2.11 Control system integration

2.11.1 Introduction

This section describes preliminary plans for the integrated control system of the Cornell ERL. Starting with requirements, it reviews the control systems used at recent accelerators and outlines the proposed three-layer architecture (see Fig. 2.11.1).

It describes the need for a Data Acquisition System (DAQ) in a control system to provide a high level of throughput, data organization, and synchronization. A justification is made for following the designs of the DESY FLASH and European XFEL control systems, based on their integrated DAQ systems, many other advantages and their similarities to the Cornell ERL. The three control system layers – Front-End, Service, and Application or Client layers – are described, with their associated major components. Controls for x-ray beamlines and the cryoplant are also described, along with a discussion of reliability, availability and maintainability. The section closes with a description of IT utilities, computing and network infrastructures.

Requirements

Cornell ERL parameters relevant to controls The key control system parameters and functional requirements are summarized in tables Tab. 2.11.1 and Tab. 2.11.2.

State of the art

Three-tier model In the current standard model, an accelerator control system consists of a distributed network of computers connected via a communication protocol which is layered over Ethernet TCP/IP protocol. The system is realized in three tiers or layers: a front-end layer, a middle service layer, and an application or client layer (see the block diagram in Fig. 2.11.1).

The front-end layer handles access to all equipment controllers and sensors, hiding proprietary field or network buses from upper layers. The ubiquitous, critical timing and event system is in this layer. The service layer allows for efficient and centralized data acquisition and buffering of accelerator information. The service layer decouples the front-end systems from the client application layer by making device and accelerator-wide abstraction possible as well as providing utility services such as directory, alarm notification, and data archiving services. The client application is responsible for presentation of the machine state to the operator and also provides an environment for developing programmed control applications.

Recent and under-construction accelerators The implementation of the tiered model varies by site. EPICS (Experimental Physics and Industrial Control System) [1] is widely employed at accelerators world-wide but is rooted in an older two-tier paradigm and is often supplemented with additional components to create a system resembling the three-tier model. For example, at SNS (Spallation Neutron Source) EPICS is supplemented with a Java based application and service programming framework called XAL [2]. The XAL framework provides a modern object-oriented development environment that facilitates the creation of sophisticated Client and Service layer applications including the integration of on-line modeling. XAL has also been

Table 2.11.1: Parameters relevant for controls

Type	Quantity	Comment
Bunch repetition rate	1.3	GHz
Beams in Linac	2	Simultaneous
Linac RF stations	384	
Injector RF stations	12	
Dipole circuits	7	require $\pm 2 \times 10^{-5}$ stability
Quadrupoles	252	$\pm 5 \times 10^{-5}$
Sextupoles	49	$\pm 1 \times 10^{-4}$
Steering coils	145	$\pm 2 \times 10^{-5}$ bipolar, normal conducting, for global orbit correction
Steering coils	~ 40 ea	≤ 250 Hz, for fast feedback
Superferric steering coils	136	$\pm 2 \times 10^{-5}$ bipolar
Vacion pump	200	Read status + current (log)
NEG pump	~ 200	Monitor during activation
Ion gauges	~ 40	Status and pressure (log)
Gate valves	~ 40	Monitor status
BPM	252	Read out and record
Ion clearing electrodes	$\sim$150	Status and voltage
Thermocouples	$\sim$1000	Read and record
Cryoplant	< 1000	Control + monitor points
Utilities (water cooling system)	~ 100	Pressure, temp, flows
Network max exten	$\sim$1	km from backbone switch

Table 2.11.2: Key functional requirements for controls

Acquire and present data from $\sim 10,000$ dynamic data points
Manage $\sim 100,000$ configuration parameters
Automated management of ~ 400 SRF cavities
Precise frequency reference and event timing distribution systems
Precise orbit control and feedback based on hundreds of BPMs and steering elements
Orbit control to < 1.0 micron at insertion devices
Physical extent approximately one kilometer
Incorporate high-rate data streams providing information about the transient behavior
Maintain a historic record of accelerator configuration and data
Centralized monitoring and management of all infrastructure and equipment
Secure, remote access

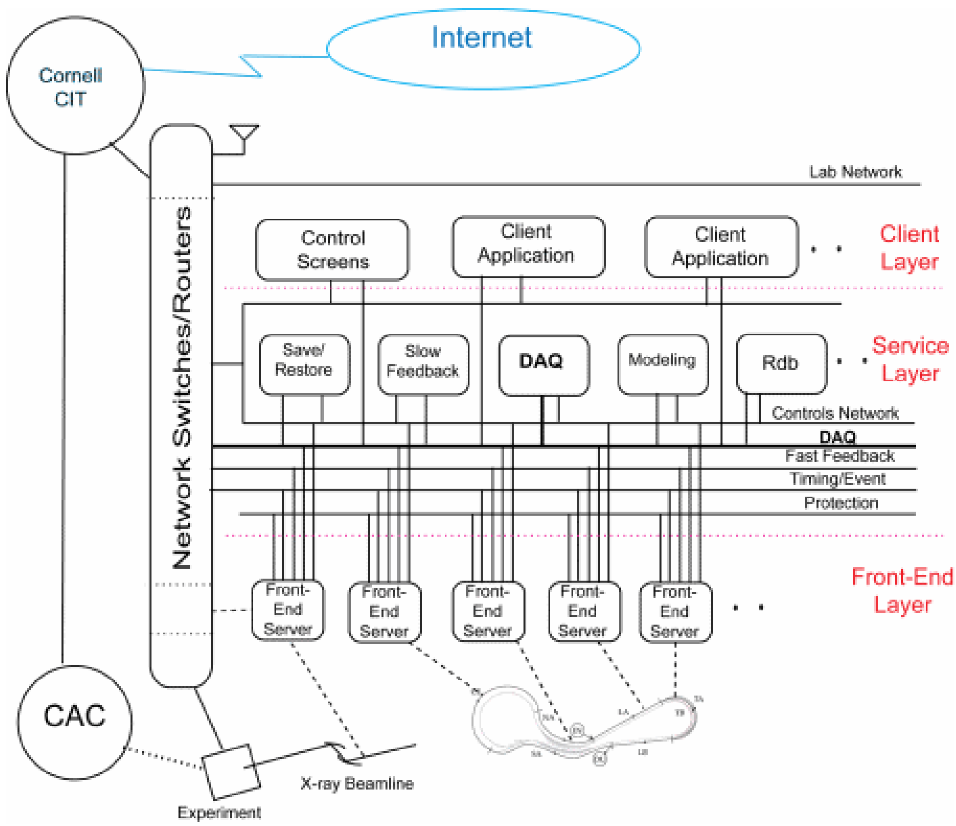

Figure 2.11.1: Overall Diagram of Proposed Three Layer Cornell ERL Control System

adapted for use with the EPICS based controls at LCLS (Linac Coherent Light Source)[3]. Rooted in work at the ALS (Advanced Light Source) the Matlab Middle Layer Toolkit (MMLT) [4, 5] extends EPICS with a Matlab based framework for the realization of client and service layer programs, including an interface to accelerator simulation tools in addition to traditional control functions. The MMLT is used with EPICS at a number of accelerators around the world. An example of a non-EPICS based realization of the standard control system model is DOOCS (Distributed Object-Oriented Control System [6]) developed at DESY beginning in the late 1990s. In this case, an entirely object-orientated approach to software development was adopted from the start and DOOCS has proven well suited to the realization of the three-tier model with the same core software functioning effectively at all three levels.

DAQ concept At DESY, DOOCS has been extended to include an integrated Data Acquisition System (DAQ). Previous control system implementations have been primarily geared to monitoring and control at low rates, typically up to approximately 10 Hz. A modern accelerator has many devices capable of generating data at rates that are orders of magnitude greater than this. While in previous control systems these devices were handled on an ad hoc basis, the DAQ system at DESY sets out an integrated approach to dealing with such devices. The DESY DAQ system [7, 8] seeks to regularize, centralize and synchronize the collection of data from high throughput devices. The DAQ system is implemented as a middle layer service thereby simplifying the task of creating control applications.

2.11.2 Cornell ERL control system architecture

Need for a DAQ in the Cornell ERL Control System The Cornell ERL will contain a number of subsystems capable of generating high rate data streams. Beam Position Monitors (BPMs), Beam Size Monitors (BSMs), Beam Loss Monitors (BLMs), Low Level RF (LLRF), and Machine Protection System (MPS), are examples of such systems. While a statistical representation of their streams (average, rms, etc.) is sometimes adequate, the analysis of beam dynamics and events, such as RF Cavity quenches or beam losses, requires the ability to capture and analyze the associated signal transients. Successful ERL operation will depend on the efficient capture and handling of these disparate data streams. A DAQ system is an infrastructure for doing just this. It requires Front-End hardware capable of streaming high-rate data in a way that is uniform across subsystems. It requires a network infrastructure capable of transporting the data with high throughput, and it requires a middle layer service capable of collecting and synchronizing the data while simplifying its presentation to accelerator physicists and client layer control applications. By making a DAQ system part of the control system plan from the beginning, one can ensure that due consideration to its requirements is given in the design of instrumentation, infrastructure, and software.

Choice of the European XFEL control system as a design model for the Cornell ERL The FLASH project at DESY operates a very successful DAQ system. The next generation of this DAQ system is currently under development to serve the European XFEL project. The XFEL control system [9], with its DAQ, serves as an excellent model for the ERL control system. The design of the Cornell ERL control system will follow closely the XFEL control system to enable re-use of hardware and software developed for the XFEL. Additionally, Cornell has a long standing relationship to DESY, including a role in the original creation of the FLASH DAQ system, which increases confidence in being able to successfully model ERL controls on the XFEL system.

DOOCS and EPICS software frameworks As mentioned earlier, the XFEL control system is built on DOOCS. At the same time, the ability to communicate with EPICS servers is included in DOOCS. Currently in operation is a control system for the Cornell ERL injector prototype which is based mainly on EPICS and includes DOOCS servers, affording the opportunity to understand the issues related to the use of both systems together. The ERL control system will include both EPICS and DOOCS components in a configuration yet to be determined.

Data transport The three tiers of the Cornell ERL Control System will be tied together via a data transport protocol which is layered over TCP/IP. The data transport protocol provided in EPICS is called Channel Access. It has demonstrated reliable, efficient communication with EPICS front-end servers (IOCs) with which it is well integrated. It is less suited for implementing communication between middle layer services and client applications. Work is currently under way on a next generation of EPICS, EPICS4 [10], and a new version of Channel Access [11]. Data transport in EPICS4 will offer greater flexibility for interlayer communication. The Cornell ERL control system designers will follow developments in this area very closely. The DOOCS control system software offers a simple and flexible data transport model. This model is based on a data object that has built-in network visibility. From the application programmer viewpoint one simply uses these network visible objects in place of local objects.

This approach has proven to be a powerful solution for both implementing layered services and supplying high-rate data acquisition at DESY. Work is ongoing at DESY towards further development of the high performance data transport in DOOCS. DOOCS is under study to better assess the feasibility of implementing ERL controls using DOOCS based software. It is preferred to have a single data transport protocol providing inter-communication between all tiers. This protocol must also be suited to implementing the high data rate DAQ component of the control system. ERL control system software choices will seek to take advantage of existing software solutions which best provide data transport suited to the range of control systems functions.

Front-end layer

The front-end layer provides monitoring and control connections to the diverse collection of accelerator hardware while presenting a standard interface to the control layers above it. There are a number of proven solutions for this layer, including EPICS and DOOCS, and both of these will have roles in the control system we are planning. EPICS will used this for illustration.

An EPICS Front-End server or Input Output Controller (IOC) runs on either a common off-the-shelf (COTS) computer or an embedded processor, communicating with the target hardware via the protocol of the target, and making the target hardware visible to the control system. For intelligent devices, real-time control is often realized by the devices themselves with EPICS providing only management and monitoring functions. For devices with less demanding requirements, real-time control can be provided by the EPICS IOC. The intelligent devices and IOCs are interconnected via the network infrastructure, configured to manage traffic and ensure reliability.

The hardware at the lowest level of the control system is dominated by serial and network connected devices with embedded intelligence. Examples of such devices include:

- Programmable Logic Controllers (PLCs) for general purpose inputs and outputs as well as for equipment and personnel protection.
- Low-Level RF (LLRF) digital controllers (one per RF cavity)
- RF power tubes
- Beam Position Monitor processors (one per BPM)
- Magnet Power Supplies
- Vacuum pump controllers and gauges.

For the most demanding instances of control hardware a hardware framework based on emerging ATCA and uTCA standards is envisioned [12]. This hardware, an outgrowth of the telecom industry, represents a forward looking replacement for VME and PCI crates and enables COTS-based solutions for demanding high-bandwidth instrumentation and control hardware.

Front-End servers run independently of the upper control layers but rely on upper layers for configuration, accumulation, and archiving of data. EPICS, with its mature suite of software tools and wide range of supported devices, facilitates the development of accelerator subsystems prior to the development of the upper control systems layers and prior to the integration of subsystem controls into the global control system.

Timing and event system The timing and event system is distributed to all parts of the ERL and, with the exception of RF phase control, is the basis for synchronizing all aspects of ERL behavior. The master timing generator is synchronized to the RF Master Oscillator and through a combination of fiber optic and coax cabling, distributes a reference timing signal to front-end equipment, which is stabilized to 10 ps. Incorporated into the hardware infrastructure, clock and timing relays in the front-end equipment utilize the distributed reference to provide sufficient timing precision to control all critical sequences in the functioning of the ERL. A second synchronization signal is planned for distribution to allow local phase-locked loops in subsystems to establish and maintain synchrony.

Event triggers are generated and distributed locally to initiate actions that are time sensitive. Event triggers have 10 ps precision, based on the master RF clock, and include a timestamp with sufficient number of bits to uniquely identify each RF clock frequency cycle. Event triggers can be programmed and generated for any purpose that is needed. Timestamps are also generated locally for asynchronous events and are included with every element of data that is collected by the DAQ system. These timestamps are central to reconstructing sequences of events, for instance, in the case of a post-mortem dump resulting from a fault signaled by the MPS.

Low-level RF system Each cavity has a dedicated Low-Level RF controller that implements high-speed controls loops implemented in firmware. A connection to the Timing System provides event triggers. There is an embedded front-end server as well as an additional DAQ interface to support capture of transient records as discussed in §2.5)

RF power controls A PLC provides monitoring and supervision for RF power devices. The PLC is connected to the control system through a front-end server but provides greater determinacy than network dependent supervision could provide, as described in §2.5.

Other sub-systems Beam Position and Beam Size Monitors have a dedicated signal processor capable of providing high rate data to DAQ and Orbit Feedback systems. A connection to the timing system provides event triggers. There is an embedded front-end server as well an additional DAQ interface, as explained in §2.10. Laser front-end controls are both PLC-based where required for equipment protection and front-end server based. Controlled hardware includes motors related to optics adjustments as well as laser beam characterization instruments.

Magnet power supplies are controlled through PLCs connected to front-end servers, as well as directly from front-end servers. Magnet power supplies that are part of the Orbit Feedback system additionally have connections to the Timing System to receive event triggers.

Vacuum pumps and instrumentation are low bandwidth devices connected to the control system through PLCs connected to front-end servers as well as directly to front-end servers. The Machine Protection System is implemented in dedicated hardware to achieve the required response time and reliability. However the system can be configured and monitored via front-end controls. It also provides data to the DAQ system as discussed in §2.12 . The Personnel Protection System will be implemented in dedicated hardware to achieve the required reliability and be entirely independent of the ERL Control System except for a gateway to allow monitoring, alarm handling, logging, and archiving.

Utilities and infrastructure Monitoring and supervision of utilities and infrastructure will be integrated into the ERL Control System. Many utilities systems, e.g. HVAC, have their own local controls supplied by their manufacturer and are required for autonomous operation and maintenance. In these cases the ERL Control System will have primarily a monitoring role, but with carefully defined control function to provide global precision environmental feedback and stabilization. There will be thorough logging capabilities that will be remotely accessible by technical staff with maintenance responsibilities.

Service layer

Data acquisition system (DAQ) For ERL feedback, monitoring and automation, the DAQ systems will provide access to data from across the ERL physical layout. These different systems and monitors will have different characteristic times of interest and thus differing data rates. However, data from these systems must be synchronized across subsystems and to the beam for analysis of transients, beam loss and commissioning diagnostics. The DAQ will centrally aggregate the data from its distributed sources, provide universal timestamps, synchronize the data to events, and make the data available to analysis programs and feedback devices. The DAQ architecture and design will closely follow that of the European XFEL DAQ [9].

Data may often be represented by statistical measurements delivered at low rates. Hundreds of kBytes of data can provide a snapshot of the machine state, and such data delivered at 10 Hz rate is sufficient for many aspects of machine operation. Transient events, however, whether unexpected, experimental, or diagnostic, may generate in excess of a GByte of data per event. Efficient operation of the accelerator requires the rapid capture and presentation of data from such events. We estimate the DAQ system will need to handle peak data rates in excess of a GByte per second. The synchronization of data from such events will, in some cases, require time stamping of data with sub-nanosecond resolution.

Archiving Data from the accelerator will be archived to RAID disks for immediate accessibility. The data will be characterized for a reduction process in a central database. This reduction characterization will typically take the following form: after one month, reduce every ten samples of a particular parameter to their average. As the data age, they will be moved to other storage systems. The retrieval of data from various sources for a client will be handled in the service layer, and the source will be transparent to the client.

Feedback and monitoring Many accelerator systems will need continual monitoring loops; some of which will need actions performed (feedback) based on the monitoring information. The rates for these loops will vary and in large part the response time desired will dictate how the loops are implemented. The faster loops ($> 30\,\text{Hz}$) will be implemented in front-end hardware (FPGAs) or front-end servers, either DOOCS or EPICS based. This placement is dictated by reducing data transfer latency and overhead. Moderate speed feedback and monitoring loops (between 1 and 30 Hz) can be added at the DAQ level. Implementing at the DAQ level offers the benefits of its synchronization, shared memory and distributed input capability. Slower loops may be implemented in the DAQ or other service layer code. At all levels, code for PID type control will be available which can be used in most of the feedback cases. Other algorithms will be implemented as needed.

Automation Automation services involve having a series of tasks completed in a specific order or based on states of accelerator hardware or subsystems. These automated events can be user- or machine-triggered. User-initiated automation will be implemented with well established sequencer software based on State Notation Language (SNL). Machine-initiated automation will be based in hardware (FPGAs, PLCs), in servers (EPICS or DOOCS), or in Finite State Machines (FSMs). Automation requiring machine synchronization can be implemented in the DAQ.

Modeling An accelerator model service will be provided in the service layer. This service will be able to provide accelerator model information to applications and other services. The model can be generated from current machine conditions as an online model, from archived machine conditions, or from accelerator physics applications. The model can then be used for analysis or to test changes to the machine in a predictive mode.

Alarm handling service A large accelerator is a capable of generating many simultaneous alarms of different severity and exhibiting various degrees of interrelatedness. Providing a hierarchical, filtered view of the accelerator alarm state is critical to efficient accelerator operation and appropriate alarm response. Alarm conditions in the front-end layer are collected by the service layer alarm handler. The alarm handler applies filters to alarm sets and provides for notification to client layer applications as well as supporting subscription to e-mail, text and voice notification. Alarm configuration is centralized. We plan to use relational databases and web technologies such as Java Message Service which are well suited to implementing an alarm handler middle layer service [13].

Save and restore service The Save and Restore Service provides for capturing all settings associated with a particular machine state and for returning to that state upon demand. In addition to servicing the client layer it provides a web interface for management and comparison of saved machine states.

Web services An electronic logbook will be provided for creating, editing and viewing operational and experimental notes. The logbook capabilities include:

- Secure viewability on- and off-site
- Robust accessibility during critical machine failures
- Automatic acceptance of entries created by client and service layer software
- Capability of alerting groups of people to entries in their area of expertise
- Attachments of images and other supporting documentation

Several free open-source electronic logbook options are available which either meet all these requirements as is, or can do so with minor modification. A selection for deployment will be made at a later time. Important experience is being gained through development of the electronic log for the ERL prototype injector.

The control system will also provide support for comprehensive web-based monitoring as well as various message services (e-mail, text, voice) for alarm, event notification, and other operational needs.

Relational data base Storage in a database is needed for accelerator monitored data and input configuration data. The monitored data needs to be archived and intelligently filtered for permanent storage. This data must remain accessible to client applications continually and the interface to both the archived data and the latest dynamic data read from the accelerator should be seamless. Client applications will need complicated and varied data from the accelerator, so the database must have robust and adaptable querying capability.

The synchronization of the control system configuration with the actual state of the accelerator facility is a critical and difficult-to-achieve goal. It is important that configuration data be kept up to date across all systems and services, so ideally this configuration data should have a single point of entry from which the data is then propagated to all relevant systems. An accelerator configuration database will include among its capabilities the automated generation of software configuration files.

Client layer

There are many applications in the client layer, the most important examples of which are described in the following paragraphs. The control screen environment for operations and monitoring must meet the following criteria:

- A rich group of screen building block components should be available
- It should be easily extensible with new tools, code, and plug-ins
- There should be an easily accessible central location for screens so that keeping distributed screens up to date will be straightforward.

We are evaluating the jDDD[14] screen tool-set and display program, as used at DESY, as it meets all of these requirements. jDDD also offers the further advantages of including built-in Java Web Start capability and Subversion repository support. EPICS also provides a similar set of tools for generating controls screens.

A client application that connects to the alarm service will make alarm information widely available and viewable. The control and acknowledgment capability will have appropriate access control.

There will be a save-restore client that will provide an interface to the save-restore service. This client will allow full and partial saving and restoring of acceleration sections. The user will be able to compare saved conditions to each other and to the current machine state, as well as view a history of changes loaded into the accelerator.

A channel browser client will provide users with the ability to search for and obtain information about control system channels. A tool to plot various control system channels in real-time and in combination with archived data will be available. The same or another tool will be used for viewing historical data and to examine beam transient or loss data.

It is important for students, users, scientists and staff to develop analytic and experimental programs that interface to the control system quickly and efficiently. The control system at the ERL should leverage their existing knowledge base as much as possible. To that end, control system interfaces to commonly used programming languages and tools will be provided. C/C++ and Java will provide the backbone of application programming languages. Matlab and LabView will provide interactive experimental environments. In addition, Matlab and Python/Jython will provide scripting support.

The control system will permit remote (off-facility) access, under extremely limited and controlled circumstances, through fully authenticated, secure connections. Remote users will have control permissions which vary with user, accelerator state and instance.

Capture of post-mortem information in the event of a fault is handled as a standard capability of the DAQ system. The MPS trigger, and any other error conditions that needs to be configured become event triggers for the DAQ, and the entire array of ERL information flowing into the DAQ prior to the trigger is automatically captured and stored for analysis. Software for convenient analysis of these dumps will be written.

X-ray beamline and experiment

Control requirements X-ray experiments require exquisite precision and stability to exploit the full potential of the ERL. Micron-sized x-ray pulses with psec timing are needed with stability requirements that are one tenth of these parameters. This requires similar stability in all dimensions of the electron beam and of the undulator where the x-rays are generated. In addition the experiments require considerable flexibility on the part of the ERL, to change undulator gaps and phases as needed, without compromising the stability of beam alignment or operating conditions for any of the other experiments. Also, the x-ray beamline and experiment controls have to function autonomously from the accelerator controls when their shutter is closed, with complete freedom to set up, align and test beamline and experiment equipment.

X-ray beamline control approach and control requirements The beamline controls are not separate from the ERL controls in that they share the same network, computing technology, controls center, and development staff. They will be logically distinct, as implemented in the controls software, with a carefully defined gateway between the two sides. This has been implemented successfully at light sources such as the ALS [15]. As the experiment calls for changes to undulator settings (e.g. gap and phase) across the gateway, previously defined feed forward and feedback mechanisms fully compensate the ERL orbit parameters to negate the effect. Precision beam stabilization may be required across the ERL to x-ray beamline divide, but in most cases stabilization of the electron beam alone should be sufficient as the electron beam instrumentation provides greater accuracy than the x-ray instrumentation.

X-ray experiment data acquisition and computation needs In the new era of pixel detectors, the potential for data rates from x-ray experiments is enormous. At full rate, not only would network and storage capacities be exceeded, but processing all the data off-line or after the experiment finished would be entirely impractical. Clearly, methods of drastically reducing the data from these detectors, in real time, will be developed before operations begin. With this assumption, in this PDDR, we are leaving to the experimental collaborations the responsibility of data acquisition, on-line data reduction, storage, processing and analysis of x-ray experiment data. At the same time there will be on-site capacity for low data rate experiments, and particularly, for experiments that choose to move reasonable amounts of data (e.g. Gbytes to a few Tbytes) in real time to off-site locations. Cornell affiliated experiment collaborations may choose to process data at the Cornell Center for Advanced Computing (CAC), and for this, we will have 10Gbit network connectivity to address their needs. Most other experiments are expected to provide their own local data acquisition, processing and storage capabilities, transporting their resulting data for further analysis at home institutions via portable storage devices. This is entirely practical with current data storage technology.

Cryoplant controls

The cryoplant will be delivered with its own autonomous turn-key control system, located within the cryoplant itself. It will be created by the cryoplant manufacturer and will be sufficient for its independent commissioning and operational function. In addition to autonomous operation from a cryoplant control center, there will be a gateway between it and the accelerator control system, which will allow monitoring, logging, archiving, and remote control functions for the cryoplant by the ERL control system, including routine operation from the ERL control room. The gateway will also pass information regarding changes in operational status, load requirements and alarm conditions. Timing requirements and data rates are modest and soft control over network will provide sufficient performance.

Although consideration will be given to compatibility between the two control systems, cryoplant controls will be primarily the responsibility of the plant manufacturer. The requirements given to the manufacturer will be carefully specified to ensure desired operational performance under conditions existing at this site, including recovery from power irregularities, and the need for remote operation from the ERL control room. In this we will be guided by the wide experience at our sister laboratories through close consultation with their experts.

Reliability, availability, and maintainability

The necessary high availability of the Cornell ERL will require the design of the control system to address issues of reliability and maintainability in multiple dimensions. A prerequisite of a high availability system is reliable hardware components, thus control hardware will make use of commercial components with demonstrated reliability. Reliability will be enhanced through redundancy when appropriate for critical systems.

When control system failures do occur it is essential that the time required to identify and recover be minimized. To this end extensive hardware status monitoring will be essential for all systems. The control system will provide for automatic and manual remote switchover to spare systems as appropriate. Recovery related disturbances will be reduced through the use of systems which support hot-swap of components.

Controls software contributes to availability in multiple ways. First, the software itself must be reliable, a goal that can be approached through modern design methodology as embodied in object oriented languages. The software must be verified on test stands and have defined release and bug tracking systems. Software changes will be documented by a version control system. Software developers will be provided with a modern integrated development environment to facilitate code development and management of code changes.

Coordination of hardware and software to achieve high availability requires a dedicated infrastructure of a relationship database and organizational procedures which insure that the database continually reflects the state of the accelerator, its equipment and current configuration. The Alarm Handler Service will facilitate rapid identification and recovery of failures. The accelerator database will help reduce failures related to configuration conflicts and errors while facilitating failure recovery and preventive maintenance.

Software and firmware management

The control system will necessarily consist of a wide variety of programs, services and codes, which will have a complicated inter-dependence such that changes made in one may require changes in many of the others. Management of the software and firmware will include use of a central-version control repository such as Subversion (SVN). SVN allows for development branches that can be tested before moving changes into a production version. SVN also provides the means to distribute source code remotely to facilitate code collaboration offsite.

Use of an integrated development environment (IDE) will be encouraged for rapid development and integration of resources and application programmable interfaces (APIs). Many modern IDEs, such as Eclipse and Netbeans, include the capability to edit, build, test and profile large-scale projects. They also support version control software and conflict resolution. The IDE can provide a useful platform for fast, collaborative, and coordinated code development.

2.11.3 Computing infrastructure

Computing technology is changing so rapidly that plans for computing infrastructure can be based only on what one could build today, while making extrapolations into the future for storage capacity, processor speed, and network performance. When it comes to construction, what is actually built will be based on the best technologies available at the time and incorporate best practices learned from other facilities.

The CLASSE laboratory computing facility provides the IT infrastructure for the Cornell ERL project. Although staffed and managed independently from the rest of the university, it exists within the context of the broader Cornell University IT structure, using Cornell's backbone network for connection to the Internet and taking advantage of university provided services, such as e-mail and scheduling. Although there will be computing and data storage facilities for scientific analysis on-site, the bulk of data processing and analysis for x-ray experiments will be the responsibility of those doing the experiments and is expected to be handled off-site.

Physical infrastructure and basic services The main computing facility consists of a server room in Wilson Laboratory. This is the network hub and location of central ERL Control System servers. Equipment that does not need to be located directly on site can be located in the Physical Sciences Building and Newman Laboratory located across campus. Cornell will provide high-speed network connectivity to the Center for Advanced Computing (CAC) located approximately a kilometer from Wilson Lab., as needed, for experimental data analysis.

The CLASSE laboratory computing facility will provide central support for all the basic services required for laboratory IT such as: system administration, workstation hardware/software support, file storage, backup, and Web services.

Compute clusters, CAC, and storage Although not intended for data analysis of x-ray experiments, there will be compute clusters to meet the computational needs of CLASSE and the Cornell ERL. For computation critical to ERL operations, such as on-line modeling, a computer cluster will be located on site, but for non-critical computations such as simulations, there will be one or more compute clusters in the Physical Sciences Building with a batch queuing system for efficient shared use.

Beyond locally available facilities, the CAC can provide high-performance computing systems and data storage to this project as well as other researchers at Cornell and their collaborators. It also has high speed network connectivity to the TeraGrid, National LambdaRail, Internet2 and the New York State Grid.

Networks The ERL Control System requires extensive network segmentation and throughput. Centrally located core switches, with level three routing, are planned for network partitioning and connectivity of the backbone network. These switches isolate major traffic flows and separate the control system network from general purpose laboratory network functions for high availability control operation.

The backbone network is based on 10 Gbit, or faster, Ethernet, with 10Gbit trunk fibers radiating out from backbone switches to local front-end switches which connect 1Gbit Ethernet to equipment shelves and individual pieces of equipment. Multiple trunks are used where needed to provide adequate throughput and latency, separating types of traffic where appropriate. The Ethernet network is modular enough to permit easily implemented network hardware upgrade over time as new traffic patterns and potential bottlenecks surface.

The Cornell ERL will have wired and wireless network deployed throughout the entire facility, for general purpose and maintenance purposes. Segmentation into appropriate subnets can provide protection of the control system as well as remote maintenance of control system and ERL components. In experimental areas separate wired and wireless network access will be provided for the needs of outside users while maintaining isolation from the ERL Control System and laboratory staff networks.

Access and security The overall strategy for access and security is to provide robust authentication for everyone within the facility and to establish appropriate authorization for access based on users' roles and needs. Routers and firewalls will provide separation of the controls network from the general laboratory network and from outside the laboratory. At the same time, they will selectively allow expert remote access from the outside and expert access to remote resourcesacross the boundary, as needed, in a carefully restricted and monitored fashion. Intrusion detection, monitoring, and logging will be deployed to maintain the security and integrity of the controls network. Registration of visitors will permit appropriate restricted access.

References

[1] EPICS main Web site: http://www.aps.anl.gov/epics/.

[2] Galambos, J., *et al. XAL Application Programming Framework*. In *9th International Conference On Accelerator And Large Experimental Physics Control Systems (ICALEPCS 2003)*, pages 332–336. Gyeongju, Korea (2003).

[3] Dalesio, L. B. *NSLS II Control System*. In *11th International Conference On Accelerator And Large Experimental Physics Control Systems (ICALEPCS 2007)*, pages 253–255. Knoxville, Tennessee, USA (2007).

[4] Portman, G., J. Corbett, and A. Terebilo. *An accelerator control middle layer using MATLAB*. In *Particle Accelerator Conference (PAC 2005)*, pages 4009–4011. Knoxville, Tennessee, USA (2005).

[5] Corbett, J., A. Terebilo, and G. Portmann. *Accelerator Control Middle Layer*. In *The twentieth biennial Particle Accelerator Conference (PAC 2003)*, pages 2369–2371. Portland, Oregon, USA (2003).

[6] Grygiel, G., O. Hensler, and K. Rehlich. *DOOC: A Distributed Object Oriented Control Sustem on PC's and Workstations*. In *International Conference On Accelerator And Large Experimental Physics Control Systems (ICALEPCS 97)*. Beijing, China (1997). See also http://tesla.desy.de/doocs/.

[7] Agababyan, A., *et al. Data Acquisition System for a VUV-FEL Linac*. In *The 5th International Workshop on Personal Computers and Particle Accelerator Controls, PCaPAC 2005*. Hayama, Japan (2005).

[8] Agababyan, A., *et al. The Data Acquisition System (DAQ) of the FLASH Facility*. In *11th International Conference On Accelerator And Large Experimental Physics Control Systems (ICALEPCS 2007)*, pages 564–566. Knoxville, Tennessee, USA (2007).

[9] XFEL Control System, see please: http://xfel.desy.de/tdr/tdr/.

[10] http://epics-pvdata.sourceforge.net/.

[11] Žagar, K., *et al. Evolution of the EPICS Channel Access Protocol*. In *12th International Conference On Accelerator And Large Experimental Physics Control Systems (ICALEPCS 2009)*, pages 40–42. Kobe Japan (2009).

[12] Larsen, R. S. *Advances in Developing Next-Generation Electronics Standards for Physics*. Slac-pub (2009).

[13] Kasemir, K. *SNS Alarm System (BEAST); EPICS Meeting in NFRI, Daejeon, Korea* (2009). http://www.aps.anl.gov/epics/meetings/2009-07/index.html.

[14] Sombrowski, E., *et al. JDDD: A JAVA DOOCS Data Display for the XFEL*. pages 43–45 (2007).

[15] Steier, C., *et al. Operational experience integrating slow and fast orbit feedbacks at the ALS.* In *The ninth European Particle Accelerator Conference, EPAC'04,*, pages 2786–2788. Lucerne, Switzerland (2004).