# ACCELERATOR CONTROL AND GLOBAL NETWORKS – STATE OF THE ART*

D. Gurd, SNS, ORNL/LANL, Oak Ridge, TN, USA

*Abstract*

As accelerators increase in size and complexity, demands upon their control systems increase correspondingly. Machine complexity is reflected in complexity of control system hardware and software and careful configuration management is essential. Model-based procedures and fast feedback based upon even faster beam instrumentation are often required. Managing machine protection systems with tens of thousands of inputs is another significant challenge. Increased use of commodity hardware and software introduces new issues of security and control. Large new facilities will increasingly be built by national (e.g. SNS) or international (e.g. a linear collider) collaborations. Building an integrated control system for an accelerator whose development is geographically widespread presents particular problems, not all of them technical. Recent discussions of a "Global Accelerator Network" include the possibility of multiple remote control rooms and no more night shifts. Based upon current experience, observable trends and rampant speculation, this paper looks at the issues and solutions - some real, some probable, and some pie-in-the-sky.

## INTRODUCTION

The phrase "state-of-the-art" can mean either of two quite different things. Whereas literally it describes current practice, it is also often used to describe "cutting edge" concepts, perhaps not yet quite ready for prime time. This paper will discuss some trends in accelerator controls which may be expected to lead from current practice to practical systems of the future.

Three main factors drive developments in accelerator control: the ever increasing scale and complexity of accelerators themselves; their ever more demanding reliability requirements and the fast pace of technology change in our discipline. Trends can be discerned by following the proceedings of the biennial accelerator controls conferences – The International Conference on Accelerator and Large Experimental Physics Controls Systems (ICALEPCS). Last year's conference was held in Gyeongju, Korea, and next year's is scheduled for Geneva, Switzerland. Many of the observations in this discussion are derived from those conferences. Although this conference is focused on linacs, the techniques and technology of linac control systems do not differ significantly from those applied to circular machines, and the following discussion applies equally to both.

## THE CONTROLS "STANDARD MODEL"

For many years, the controls community has referred to a "Controls Standard Model." Although this three-layer distributed model has evolved in the details of its various implementations, it has remained surprisingly constant for over a decade. A good example, drawn from the LHC Design Report, is the architectural representation of the LHC control system shown below in figure 1. The three layers are known respectively as the "presentation" tier, the "application" or "server" tier and the "resource" tier.

Although well-established institutions (DESY, CERN, SLAC, BNL, Fermilab, etc) have built new facilities based upon this standard model but using legacy control system tools and technologies, it is not an exaggeration to suggest that the last decade has been dominated by the use of EPICS – the Experimental Physics and Industrial Control System [1] – developed jointly by Los Alamos National Laboratory and Argonne National Laboratory and embracing the contributions of many collaborators. The EPICS collaboration includes well over 100 licensees, and most "green-field" facilities started in the past ten years – including KEKB, SNS, ISAC at TRIUMF, SLS and Diamond – have opted to use EPICS, benefiting from a reliable, high-performance base and well-defined user interfaces that allow easy interfacing and widespread sharing of locally-developed applications and embellishments. As many as 50% of papers presented at recent ICALEPCS meetings come from the EPICS community, and the semi-annual EPICS collaboration meetings sometimes draw over one hundred participants. More recently, EPICS, which had been available free only to not-for-profit institutions and two competitively licensed companies, has been released as open software and distributed to industries and individuals, expanding its use even further. Several companies advertise products and instruments with EPICS drivers, and industry has been successfully contracted by SNS, SLS and others to develop EPICS-based subsystems for integration into otherwise home-built systems.

EPICS represents one possible approach to implementation of the standard model. Although EPICS users select from a variety of available tools, all implementations have two things in common – a communication protocol known as "Channel Access" and a common distributed database design. Without these it isn't EPICS; but after these almost anything goes. EPICS continues to be selected for new machines because its open architecture allows the use of modern (state-of-the-art) technology at all levels. Channel Access is layered upon the Ethernet TCP/IP protocol, and so EPICS has benefited from the ever-improving performance and cost

* Work supported by the US Department of Energy under contract DE-AC05-00OR22725

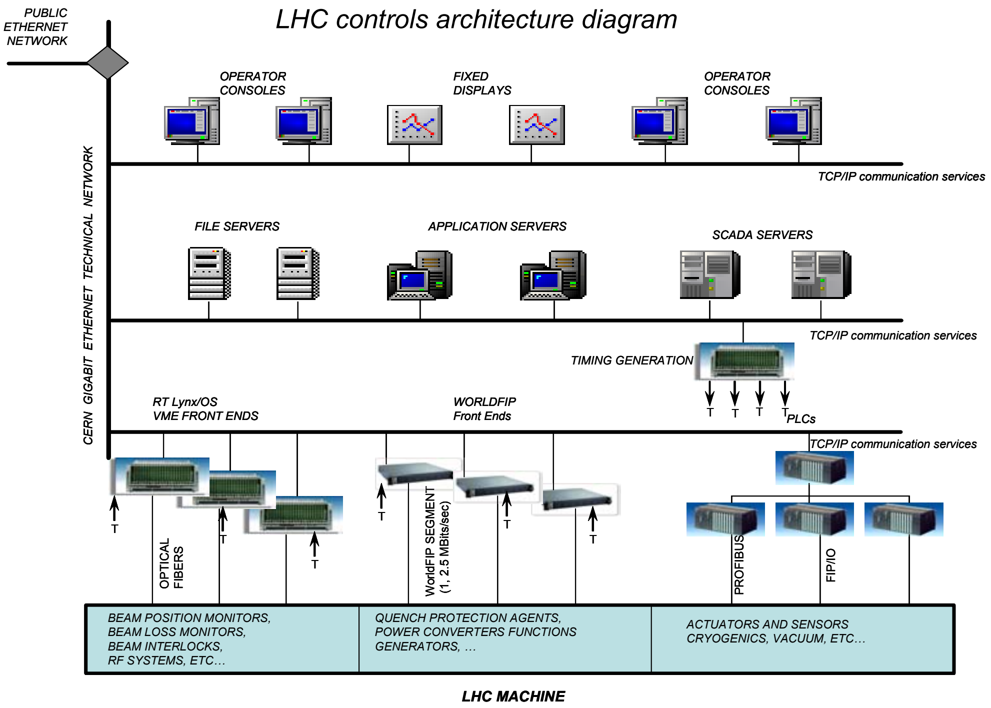
LHC controls architecture diagram
PUBLIC ETHERNET NETWORK
CERN GIGABIT ETHERNET TECHNICAL NETWORK
OPERATOR CONSOLES
FIXED DISPLAYS
OPERATOR CONSOLES
TCP/IP communication services
FILE SERVERS
APPLICATION SERVERS
SCADA SERVERS
TCP/IP communication services
TIMING GENERATION
T
T
T
T
PLCs
RT Lynx/OS VME FRONT ENDS
WORLDFIP Front Ends
TCP/IP communication services
T
T
T
T
T
T
OPTICAL FIBERS
WorldFIP SEGMENT (1, 2.5 MBits/sec)
PROFIBUS
FIP/IO
BEAM POSITION MONITORS, BEAM LOSS MONITORS, BEAM INTERLOCKS, RF SYSTEMS, ETC…
QUENCH PROTECTION AGENTS, POWER CONVERTERS FUNCTIONS GENERATORS, …
ACTUATORS AND SENSORS CRYOGENICS, VACUUM, ETC…
LHC MACHINE

Figure 1: The LHC Control System Architecture as an example of the "Controls Standard Model" Ref. [2]

competitiveness of Ethernet hardware. New designs use off-the-shelf one-gigabyte backbone technology and fast switches. The well-defined channel access interface has also made it possible to incorporate modern fieldbuses and I/O devices as they become available. Most importantly, the EPICS open architecture has made it possible to incorporate commodity PC technology at all three layers of the standard model: Linux-based presentation layer operator interfaces; Linux-based application servers and Linux- or Windows-based I/O controllers. This openness is one of the most important features of the state-of-the-art in accelerator controls.

In a sense, then, (the sense in which it continues to be selected) EPICS *is* the "state-of-the-art" for accelerator control. Nonetheless the EPICS community is aware that it is working with technology and an architecture that is over fifteen years old. While "state-of-the-art" in one sense, it is most assuredly no longer "cutting edge." Moreover, it is clear that EPICS "out-of-the-box" will not be adequate to address the formidable challenges of potential new machines, such as the proposed linear collider. As a result the EPICS community has formed an international "EPICS 2010" committee to examine the requirements for a control system for the next decade, and to determine how EPICS might evolve to meet those requirements. Participants from outside the EPICS community have been included, and the underlying idea is that the outcome of these discussions might have nothing but the name in common with the EPICS of today.

The EPICS design sacrifices much on the altar of performance. The emphasis has been on optimizing the underlying infrastructure, and the application layer has been largely ignored. As suggested by the name of its protocol, all communication is at the level of a "channel," – a single process variable (PV) or signal – analog, binary or "soft." EPICS has no inherent concept of an accelerator model or of a group of related signals making up a "device." There have been various attempts to redress this deficiency, the most successful being the "cdev" protocol from JLab which found some acceptance both inside and beyond the EPICS community. EPICS purists, however, have seen the device concept to be unnecessary at the lower communication layers, and have been unwilling to pay even its rather small performance penalty. Physics users and application programs however think in terms of devices, and devices are "objects" with an appeal to computer science purists. At SNS accelerator objects and object-based application programs have been successfully grafted on to the control system at the top using the home-grown, Java-based XAL accelerator object library, toolkit and framework and a native Java Channel Access is under development. The desire for a more integrated object-oriented approach, however, has resulted in other, non-EPICS, implementations of the controls standard model. We use LHC as an example, although there are many others using similar concepts, such as the TANGO system developed at ESRF and recently adopted by Soleil. [3]

The similarities in technologies between the overall hardware architectures of the EPICS-based SNS control system and the non-EPICS LHC design illustrated in Figure 1 are striking. Besides the obligatory three layers, both are based upon a gigabit Ethernet communications backbone. At the "resource tier" both use VME and PC-based Front End Computers (FECs) running a standard real-time kernel as well as Programmable Logic

Controllers (PLCs) for slower or non-synchronized I/O. Both use a limited number of standard fieldbuses. (Specific PLCs and fieldbuses are different, reflecting different European and American preferences and availabilities.) In both cases the upper layers – application servers and operator consoles – are based upon PCs running the Linux operating system.

The differences are in the software architecture, and in particular in the use of an elaborate "Controls Middleware" which is the collection of protocols, frameworks and Application Programming Interfaces (APIs) that allow the various layers to communicate. For LHC, this software infrastructure is built around a standard Machine Control "Framework" (J2EE), which binds the three levels together and provides all the needed common services, freeing the applications programmer to concentrate on the applications. The LHC design is object-oriented and uses the widely-accepted higher-level, standardized "Common Object Request Broker Architecture" (CORBA) protocol. The concept of an accelerator device is inherent in the protocol, and an accelerator model is implicit in the implementation. "Wrapping" the communicated objects incurs a performance overhead cost, but the payback is in the more natural relationship between the accelerator model and applications on one hand and the common framework developed for the front end computers on the other. The LHC software design is surely state-of-the-art, and variations on this approach are widespread in the community. The most recent ICALEPCS featured a session devoted entirely to middleware and frameworks. At LHC, a similar framework (UNICOS) provides the unifying infrastructure and services for the PLC-based industrial controls.

## COMPLEXITY

It is interesting that notwithstanding reduced hardware costs, substantially improved performance and the availability of free control system toolkits such as EPICS, the cost of accelerator control systems as a fraction of total facility costs has not changed significantly over the last three decades. The reason is that as these costs have gone down, machine complexity and corresponding demands on their control systems have increased equivalently. In big new machines, complexity manifests itself in sheer size and numbers. In smaller ones, often requiring control of very high power beams or having very demanding up-time requirements, the issue may be a need for complex automation, redundancy or complicated control and feedback algorithms. Increased accelerator complexity challenges control systems in several ways:

### *Scale:*

The most obvious impact of big machines is big control systems. As the number of distributed processors goes up, the amount of data to be communicated and therefore network bandwidth requirements goes up correspondingly. Network technology has so far kept pace. Gigabit Ethernet is readily available with appropriately fast switches and routers and the switched architecture gives considerable flexibility in controlling traffic patterns. It is essential to consider the scalability of the software architecture. As an example, the EPICS Channel Access communication protocol does not assume a name service and uses network broadcasts to make connections. This approach does not scale well – the number of broadcasts could overwhelm slower devices on the network – and the introduction of name service (already developed at JLab) will be required for larger systems.

### *Configuration Management:*

As numbers go up, managing the control system configuration becomes increasingly difficult. State-of-the art control systems use a number of computer-based tools to keep track of their control system configuration, including hardware and network configuration and software revision control. The more processors there are the more likely that multiple software versions require tracking. Hardware configuration is commonly maintained in a relational database – Oracle is the preferred product in the accelerator world – and software revision and release control is often managed by the Concurrent Version System (CVS) or a similar system. The APS at Argonne has developed a powerful on-line tool that populates a configuration database automatically whenever a front end processor is rebooted, assuring correctness by using the same files as were just loaded.

### *Data Management*

Large machines produce large amounts of operational data that needs to be saved for subsequent analysis. The recent Au/Au run at RHIC archived about 6 Gbytes/day for the duration of that six month run in addition to 3 Gbytes/day of post-mortem data. Estimates for SNS are comparable (without, at present, the post-mortem), using a new EPICS archiver designed to log 10K process variables/sec. The LHC design document anticipates routine logging of $10^5$ – $10^6$ variables with frequencies of up to 0.1 Hz, in addition to "several gigabytes" of post mortem data on events such as quenches. The data is inherently "bursty," so very high speed disk access is required. Here again we are saved by technology. For example, relatively "low-cost, entry-level" (advertiser's words) Fibre Channel-based storage arrays are available that achieve 320 Mbytes/sec data transfer rates and have a capacity of 6 Terabytes. Low demand systems can be built with commodity servers; high demand systems make use of "write-once" technology for improved throughput. Design of the archive files is particularly challenging. A compromise has to be reached between optimizing the rate of data acquisition and that of data retrieval. Acquiring these large amounts of data is challenging; managing the terabytes of accumulated data even more so. (A rough estimate based upon an early NLC design and – probably unrealistically – assuming 120Hz acquisition from 15,000 BPMs and 10,000 Klystrons and as many other miscellaneous channels anticipated acquiring over 30 *petabytes* of data/year! [4]) Strategies for long-term storage and data decimation or compression are required. Happily, our problem is still relatively small in comparison with that of the high energy physics experimenter.

### *Fast Feedback and Automation*

Modern accelerators are frequently loss-limited and too complex to operate safely manually. A high level of automation is required, as is fast feedback and/or feedforward. Model-based control requires the on-line availability of an accelerator model, leading to the widespread use of object-based programming languages and communication protocols to facilitate linking the model to the hardware. Fast feedback generally operates at speeds not achievable by the control system TCP/IP infrastructure, so separate point-to-point links (for example at SLC) or reflective memory (for example at APS) may be used. Algorithms typically execute in dedicated Digital Signal Processors (DSPs) or Field Programmable Gate Arrays (FPGAs), fortunately also keeping up in speed and memory size.

### *Reliability*

State of the art control systems are expected to be reliable. Goals of 98.8% scheduled up-time are often set. Up-time requirements for the new machines demand it. Controls hardware is inherently more reliable than other accelerator components, and the less reliable components (power supplies, for example) can (with money) be made redundant. New facilities using VME have adopted the VME64 standard that permits "hot-swapping" for faster repair times, but software drivers need to be changed to take advantage of this feature. The widespread use of robust industrial controllers (see below) also contributes to improved reliability. Small accelerators – in hospitals, for example – perform reliably, in general by being very simple. (I recently asked a hospital technician operating a surgical robot how often he had to reboot. He didn't even understand the question!) It is harder to achieve reliability in large, complex systems. Reliability issues are most likely to occur in the software, and here configuration management is extremely important.

## SOME COMMON APPROACHES

### *Open Systems*

A trend in state-of-the-art accelerator control systems is the use of "open" or non-proprietary software systems. Until very recently, EPICS depended upon the proprietary VxWorks real-time kernel. That constraint has been removed, however, and some recent implementations of EPICS – notably the Canadian Light Source – have been successfully implemented with the open RTEMS kernel. Indeed, EPICS itself has become open and is available to anyone. The LHC J2EE framework described above is also a vendor-independent industrial standard which includes both the Java Message Service (JMS) and Enterprise Java Beans (EJB). Perhaps the most widely adopted open standard is the Linux operating system which is, at present, the system of choice for state-of-the art accelerator control systems.

### *Industrial Systems*

What seems obvious now wasn't always so. Accelerators include many subsystems that do not differ in any important way from common industrial process control systems. These are the subsystems that do not require synchronization with the beam. Indeed beam synchronous systems – RF, Beam Instrumentation, ramping and/or kicker magnets and a few others – are in the minority. It makes sense to use industrial process control for industrial processes – vacuum, cooling, cryogenics, power, conventional facilities – and modern control systems do just that. All new facilities use PLCs and standard industrial fieldbuses for these systems, nonetheless integrating them at some level with beam synchronous subsystems so data can be correlated as required. CERN has carried the use of industrial controls to the limit, employing complete commercial SCADA systems for operator interaction and deploying the UNICOS framework as an infrastructure.

### *Ethernet as a Fieldbus*

One aspect of the use of commercial controllers is an increasing number of devices connected directly to the Ethernet. Architecturally this is very attractive; however caution must be exercised. Commercial devices may on the one hand be putting unwanted traffic on the network; and on the other they may not be able to handle the high data rates already on the network. They may be confused by broadcast or multicast messages, or their processors may have no time for their own work while they make decisions about unsolicited traffic. The use of Ethernet controllers requires careful thought about network architecture and configuration in order to derive the benefit from these products without experiencing the potential problems. Nonetheless it is fair to say that new systems incorporate more and more Ethernet-based controllers.

### *PC-based Controllers*

More and more devices come with "smart controllers," many of them PC-based. At the SNS, nearly all beam diagnostic devices are controlled individually by Windows-based "Network Attached Devices" (NADs). Hundreds of them. These devices run with a stripped-down version of Windows and EPICS core; have been configured with Labview© for initial front-end processing; use memory-mapping to transfer massaged data to EPICS and then talk to the control system transparently like any other IOC.[5] The deep integration of commercial software packages such as Labview© is another related and notable trend. The presence of hundreds of PCs on the control system network presents security and configuration management issues, but not significantly different from those presented by (say) VME-based processors.

### *Timing*

Timing system requirements are similar from accelerator to accelerator: some “events” for synchronization of RF and data acquisition, time-of-day for eventual data correlation and synchronous distribution of mode, magnet ramp information, etc. While the requirements are similar, specifications on resolution, stability (jitter), number of events or length of distributed messages become increasingly demanding. The jitter specification for LHC, for example, is $10^{-10}$. The timing system at the SLS has achieved excellent performance using commercial Ethernet components to transmit 8 bit event codes at 50 MHz (20ns resolution) [6]. This system has proved very portable. The software was adapted from the APS, and the entire system is being imported by Diamond.

## COLLABORATION

Perhaps the most significant development in new and planned large machines is not technical at all, but rather more sociological. As accelerators become larger and more expensive to build, it is clear that new projects can only be accomplished collaboratively. CERN has long provided one model for collaboration in accelerator building. SNS – a collaboration between six American National Laboratories – provides a different model, in which the control system itself was developed collaboratively by the partner laboratories. In anticipation of the international collaboration that will be required for a linear collider project, a series of international workshops has been held to discuss the idea of a “Global Accelerator Network” (GAN) [7]. The name refers both to a network of collaborating institutes, and to the communication networks that would be required to allow remote participation by distant laboratories in the commissioning and possibly operation of the facility itself. The technology for remote monitoring already exists, and remote access to modern control systems is generally implemented, subject to the security restrictions of individual institutions. One extreme idea was to have three complete control rooms, making extensive use of video-conferencing technology, one or more each in Europe, Asia and America, with machine control rotating around the world from continent to continent. Day shifts only! That concept requires video-conferencing technology well beyond the current state-of-the-art, and probably never usefully realizable. A remote Main Control Room is neither state-of-the-art nor cutting-edge. It is pie-in-the-sky. More practical is the ability for distant subsystem contributors to assist with the commissioning of their subsystems. This capability was found to be very useful for SNS, and existing networks can already meet most requirements. The principle obstacle, and likely show-stopper, will be meeting ever-more-stringent computer and network security requirements.

## CONCLUSIONS

The “Controls Standard Model” has proven extremely adaptable and continues to be the model for new and yet-to-be-designed accelerators. Computer and communications technology has so far kept up with the requirements of ever larger and more complex systems. Commodity and industrial equipment play an increasing role. Notwithstanding its older software architecture, EPICS continues to be selected for many new machines under construction. The use of software “frameworks” and object-oriented techniques is increasing as more effort goes into applications and less into more routine tools. The biggest challenges for the very large accelerator control systems of the future include stringent reliability goals, large volumes of data and collaboration management. Accelerator Control is an exciting discipline; systems are built using rapidly-changing technologies and it is always tempting to use the “latest and greatest.” It is wise, however to keep in mind that “better is the enemy of good enough,” and to select mature technologies that can meet requirements when designing real machines. The true state-of-the-art isn’t necessarily the sexiest or the most cutting edge – it is the one that really works.

## ACKNOWLEDGEMENTS

This paper has been informed by discussions with many colleagues to whom I am very grateful. I would like particularly to acknowledge the contributions of Hamid Shoaee (LANL), Larry Hoff (BNL) and Bertrand Frammery (CERN).

## REFERENCES

[1] Experimental Physics and Industrial Control System http://www.aps.anl.gov/epics

[2] LHC Design Report. Chapter 14. (B. Frammery et al)

[3] A. Götz et al. TANGO – “A CORBA-based Control System” http://icalepcs2003.postech.ac.kr/db/proc_papers/MP705/MP705.pdf

[4] Hamid Shoaee. Private Communication

[5] D. Thompson and W. Blokland. “A Shared Memory Interface between Labview and EPICS” http://icalepcs2003.postech.ac.kr/db/proc_papers/TU514/TU514.pdf

[6] Timo Korhonen and Martin Heiniger. “Timing System of the Swiss Light Source,” http://arxiv.org/abs/physics/0111173

[7] Proceedings of the GoToGAN Workshop (Trieste, October 2003) http://www.elettra.trieste.it/cotogan2003/contributions.html