CHAPTER 14
CONTROL SYSTEM

14.1 INTRODUCTION

The LHC control system will be an extension of the infrastructure in use today for the Injector Chain and the Technical Services. Controls projects are underway for additions and upgrades to the existing CERN accelerators – the LEIR reshuffling as an ion accumulator for LHC – the CNGS and the LHC transfer lines at the SPS. Evolving control technologies and diminishing costs are already leveraged in the infrastructure.

Nevertheless the LHC will present unique challenges for the control system. The hardware will be of unprecedented complexity and will require a precise and unambiguous description. Failures will lead to long cryogenic recovery times and long magnetic conditioning cycles. Operational efficiency will depend on both the technical services and the accelerator hardware control systems. The high stored energies and beam powers will impose rigorous operational procedures and sophisticated diagnostics, while the challenging design parameters and the large dynamic effects will only be mastered by a flexible control strategy, adapted to changing circumstances as the knowledge of the LHC improves.

In order to achieve good beam performance and safe operation the beam control system must provide for real-time automation which is a novel requirement at CERN.

This chapter presents an overview of the control system architecture, a review of the different layers starting at the hardware level and concludes with an outline of some important high level services. Particular emphasis is given to software frameworks and the impact of modern industrial control technology.

It is anticipated that the LHC will be commissioned and operated from a new CERN Control Room (CCR). This will bring together control and operation of the site services, cryogenics, injector chain and the new machine. The impact of the reorganisation of the existing control rooms is not discussed in this report.

14.2 ARCHITECTURE

The LHC control system architecture is largely based on standard components employed worldwide for the control of accelerators and used during recent years of efficient operations of the CERN injectors (PS, SPS) and of the LEP machine. However, two major changes will take place for the LHC era:

- The consistent use of object oriented technologies and design principles for the control of beam-related systems,
- The wide use of industrial controls solutions for the supervision of complete subsystems of the LHC.

14.2.1 Overall Architecture

As shown in Fig 14.1, the LHC control system has three hierarchical layers of equipment communicating through the CERN Technical Network, a flat Gigabit Ethernet network using the TCP-IP protocol (Sec. 14.2.2).

Starting from the bottom of Fig 14.1:

- At the equipment level, the various actuators, sensors and measurement devices are interfaced to the control system through three different types of front-end computers:
  - VME computers dealing with high performance acquisitions and real-time processing; these employ a large variety of I/O modules. Typically, the LHC beam instrumentation and the LHC beam interlock systems use VME front-ends.
  - PC based gateways interfacing systems where a large quantity of identical equipment is controlled through fieldbuses, such as the LHC power converters and the LHC Quench Protection System.
  - Programmable Logic Controllers (PLCs) driving various sorts of industrial actuators and sensors for systems such as the LHC Cryogenics systems or the LHC vacuum system.
- At the heart of the control system, powerful UNIX servers host the operational files and run the LHC applications:
  - Application servers hosting the software required to operate the LHC beams and running the Supervisory Control and Data Acquisition (SCADA) systems.
- Data servers containing the LHC layout and the controls configuration as well as all the machine settings needed to operate the machine or to diagnose machine behaviours.
- Central timing which provides the cycling information of the whole complex of machines involved in the production of the LHC beam and the timestamp reference.

Processes running in these servers will communicate with the equipment access machines using various mechanisms and protocols such as the Common Object Request Broker Architecture (CORBA) and TCP-Modbus.

- At the control room level, consoles running the Graphical User Interfaces (GUI) will allow machine operators to control and optimise the LHC beams and to supervise the state of key industrial systems. Dedicated fixed displays will also provide real-time summaries of key machine parameters.

---

![Control System Architecture](image)

**Figure 14.1: Control System Architecture**

### 14.2.2 Network

The control system for LHC relies on the CERN Technical Network. This network uses IP addresses of the form `172.18.xxx.xxx`. This address prefix prevents any direct communication between the technical network and the Internet. The infrastructure is interconnected with the General Purpose Network so that communications from within the Organization are still possible; however, the technical network can be completely isolated if required.

The technical network is a highly sub netted, routed network that implements a high performance redundant backbone, reduces parasitic traffic and keeps the overall structure maintainable. As shown in Fig 14.2, the basic distribution is based on a redundant Gigabit Ethernet backbone using fibre optical distribution. The core consists of two redundant backbone routers located in the CERN computer centre and in the Prévessin control room, where most of the fibres terminate. These two central routers are interconnected in a redundant way to a series of regional routers located in the following buildings:

- The computer centre (building 513) for central services like Oracle databases and main file servers,
- The Meyrin control room (building 354) for the control of accelerators located on Meyrin site,
- The technical control room (building 212) for the supervision of the 2 CERN sites,
- The Prévessin control room (building 874) for the SPS control and other technical services located on the Prévessin site,
- Each LHC pit (SR building) for the LHC machine and technical services control.
In each technical building we find one or more “star points” in order to respect the maximum cable length of 100 m for 100-BaseT connections between the end-node and a high performance Fast-Ethernet switch. This usually has a 100Mbps uplink to the closest regional router that may be upgraded to Gigabit if needed. Close to regional routers, 1Gbit/s Ethernet may easily be made available if needed.

Active equipment in the network architecture is installed with a double power supply connected to two independent power sources (normal and secured) in order to ensure proper connectivity even in case of a power-cut on either 220V distribution.

Because the technical network can be completely isolated from the external world, independent dedicated redundant name and time servers have been implemented. These servers are installed at different locations and connected to different regional routers to ensure full redundancy of the system.

It must be noted that because the technical network infrastructure is interconnected with the general purpose network, security break-ins can be attempted on the devices connected to this infrastructure. End nodes security survey and updates must not be forgotten.

14.3 EQUIPMENT ACCESS

14.3.1 The VME and PC Front End Computers

Most of the non-industrial accelerator hardware in the PS, SPS and LHC sites is already connected to the control system via VME or PC based Front End Computers (FEC). These systems run a real-time operating system called LynxOS from LynuxWorks [1] or Red Hat Linux [2].

There will be several hundred FECs distributed in the surface buildings of the LHC and in some cases in the underground areas. Wherever possible they will be diskless to increase reliability and will boot over the network. They will be connected to the remote reboot and terminal server systems so that they can be managed remotely by the operators and system administrators.

A set of commercial or CERN made hardware interface boards is being standardised for the LHC, together with the necessary device drivers (e.g. timing generators, timing receivers, beam interlock controllers, digital acquisition boards, WFIP cards, digitisers, etc.). The type of hardware (VME or PC), as well as the Operating System (Linux or LynxOS), will be selected according to the performance needed and to the availability of the specific drivers.

These FECs will execute the equipment access software which is part of the resource tier (Fig. 14.1). This software accesses data of the hardware interface boards and provides it either via subscription or command/response to any software agent in the control system.

A dedicated FEC software framework (Sec. 14.7.1) has been developed in order to simplify and standardise the software running in the FECs.
14.3.2 The PLCs

PLCs are increasingly used for controlling industrial equipment for LHC. This type of controller can be chosen when the process is not synchronised to accelerator timing and when the sampling period required is not smaller than 100 msec.

PLCs offer ease of programming process logic via standard high-level languages (IEC-61131-3), cost efficiency, high reliability and adaptation to industrial environment. They are used in typical industrial controls fields such as electricity, water, gas, cooling, ventilation, but also in accelerator-specific fields of machine cryogenics, interlocks, vacuum, radio-frequency and beam extraction.

PLCs are generally part of a complete subsystem including supervision (Sec. 14.7.4). In a few cases they are accessed through VME FECs. As with any other FEC in the Controls Infrastructure, PLCs benefit from generic facilities such as remote reset, monitoring and optionally clock synchronisation.

Only PLCs from Schneider and Siemens are used for LHC as recommended by the CERN Controls Board.

14.3.3 The Supported Fieldbuses

The two fieldbuses used in the LHC accelerator, namely WorldFIP and Profibus, are part of the three fieldbuses recommended at CERN. They are both supported by an internal CERN service. These fieldbuses typically allow longer distance and more robust protocols than Ethernet. In addition they are the only means to connect equipment located in the LHC tunnel or other radioactive areas.

*The WorldFIP Fieldbus*

Tab. 14.1 summaries the LHC WorldFIP installation needs. WorldFIP is selected when the following features are required:

- **Determinism (up to 10 µs):** is required for:
  - The real-time control and synchronisation of LHC equipment,
  - High precision time distribution,
  - The management of periodic data.

- **Robustness in severe environments and in particular:**
  - Its resistance to electromagnetic noise (level 3),
  - Its good resistance to high radiation levels (based on robust semiconductors and a magnetic coupling ensuring galvanic isolation).

- **Data Rates:**
  - WorldFIP will be used to control largely distributed LHC systems at high data rates (1MBits/s and 2.5 MBits/s),
  - High load factor possible (70 to 80% of the network bandwidth)

<table>
<thead>
<tr>
<th>LHC System</th>
<th>Required Time precision (ms)</th>
<th>Cable length (km)</th>
<th>Data rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magnet Protection</td>
<td>1</td>
<td>61</td>
<td>1 MBit/s</td>
</tr>
<tr>
<td>Power Converters</td>
<td>0.01</td>
<td>45</td>
<td>2.5 MBit/s</td>
</tr>
<tr>
<td>Beam Instrumentation</td>
<td>500</td>
<td>50</td>
<td>31.25 KBit/s</td>
</tr>
<tr>
<td>Radio Frequency</td>
<td>1</td>
<td>5</td>
<td>1 MBit/s</td>
</tr>
<tr>
<td>Cryogenics</td>
<td>500</td>
<td>90</td>
<td>1 MBit/s</td>
</tr>
<tr>
<td>Survey</td>
<td>1000</td>
<td>44</td>
<td>31.25 KBit/s</td>
</tr>
</tbody>
</table>

*The Profibus Fieldbus*

Profibus has been selected for several applications in the LHC, such as fast kicker magnets, cooling and ventilation, vacuum, cryogenics, magnet and power interlock controllers.
The main reasons to select Profibus for these applications are:

- The robustness of the protocol and simplicity of configuration.
- The large variety of remote I/O systems available with Profibus.
- The ease of integration with Siemens PLCs.
- The availability of radiation-tolerant remote I/O on Profibus.
- Its capacity to be used as an instrumentation bus (Profibus-PA) with a wide range of instrumentation products offering Profibus as standard means of connection.

14.4 SERVERS AND OPERATOR CONSOLES

The upper layers of the LHC control system (Fig. 14.1) will be deployed on operation consoles and fixed displays, files and applications servers to meet the requirements of the LHC applications software.

The servers will be used to run the LHC middle-tier software, to host operational programs and data files, and also to offer specific services (web services for operation, fixed displays, SCADA servers, Database servers, etc.). The servers will run the Linux operating system. Emphasis will be put on the hardware reliability and availability issues by selecting multi-CPU architectures with redundant and hot swappable power supplies, discs and fans. Mirroring and RAID techniques will be used to ensure data integrity and error recovery.

The UCR fixed displays and operator consoles will both be based on desktop PCs with a large amount of memory. They will run GUI applications from their local disks and use one to three screens to display the data. At present the choice of the Operating System between Linux and Windows has not yet been made.

More than 20 fixed displays and around 10 consoles will be needed in the CCR control room to operate the LHC machine. Around 50 servers located nearby and in the LHC surface buildings will be used to deploy and run the operational LHC software.

14.5 MACHINE TIMING AND UTC

14.5.1 Central Beam and Cycle Management

The LHC beam production involves a long chain of injectors which need to be tightly synchronised. Moreover, the different types of beam to be produced for LHC are only a subset of the beams that the injectors have to produce, e.g. beams for fixed target physics, for Isolde, for AD. All these beams are produced in sequences of typically a few tens of seconds. The composition of these sequences changes many times a day and the transition between different sequences has to be seamless.

To manage the settings of the machines and to synchronise precisely the beam transfer from one injector to the other up to the LHC, a Central Beam and Cycle Manager (CBCM) is required. The CBCM will:

- Deliver General Machine Timing (GMT),
- Provide Beam Synchronous Timings (BST),
- Elaborate the “telegrams” specific to each machine and broadcast over the GMT networks.

A “telegram” describes the type of beam that has to be produced and provides detailed information for sequencing real-time tasks running in the FECs and for setting up equipment.

The CBCM drives seven separate GMT networks dedicated to the different CERN accelerators, including the LHC and four BST networks of which three are for the LHC and one for the SPS. To achieve extreme reliability, a CBCM is running in parallel on two different machines, with a selector module capable of seamlessly switching between them.

14.5.2 Timing Generation, Transmission and Reception

Several hardware timing modules are being built either to produce reference clocks and timing events or to extract the timing pulses or the interrupts needed to drive the equipment from the distributed timing data. The deployment of these modules will start on the PS and on the SPS in 2004:

- The CTSYNC module provides the “master” clocks from a 10MHz oscillator which has a stability which is better than $10^{-10}$. 

• The CTGU module encodes events to be transmitted over the GMT network: the millisecond events, machine timing events, telegrams, and the Universal Coordinated Time (UTC) events.
• The CTGB is the driver module for the BST network; it uses the TTC technology [3]. The CTGB will also send the 1 Pulse per Second (1PPS) UTC time and some telegram events over these networks, hence its inclusion into the CBCM.
• The CTRP is a GMT multi-channel reception module. It comes in three formats: PMC, PCI, and VME and can generate timing pulses with very low jitter. The CTRP recognises the various types of timing events: UTC time, the telegrams, the millisecond and other machine events, and can use them to trigger pre-programmed actions.

14.5.3 UTC for LHC Time Stamping

A key requirement for the LHC control system is the need for a timing reference to timestamp the accelerator data. UTC has been adopted as the standard of date and time for all CERN accelerators [4, 5]. UTC is the basis for the worldwide system of civil time. The motivation behind this decision was to eliminate the problems associated with the changes between the winter and summer standards, particularly for the change in October (CET changes from UTC+2H to UTC+1H) when the time is repeated for one hour.

14.5.4 UTC Generation, Transmission and Reception

The source of date and time for all CERN accelerators will be a Global Positioning System (GPS) time receiver connected to the CTGU. The GPS provides UTC referenced date and time plus a very accurate 1 PPS tick, which is used by the CTGUs to synchronise the UTC time.

At initialisation the CTGUs receive the date and time information from the GPS receiver then run independently, using the clock delivered by the CTSYNCH until a manual resynchronisation is made. CTGUs provide the time every second in UNIX standard format. UNIX time can then easily be converted by software into year, month, day, hour, minute and second.

UTC time can be provided by the CTRP modules to timestamp the data to be logged with a 25 ns resolution. Using an optional CERN-developed chip the resolution can be brought down to 1 ns. The required time-stamping granularity for the different LHC systems [6] is shown in Tab. 14.2. The 1 ns performance is required principally in the injector chain.

<table>
<thead>
<tr>
<th>System</th>
<th>Timestamp Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam Dump</td>
<td>&lt; 0.05 ms</td>
</tr>
<tr>
<td>Beam Instrumentation</td>
<td></td>
</tr>
<tr>
<td>Radio Frequency</td>
<td></td>
</tr>
<tr>
<td>Injection (Kickers...)</td>
<td></td>
</tr>
<tr>
<td>General Machine Timing</td>
<td></td>
</tr>
<tr>
<td>Machine Interlocks</td>
<td>1 ms or better</td>
</tr>
<tr>
<td>Quench Protection</td>
<td></td>
</tr>
<tr>
<td>Power Converter</td>
<td></td>
</tr>
<tr>
<td>Feedbacks (orbit, tune ...)</td>
<td>1 – 10 ms</td>
</tr>
<tr>
<td>Cryogenics</td>
<td></td>
</tr>
<tr>
<td>Vacuum</td>
<td></td>
</tr>
<tr>
<td>All other systems</td>
<td></td>
</tr>
</tbody>
</table>

14.5.5 NTP Time Protocol

PLCs will use the Network Time Protocol (NTP) [7] to synchronise the time (Sec. 14.2.2). NTP provides accuracies typically within a few milliseconds on LANs and up to a few tens of milliseconds on WANs. This accuracy is sufficient to properly tag most of the PLC data. Some PLCs that require higher precision for time-stamping their data will use the high precision 1PPS tick from the CTRP; in this case NTP will be used only to relate this tick to UTC time.
14.6 DATA MANAGEMENT

The LHC Machine will be of unparalleled complexity in terms of number of components and in the manipulation of operational parameters. A large number of signals will be made available to the control room, including over 100,000 from the cryogenics, machine protection and beam monitoring systems. Over 1600 electrical circuits will power a wide variety of magnet systems. The quantity of data to be recorded for diagnostics after a “beam abort” has been estimated as a few gigabytes; during Hardware Commissioning it is anticipated that 1 terabyte of information will be accumulated. Operation of the machine will rely on stringent control of the hardware settings. A complete beam operational history will be created in order to build a long term understanding on the accelerator performance.

Such a variety and volume of data cannot be exploited without data management tools. Building on the experience of LEP construction and operation [8] Oracle has been selected as the strategic choice for data management at the LHC.

14.6.1 Offline and Online Data Repositories

As shown in Fig 14.3, the LHC Reference Database is being populated during the machine construction in order to manage the information essential for integration studies, installation and hardware commissioning [9]. This offline Database will be one of the sources of data that is required to configure the online LHC control room application software providing essential information such as the layout, accelerator optics and calibration information. The electrical circuit data, described in the next section, are an example of the use of the database both during construction and operation.

![Figure 14.3: LHC Reference Database](image)

While the Reference Database will be an offline source of information during machine operation, several online databases - alarm archives, periodic data logging, Post Mortem event archive and accelerators settings management - will be required to capture the operational history. These are described in the later sections of this chapter.

14.6.2 Electrical Circuits

The powering layout of the LHC is extremely complex, comprising 1232 main dipole magnets, about 450 quadrupole magnets and several thousand corrector magnets, powered in 1612 electrical circuits. The current feeding the magnets ranges from 60 A (for small correctors) to 12 kA for main dipole and quadrupole
magnets, while the energy stored in the superconducting magnets is about 10 GJ. About 80,000 high current connections have to be made and tested in the tunnel during the installation and commissioning phases.

To ensure the connection correctness of the superconducting bus bars around the 27-km long machine, a detailed description of the electrical circuits and their elements has been added to the layout description of the machine in the LHC Reference Database [10]. This description has been linked to the existing data describing the physical and magnetic layouts. Changes of this powering information are very rare (static data) and updates have to be performed only if there are major changes in the physical or powering layout of the machine.

Information about the powering will be provided to users to:
- Perform the interconnections between cryo-assemblies in the long arc cryostats,
- Configure the powering interlock PLCs (section 15.3.7),
- Create MAD input files to calculate the beam optics,
- Display detailed circuit and equipment information via the web and the LHC equipment catalogue,
- Configure hardware state and beam parameter control room application software.

Integrity will be ensured by using the same information to generate the machine optics, make the circuit connections during assembly, configure the behaviour of the interlock system and operate hardware and beams.

14.6.3 Control System Configuration

The control system is composed of a large number of application programs, a middleware layer, VME- and PC-based FECs with control software, industrial PLCs, fieldbuses, and hardware interface modules. These components have shared characteristics, associated with their type or class, and specific addresses and parameters associated with their instances. All these data will be stored and described in an Oracle relational database, the Controls Configuration Database.

This centrally managed configuration information permits the use of generic software on all levels. A layer of Java and C++ interfaces will enable all control room software to be data-driven, by fetching at runtime the layout, interconnection, access methods, and parameters of remotely controlled LHC equipment. All FECs can be automatically configured and bootstrapped with files generated from the database.

Thanks to a generic design, the database also holds the controls configuration of the injection accelerator chain of the PS and SPS complexes. This common source of data will make it possible to have uniform controls data management throughout the accelerator chain, as well as effective sharing of tools and methods that use this data [11].

On top of the Configuration Database, a Web browser service will allow exploration of all the information about the underlying controls hardware and software.

14.7 COMMUNICATION & SOFTWARE FRAMEWORKS

14.7.1 FEC Software Framework

The software running in the LHC FECs will be developed using a specific framework, known as FESA. This framework is a complete environment for the equipment specialists to design, develop, test and deploy real-time control software for the FECs. This framework is also the new standard for the LHC injector chain.

The recurrent components of the equipment control software have been formalised into a generic model. It defines the structure of the real-time part of the software which handles hardware and interrupts together with the structure of the server task that handles external requests. In order to enhance the productivity and the maintainability of the code, a clear separation between the generic and the equipment specific parts is enforced and automatic code generation is extensively used. As a result, the ratio between the generic/generated code and the developer specific code is usually around 10 to 1. An object-oriented design and a C++ implementation have been used for supporting this approach.

For building the control software, the programmer is provided with a set of configuration tools for the creation and evolution of any FESA component. These tools are directly linked to the control system configuration DB (Sec. 14.6.3). They are used for the definition of internal data structures, syntax of the external command and binding between hardware events or external requests and treatment code.
External requests are handled by dedicated server tasks that fully implement the device access model of the Controls Middleware, which is described in the next section. Compound data types are supported for data coherency and for performance purposes.

System-level services are provided for the operation of the FECs: handling of errors from the equipment software, monitoring of the processes, etc. These dedicated services are made for preserving the real-time behaviour of the FECs and they support remote control for adjusting parameters, such as the trace level of a task, or executing corrective action like restarting a process.

The developers and the users (equipment-specialists, operators, and maintenance team) are also provided with a set of interactive tools, test programs and Java libraries which allow immediate testing of all the functionaility of the software during its development and operation. These facilities are all generated in an automatic manner, without any specific code required.

The FESA framework is built for LynxOS and Linux platforms. The software tools are written in Java and are thus platform independent. No commercial software requiring a run-time license has been used.

14.7.2 Controls Middleware

The Controls Middleware (CMW) [12] is the ensemble of protocols, Application Programming Interfaces (API) and software frameworks, which allow seamless communication between the software entities of the control system. Two conceptual models are supported: the device access model and the messaging model. The device access model is mainly used in the communication between the resource and middle tiers while the messaging model is mainly used within the business tier or between the business tier and applications running in the presentation tier.

Device Access Model

Fig. 14.4 depicts the full architecture of the CMW device access model. The typical use of this communication model is between Java applications running in the middle tier and CMW equipment servers running in FECs. These processes communicate together through the Remote Device Access (RDA) API [13] which is available for both Java and C++ languages.

Within the device access model all accelerator equipment can be accessed as “devices”. A device is a named entity within the control system, which normally corresponds to a physical device (e.g. beam position monitor, power converter) or to a virtual controls entity (e.g. transfer line). Conceptually each device has a number of properties that characterise its state. By getting the value of a property, the state of the device can be read. Accordingly, a device can be controlled by setting one of its properties with the required value. Get
and Set operations can be either synchronous or asynchronous. In addition to the Get and Set operation on device properties, it is possible to monitor changes to the property via listeners (callbacks).

It is often necessary to specify under which conditions the operation has to take place, for instance to which cycle of the machine the operation applies. These conditions are commonly referred to as context or filter.

A list of properties, together with their type description and the semantic of the assessors, is referred to as a contract. Contracts typically serve a specific purpose such as measurements or settings of a device.

**Messaging Model**

Complementary to the device access model, the messaging model allows any software process to send and receive asynchronous messages in a loosely coupled way. Various application programs for accelerator controls naturally exchange data via asynchronous messages: timing and logging events, notifications and alarms are typical examples of data which is asynchronously generated from a various set of producers, which is potentially of interest for multiple consumers.

The LHC control system software relies on the Java Message Service (JMS) [14] as the messaging solution for Java based controls application. The JMS specification adds a standard and vendor-independent API that enables the development of portable, message-based applications in the Java programming language. Moreover, JMS is a strategic technology for the Java 2 Platform Enterprise Edition (J2EE).

The LHC Alarm Service [15] is a typical example of a messaging system relying on the publish-and-subscribe paradigm: it subscribes to alarm notifications being published by several processes and, after processing, redistributes the result to dedicated consoles and any other controls software component that subscribed to the appropriate alarm content hierarchy.

14.7.3 The J2EE Framework for Machine Control

Applications controlling the LHC beams must handle a great variety of tasks such as visualisation of data and significant computation, together with database and equipment access. These applications rely on several services such as security, transactions (to make sure that complex operations are performed atomically), and remote access or resource management. Those requirements dictate a move towards a modular and distributed architecture. Such architecture offers a clear separation between the GUI (the user interface), the control (the core business of the application), the model (the abstract model of the accelerator control) and the devices (the physics equipment) that are controlled.

For LHC, a 3-tier architecture is being implemented where the middle-tier is responsible for providing all services and for coordinating the client applications running on the operator consoles. In this architecture shown in Fig. 14.4, applications are running both in the operational consoles of the presentation tier and in dedicated servers of the middle-tier. The consoles are responsible for the GUI applications and translate operator’s actions into commands invocation in the middle-tier. The middle-tier, through its centralised and shared processing power is in charge of accessing databases and accelerator devices. The middle-tier also ensures the coherency of operator actions and shelter resources. It enforces separation between presentation and application logic and provides shared services to many clients.

To achieve this new programming model, a platform is needed, to support the middle-tier. The platform provides the infrastructure with all the necessary common basic services, i.e. all parts of a control system that are not specific to accelerator controls. By using such a platform, the developers can concentrate on writing code for the accelerator controls components such as parameter management, setting generation, cycle handling, trim management; they do not have to write system level services. The platform of choice today is the J2EE [16]. J2EE is an industrial standard defined by a set of specifications and APIs, implemented by many vendors. J2EE is based on the Java programming language and on a component model known as Enterprise JavaBeans (EJB). Components are the key technology to write a modular object-oriented distributed application. EJB components are "Lego pieces" that implement a set of interfaces allowing them to run inside a so-called "EJB container".

As shown in Fig 14.5, developers write accelerator-specific code in the form of EJB components, and the EJB container provides the necessary infrastructure to run them, such as remote accessibility, components management, concurrency support, fault tolerance, high availability, scalability, resources pooling, transaction management and security.
To summarise, the 3-tier approach and the J2EE framework are the basis on which to build the control for the LHC. Efforts may be focused on the LHC control challenges and not on the infrastructure. It exploits existing technology and provides the modular and distributed architecture needed.

14.7.4 The UNICOS Framework for Industrial Controls

In the past decade many CERN control systems, particularly those for technical services, have been built with standard industrial control components, often by industry following a technical or functional specification. Reflecting this trend, a single industrial supply has been made for the procurement of the hardware and software for the control of the cryogenic equipment and experimental magnets installed in the LHC accelerator complex. This control system is referred to using the acronym UNICOS, standing for UNified Industrial COntrol System.

UNICOS builds on a classic industrial controls architecture using the PVSS [17] SCADA in the Supervision layer, Schneider Quantum PLCs for process control at the Control layer and Schneider Premium PLCs to connect process channels in the Field layer (an alternative is Quantum remote I/O). Communication is based on Ethernet and FIPIO. The software design is an evolution of an object oriented philosophy used with former control systems [18, 19]. In this approach each process device - sensor, actuator, or set of devices constituting a process entity - is modelled in an object. This object integrates the process behaviour and the GUI.

The UNICOS Object Model

In the UNICOS concept the object implementation is split into two parts:

- The GUI functionality - programmed in the SCADA,
- The process behaviour - programmed in the PLC.

The GUI part includes the interaction with the operator by means of graphical components and dedicated panels called “faceplates”; these GUIs inform the operator of the object status, and allow him to send orders. The PLC part contains the process behaviour of the object. The object is linked to the plant through I/O channels that may be linked to the PLC via either a fieldbus or a direct I/O connection. The SCADA and PLC object parts are connected together by the communication layer based on the TCP-Modbus protocol.

As shown in Fig. 14.6, each object has several interfaces and receives the following information:
- Requests from the operator via the SCADA object. These requests are transmitted to the PLC by the manual requests through the communication middleware.
- Configuration parameters (GUI or PLC) set during the programming phase and accessible for modification by a program specialist.
- Information from the process (process inputs), consisting of analogue or binary values from sensors and status of other objects.
- Orders from the control logic programmed into an object of a higher hierarchical level via the Auto Requests.

Three main types of object are defined in the UNICOS architecture:
- Input-output objects which provide the interface to the plant. They link the devices and actuators to the control system. Some basic treatments may be embedded in these objects. Input/output channels are exclusively accessed through such objects.
- Field objects that represent hardware elements such as valves, heaters, motors, and other devices. Each type of field object has its own associated process logic. This logic integrates specific functions (alarm generation, transition between set points, interlocking).
- Process control objects that control equipment units grouping several field objects and/or other process control objects. The process logic of these objects is split between a standard part insuring a homogeneous interface to the environment, and a specific part to handle the process to control.

Implementation

The industrial supply includes the development of the PLC and SCADA components for each type of object as well as the communication mechanisms between the PLCs and the SCADA. It also includes a SCADA framework offering homogeneous user interfaces and accelerating application development for the supervision layer. The PLC objects were developed in IEC-61131-3 languages and all SCADA modules have been developed using ActiveX and proprietary PVSS scripts. In order to optimise the communication bandwidth between the PLCs and the SCADA, an event-driven protocol based on TCP-Modbus has been developed by the supplier.

The user application development process is considered as an integral part of the software framework. This is split into 3 distinct phases: specification, software development and acceptance/maintenance.
14.7 CONTROL ROOM SOFTWARE

14.7.1 Software for LHC Beam Operation

The LHC control room applications will be based on a set of common control system services and components, in order to avoid duplication of effort and solutions. The operational software development process relies on common development tools, guidelines and procedures for the construction, testing, integration, deployment and change management.

UCR Software Requirements

The LHC aims at injecting, accelerating and then colliding beams with well controlled beam parameters in an efficient, reliable and reproducible manner. This is a non-trivial task, since the small aperture, the high stored beam energy and the sensitivity of the machine to beam losses impose very tight accelerator physics constraints. The superconducting magnets will generate field errors that have large static and dynamic components. Thus, the destructive power of the LHC beams and the low tolerances to beam losses will place very stringent demands on the LHC control system. A non-exhaustive list of the main software applications for LHC is outlined below:

- **On-line monitoring** of the state and tolerances of every LHC component and interlock systems.
- **Measurements** to allow for synchronised measurement acquisition as part of scans, feedback, or at a given point in the ramp.
- **Real time control** to implement feedback on key beam parameters (oribt, tune, chromaticity) plus the feed-forward of corrections from the reference magnets.
- **On-line model** including calculation of parameter variations and feed-forward from machine into model.
- **Automatic sequencer** to allow complicated sequences for injection and ramp to be performed systematically.
- **Fixed displays** including transfer lines, beam size, bunch currents, mountain range turn-by-turn, tune FFT, beam loss monitors, global orbit plus crossing angles, beam separation, transverse position along a batch.
- **Settings management system** to deal with the generation and management of complex settings.
- **Control of the main beam parameters** implying fast, accurate control in terms of appropriate parameters in both the physics coast and during injection and ramping.
- **Trim facility** to allow for trim incorporation, persistent versus geometric correction, scaling, accumulation, and smoothing out of errors and corrections, time and history dependency of errors.
- **Communication with experiments** and other major systems (cryogenics, vacuum, technical services).
- **Scripting environment** for ad-hoc rapid application development for machine development.
- **Miscellaneous support applications** such as Electronic Logbook.
- **Console Manager** to launch and manage controls applications in a unique manner.

The Software Development Process

The current software development follows the Unified Software Development Process [20]; a practical software process model followed by many industrial OO projects. Java is the programming language chosen for the implementation of the high-level services and control room applications, as it enables platform independent development. XML is playing an increasingly important role in the exchange of data.

A suite of selected software tools for code optimisation, quality insurance and testing is provided to the developers to guarantee the quality of the control room software. Software configuration management facilities are provided on top of the archive engine (RCS) to provide version management services, making it possible to trace and identify any component of an operational application, and to deliver consistent operational software releases. In addition tools for code building and distribution are available to release operational software components in a multi-versioned repository residing on dedicated fileservers (Sec. 14.4).

Operational software, once released, is deployed locally to the operators’ consoles using standard Java distributed deployment technology (e.g. Java Web Start) [21], which guarantee automatic delivery of
software updates without operator intervention. Local deployment ensures the speed and performance required by operations.

14.8.2 Software for LHC Industrial Systems

Several LHC accelerator industrial subsystems will be accessible from the UCR consoles via a SCADA supervision interface. SCADA systems are used in a wide variety of industrial domains and therefore typically provide a flexible, distributed and open architecture to allow customisation to a particular application area. In addition to a set of basic SCADA functionality, these systems also provide a strong support for GUI tools, a set of standard interfaces to hardware as well as an API to enable integration with other applications or software systems. PVSS [17], the industrial SCADA product recommended at CERN, has the following specific features:

- It can run in a distributed manner.
- It has multi-platform support (Linux and Windows).
- It is device oriented with a flexible structural concept.
- It has advanced scripting capabilities.
- It has a flexible API allowing access to all features of PVSS from an external application.

The SCADA development process, in particular when based on frameworks such as UNICOS (Sec. 14.7.4), can be based mostly on configuration and graphic editors rather than programming. A unique device configuration database contains all characteristics of the devices, including device addressing, alarm and data logging parameterisation.

14.9 SERVICES FOR OPERATIONS

14.9.1 Analogue Signals Transmission

While most signals from the LHC will be digitised for internal usage by the respective accelerator systems there are a number of them which are required for visualisation during tuning and measurement. These include some 200 signals, mainly in the 0 to 10 kHz range from the RF system and over 300 signals with a typical bandwidth of 50 MHz from the kicker systems for injection and beam extraction. In the injector chain, the nAos system [22] is successfully used to fulfil the need for coherent acquisition and display of analogue signals. The system’s main feature is the digitisation of signals in acquisition crates as close as possible to their sources, therefore ensuring optimal fidelity. Whenever possible, in order to save costly oscilloscope modules, signals are multiplexed allowing some 100 signals to be serviced by one crate containing four two-channel oscilloscope modules. The triggers for all the oscilloscope modules are elaborated from the GMT distribution (Sec. 14.5.1), ensuring a precise time-correlation between signals. The digitised signals are sent through Ethernet to control room consoles, where a GUI application (Fig. 14.7) allows operators to monitor them in four-channel “virtual oscilloscope” screens.

The nAos system is currently being upgraded to more modern, commercial hardware and software platforms. The new version of the system is called Open Analogue Signals Information System (OASIS) [23]. It will implement all the above features, including arbitration of resource (i.e. oscilloscopes), allocation to clients and the additional functionality for the LHC requirements. This approach will ensure that signals from the injector chain may also be integrated into the LHC operation; the older, lower energy machines are more richly instrumented by this system. As with all LHC systems OASIS must provide information to the Logging (Sec. 14.9.3) and Post Mortem (Sec. 14.9.4) systems. This offers a very attractive possibility of flexibly connecting signals which should prove very useful in understanding malfunctioning of complex systems. Analogue signals will be monitored by Compact PCI-based acquisition systems; the total number of crates will depend on the amount of multiplexing achievable with regard to the number of clients.

14.9.2 Alarms

Operating the LHC will require many services to help both operators and equipment specialists to understand the many complex situations which will arise during maintenance or operation of the CERN accelerator complex. The detection, collection, management and distribution of information concerning
abnormal situations ranging from severe alarm states to warning states, hereafter referred to as Fault States (FS), will require one global system. The system will accept information concerning any problem associated with the CERN accelerator complex and offer this information, in a structured way, to any interested party. The core part of this system will be the LHC Alarm SERvice (LASER), [15, 24]. As a result of a detailed technology survey, a 3-tier architecture was chosen and is now being implemented according to the J2EE specification (see 14.7.3).

Fault State (FS) detection is performed (Fig. 14.5) in the resource tier by surveillance software written by equipment specialists, application programmers and operators. The LASER system offers a standard interface for these sources to push a FS, by means of either a C or Java API [25] based on the messaging system. FS time stamping at the point of detection, using UTC time, down to the microsecond level if needed, will be crucial in correlating data. The middle-tier will collect the FS by subscribing to all FS sources. A number of services and facilities will be offered at this level including:

- FS persistence;
- Logging of FS transitions using the LHC Logging facility;
- FS dependency and reduction management;
- FS browsing and interaction;
- Administration of the overall system;
- User and configuration management;
- Scaling, using clustering facilities;
- FS distribution according to a FS hierarchy, representing the interest of users.

Finally, the presentation tier will cater for clients interested in those services via a Java client interface. The major user of this interface will be the alarm console, which will be used by equipment groups and control centres to select, receive and display FS, and in order to access all LASER services. Important configuration facilities will allow the alarm console to be personalised.

14.9.3 Logging

The process of capturing and recording information on the operation of the LHC is generally referred to as “LHC Logging”. Logging of information in a centralised storage medium is an operational requirement and has proven to be essential in previous projects, including LEP and the String2 installation.
The information of interest that needs to be logged is various and heterogeneous, but usually concerns values of variables that evolve over time, for example cryogenic temperatures, vacuum pressures, magnetic field measurements, bunch intensities, or beam profiles. The total number of logging variables will be in the order of $10^5$-$10^6$ with logging frequencies up to 0.1 Hz.

The main purpose of LHC logging is to improve the machine performance and that of the individual subsystems. Therefore the recorded data will be stored and kept accessible over the consecutive years, in order to permit comparison of historical data, off-line analysis to search for data correlations, and compilation of operational statistics.

For each of the logged records of the data variables, the date-time value is a key attribute, and therefore correct “time stamping” is vital for the exactness of the data. For time stamping, the usage of UTC is endorsed throughout the LHC Logging system (Sec. 14.5.3), with microsecond precision where applicable.

In order to enable effective exploitation of the logged data by users such as equipment engineers, control room operators, machine physicists and CERN managers, the LHC Logging system will employ web-enabled interfaces to graphically visualise the information and to extract it to common desktop tools. It will be essential to be able to correlate logging data to the other major operational data systems: Alarms and Post-Mortem.

The functional specification [26] and architectural design [27] of the LHC Logging system have been documented. The central storage medium will be a dedicated Oracle database instance. Input of the logging data and its description will be done through formally defined XML files, which may be generated from Java, C or C++ programs.

Logging of data will start as of the commissioning of the first QRL sector and will ultimately cover the complete lifetime of the LHC.

14.9.4 Post Mortem

The LHC will be protected by over 10000 interlock channels, thousands of quench detectors and around 3000 beam loss monitors. Many tens of thousands of signals will be available for the operators and engineers to interpret the circumstances of Beam and Power Aborts. The Quench Protection System alone will provide around 80000 signals for Alarms, Logging and Post Mortem purposes. Recovery from a perturbation to the nominal magnet cycling will require a minimum of 2 hours (7 TeV back to 7 TeV) which precludes learning by trial and error as practiced with normal conducting machines.

The LHC Post Mortem System (Fig. 14.8) [6, 28, 29] is required as a suitable diagnostic tool. The purpose of this system is:

- To ensure comprehensive monitoring of machine protection systems.
- To improve the operational efficiency of the LHC by:
  - Providing diagnostics of power and beam aborts - the aim is to quickly identify the origin of failures in order to initiate appropriate actions and restore operation,
  - Building long term understanding of accelerator operation,
  - Providing diagnostics for beam losses resulting from equipment malfunction.
- To explain the mechanism if damage occurs.

To achieve these aims the post mortem data must be complete and coherent across systems.

Whilst it is intended to make full use of diagnostics from Alarms and Logging these systems will not provide all the facilities required for the understanding of quenches and beam losses in the LHC. In particular the underlying hardware must capture transient data pertaining to conditions prevailing before, during and after such events. These transients may be relatively slow when they are provoked by quenches; indeed the Logging system will gather some pertinent information, such as cryogenic temperatures. However in the case of beam losses the timescales are extended downwards to turns of the machine.

In order to satisfy the essential aims of being complete and coherent across all LHC systems the following essential ingredients are required:

- Every LHC equipment and diagnostics system must implement a circular Post Mortem buffer of appropriate depth holding the latest data (for example 1000 turns for beam instrumentation).
- This data must be time stamped using a common clock with a precision related to the corresponding process (see Tab. 14.2).
• The Post Mortem buffers must be frozen by the Post Mortem timing event or by self-triggering in the case of the protection systems.
• Data must be self describing so that it can be managed by the event builder and analysed by generic tools.

![Post Mortem Architecture](image)

Figure 14.8: The LHC Post Mortem Architecture

The Post Mortem events will be large - several gigabytes - and therefore analysis must be automatic in order to generate digested information for operators. After analysis the information must be stored in order to build up the longer term history and understanding of the machine. The most relevant data will be stored for the lifetime of the LHC.

14.9.5 Support for Real-Time Feedback Loops

To operate the LHC efficiently with high intensity beams, a number of key beam parameters must be stabilised using digital real-time feedbacks and fast feed-forward from reference magnet measurements. It is anticipated [30] that an 80% reduction of the persistent current decay effects can be achieved with feed-forward but that real-time feedbacks will be required for orbit, tune and chromaticity. The expected sampling frequencies of the feedbacks lie in the range of 10 to 100 Hz. For chromaticity the main difficulty lies presently in the possibility to measure this parameter in a non-destructive way at a reasonably high frequency.

It has been shown [31] that a properly designed feedback loop can control orbit distortions during injection and at the beginning of the ramp. For example, a proportional–integral control loop that samples at 10 Hz and that has a total time delay of 100 ms can reduce orbit errors at a frequency of 0.1 Hz with a gain of 8. A considerably better performance can be obtained when the sampling frequency is increased and/or when the total time delay is reduced.

From the point of view of infrastructure, robust operation of feedbacks mainly concerns networks and FECs. A global orbit feedback in particular will involve more than 100 FECs of the beam instrumentation and power converters. Despite the large size of the LHC ring and the distribution of the equipment over the machine, the network performance must be stable and guarantee delays in the range of a few milliseconds for data sent between any points of the LHC. Large variations of time delays due to the computation, control or communication must be avoided, if necessary by allocating dedicated resources to feedback operation.

The LHC FECs will provide good real-time support based on the LynxOS operating system. Networking topology and resources will be organised to meet the service requirements. At present no specific infrastructure for feedback is foreseen.
14.9.6 Control System Monitoring and Diagnostic Tools

The monitoring tools used by accelerator operators, experimental areas operators, technical services operators and system administrators allow operators to detect, understand and repair faults or remotely reboot the computers. The current system (Fig 14.9) monitors a few hundred UNIX, Linux and LynxOS computers, 365 days a year, 24 hours a day.

In order to cope with the growing and evolving controls environment as well as to conform to the new application software standards (Sec. 14.7.3) a new design of the monitoring tools is currently underway. A new surveillance program will inject alarms directly into the LASER system (see 14.9.2). The new operator interface will be based on the same technology as the LHC Alarm Console and will allow the operator to view the state of all the computers being monitored, either good or bad. The operator can then select one of them to perform more detailed checks or restart a failing program or even reboot the computer.

The LASER archive will store all these monitoring alarms. So, after a major event for example a UPS failure on an LHC site, the operators and system administrators will be able to check at what time computers stopped functioning due to loss of power and when the control system was restored.

As part of the upgrade, the surveillance will be extended to Sun Workstations, Windows Servers and PLCs in order to cover all the main components of the LHC Control System.

REFERENCES

[9] Installation and Integration Database. R. Saban Presented at the 16th Meeting of the LHC Controls Project, CERN. September, 2001
[26] The LHC Logging System, functional specification, EDMS Id 358869
[27] The LHC Logging, design specification, EDMS Id 367787