Control System Architecture: The Standard and Non–Standard Models*

M. E. Thuot, L. R. Dalesio, Los Alamos National Laboratory

Abstract

Control system architecture development has followed the advances in computer technology through mainframes to minicomputers to micros and workstations. This technology advance and increasingly challenging accelerator data acquisition and automation requirements have driven control system architecture development. In summarizing the progress of control system architecture at the last International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS) B. Kuiper asserted that the system architecture issue was resolved and presented a "standard model".[1] The "standard model" consists of a local area network (Ethernet or FDDI) providing communication between front end microcomputers, connected to the accelerator, and workstations, providing the operator interface and computational support. Although this model represents many present designs, there are exceptions including reflected memory and hierarchical architectures driven by requirements for widely dispersed, large channel count or tightly coupled systems. This paper describes the performance characteristics and features of the "standard model" to determine if the requirements of "non-standard" architectures can be met. Several possible extensions to the "standard model" are suggested including software as well as the hardware architectural features.

I. INTRODUCTION

Advances in computer technology, changes in the computer marketplace, and demanding control requirements [2] have motivated control system architecture development. The reduction in prices of powerful, user-friendly, networkable workstations coupled with the ever increasing cost and complexity of software stimulated new designs with a philosophy of control system evolution rather than totally new design, even on entirely new facilities[3]. This evolutionary design philosophy includes standardized structures and the use of open software standards to provide much greater flexibility to expand the size and automation of a system, to accommodate new high performance platforms, to reuse software developed previously, and to share software developed by other laboratories and industry. "The recent and continuing efforts of standardization at all levels on protocols and other interfacing conventions means that the plugged in equipment and other gadgets may be exchanged for newer versions, using entirely different internal technologies, which may then increase performance" [1] or functionality of the entire control system. These changes have driven the designers of computer control systems toward a standardized modular architecture.

II. THE STANDARD MODEL

The standard model employs a workstation/personal computer as the operator station, a local area network for data communications and front end micro–computers connected to the accelerator through signal conditioning and/or remote input/output interfaces. In a recent literature review, over three dozen systems world–wide were identified as employing this standard architectural model.

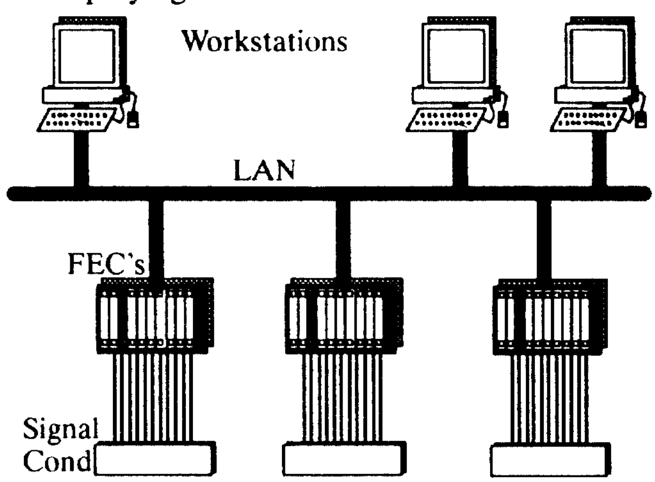

Figure 1. Three basic components of the standard model: the operator interface, data communication, and the front–end computers.

The operator interface provides the operator with a current view of the control process, historical data, alarm information, and a number of physics

Los Alamos National Laboratory is operated by the University of California under contract W-7405-ENG-36 for the U. S. Dept. of Energy.

models to help maintain and predict the operation of the machine. The communication layer provides data transport between the distributed front-end computers and between the front end and the operators. The front-end computers provide distributed intelligence for data acquisition, data conversion, supervisory control, interlock enforcement, closed-loop control and sequential control.

III. OPERATOR INTERFACE FACTORS

The most important factors in selecting an operator interface are: performance, user-friendly interface, cost to configure and maintain displays. To monitor a process, 10 Hz updates to the operator offer feedback in real-time (human perception). Responses at less than 10Hz may be acceptable in many cases. Feedback to an operator action should also occur within 100 milliseconds to give the operator an immediate feedback that the action has occurred. Most modern operating systems (UNIX, VMS, etc.) on moderate performance workstations can provide an off-the-shelf platform that supports this level of performance. Slower system response could result in the operator giving multiple commands to take the same action, not to mention operator frustration. To take full advantage of a windowing environment , display call-up needs to occur quickly, less than 100 milliseconds is optimal, up to four seconds may be acceptable. When the delay is too long, the operator will resort to multiple dedicated displays. This will increase the system cost and overhead. The workstations may also support physics modeling codes that provide higher level data analysis and interpretation. These codes may take many seconds to run. The operator station needs to have adequate computational performance to provide the operator with adequate response from physics modeling codes - ideally in under four seconds.

Consistency in the operator interface is required to reduce the number of interactions the operator needs to learn to monitor and control the many diverse processes. This can be done by minimizing the number of interactions available, by standardizing the meaning of symbols and colors and by consistent display layout.

Display configuration and maintenance cost is based primarily on the quality of the tools provided to create the operator interface displays. An interactive display builder will provide the quickest creation and modification of displays with the highest reliability as it will only entail interactive editing of graphical objects. Hard coded displays will take the longest to create with the highest cost as they will require editing, compilation, debugging and activation to verify position, color, and shape, and function.

IV. DATA COMMUNICATION FACTORS

The communication layer has several features of importance: reliability, throughput, cost, and connectivity. The most widely used communication media is 802.3 standard Ethernet using TCP/IP. It provides a data communication rate of approximately 350K bytes per second per subnet. (35% utilization of a 10 Mbit media to reduce the collision rate). Throughput per node can be enhanced through the use of routers and bridges to isolate traffic on any subnet. The cost for an Ethernet communication interface is less than $500 per node and in many cases interfaces are included on the front end controller CPU board. Higher bandwidth network technology like FDDI is also available. Using TCP/IP, FDDI has an approximate 8 Mbyte throughput (80% of the 100 Mbit media; token ring does not need a collision margin). The cost per node is approximately $5,000. It is possible to mix FDDI and Ethernet using commercial bridges and routers. Efficient protocols, intelligent buffering, blocked message construction, and data compression can also help reduce the communication utilization. Buffering must not however, introduce excessive latency for operator notification (100 milliseconds or more).

The physical layer is only one aspect of network/system performance; there is also the use of a communication layer. The communication layer provides a means to isolate the various functional modules of an application, for example, isolating details of the data acquisition function from the data archiving function. If there is no imbedded knowledge of the location of some piece of data, system growth or re-configuration will only impact the portion of the application that is being modified. If there is embedded knowledge, a slight modification could cause a perturbation in the entire control network. B. Kuiper warned designers "to take appropriate measures to safe guard the upper part of the control system from importing the intricacies and diversity of the far front-end". [1]

There are two primary methods of moving data between nodes of a network: polling data into a target node and notification on change of state. Polled updating of a centralized data node or display is conceptually simple, provides redundant data for improved data security, and consumes a lot of front end computer cycles and network bandwidth. Polling requires the continuous communication of all data channels, so higher update rates use more network bandwidth, while lower rates increase the latency between a change of state and operator notification. Polling improves the data security, but makes acquisition of beam synchronous data in a generalized way more difficult. Variability in data latency in polled data systems will have a deleterious effect on the stability of closed loop control.

In contrast, notification on change of state significantly reduces the needed communication bandwidth for discrete (binary) variables and slow analog signals with reasonable deadbands. Beam diagnostics data however, may need to be sent on every sample. Notification on change requires guaranteed delivery of notifications, where polling may to some degree, compensate for a lost message. Event driven acquisition, a variant of notification of change, is an efficient method to provide stable closed loop control data. The best system design will support both time driven and notification on change to balance data communication efficiency and to insure data integrity.

Connectivity is extremely important in providing maximum flexibility for control and monitoring. There are important cases where front-end controllers need information from each other to provide optimization, closed-loop control and sequential control. A lack of point to point connectivity will result in added latency for these inter-computer control strategies and may result in an inability to provide needed control.

V. FRONT END COMPUTER FACTORS

The most important aspects for the front-end computers are performance and ease of configuration. Single board computers running a real-time operating system provide a high performance, general use environment. In a physical memory mapped environment, no operating system overhead will be added for paging or swapping virtual memory. Response to outside stimuli can occur in less than five microseconds when action can be provided in an interrupt routine and about thirty microseconds when a context switch is required. The use of a configuration database or class library can provide an easy to configure and more reliable application since the base software for all front-end controllers is identical For example, in the EPICS control system software being produced by a collaboration of Los Alamos, Argonne, Lawrence Berkeley and Superconducting Super Collider Laboratory, [5] processing an input has been timed at about 80 microseconds per signal (read, convert, check for alarms, notification on change of state). It is easy to achieve 10 Hz closed loop operation of hundreds of control loops in the EPICS operating environment. 100 Hz operation of 10s of control loops is also possible. Kilohertz bandwidth closed loop control using DSPs and MHz operation using wide bandwidth hardware feedback is also possible using a VME/VXI front end controller backplane to monitor and control setpoints at slower rates. The ability to reduce data in the front-end controllers allows the system computational requirements to be distributed over many front-end computers. Moving the data conversions, closed-loop control, interlocks and sequential control closest to the physical I/O provides the highest performance possible. It also improves reliability by reducing the number of control system components required to maintain control in any local area.

Signal conditioning and field instrumentation must be selected for performance, cost, and reliability. There is a wide variety of field instrumentation techniques available. Using the backplane of the front end computer for communication to the field instrumentation provides the highest throughput. This is very useful for high repetition rate and short latency responses like those required for beam instrumentation. There are also a variety of commercial field buses that provide wider distribution of the I/O, better environmental tolerance, and short instrumentation cable runs. Industrial buses can also provide I/O redundancy, hot swap, and convenient field cable connections.

This significantly reduces installation and maintenance cost and down time.

It is worth mentioning the need to correlate data taken in an accelerator. Three system design approaches are: distribute the data acquisition and provide a correlation identifier, e.g., a time stamp, control the data collection rate by triggering data acquisition system wide, or collect all of the data at a single point. In the case where the data is identified with a time stamp as belonging to a unique event, data collection can run at the rate of the data source event. In the case where the data is taken in complete synchronous sets at one time, the data acquisition is synchronous in the entire system and therefore can only be gathered at a rate limited by the availability of a complete data set. Finally, in the case where the data is collected to a single node, the limiting factor is predominantly the transfer and processing rate in that node and a further limitation is that all data in the synchronous set must be connected to that node.

VI. NON-STANDARD MODELS

There are a number of system design problems that are not optimally addressed by the standard model as defined above. Many of these issues can be addressed as extensions to the standard model however. Three will be addressed as examples of the flexibility of this basic architecture: large area and high signal count systems, requirements for fast global data access, and the distribution of control system data to a large multiple node user community.

The SSC has an estimated 445,000 signals distributed over a fifty mile ring. There will be five machines separated into fifteen sectors: linac (1), low energy booster(1), medium energy booster(1), high energy booster(2) and collider ring(10), where each sector must be capable of independent operation. Reliability, performance, and cost are major issues. The control system availability must exceed 99.3% to meet operational goals. To meet stringent reliability and wide area requirements telephone communication network technology was selected to provide each front end computer with communication links.[6] As with the more common Ethernet LAN, this wide area network provides point to point capability between front-end computers within a sector as well as the ability to configure a direct connection to any other sector. All of the front-end computers are connected to a high speed router in the sector and a high speed router in the main control room through a 155 Mbps OC3 communication link. This maintains the original standard model node concept of point-to-point communication and uses routers for sub-net isolation. It replaces the Ethernet and FDDI technology typical of the standard model, with broad bandwide area high reliability telecommunications gear. An extra level of flexibility is also provided since at each controller node multiple T1 (1.54 Mbps) channels may be allocated for data intensive functions, like archiving.

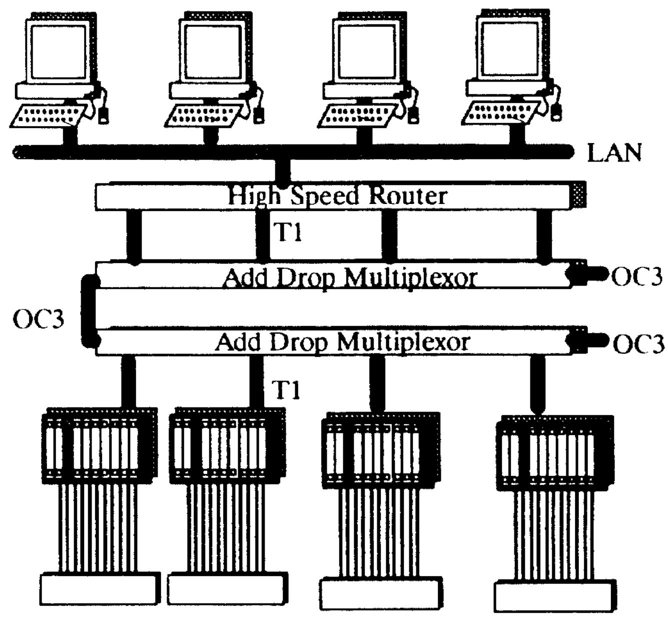

Figure 2 - SSC Architecture with additions to the standard model.

The global control beam steering problem at APS requires collecting beam position (BPM) data and providing feedback control at 4Khz. This performance issue is addressed by using an additional data communication path in a reflected memory scheme to each of 20 VME BPM controllers to provide position readback for all BPMs to all 20 controllers within 50 microseconds.[7] A correction is formulated by individual digital signal processors that solve the correction matrix for magnet control. The additional communication bus overlays the standard model control network that provides general monitoring and supervisory control.

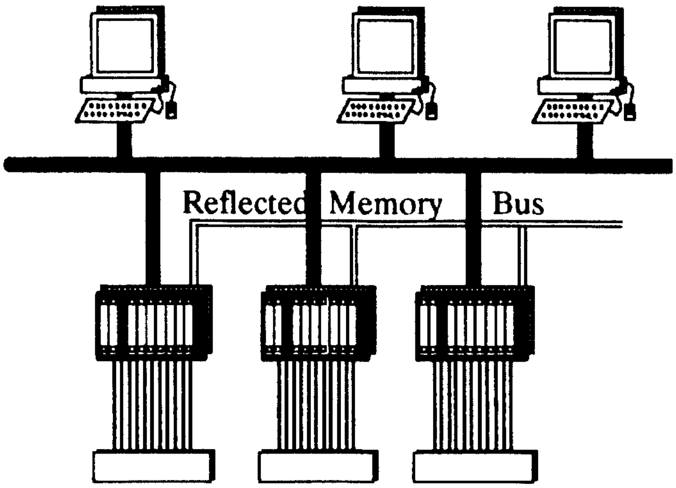

Figure 3 – APS Architecture with Global Data Path

With many user facilities, there is a need to send status information to a large number of users. In the standard model systems, this would place a high burden on the front-end computers for duplicating information to non-critical locations. With the addition of a data gateway, a minimal load is added to the standard model control network, while providing isolation for the control network from the user demand for data. There is an additional latency added to this data, a possible throughput bottleneck, and the potential for a single point of failure. If this function was part of the machine control, these three limitations would be of paramount importance. However, for providing status monitoring, these limitations are not critical.

VII. CONCLUSION

The standard model architecture has been used very successfully on dozens of distributed control systems with thousands of data channels. It provides performance, flexibility and cost benefits when implemented with present workstation, LAN and VME/VXI microprocessor technology. Standardization of network protocols (TCP/IP), open software standards, communication layer protocols, [4] workstation operating systems, and POSIX compliant real-time operating systems provide the ability to expand the size and automation of a system as requirements change, the ability to accommodate new high performance platforms as technology advances and most importantly, to share and re-use software.

The standard model has demonstrated an ability to meet demanding requirements by accommodating overlays of alternate technology while leaving the basic structure and function unchanged. This ability to adapt gives the software designer some level of assurance that programs designed for a local application may indeed find extensive use at other facilities using standard model architectures.

VIII. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the direct contributions and many productive discussions with the members of the EPICS co-development collaboration. We specifically thank M. Knott, M. Kraimer, and F. Lenkszus at Argonne National Laboratory, S. Lewis at Lawrence Berkeley Laboratory, D. Gurd, D. Murray and S. Hunt at the Superconducting Super Collider Laboratory and J. Hill and A. Kozubal at Los Alamos National Laboratory.

IX. REFERENCES

[1] B. Kuiper, "Issues in Accelerator Controls", Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems", (ICALEPCS '91), Tsukuba, Japan, November 1991, pp. 602–611.

[2] R. Humphrey, "Lessons from the SCL for Future LC Control Systems", Proceedings of ICALEPCS '91, Tsukuba, Japan, November 1991, pp. 14–18.

[3] W. McDowell, et. al. "Standards and the Design of the Advanced Photon Source Control System", ICALEPCS '91, Tsukuba, Japan, November 1991, pp. 116–120.

[4] J. Hill, "Channel Access: A Software Bus for the LAACS", ICALEPCS '89, Vancouver, BC, Canada, October 1989, pp. 352–355.

[5] L. Dalesio, et. al. "EPICS Architecture", ICALEPCS '91, Tsukuba, Japan, November 1991, pp. 278–282.

[6] S. Hunt, et. al. "BubbaNet" – A High Performance Network for the SSC Accelerator Control System", these proceedings.

[7] F. Lenkszus, et. al., "Data Acquisition of Beam Position Monitor for the Advanced Photon Source", these proceedings.