THE ILC CONTROL SYSTEM

J. Carwardine, C. Saunders, N. Arnold, F. Lenkszus (Argonne), K. Rehlich, S. Simrock (DESY), B. Banerjee, B. Chase, E. Gottschalk, P. Joireman, P. Kasley, S. Lackey, P. McBride, V. Pavlicek, J. Patrick, M. Votava, S. Wolbers (Fermilab), K. Furukawa, S. Michizono (KEK), R.S. Larsen, R. Downing (SLAC)

Abstract

Since the last ICALEPCS, a small multi-region team has developed a reference design model for a control system for the International Linear Collider as part of the ILC Global Design Effort. The scale and performance parameters of the ILC accelerator require new thinking in regards to control system design. Technical challenges include the large number of accelerator systems to be controlled, the large scale of the accelerator facility, the high degree of automation needed during accelerator operations, and control system equipment requiring 'Five Nines' availability. The R&D path for high availability touches the control system hardware, software, and overall architecture, and extends beyond traditional interfaces into the technical systems. Software considerations for HA include fault detection through exhaustive out-of-band monitoring and automatic state migration to redundant systems, while the telecom industry's emerging ATCA standard—conceived, specified, and designed for High Availability—is being evaluated for suitability for ILC front-end electronics.

INTRODUCTION

The International Linear Collider (ILC) is a 500-GeV center-of-mass electron-positron collider [1,2]. As shown in Figure 1, the accelerator complex comprises the following major elements:

- Two 11-km-long 250-GeV linacs comprising 16,000 L-Band superconducting RF cavities housed in 2000 cryomodules and powered by 640 RF klystrons.
- An injector complex comprising a polarized photocathode electron gun, an undulator-based positron source, and 5-GeV electron and positron damping rings, each 6.7 km in circumference.
- A 4.5-km beam delivery system with one interaction region and two detectors in push-pull configuration.

The accelerator will operate at a 5-Hz pulse repetition rate, each 1-ms pulse comprising ~3000 microbunches.

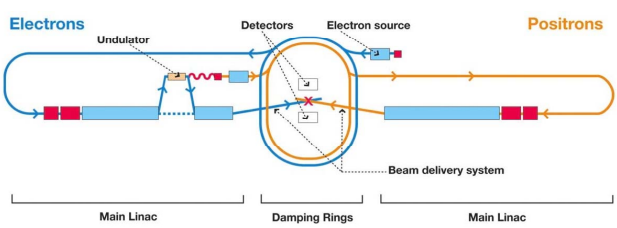

Figure 1: ILC accelerator schematic layout.

CONTROL SYSTEM CHALLENGES

In addition to providing conventional accelerator control system functionality, the control system must address several challenges that arise from the large scale and complexity of the ILC accelerator facility:

- Accelerator operations will rely heavily on automation for routine activities such as machine start-up, commissioning, tuning, and automated operation of the superconducting RF technical systems. Similarly, there will be extensive reliance on beam-based feedback, including many feedback loops running pulse-to-pulse at 5 Hz.
- An availability goal of 85% has been set for accelerator operations over a 5000-hr/year schedule. To meet this goal, a design availability goal of 99% has been allocated to the control system.
- Infrastructure and tools will be required to support worldwide remote participation in accelerator commissioning, operation, and machine support.
- Distribution of precision (sub-picosecond) RF phase reference and timing fiducials over 10s of km [3].
- The control system must be able to integrate and support technical equipment provided to the project through an in-kind funding model.

To meet the needs for automation and control, we propose to implement automation and feedback engines as control system services. Embedding functionality and APIs into the control system infrastructure will simplify development of high-level applications and allow the control system to better coordinate resources and control system activities.

A generalized control system framework is proposed for automation and feedback that would make it possible, for example, to implement a 5-Hz synchronous feedback loop with any subset of monitoring and control points anywhere in the machine. To accomplish this, all readbacks and control points from all technical equipment would have to be synchronized at 5 Hz.

Meeting control system availability goals will require increased attention to standards-based solutions, online diagnostics, resource monitoring, and configuration management.

FUNCTIONAL MODEL

The control system functional model includes three tiers, where a logical 'Services' tier is introduced between the conventional Client and Front-end tiers of older two-tier control system models. The functions of each tier are summarized below [4].

Client Tier

The Client tier comprises applications that implement the Human Machine Interface to the accelerator and technical systems and range from engineering-oriented control screens to high-level physics control applications to configuration management applications. Engineer-oriented consoles are focused on the operation of the underlying accelerator equipment. High-level physics applications will require a blend of services that combine data from the Front-end tier and supporting data from the relational database in the context of high-level device abstractions (e.g., magnets, BPMs).

It should be noted that tools for developing, deploying, and interacting with high-level applications are available at the Client tier, but that the applications themselves are instantiated in the Services Tier.

Services Tier

The Services tier provides services that coordinate many typical control system activities while providing a set of well-defined non-graphical interfaces.

An intrinsic component of the Services tier is an online relational database that includes engineering and physics models of the accelerator, which makes it possible to relate high-level machine parameters with low-level equipment settings in a standard and centralized way.

This centralization of control provides many benefits in terms of coordination, security, automation, optimization, and conflict avoidance. For example, a parameter save/restore service can prevent two client applications from simultaneously attempting to restore a common subset of operational parameters.

A suite of Services APIs will provide the primary means by which high-level applications interact with the control system.

Front-End Tier

The Front-end tier provides access to the field I/O and underlying dedicated fast feedback systems. This tier is configured and managed by the Services tier, but can run autonomously. For example, the Services tier may configure a feedback loop in the Front-end tier, but the loop itself runs without direct involvement. The primary abstraction in this tier is a channel, or process variable, roughly equivalent to a single I/O point.

PHYSICAL MODEL

In this section, we describe a physical model for implementing the functional model. The main elements are shown in Figure 2 and are described bottom-up, starting at the technical equipment tier.

Technical Equipment

It has been common practice at accelerator facilities for the control system to accommodate a wide variety of interfaces and protocols, leaving the choice of interface largely up to the technical system groups. The large scale of the ILC accelerator facility means that following this same approach would almost certainly make the controls task unmanageable, so we anticipate following an approach of specifying a limited number of allowable interface options for technical equipment (physical, protocol, technical equipment command and response).

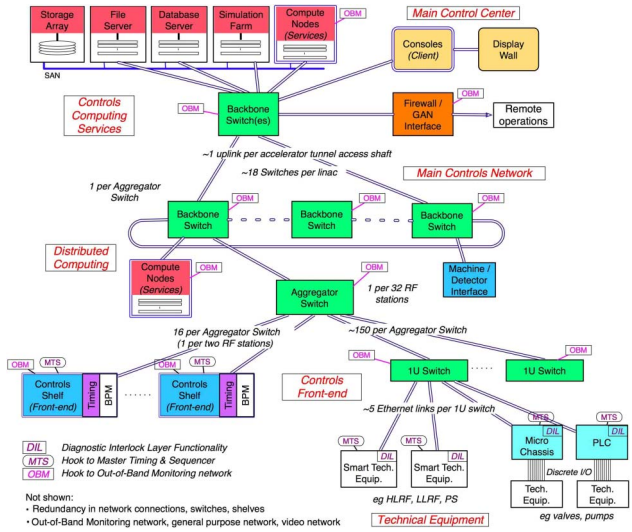

Figure 2: Control system physical model.

Controls Front End

The controls front end contains the following three main elements:

1U Switch: Aggregates the many Ethernet-controlled devices in a rack or neighborhood of racks. Some of these devices will speak the controls protocol natively, while others will have proprietary protocols that must be interfaced to the control system. It is assumed these 1U switches will reside in many of the technical equipment racks.

Controls Shelf: Consists of an electronics chassis, power supplies, shelf manager, backplane switch cards, CPUs, timing cards, and instrumentation cards (mainly BPMs). The controls shelf serves several purposes: (1) hosts controls protocol gateways, reverse gateways, and name servers to manage the connections required for clients to acquire controls data; (2) runs the core control system software for managing the various Ethernet device communication protocols, including managing any instrumentation (BPM) cards in the same shelf; and (3) performs data reduction, for example, so that full-bandwidth RF/BPM waveforms need not be sent northbound in the control system.

Aggregation Switch: Aggregates network connections from the 1U switches and controls shelves and allow flexible formation of VLANs (Virtual Local Area Networks), as needed.

Distributed Computing

The distributed computing tier is the highest level of backbone switches in the underground tunnels. The switches will most likely be configured in a hybrid loop and mesh topology to allow for the large-scale movement

of data necessary for global feedback. Dedicated compute nodes associated with each backbone switch allow localized instantiation of control system services, such as monitoring, data reduction, and implementation of feedback algorithms.

Controls Computing Services

Conventional computing services dedicated to the controls system will include storage arrays, file servers, databases, and compute nodes. The overall philosophy is to develop an architecture that meets the requirements, while leveraging the cost savings and rapid advancements in the performance of COTS components [5].

To give some idea of the scope, preliminary front-end component counts for the overall accelerator complex are summarized in Table 1.

Table 1: Summary of Controls Equipment

Controls Equipment	**Counts**
1U Switch	8356
Controls Shelf	1195
Aggregation Switch	71
Controls network backbone switch	126

CONTROL SYSTEM SERVICES

From the perspective of a user of the client tier, the Services tier is largely invisible. The goal of the Services tier is to provide services that manage the execution of logic in the problem domain and leave the problems of user interaction and graphical presentation of data and status to the Client tier [6]. This approach can be considered in part because it should be possible to create a well-defined interface for common control system functions, similar to efforts in the business world to create standard interfaces for business transactions. Figure 3 shows a functional view of the Client and Services tiers.

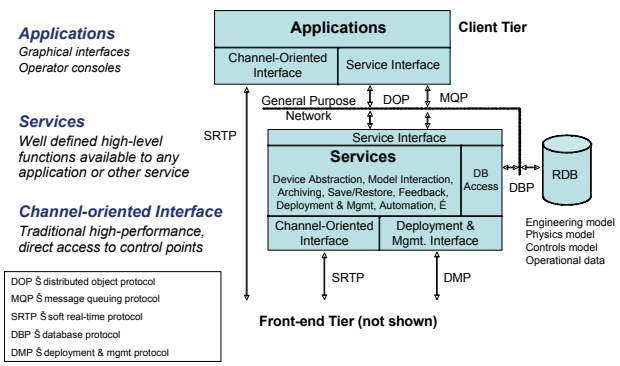

Figure 3: Functional view of Client and Services tiers.

The Services model allows rapid prototyping of high-level applications through composition, while maintaining impedance to changing the core functions. It also supports activities that are not well suited to channel-oriented interfaces either because they may involve a sequence of interactions and require access to multiple control system parameters, or because they would be added and removed frequently during operations and therefore may require dynamic allocation (network latency and/or CPU loading). Table 2 is a list of possible services.

Table 2: Examples of Control System Services

Script execution service	Device server
Archiving service	Data concentrator
Math & logic functions	Feedback/dynamical control
Logging service	Video & image processing
Save, compare, restore	Out of Band monitoring
Alarm management	Exception handling
Relational database calls	Resource management
Locking (e.g., data channels)	Authentication
Data processing & visualization	Notification (e.g., email, sms)
Event sequencer / synchronizer	

Depending on the circumstances, any particular service or instance of a service might be instantiated in any of the controls computing resources, including front-end computers, distributed computing nodes, controls computing center, and control room workstations.

Control System Architecture Research and Development

Research into accelerator controls architecture and the role of services is underway. The ILC Reference Design Report describes an architecture in which the applications are split into a services tier and client GUI tier. How best to meet the needs of operations while increasing manageability, coordination (conflict avoidance), automation, and optimization is an active subject, even outside the ILC. Applications such as CSS (Control System Studio) offer a new platform in which to create GUIs. Services deployed outside the CSS session can be invoked using web services, CORBA, ICE, or other distributed object technology. Work is underway to evaluate the suitability of web services and ICE. It is also necessary to define a framework for the stateful orchestration of services and how that would be implemented. There is similar research currently underway within the neutron and synchrotron experimental communities to orchestrate the many software components involved in conducting beamline experiments. This work is similar, although perhaps with different requirements in mind. In particular, the ILC must implement a general-purpose, distributed-feedback service infrastructure, perhaps including an embedded numerical engine, such as Matlab or Octave, within a service.

AVAILABILITY

Several factors lead us to pay particularly close attention to the control system availability requirements:

- In order to meet 99% overall availability, each one of the 1000+ front-end crates must provide at least 'five nines' (99.999%) availability.
- Travel times to the equipment location increase recovery times when human intervention is required.
- Integration problems such as resource conflicts with the large number of hardware and software components must be avoided.
- Extensive reliance on automation and feedback increases the number of controls channels that would be active at any one time, making a channel failure more likely to impact accelerator operations.
- Downtime attributed to the control system would include not only time to mitigate faults in the control system itself, but also any resultant time to recover accelerator operations.
- Manual implementation of widespread fixes or revisions would be time-consuming and error prone.

As depicted in Figure 4, accelerator availability can be divided into three components: downtime event rate; time taken to recover machine operations after a downtime event; and the time to reestablish machine operations after a shutdown or accelerator studies.

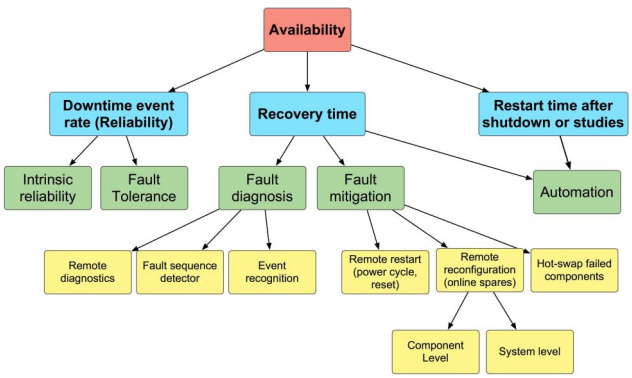

Figure 4: Components of accelerator availability.

Accelerator technical equipment has commonly been designed and implemented using 'good engineering practice' with the expectation that reliability will be improved reactively and incrementally as any systematic problems are discovered once the machine is in operation. We plan to take a more proactive approach.

The top-down work of determining control system failure modes, detailed availability requirements, cost-benefit analyses, and priorities has not progressed far enough to present conclusions here. Instead, we begin by describing what we believe is an ideal picture in terms of abstract features and capabilities. The goal is to communicate a vision that fits the myriad of techniques.

Correlating control system faults and machine downtime is complex and implementation specific. Not all control system failures cause accelerator downtime, although they might result in reduced performance (e.g., if a particular control point can no longer be adjusted), or loss of functionality (e.g., loss of archived data). It can therefore be expected that not all control system elements would require the same attention from a high availability perspective. As shown in Figure 5, one goal of our R&D program is to understand the cost penalty and relative benefit of implementing various high availability tools and techniques. The goal is to make sound design decisions based on information from the high availability R&D program coupled with information from failure modes and effects analyses [7].

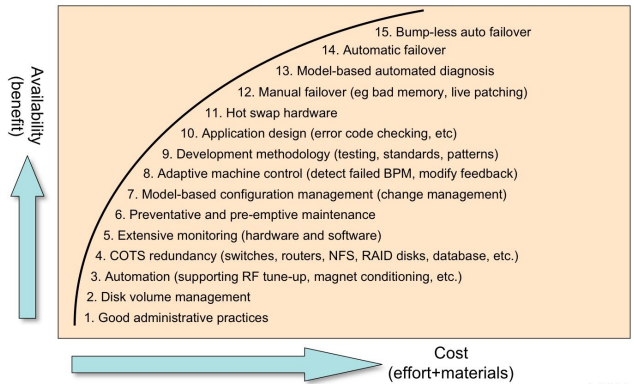

Figure 5: Relative cost benefit of techniques.

Configuration Management

Configuration management is the act of maintaining the state of a system over time. For a control system, this encompasses controls front-end code, driver deployment, network switch port configuration, FPGA programming, field-replaceable hardware modules, server software deployment, and many other categories of configuration. Configuration is traditionally handled in a somewhat ad-hoc manner, and we propose to integrate this as much as possible using a model-based approach. Successful configurations of the control system hardware and software should be repeatable and rollback-capable to the extent that such an enterprise is possible. An important aspect of configuration management is the ability to introspect hardware and software in order to verify that the desired configuration has been reached.

Out-of-Band Monitoring and Diagnostics

Figure 2 represents some of the traditional physical components of a control system. What is perhaps less familiar is the extensive use of what is referred to as out-of-band monitoring and management in the telecommunications industry. This is typically a completely separate network dedicated to the monitoring and management of resources in the system, both hardware and software.

Network hardware typically has SNMP (Simple Network Management Protocol) and CLI (Command Line Interface) management interfaces. Commodity server hardware is now routinely shipped with SNMP and IPMI (Intelligent Platform Management Interface) interfaces, as well as the ability to remotely power-cycle the hardware using a LOM (Lights Out Management) interface.

The management interfaces for switches, routers, and servers should also be available for controls IOCs and Device Servers, and even technical equipment such as

power supplies and modulators. With these management interfaces, one can:

- Identify hardware modules in the chassis;
- Monitor module voltages, temperatures, and other health and status measures;
- Control power, reset, and self-test of individual modules, control hot-swap LEDs and prepare a module for extraction and replacement; and
- Upload and install new code and firmware to boot-up up memory and FPGAs.

Additionally, it should be possible to get to all console ports. Traditionally done with a terminal server network, it is now possible to carry console traffic shared over an existing Ethernet interface on the equipment using the IPMI standard known as SOL (Serial Over LAN).

Software components themselves must also implement a management interface. This interface should permit remote execution and termination of the process, the ability to checkpoint the state of the process where reasonable, and the ability to gracefully quiesce activity. Additionally, it should be possible to request a component self health-check. This can be as simple as a heartbeat response, or more complex such as involving a check of the application's internal state. Related to this, the heap and/or buffer memory allocation used by the software component can be monitored at run time to identify memory problems related to leaks or excessive load.

Hot-Swap, Redundancy, and Failover

In order to implement hot-swap, redundancy, and failover in hardware and software, the previously described capabilities of configuration management and out-of-band monitoring and diagnostics must be in place. For example, in order to hot-swap hardware we must command any dependent software to stop or failover, and then activate the hot-swap LEDs in preparation for sending out a technician. Much of the hard work in this area involves the application-specific modifications necessary to gracefully capture and control the operational state of software components, also known as checkpointing and lifecycle management.

Availability Research and Development

Achieving high availability in electronic and computing systems requires bringing together a wide variety of techniques, each of which addresses a small piece of the overall picture. Since there is neither a broad experience base nor a comprehensive framework of standards and techniques for implementing high-availability accelerator control systems, we must look to other industries. The telecommunications industry has recently introduced two open standards that represent decades of best practices in that industry. The Service Availability Forum (SAF) and Advanced Telecommunications Computing Architecture (ATCA) specifications focus on software and hardware respectively, and offer perhaps the most comprehensive starting point for high availability designs [8,9].

High-availability R&D for ILC controls must develop a top-down set of requirements and priorities as well as a bottom-up set of techniques. Implementations of the SAF and ATCA specifications allow us to begin prototyping using a well-supported set of standards. We must also assess control system failure modes (many of which differ from telecom failure modes), understand how to mitigate them, and assess the cost-benefit of implementing these standards. It is yet to be determined the degree to which SAF and ATCA meet our priorities compared with other standards such as VITA VXS and VPX.

CONCLUDING REMARKS

Worldwide participation through an open process is key to the ILC Global Design Effort as a whole. On controls, an international effort of coordinated research and development is underway, with active work on the topics described in this paper and on other topics such as the timing system. Collaborations are also being established with controls groups at new and existing accelerator facilities. We invite research groups and accelerator groups to participate in these and other controls activities.

ACKNOWLEDGMENTS

Work in the U.S. is supported by the U.S. Department of Energy under Contract Nos. DE-AC02-06CH11357, DE-AC02-76CH03000, and DE-AC02-76SF00515.

REFERENCES

[1] http://www.linearcollider.org/cms/
[2] International Linear Collider Reference Design Report, ILC-REPORT-2007-001, 2007.
[3] F. Lenkszus et al., Timing & RF Phase Reference for ILC Reference Design, ILC-NOTE-2007-025, 2007.
[4] C. Saunders et al., Control System Architecture Model for ILC Reference Design, ILC-NOTE-2007-022, 2007.
[5] M. Votava et al., Commodity Computing Architecture for ILC Reference Design, ILC-NOTE-2007-026, 2007.
[6] A. Gotz, D. Schmidt, M. Clausen, "Middleware in Accelerator and Telescope Control Systems," Proc. of ICALEPCS 2003, Gyeongju, Korea, Oct. 13-17, 2003, p. 322 (2003); http://www.jacow.org.
[7] E. Marcus, H. Stern, *Blueprints for High Availability*, Second Edition, (Wiley Publishing Inc.: 2003).
[8] http://www.saforum.org
[9] R.W. Downing, R.S. Larsen, High Availability Instrumentation Packaging Standards for the ILC and Detectors, SLAC-PUB-12208, 2006.