWorks real–time operating system. In this way, windows can be opened at an OPI for software development, actual run–time applications, database configuration, electronic mail, etc., all at the same time. This streamlines software development, database servicing, and system trouble shooting.

IV. Possible Improvements

Due to the modular nature of the components of the system and the adherence to standards, the system can be improved and upgraded in a piecemeal fashion. For example, we may find that the Ethernet LAN may suffer from heavy X–windows traffic, causing channel access data to be delayed. It should be possible to add an additional X–only network or bridge the existing Ethernet segments to an FDDI loop.

As improved versions of the IOC processor become available, or if additional capabilities demand an entirely new type of processor, the new modules or devices can be added to the system as long as the channel access protocol is carefully followed. In fact, entirely different versions of the OPI workstation or any other processor can be substituted or added in the future. In this way, upward compatibility is preserved for all three major components of the system, OPI, IOC, and interconnecting network.

V. Development Status

The original development work on this system was done by the GTA controls group at LANL. Further development and improvement of the system is being carried out by both LANL and by the APS control group at ANL. LANL has two test stands operating and is beginning to interface to the initial GTA subsystems. ANL has test stands operating for linac and rf systems. Current effort is centered on improvements to the IOC core software and transition from the original VaxStation OPI to the UNIX– based workstation with X–windows.

VI. Future Directions

Due to the obvious advantages of the X–windows application, the APS experimental community is anxious to adapt the control system components and software to the control and monitoring of their photon beamlines. They are also considering its possible application to control of experimental data gathering.

VII. References

[1] N. D. Arnold, "I/O Subnets for the APS Control System" these proceedings.