D.2 Controls

2.1 INTRODUCTION

This section outlines the control system for the DIAMOND project. The choice of control system toolkit is presented and a description of the chosen solution. The requirements for the control system are considered in the generic sense, for overall functionality, the applications and the underlying hardware interfaces. For each of these, the design is not sufficiently well advanced to define detailed specifications but in each case general requirements and likely solutions are presented.

2.2 SCOPE OF DIAMOND CONTROL SYSTEM

The DIAMOND control system will be a site-wide monitoring and control system for the accelerators, beamlines and conventional facilities. The control system will extend from the interface of the equipment being controlled through to the operator. It will include all hardware and software between these bounds: computer systems, networking, hardware interfaces and programmable logic controllers.

The control system will not include any control or data acquisition for the experimental stations, which is addressed in Part B subsection 10.5. It further will not include the personnel protection system, which is a separate system that will be monitored by the control system and is addressed in Part D section 5

2.3 REQUIREMENTS

2.3.1 Control System User requirements

The control system must support several user groups, each with varying requirements.

(a) Operators

The accelerator operators are the principal users of the control system. The control system must be a complete and consistent interface to perform any function on the accelerator complex. All data and interfaces must be consistent in how data is presented and how equipment is seen to behave. The operation of the accelerators requires real-time control and monitoring of the equipment, archiving, alarm handling, sequencing, backup and restore for routine operation. Alarm and error messages should be supported by information regarding recommended courses of action. The Control System should allow the automation of onerous plant operating tasks. The Control System should provide applications which encourage and facilitate the keeping and passing of operation logs, particularly from shift to shift.

(b) Accelerator Physicists

The accelerator physicists requirements of the control system include all the routine operations of the control system together with the ability to integrate programs developed to support particular experiments. Functionality is required to allow easy acquisition of data produced as part of an experiment, and to provide the ability to switch between the accelerator and an accelerator model. Data retrieved from the control system must be acquired with sufficient time accuracy to enable accurate correlation.

(c) Technical Groups

The technical groups require diagnostics to enable maintenance such as calibration and fault finding. Access to the control system is required in the Main Control Room, local to the equipment and potentially in the offices, laboratories and off-site. Applications must provide all diagnostic information necessary to assist in commissioning and debugging of equipment. They must provide useful fault diagnosis facilities to assist with plant equipment maintenance and maintenance of the Control System itself (both hardware and software). An easy interface to databases of equipment properties, manufacturers, documentation, cabling data and fault histories is required, as well as access to information clearly identifying the geographical location of equipment and a system of fault prediction facilities to allow for scheduled maintenance of components likely to fail.

(d) Experimenters

The end users of the experimental station require a straightforward graphical interface to the control system. They also require good integration of control system parameters with the experimental control and data acquisition systems. This is particularly necessary in the case of synchronising scanning of a sample with changing a parameter on an insertion device in the storage ring e.g. the gap of an undulator. Experimenters require clear information on light source status and performance, timing signals and may require remote (i.e. off-site) access to experiments and beamlines.

(e) Control System Engineers

Control system engineers require current and archived data on the status and behaviour of the entire control system. Information required includes CPU loading, network loading, application monitoring (for frozen/crashed applications), connectivity status and reports of any control system faults.

(f) Facility Managers

The Control System should be capable of producing operating reports and statistics in a form which can then be imported into software applications (i.e. spreadsheets, web-based tools etc.) used by management.

Information required could include number of hours of beamtime supplied to users and unplanned beam dump statistics – how often these events occur, time taken to restore beam, reason for beam dump, and signs of common modes of failure.

(g) Staff and Public

A wide range of other groups will require information from the control system. These include technical and scientific groups on- and off-site. These groups should be served through a web service as the user interface.

2.3.2 System requirements

(a) Architecture

The control system will adopt the standard two-layer client server architecture, to ensure scalability and avoid performance limitations. It should use a proven tool kit with well-defined interfaces at both the server and client to enable integration and development. It should enable the use of hardware and software already developed for specific light source requirements.

(b) Network

The connection of the control system layers should use standard network components. These should be in a redundant configuration and include provision for network evolution, i.e. the development to Gbit standards.

(c) Operator Interface

The operator interface will be either workstations or PCs running Unix or NT as these are the established standards. The control system should seamlessly integrate with office systems through a gateway process to maintain security.

(d) Equipment Interface

The equipment interface will provide the physical connection to the equipment being controlled through a variety of interfaces. The preferred standard will be VME because of physical and electrical performance. The PC architecture is particularly discouraged as the equipment interface; however, Compact PCI may be an acceptable alternative to VME.

(e) Database

The control system must include a relational database as a central repository for all configuration information. This should include all static information about accelerator components such as coefficients to calculate field magnetic strength from current. Consideration should be given to extending the database to include all technical information to enable subsequent support and maintenance. At the application level, there should be a unified and seamless interface to both the static and dynamic data.

2.3.3 General requirements

(a) Performance

The update rate of channels should be at least 10Hz to enable analogue style control for all parameters. On display screens, an update of 1000 channels per second should be achieved. The update of channels should be on data change outside bounds to manage the use of network bandwidth. The loading of control panels or applications should take less than 1 second to impose no significant delay in carrying out a control operation.

(b) Reliability

The control system should be designed as a high reliability system to have greater that 99% availability for 24x7x52 operation enabling facility operation for in excess of 6000 hours per year. This requires a long MTBF and a MTTR. These call for systems to be modular in design and distributed. Single point failure systems with long time to repair such as servers and network infrastructure will be required to be in a redundant configuration.

(c) Scalability

The control system must be scaleable. As the control system expands with the build-up of the accelerators and during the life of the facility this must not impact on the performance.

(d) Openness

The control system should use open standards and an open architecture. The life expectancy of an accelerator complex means that the control system will have to evolve to incorporate upgrades and new technology. The control system must enable seamless integration of systems at both the server and client side through well-defined APIs.

(e) Security

The control system must incorporate security to manage the access requirements of the different classes of user. It must also include security and tools to manage security of the computers and networks systems used.

(f) Integrated

The applications that comprise the DIAMOND Control System should have a consistent "look and feel". Related functions should be linked to reduce the number of mouse clicks a user has to perform. For example, trends should be accessed by clicking on a process value hotspot displayed on a plant synoptic value.

(g) Seamless

The source of the data should be hidden from the users. For example, it should be possible to display data from the Control System, from the associated Relational Database and from an Accelerator Model on one full-screen synoptic.

(h) Ease Of Use

All Control System facilities which need to be accessed directly should be accessible via menus, where the menu titles give a clear indication of the facility being called up. It should never be necessary for a user to have to remember the name of a program or of data files in order to use the system.

Comprehensive online help facilities should be designed into the system, and these should be context-sensitive so that the operator does not have to navigate through a large selection of Help information.

(i) Ergonomics and Consistency Of Operation

Conscientious attention to common-sense ergonomics during application development will clearly pay dividends for long-term ease of use and minimise familiarisation and training costs for new users.

Production of a concise Style Guide document at an early stage in the development cycle of the project will provide a good ethos for addressing these issues.

(j) Upgradability and Flexibility

Control System applications must be designed, where possible, to enable future upgrades to be incorporated economically.

Wherever possible, application programs should be provided with command-line options, so that the functionality on offer can be increased by simply running the program with a different set of command line options.

2.4 TOOL KIT

2.4.1 Choice of Tool kit

Four options for a control system tool kit were reviewed [1]; EPICS [2], TANGO [3], a commercial SCADA system [4] and a development on the existing SRS control system

[5]. The conclusion from this was that any of the options could be made to work, but with varying levels of functionality, support and cost. However, EPICS had advantages over the other options in three key areas; user base in the accelerator community, functionality for accelerator-related systems and support for the required hardware interfaces.

2.4.2 The EPICS Tool Kit

EPICS came out of a collaboration of control groups, across a number of research organisations, to produce a tool kit to build distributed control systems. The resultant tool kit is such that it reduces software development and maintenance cost by providing: configuration tools in place of programming, a large user base of proven software, a modular design that is expandable and well defined interfaces for extension at all levels.

World-wide, EPICS has a very large user base, on a variety of accelerators, detector systems, astronomical projects and industrial processes. In Europe it has recently been applied as the control system on both the BESSY II light source and the Swiss Light Source together with other smaller projects. In the USA, EPICS has been used on the APS light source, a number of small accelerator projects and is being applied to the Spallation Neutron Source.

The use of EPICS on a diverse range of projects means that there is a large base of drivers and hardware support already available. The existence of these makes interfacing of the underlying systems less dependent on software development.

The EPICS tool kit is supported through the collaboration with software distribution and documented through the web. There are EPICS training courses run each year by LANL, and two EPICS workshops rotating through America, Europe and Asia each year; a number of individuals and companies are also available to provide support and training. In the UK commercial support is available from at least two companies.

2.4.3 Structure of an EPICS Control System

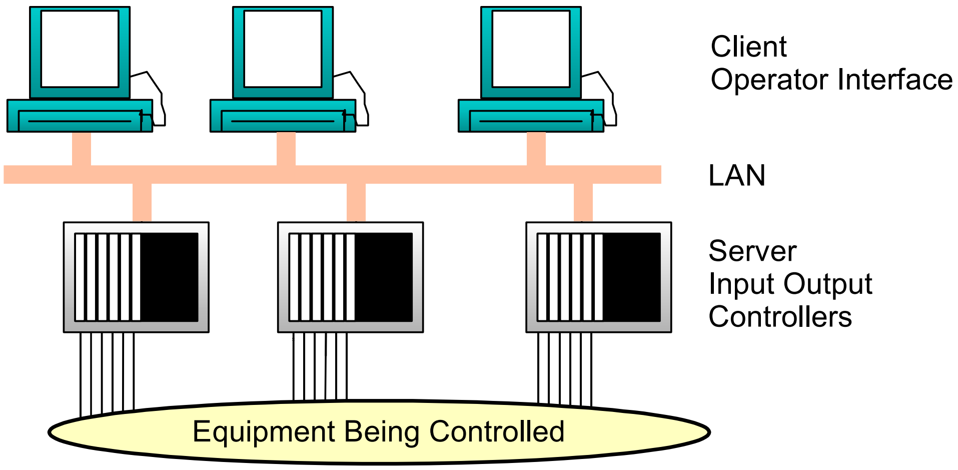

Figure 2.4-1 EPICS Model

EPICS embodies the standard client server model, Figure 2.4-1, for a distributed control system. The user consoles are one class of client that receives and processes information. The servers are the source of information and in the general case interface to the equipment being controlled. The clients and servers are physically connected using network technology and communicate with the EPICS protocol Channel Access.

2.4.4 EPICS Servers

The physical realisation of EPICS servers is typically as multiple embedded VME systems, which are called IOCs, Figure 2.4-2. These interface to the equipment being controlled, for which EPICS supports a large range of physical interface standards, protocols and devices. IOCs also support the use of an event timing signal, to time-stamp transactions and enable synchronous acquisition or control across multiple IOCs.

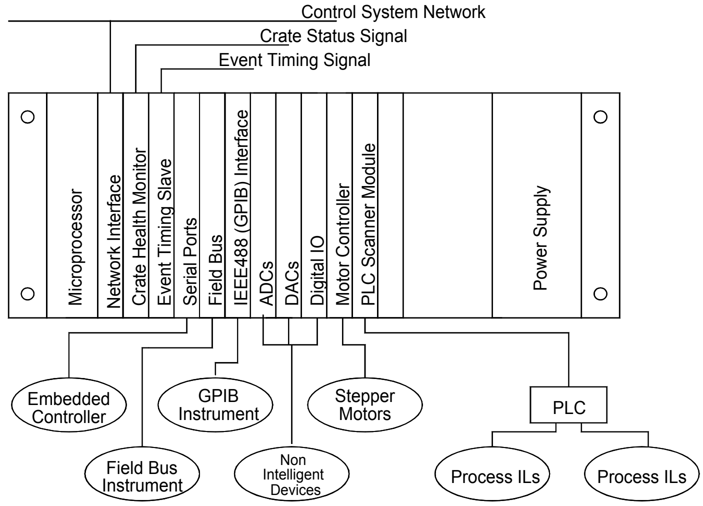

Figure 2.4-2 Example EPICS IOC

2.4.5 Server Side Processing

Within the IOC, the CA server communicates with the Process Database which uses the Record Support, Device Support and Driver Support layers to interface to the plant, fig 2.4-3. The communication from then EPICS client, over CA to the database, can be by synchronous call and reply or by the client establishing a monitor whereby the server asynchronously serves the data. The update of monitors can be on a periodic basis, on change of data or external event.

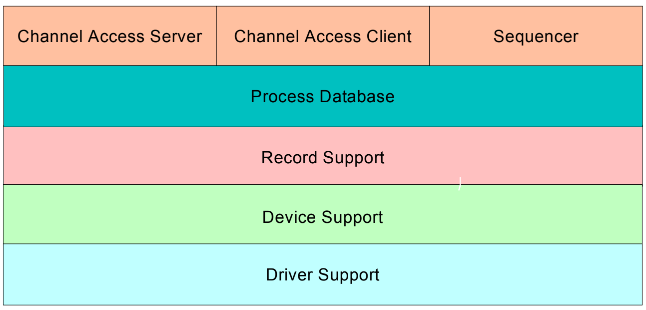

Figure 2.4-3 EPICS IOC data model

The process database is a memory resident database that defines the functionality of the IOC. The database uses the Record Support layer to perform the processing necessary to access IO, perform data conversion, alarm checking, and to update monitors. The IO operations are carried out through the Device Support layer, which handles equipment specific protocols and the Driver Support layer for the hardware interfaces. The structure provides support for interfacing to embedded controllers, field buses, IEEE488 (GPIB), DACs, ADCs, Digital IO, stepper motors, PLCs, power supplies and a range of instrumentation.

Within the IOC there is also a CA client to facilitate IOC-to-IOC communication. This is realised by linking process information from one process database to a process database on another IOC.

An IOC also contains a Sequencer to perform Finite State Machine control on the process database. The sequencer logic is defined as SNL, which is compiled to C code, then to an executable to run on the IOC. This allows for easy production on complex sequences, such as switching through the steps in bringing on a piece of equipment.

A standalone version of the CA server is available, which can be integrated into other systems without the process database and support layers. This facilitates integration of self-contained systems into EPICS, one example being the integration of LabView systems.

2.4.6 EPICS Clients

The client side of EPICS is realised on either Unix workstations or PCs running Windows and is called the OPI.

2.4.7 Client Structure

In the standard EPICS model the OPI application programs interfaced directly to the CA client. This has limitations in that it only provides access to the dynamic control data through the CA API and so limits seamless integration of data from other sources e.g. RDB. A solution to resolve this is presented in section 2.6.8 Application Programming Interface. The EPICS toolkit provides a suite of applications for the OPI. The core tools are: a synoptic user interface for control and monitoring (MEDM), an Alarm Handler, an Archiver for recording and retrieving the historical state of the control system, a backup and restore facility to take snap shots of parameter settings, a knob manager to provide attachment of physical knobs to parameters and a parameter plotting tool.

There is support within EPICS for the scripting languages Tcl/Tk, Perl and Python, Mathematica modelling package, Wingz spreadsheet, IDL graphical visualisation, LabView and Java. Data can be further served up on to web pages through a CGI server.

2.4.8 EPICS Tools

The functionality of a control system built with EPICS is defined in three places; the Operator Interface, the Process Database and the Sequencer logic. EPICS provides a number of tools to develop each of these which do not require software development. The Operator Interface applications can be produced by a number of interface-generating tools, one of which is MEDM. These tools allow for control objects to be placed on pages and animated with a connection to the control parameters. There are both text and graphical tools to produce the Process Database, which involves selecting records and drivers, and linking them to process data and alarms. The

Sequencer logic is produced from SNL, which can be defined as text or more recently in a graphical form.

2.5 DEVELOPMENT ENVIRONMENT AND TOOLS

2.5.1 Distributed development

Each developer will work on either on a local Solaris workstation or Linux PC. These will be networked to a development file server providing read-access to all the necessary development tools and applications (VxWorks, EPICS base and extensions, configuration, database files etc.) see Figure 2.5-1

Modifications will only be made to local copies of applications, which will then be checked in to a branch of the CVS repository to enable any changes to be backtracked. When the application has been fully tested and is stable, this branch will become the main development branch for the control system.

A central boot server will be used to provide all the necessary files required by the local IOCs for booting. These files will be generated from the CVS repository. This will ensure that all IOCs are running a consistent and stable version of the control system. The contents of the central boot server will be mirrored on boot servers located at each of the CIAs, which will provide local booting of the IOCs in that CIA

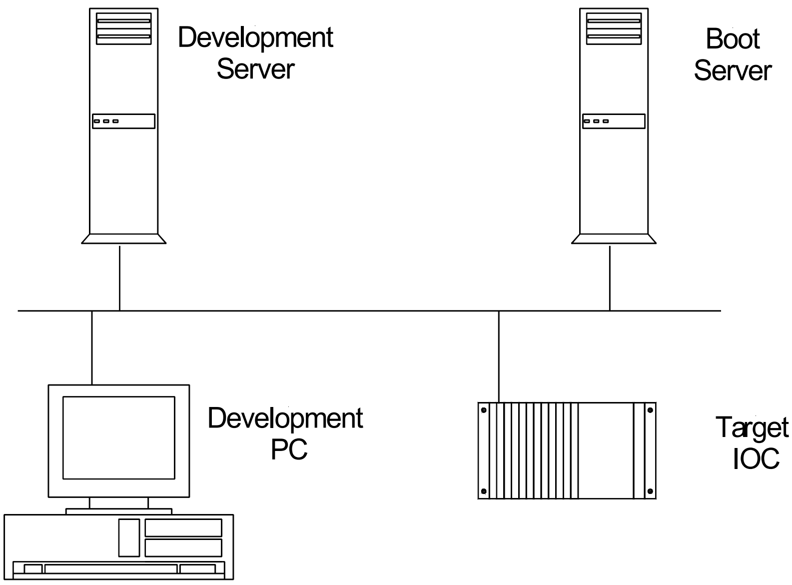

Figure 2.5-1 Distributed development structure

2.5.2 File Management with CVS

CVS [6] is a version control system for keeping track of all modifications to project source code files. CVS is widely used in both open source and proprietary software development projects, and is generally considered to be the best freely available, fullfeatured version control tool. Two features make CVS particularly suited to collaborative development across any network including the Internet:

- CVS allows multiple developers to edit their own working copies of files simultaneously, then deals with combining all the changes and notifying developers when there are conflicts.

- Remote access to source code file repositories. Developers who are project members can obtain and modify project files from virtually anywhere.

CVS is a client-server system. The CVS repository is maintained on a server; clients run on users' machines and connect to the server via the network (or Internet). Clients are available for nearly all platforms including, Unix, Windows, Macintosh, and any Java-based platform.

CVS allows project members to:
- Check out source files and directories
- View differences between versions of files
- View change log history and comments
- Commit changes made in their local copies of the source files to the main source code repository
- Update their local project files when they want to remain in sync with changes committed by other project members.

CVS has proven very beneficial to many other accelerator projects in the world, and there is a very large CVS knowledge base within the EPICS community .

2.5.3 Application Development

Most application requirements can be met through the standard tools as discussed in section 2.6. Where more intelligence is required at the application level there are EPICS interfaces to all the popular programming languages. The preferred solution will be to use C/C++, Java or scripting languages to minimise the number of supported languages.

(a) C/C++

C and C++ are high-level programming languages which have, for some time, been the de facto standard for portable open systems solutions on Unix/Linux platforms, with C++ usage increasing due to the popularity of Object-Oriented design and programming. Both languages have been widely used both for the EPICS baseline product and for driver software and other applications built on top of the baseline. For the DIAMOND Control System, the emphasis will be on re-use of existing software; it is not anticipated that the project will involve large-scale production of new C/C++ code.

(b) Tcl/Tk

Tcl is a widely-used open source scripting language. It has a simple and programmable syntax and can be either used as a standalone application or embedded in application programs. Tk is a graphical user interface toolkit that can be used for rapid development of powerful GUIs. Tcl and Tk are highly portable, running on essentially all flavours of Unix, (Linux Solaris, IRIX, AIX, *BSD*, and so on) Windows, Macintosh, and more.

Tcl/Tk is well supported and extensively used on many EPICS-based projects, particularly for GUI development.

(c) Java

Java is an object-oriented interpretative programming language with a built-in API that can handle graphics and user interfaces. Java can be used to create standalone applications. However, a more important use is in the development of applets, programs which can be embedded in a Web page. The growth of the Internet, together with Java's hardware independence, has made the language essential for web-based developments.

Currently, Java performance issues mean its usage will only be considered for applications where response time is unimportant. Generally, though, Java solutions providers are seeking to improve performance with developments such as Just In Time compilers and Java processors. If these developments yield effective performance improvements during the development phase of the DIAMOND project, then the importance of Java on the project will increase.

2.5.4 Server Development

Development of EPICS at the server level is required potentially in three places, namely record and device support, database and state notation language.

(a) Record and Device support

While there is extensive record and device support available for EPICS, addition of unsupported hardware will necessitate the developed of Device and possibly Record support layers. The EPICS toolkit provides well-defined interfaces to each of these layers and examples to aid development. Device development is carried out in C within the standard EPICS development environment.

Building and developing EPICS requires the VxWorks [7] development environment. This is currently only available for Windows or Solaris. However, given that the development environment is based on the GNU tool chain it should be possible to run these tools on Linux, making this an alternative to Solaris. The preference will be to standardise on one operating system for development, possibly Linux.

(b) Database Configuration Tools

There are several DCT's available. These DCT's allow designers to create EPICS databases by implementing them visually with a 'block diagram and link wire' approach, similar to that used in electronic schematic design packages. Two of these are being evaluated for the DIAMOND Control System, CapFast [8] and VisualDCT [9].

The database development cycle will involve importing the EPICS runtime database into the central relational database (see section 2.10) to have a single repository of all control system information.

CapFast is probably the most frequently used EPICS database configuration tool currently in use. It was originally created for electronic circuit design and adapted to EPICS. The EPICS library contains all the standard records and these are represented as graphical symbols and connected via lines. The rest of the properties of each symbol (fields of the record) are edited in a properties window.

One of CapFast's strongest features is the ability to create database templates that can accept field substitutions when instantiated in a database design.

The schematics created with the EPICS library are translated from an intermediate file into the EPICS database format with a build script and schematic compiler tool. This build process is not reversible; i.e. you cannot convert a database file back into a schematic diagram. This has the limitation that any minor changes to the database have to be entered at the schematic level and then the database rebuilt. Any manual changes to the database file will not be reflected in the schematic diagram.

CapFast may be used to create EPICS databases initially, while the newer VisualDCT package is developed, but is unlikely to be used in the long term.

VisualDCT is an EPICS database configuration tool written in Java and can therefore run under any operating system that can support a Java Runtime Environment. It was developed to provide features missing in existing configuration tools and to make databases easier to understand and implement.

VisualDCT has a powerful DB parser, which allows existing DB and DBD files to be imported with ease. The parser detects syntax errors in databases, as well as defective visual composition data or its absence. Faults in DB files are safely handled and do not raise any critical errors. VisualDCT automatically lays out all objects that have no visual composition data and saves all visual data as comments to maintain backward compatibility. The output from VisualDCT is also a DB file, with all comments and record order preserved.

VisualDCT can be considered as a rapid database development tool - intuitive database construction can be done quickly and with little effort. Visualisation of the databases makes them easier to understand, as errors are much simpler to find (e.g.

broken links are indicated by a red cross) and the hierarchical views simplify the design by splitting databases into logical blocks.

VisualDCT provides many now 'standard' features such as clipboard and undo support. Much effort has been given to synchronisation between the database and its visualisation. Every change done visually is immediately reflected in the database and vice versa; all actions, such as moving, renaming and deletion of records, which can affect database links are automatically fixed and updated by VisualDCT in real-time.

VisualDCT does have some deficiencies; some of the more advanced features that other DCT's have, such as the ability to create database templates and perform field substitutions, are not present. However, VisualDCT is under development and its shortfalls are to be corrected very soon. With these shortfalls in mind, the ease of new database creation, existing database manipulation and cross platform support, Visual DCT is the preferred choice for EPICS database configuration tool.

Visual DCT has been written within the EPICS community specifically to support EPICS, and is available free to EPICS database developers. However, the developments of VisualDCT to add some of the missing functionality will have to be funded.

(c) State Notation Language / Sequencer Tools

The sequencer is a tool within EPICS that allows the implementation and control of (one or more) state machines on the IOC. The state machines are created using EPICS SNL. SNL has a C-like syntax that has constructs for building state machines. Once the SNL source code has been written, a SNC pre-processes it into 'C' code and then compiles it to create an object file which the sequencer can run in the IOC.

VisualSNL [10] is a graphical tool used to generate EPICS SNL code. VisualSNL enables non-programmers to build state machines and enables programmers to avoid having to learn yet another language, as well as providing better documentation.

The user creates, on screen, a state transition diagram, and can add, delete, move, and edit states as well as defining transitions between states. New features are being implemented, including the ability to read an existing SNL file.

VisualSNL is written in Java and can therefore run on any operating system that supports a Java Runtime Environment.

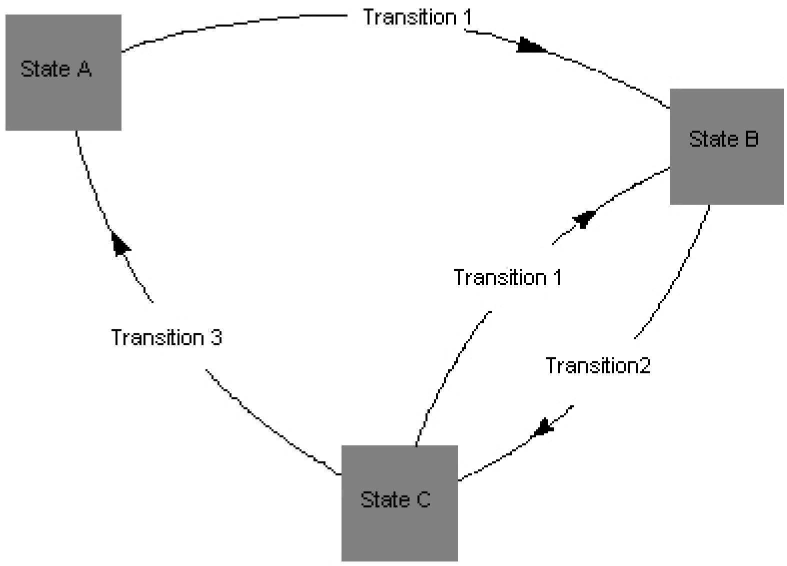

Figure 2.5-2 Example of a state machine in VisualSNL

2.6 USER INTERFACE

2.6.1 Operating Modes

During normal operations, the DIAMOND control system could be regarded as having one of three possible modes. A number of application requirements can be derived by considering these modes.

(a) Storage Ring Fill

The Control System usage reaches a peak during this state. All facilities of the system are in use. Therefore, this is where any performance implications will become critical. No individual application must make excessive demands on system or network resources.

The Control System must provide the operators with a means of consistently managing the fill process, and consistently diagnosing and handling any problems that prevent the storage ring from being filled. The SRS Sequence Manager application successfully achieves this and would provide a good basis for a similar application on the DIAMOND Control System.

(b) Steady State (including Top-up)

Routine adjustments and checks will be carried out. Some of these will be in response to requests from beamline users.

There is, in general, no need for rapid operator intervention.

Monitoring of trends may highlight certain actions being required, for example in the vacuum system. In general, though, if an event occurs that will cause the beam to dump, the time between the event and the beam dump is so short (of the order of microseconds) that there is no chance of an operator taking remedial action to prevent the beam dump.

(c) Unplanned Beam Dump

The critical application requirement after an unplanned beam dump is to provide control room staff with sufficient diagnostic information to allow identification of remedial action as quickly as possible.

The diagnostic applications must be easy to navigate, and must retrieve the data quickly. The amount of pre-dump data and the frequency at which it has been collected and stored must be sufficient to ensure quick and correct diagnosis of the problem. As a further development, beam dump trip information could be saved for long-term archiving along with operator comments on how the problem was fixed. This information could be saved as one data "package" accessible online, so that control room staff could look for similarities with past beam dumps. This information could also be a useful tool enabling management to identify common failure modes during DIAMOND operations.

2.6.2 Console Applications

The EPICS software package offers comprehensive operator display applications including :

- Tcl/Tk toolkit interface
- Motif Editor and Display Manager (MEDM).
- Extensible Display Manager (EDM).
- Array Display Tool (ADT)
- Parameter Display Page (DP)
- Knob Manager (KM).

These applications will be used to supply operator display facilities which will include.

(a) Operator Menu Bar

This will provide rapid single-click access to all key operator facilities.

(b) Plant synoptics

These full-screen plant schematic diagrams will provide operators with an at-a-glance indication of plant conditions. Each process value displayed on a synoptic will constitute a "hotspot"; clicking on a hotspot will produce a pull-down menu providing access to further information and control actions relevant to that process value. Typically, a text description of the signal, the units of measurement, alarm limits, maximum and minimum, trend, alarm history, wiring information, operator comment and web access to online help might be provided). By this means, plant synoptics will act as the launch platforms which allow operators to access a wide variety of data in a seamless manner.

Ease of navigation will be considered during the detailed design stage for plant synoptics. An overall Synoptic Menu will be provided, which lists all synoptics grouped by functional area, presenting a clear hierarchy. In addition, where appropriate, plant synoptics will contain links to other associated synoptics. The design aim will be that operators should be able to navigate around the hierarchy without the constant need to return to the Synoptic Menu.

Plant synoptics will be designed to have a simple, uncluttered appearance so as not to present more information to the operator than can reasonably be taken in.

(c) Control panels

Usually sized smaller than full-screen, these will be available with a wide variety of control widgets (radio buttons, slider bars, data entry fields with data validity checking etc.) to allow users to apply control actions to the plant.

Control panels can be configured such that a single slider bar is used to control simultaneously a number of control outputs. Mathematical functions are available to define how these combined control outputs operate in relation to one other.

(d) User-configurable tabular displays

Operators will be able to configure their own sets of tabular displays showing closely-related accelerator parameters. Facilities will be provided to save these user-configured displays with a user-chosen name, and to recall the display from a list presented in a pull-down menu.

(e) System Status schematics

These schematics will show the status of IOC's, operator monitors, printers etc. They will also display the health of key applications – so that, for example, operators are made aware quickly if alarm processing stops due to an alarm server program crash.

(f) Operator Comments facility

This will allow operators to enter lines of text comment for any plant input – to record, for example, when an input is not reading correctly due to a known fault. The presence of an operator comment for a process variable will be clearly indicated on any synoptic which displays that process variable. Individual comments will be easily readable via a suitable control panel, and it will also be possible to collate lists of comments (e.g. all operator comments entered during a shift).

(g) Signal Information Panel

Only a subset of the process variables will be displayed on plant synoptics. However, operators require rapid access to information about any process variable and the Signal Information Panel satisfies this requirement. The panel will provide a Search section and a Display section. The Search section will enable the user to carry out a name search on the relational database, using a name mask to search for either a EPICS database record name or an EPICS record descriptor. Clicking on one of the returned search results will enable the user to request further information (e.g. trend, alarm history, operator comment etc.).

(h) Message Logging

The CMLOG package available with EPICS will be used to provide a distributed message logging system. This package can be used by any application or system that needs to log messages to centralised log files and display distributed messages to users. The CMLOG package supports C++, C and CDEV application interfaces for logging messages and has C++ application interfaces for searching/retrieving messages from a dedicated logging server. Applications may send a selection rule to the server to select a subset of log messages for viewing – these rules can be in a form similar to C logic syntax or in a form similar to SQL.

A sample Message Log Browser (an X-Windows Motif application) is included with the CMLOG package. An additional browser will be developed using the supplied application interfaces once detailed requirements are established during the detailed design phase of the project.

2.6.3 Alarm Handling

The EPICS Alarm Handler package will be used to provide the following facilities:

(a) Alarm List

An Alarm List allows the users to view and manipulate current plant alarms. The Alarm List will incorporate the following facilities :

- Indication of alarm acknowledgement state.

- Alarm message which includes EPICS record name, descriptive text, alarm value and date/time of alarm generation.

- Removal of acknowledged alarms from the Alarm List when they are no longer in the alarm state.

- Access to a menu-based set of facilities from each alarm in the Alarm list. The menu would give access to further information about the alarmed signal, including :
    - Trend
    - Alarm History
    - Access to a synoptic which includes the alarmed signal.
    - Web access – e.g. a link to a text page with more details about the alarm condition and possible corrective action

    - Operator-selectable alarm inhibition to prevent use of the Alarm List from being disrupted by non-genuine alarms (e.g. "flickering" alarms being generated by a faulty switch). The names and descriptions of inhibited signals will be viewable on a separate list, from where it will be possible to de-inhibit each signal.

    - Association of each alarm with a plant area, along with the ability to display only alarms for a particular plant area.

    - Colour indication of alarm severity.

(b) Alarm Logging

All alarm messages will be logged to text file for interrogation and archiving purposes. An Alarm Log Viewer will be available, with various filtering options such as date/time, alarm severity, input name etc.

(c) Audible/visible alarms

Provision will be made for audible alarm tones, driven from software using wav files. A separate alarm tone will be available for each alarm severity.

(d) Alarm Banner

An alarm banner window will be available to display the latest n (where n is configurable) alarms in a dedicated window at the top or bottom of the screen. Alarms can be acknowledged via the banner without having to call up the main Alarm List.

2.6.4 Archiving

The EPICS software toolkit offers comprehensive short, medium and long-term data collection, archiving and retrieval through the EPICS Channel Archiver package. This package will be used to provide the following facilities.

(a) Long-term archiving

The archiver provides support for:

- Data retrievable in tabular and trend form.
- A data quality indicator associated with each item of data.
- Data compression to minimise the size of archive files.
- Dumping of data to removable storage media, for long-term storage.
- Loading of archive data from removable storage media for data analysis.
- Timely warning to operators when archive data collection is compromised by a “disk full” condition on the archive server.
- Variable data collection intervals for archiving.
- A mechanism for easily configuring large numbers of process variables for archiving (e.g. by use of name masks).
- Facilities for collecting data in user-definable data sets, where data sets can include files as well as process variable data.

(b) Historical Data Collection

This provides for short to medium-term data collection offering the following features :

- Data retrievable in tabular form and trend form.
- Data quality indicator associated with all data.
- Variable data collection intervals.
- Mathematical functions (e.g. averaging, MIN-MAX etc) applicable to historical data.

(c) Data Retrieval and Data Management Tools.

A wide variety of data retrieval and data management tools are available as standard with the Channel Archiver package, including :

- Retrieval via scripting tools, provided by the Channel Archiver Scripting Interface. TCL, Python or Perl can be used to develop automation of archive handling.

- Retrieval via native tools, with Xarr/Striptool for UNIX-based systems and WinBrowser for Win32 systems. WinBrowser also provides data export in spreadsheet format or in a format suitable for the Matlab data analysis and modelling package.

- Retrieval via a web server plugin, offered by the CGIExport client, which allows users to browse the archive via any web browser. File download in spreadsheet or Matlab format is supported by this plugin.

- Command-line tools provided by the ArchiveExport/ArchiveManager component, providing commands to manage archives and to export data to a spreadsheet, to Matlab or to the GnuPlot plotting utility program.

- The Archive Engine component of the Channel Archiver package includes a built-in Web server. By using this feature, current operational parameters can be viewed and interactive configuration can be carried out via any Web browser.

2.6.5 Sequence Manager

A sequence manager tool will be needed to automate many of the routine, high-level tasks encountered when operating a light source. These tasks include refilling the storage ring, switching the injector system on and off, opening and closing valves on beamlines etc.

The sequence manager tool should include the following functionality:

- Simple graphical user interface

- Ability to select and execute a pre-defined sequence

- Each stage of the sequence should only be available for execution if all previous stages in the execution flow have been successfully completed.

- Comprehensive reporting of any errors or anomalies that occurred during each step of the sequence.

- Each stage in the sequence should contain a clear description of its purpose and operation that can be displayed to the operator.

- The operator must be able to make decisions at key points within the sequence to alter the execution flow.

- The sequences will be written in a simple, interpreted language or script (for example Tcl) so that operators, accelerator physicists and technical specialists can write and maintain the sequences.

At present, such an application does not appear to be available within the EPICS community.

2.6.6 Plotting

The StripTool program will be used for displaying trends of current and archived data. The key features of the StripTool program are :

- A StripTool chart displaying recent live data can be scrolled back to view archive data.

- Data acquisition via both Channel Access and CDEV, thereby allowing trending of both EPICS and non-EPICS data on the same axes.

- Ability to drag signals from synoptic diagram (drawn using MEDM) into a graph window.

- Flexible configuration options, including logarithmic and linear transformations, sampling rate, graph refresh rate, plot colours, grid lines, graph legend colouring, and plot line width. The trend can also be easily reconfigured to make one or more plot curves invisible without removing the plot configuration information for that curve.

- Trends can be printed and trend data saved to file.

- Trends are customisable via X resources, giving access to a wider set of configuration options than those offered by the standard StripTool configuration facilities.

2.6.7 Physics Model

A physics modelling system will be needed to provide an interface between the control system and standard codes such as MAD or RING. These codes can mathematically model the behaviour of various accelerator systems. They can be used to aid understanding of the machine, as well as being a vital tool for optimising and operating a complex accelerator.

The system should be capable of operating in online and predictive modes.

(a) Online mode

Real-time magnet and RF data will be passed to the modelling system. Computed values of Twiss parameters, Q values, chromaticity, momentum compaction factor etc. will be made available to the rest of the control system or to specialised Accelerator Physics applications.

(b) Predictive mode

In this mode offline magnet and RF values are supplied to the modelling system rather than real-time data. This will allow predictive and "look-and-see" experiments to be performed using the machine model to predict the behaviour of the accelerator.

The interface between machine model and the rest of the control system can be achieved by implementing a CDEV service to translate CDEV requests into native machine model commands. This approach allows for easy integration with standard application such as MEDM, ALH etc.

2.6.8 Application Programming Interface

(a) Common Device interface

The CDEV application development framework [11] will be used to provide a single, consistent API to Channel Access, to the Physics Model and to Relational Databases through an ODBC or JDBC layer. This approach greatly simplifies application level programming as it allows database entries to be accessed in exactly the same way as standard channel access devices.

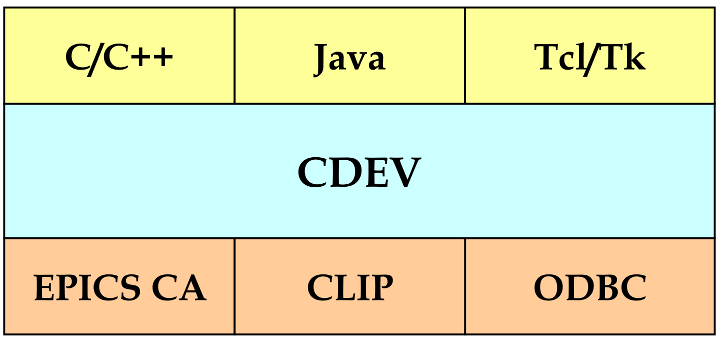

Figure 2.6-1 Application interface

CDEV is implemented both as a set of C++ classes as well as Java classes. There is also an interface to the Tcl/Tk scripting language.

The standard CDEV distribution includes service layers to implement EPICS CA and CLIP [12]. CLIP provides a generic interface to other CDEV applications. It can be used to communicate between CDEV clients and a CDEV gateway [13] system that fully implements all the required services. This approach allows full multi-language support without having to write every service in both C++ and Java.

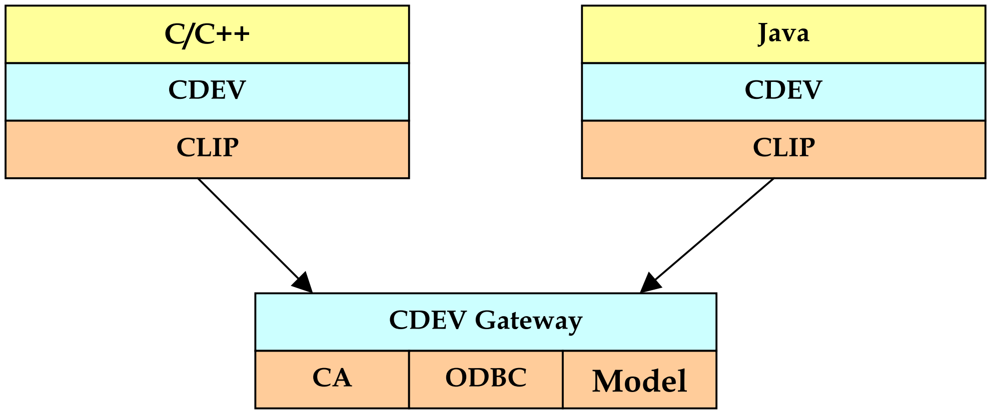

Figure 2.6-2 Application interface using CLIP

CDEV has an object-oriented view of devices. It allows a hierarchy to be created with inheritance of properties. CDEV device names are defined in an ASCII text file containing DDL.

(b) Self Describing Data Sets

SDDS [14] is a way of storing and working with data that was developed at the APS for use in the simulation and operation of accelerators. The basic approach behind SDDS is: define a standard file protocol and then design all programs to work with such files. Since SDDS is very generic in nature, it can be used for processing and displaying data from essentially any source. Parts of SDDS are linked to EPICS.

SDDS is already is use at a number of EPICS sites. In conjunction with the Tcl/Tk scripting language, it has been used to develop GUIs for configuration of the accelerators; data collection, analysis and display; experiment execution; and feedback processes, amongst others. SDDS has also been used for pre- and post-processing of data for accelerator simulations.

(c) Experiment and Data Acquisition Interface

Experiment and data acquisition systems will require a clearly-defined interface to EPICS; for wavelength selection from insertion devices, for example. Such interfaces will be achieved using middleware appropriate for distributed heterogeneous systems, (CORBA being an example); interface specification will be determined during the detail design stage of the project.

(d) External Access

General utilities and EPICS components which will be used to provide external access to Control System data include the following :

- HummingBird Exceed will be used to provide trusted users (e.g. Control System support engineers) with an X-Windows interface for off-site access to the Control System.
- SDDS, in conjunction with Tcl/Tk, will be used to develop graphical user interfaces and for pre- and post-processing of data for accelerator simulations. Security issues can be addressed by making SDDS-compliant data files available for external access via anonymous ftp.
- ActiveXCA provides an ActiveX control interface, allowing any application which is an ActiveX container access to EPICS data. Examples of ActiveX containers include Excel spreadsheets, Internet Explorer and Delphi.
- The CGIExport plugin for web servers will be used to provide web access to EPICS ChannelArchiver data. CGIExport allows the user to browse archived channel names, view online plots. Additionally, there are valuable download features; selected data can be downloaded in spreadsheet format or in a format suitable for Matlab.

Currently, the performance offered by Java is not sufficient to allow generic Control System operator/user interface facilities to be provided for users both inside and outside the Control System LAN. However, improvements in Java performance (such as developments in Just In Time compilers) during the development cycle of the project may make this a viable proposition.

2.7 INTERFACE LAYER

2.7.1 Control System Interface

All the interfaces to the control system will be through the IOCs. Most connections will be directly to an IOC, but signals, which provide interlocking functionality, will be connected to a PLC sub-system. The PLC will manage the protection and the state of the signals will be read from the PLC over a field bus.

2.7.2 Choice of interface hardware

Several possible hardware standards could be used at the interface level for the IOCs, including VME, Compact PCI and industrial PCs. All of these are compatible with EPICS and supported by the VxWorks [15] operating system. Of these standards only VME is likely to be maintainable throughout the life cycle of the DIAMOND project. VME provides the greatest flexibility in term of available interfaces and mechanical forms. It is clear that VME would have to be used in some of the interface requirement, possibly on the beam position feedback system. For these reasons the interface layer has been standardised on VME and in particular the VME64x variant.

2.7.3 Choice of Processor Board

The processor used in the IOCs will be a Power PC VME board chosen from the then-current range of the market-leading manufacturer in the area. Reliability of the processor board, robustness of the board support pack and life cycle are crucial to initial availability of the DIAMOND control system and long term reliability.

Initial work has used a PPC processor board based on the MPC604 processor, which is current today. According to the road map provided by Motorola this will not be current in 2006, but processor boards now being introduced based on the MPC7400 processor will be and these are the likely choice for the DIAMOND control system

2.7.4 IOCs Physical

Each IOC will consist of a VME64x crate containing a number of various IO boards depending on the equipment being interfaced. The VME64x extensions provide a number of advantages over VME32, including improved electrical interface, geographical addressing and hot-swap capability. Extra connectors provide a large number of spare pins, so IO signals will be brought through the P0 and P2 connectors on the VME64x backplane.

In a typical IOC the processor board will reside in the first slot. This will act as the VME bus controller and host the EPICS server software. It will have a connection to the controls network and a serial port for the system console.

An event receiver module will occupy the third slot. This enables the IOC to use an event timing signal, to time stamp transactions and to synchronise acquisition or control with other IOCs.

The second slot of the IOC should remain unoccupied so that a VME bus analyser can be plugged in for debugging or performance analysis. The remaining slots will be available for IO boards, which will depend on the equipment being interfaced to a particular IOC.

2.7.5 Choice of IO interface

Many of the IO signals will be handled by IP cards. These will reside on VME64x IP carrier boards, which can accommodate up to 4 IP cards and occupy 1 VME slot. The I/O will be performed using a variety of IP modules. The cabling to the carrier boards will be via a transition module in the back of the VME64x crate. The transition modules will connect to DIN rail mounted interfaces in the racks, where the plant wiring will be terminated. This is shown in Figure 2.7-1.

A number of accelerator and HEP experiments make extensive use of IP modules. This has led to there being very good device support from EPICS. Because the IP is an open standard, there are many companies [16][17][18] with specialist knowledge in these areas, providing IP modules for just about any I/O interface.

Due to the size of the IP modules signal conditioning and isolation will be provided

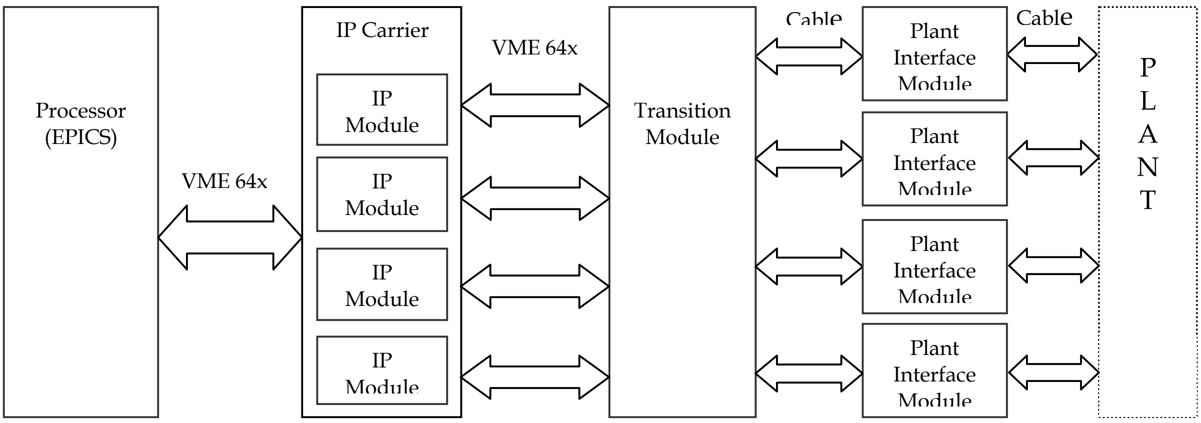

Figure 2.7-1 Block Diagram of IO using IP Modules

on the transition modules. This maintains the modularity of the IP carrier and modules, reducing system cost and component count. However, as isolation and conditioning are specific to I/O types, the IP modules and transition modules must be carefully matched, restricting system modularity. This ensures that for a signal fault the IP Carrier and modules are protected, though the fault is dealt with inside the IOC hardware.

The I/O requirements for DIAMOND will be met as follows:

(a) Analogue

DAC and ADC IP modules are available from many manufacturers, covering all conversion rates and bit counts that will be required. There are also several manufacturers that supply DACs and ADCs with past conversion memories, of varying sizes, which would allow for the retrieval of post mortem data.

(b) Digital

Digital I/O IP modules are available from many manufacturers, both as exclusively input modules, exclusively output modules and as mixed I/O. There are options for various input and output voltage levels, including various logic levels and relay driven.

(c) Serial

Serial IP modules are available from many manufacturers covering a variety of standards, such as RS232, RS422, RS485, etc.

(d) Field Bus

The DIAMOND control system has the potential requirement to support several field bus protocols. There are IP modules available to support most field buses including Profibus, CAN bus and GPIB. The newer field buses operate over networking protocols so require a network connection. Some Power PC processor boards have a secondary network connection that would enable the network field bus to be physically separate from the controls network.

(e) Motion Control

There are several systems that require the use of motors (usually stepper motors). Several IP module manufacturers make IP modules that specialise in motion and motor control. Some stepper motor manufacturers also supply their motors with serial interfaces. Direct VME64x motion controllers are also available.

(f) Specialist

There are parts of the DIAMOND control system, which will require specialist I/O interfaces, such as potentially magnet power supplies. Some of these interfaces are available in a VME standard, which allows them to be controlled directly from the IOC. However, there are also specialist IP modules available with a multitude of functionality, including from DSPs and FPGAs that will allow the creation of specialist interfaces.

2.7.6 Hot-Swap

The use of VME64x enables Hot-swap functionality which is the ability to unplug a board from a correctly configured and running IOC, replace a defective component, say an IP card, on the board and replace it into the running system. The system will recognise that the board has been removed and replaced and resume normal operation.

Hot-swap capability facilitates increasing overall system uptime, thereby maintaining a high overall MTBF, especially in large installations.

(a) Hot-swap Physical

Electrical protection is taken care of by the VME64x standard. This ensures that boards can be safely removed and inserted into VME crates while the power remains on. Both the crate and the board must comply with the standard. The use of VME64x compliant IP carrier boards will mean that any IP cards can be hot-swapped.

All I/O should be brought through the VME backplane rather than the front panels, to permit access to the boards, and simplify their removal and replacement.

(b) Hot-swap Software

The software must be able to respond robustly to the removal of one or more VME boards and their replacement with functionally equivalent hardware in the same slots. Device drivers will:

- Detect the removal of a card, either by polling or handling an interrupt,
- Trap bus errors caused by attempting to access missing hardware,
- Detect the reinsertion of the board and perform any necessary re-initialisation.

The software will not be required to handle situations where new VME cards are introduced, or cards are moved from one slot to another. In these cases the system will need to be reconfigured, which typically requires at least a re-boot of the IOC.

2.7.7 PLCs

PLCs will be used to perform subsystems control requiring complex logic, such as interlocking and vacuum valve control. Removing the protection logic from the IOC simplifies the design by removing the need for watchdog timers to protect against software failure. It further provides operational flexibility by allowing the plant to remain functional while the IOC is rebooted. This works by holding analogue and digital outputs at their present state and monitoring for fault conditions during reboot.

Communications between the IOC and PLC will be via a field bus. The choice of PLC and field bus will be settled during the detail design phase. PLCs are widely used in other accelerator facilities and there is a large EPICS support for many different field buses and PLC makes.

2.7.8 Interface Requirements of Underlying System

Following from initial discussions with the technical group an assessment has been made of the equipment interfaces required. From experience on the SRS and other light sources, the likely interfaces have been considered for each piece of equipment. The numbers and types of interfaces must be considered a baseline guide as inevitably, as designs for the underlying systems evolve, further equipment that requires controlling or monitoring will be identified and the requirement will tend to grow.

Table 2.7-1 shows the control system interface requirement broken down into each of the underlying subsystems and its likely interface. The table columns are as follows:

- System : The DIAMOND subsystem
- No. Sys: the number of subsystems needed to support DIAMOND
- Equipment: The equipment required to support the subsystem
- Per Sys: The numbers of equipment in the subsystem
- Add: The number of addition I/O required by the subsystem
- Total: The total amount of I/O lines needed for the subsystem
- IF Type: I/O the interface types required for the equipment in the subsystem

Table 2.7-1 Interface number and types for the technical systems

System	**No. Sys**	**Equipment**	**Per Sys**	**Add**	**Total**	**Possible IF**
SR Arc Vacuum	24	Ion Pump	14		336	S
	24	TSP Pump	2		48	S or D
	24	Neg Pump	4		96	S or D
	24	Pirg	4		96	S
	24	IMG	4		96	
	24	Valve	2	1	49	D via PLC
SR Straight Vacuum	24	Ion Pump	2	1	49	S
	24	TSP or Neg Pump	2	1	49	S or D
	24	Pirg	1	3	27	S
	24	IMG	1	3	27	S
SR Roughing Vacuum	24	Turbo Pump	3		72	D Via PLC
	24	Pirg	2		48	S
	24	IMG	2		48	S
	24	Valve	4		96	D via PLC
SR Magnet	1	Dipole	1		1	A,S
	24	Global IL	1		24	BP
	24	Quad	10		240	A,S
	24	Sext	7		168	A,S
	24	Multipoles on Sext	35		840	A,D
	24	Fast Correction	8		192	A,D
	24	Water Flow Meters	5		120	A
SR Vessel Protection	24	Vessel Protection Module	1		24	BP
	24	Global Il Module	1		24	BP
	24	Beam Loss Monitor	20		480	C
	24	Water Flows	5		120	A
	24	Temp Transducers	24		576	A
Insertion Devices PM	7	Motor Drive	1		7	S
	7	Trim PSU	1		7	A,D
Insertion Devices SC	1	Main PSU	1		1	S
	1	Trim PSU	1		1	S
	1	Quench protection	1		1	A,D
	1	Cryogenic Plant	1		1	SI
Front End Vacuum	24	Ion Pump	4		96	S
	24	TSP or Neg Pump	4		96	S or D
	24	Pirg	4		96	S
	24	IMG	4		96	A,S
	24	Valve	3		72	D
	24	Water Flows	5		120	A
	24	Radiation Shutter	2		48	D via PLC
SR Diagnostics	24	EBPM I Tech	2		48	BP

PART D: INFRASTRUCTURE

System	No. Sys	Equipment	Per Sys	Add	Total	Possible IF
	24	EBPM Begoz Arc	5		120	A,D
	8	EBPM Begoz Straight	2		16	A,D
	24	Photon BPM	2		48	A,S
	2	Total Current Monitor	1		2	A
	1	Screen and Camera	1		1	S,V
	1	Tune Measurement	1		1	G
	1	Charge per bunch	1		1	BP
	1	Streak Camera	1		1	SI
	2	Jaws	1		2	S
SR RF	3	Amplifiers Fundamental	1		3	A,D,S
	3	Cavities Fundamental	1		3	A,D,S
	1	Low Level RF	1		1	BP
	1	DC Supply	3		3	A,D,S
	1	Cryogenic Plant	1		1	SI or O
	1	Harmonic Cavities	1		1	A,D,S
SR Personnel Safety	1	Safety System Interface	4		4	BP
SR Radiation Monitors	1	Monitors	48		48	A,S
SR Traverse Feedback	1	Pick up	1		1	A
	1	Signal processing	1		1	BP
	1	Amplifier	1		1	A,D
	1	Kicker	1		1	A,D
Linac Diagnostics	1	FCM	1		1	A,D
	1	BPM	1		1	A,D
	1	Fast Charge Monitor	1		1	A,D
Linac Vacuum	1	Ion Pump	3		3	S
	1	Pirg	2		2	S
	1	IMG	2		2	A,S
	1	Valve	2		2	D
	1	Radiation Shutter	1		1	D
Linac RF	1	PSU	1		1	A,D,S
	1	RF Amplifier	1		1	A,D,S
Linac and Flight Path Personnel Safety	1	Safety System Interface	1		1	BP
Linac Radiation Monitors	1	Monitors	5		5	A,S
Flight Path Diag	1	Screen and Camera	2		2	S,V
	1	Faraday Cup	2		2	A
	1	FCM	1			
	1	Electron BPM	1		1	A,D or BP
	1	Emmitance Measurement	1		1	SI

System	No. Sys	Equipment	Per Sys	Add	Total	Possible IF
Flight Path Vacuum	1	Ion Pump	6		6	S
	1	Pirg	1		1	S
	1	IMG	2		2	A,S
	1	Valve	1		1	D
	1	Radiation Shutter	1		1	D
Flight Path Magnet	1	Dipole	2		2	A,S
	1	Quad	4		4	A,S
	1	Corrector	8		8	A,D
Transfer Path Diagnostic	1	Screen and Camera	2		2	S,V
	1	Faraday Cup	2		2	A
	1	Electron BPMS	1		1	A,D or BP
	1	PCM	1			A,D
	1	Emmitance Measurement	1		1	SI
Transfer Path Vacuum	1	Ion Pump	6		6	S
	1	Pirg	1		1	S
	1	IMG	2		2	A,S
	1	Valve	1		1	D
Transfer Path Magnets	1	Dipole	6		6	A,S
	1	Quad	21		21	A,S
	1	Corrector	8		8	A,D
Booster Vacuum	1	Ion Pump	55		55	S
	1	Pirg	8		8	S
	1	IMG	8		8	A,S
	1	Valve	5		5	D
Booster Magnet	1	Dipole	1		1	A,S
	1	Quad	2		2	A,S
	1	Sext	1		1	S
	1	Steering	32		32	A,D
Booster RF	1	Cavity	1		1	A,D,S
	1	Amplifier	1		1	A,D,S
	1	DC Supply	1		1	A,D,S
Booster Diagnostics	1	Electron BPM	24		24	A,D or BP
	1	Fast TCM	1		1	A
	1	Screen and Camera	1		1	S,V
	1	Tune Measurement	1		1	G
	1	Jaws	1		1	S
Booster and Transfer Path Personnel Safety	1	Safety System Interface	1		1	BP
Booster Radiation Monitors	1	Monitors	5		5	A,S

System	No. Sys	Equipment	Per Sys	Add	Total	Possible IF
Timing	1	Master Timing Generation	1		1	BP
	1	Timing Distribution and Delay	20		20	BP
	1	Event generation	1		1	BP
Injection extraction and Kickers	1	Booster Injection PSU	1		1	A,D
	1	Booster Extraction PSU	1		1	A,D
	1	Booster Kicker	4		4	A,D
	1	SR Injection PSU	1		1	A,D
	1	SR Kicker PSU	4		4	A,D
Conventional Services	1	Air Con Building	1		1	SI or O
	1	Service Water Temp	1		1	SI or O
	1	Mains Monitoring	1		1	SI or O
Wide Band Multiplexer	1	Input Channel Selector	18		18	D
	1	Output Selector	2		2	D
Analogue Multiplexer	1	Channel Selector	38		38	D
Status Monitor	1	Generation	1		1	SI
Beamline Vacuum	7	Pirg	8		56	S
	7	IMG	8		56	S
	7	Ion Pump	8		56	A,S
	7	Valve	8		56	D
	7	Radiation Shutter	1		7	D
Beamline Motion	7	Motor Drives	20		140	S
Beamline COSH	7	COSHH Systems	1		7	S
Beamline Personnel Safety	7	Safety System Interface	1		7	BP

Table 2.7-2 Key to interface types in Table 2.7-1.

IF Code	Interface Type
A	Analogue
D	Digital
BP	Back plane
V	Video
G	GPIB
S	Serial, RS232, RS422, RS485 or Field Bus
SI	System integration, e.g. Labview
O	Sub System Protocol, e.g. OPC
SC	Scaler, counter timer

From Table 2.7-1 a summary of the required numbers of interface can be produced and these are shown in

Table 2.7-3 and shows a total of 5633 interfaces. These are then mapped on to IOCs as shown in Table 2.7-4. This gives a total requirement for 212 IOCs to support the accelerator and the first 7 beamlines.

Table 2.7-3 Summary of Interface requirements

Equipment	**Linac**	**Transfer Paths**	**Booster**	**Storage Ring**	**Beam Lines (7)**	**Total**
Main Magnets	6	27	4	553		590
Steering Magnets	8	8	32	1032		1080
RF	2		3	16		21
Vacuum	21	10	76	1761	231	2099
Diagnostics	9	7	28	240		284
Pulsed PSUs			6	5		11
Personnel Safety	1		1	4	7	13
Vessel Protection				744		744
Beam Loss Monitoring				480		480
Radiation Monitoring	5		5	48		58
Ids or Motors				18	140	158
Miscellaneous				88	7	95
Totals	52	52	155	4989	385	5633

Table 2.7-4 Mapping of interface requirements on to IOCs

IOCs	**Linac**	**Transfer Paths**	**Booster**	**Storage Ring**	**Beam Lines (7)**	**Total**
Main Magnets		1	1	25		27
Steering Magnets		1	1	24		26
RF	1		2	5		8
Vacuum	1	2	4	48	7	62
Diagnostics	1	2	4	25		32
Pulsed PSUs			1	1		2
Personnel Safety	1			2	2	5
Vessel Protection				24		24
Radiation Mons	1			2	2	5
Ids or Motors				7	7	14
Miscellaneous				5	2	7
Totals	5	6	13	168	20	212

2.8 CONTROL SYSTEM SUPPORT SERVERS

In addition to EPICS IOCs and EPICS OPIs, a third class of machine can be identified; the Control System Support Server. These machines will host any files, databases, environments etc. which cannot be held on IOCs (since these are memory-only systems) and are inappropriate for storage on the OPIs themselves (for example, cumulative files requiring large amounts of disc space and appropriate media back-up facilities). For each Control System Support Server, a hardware platform and configuration appropriate to its function will be selected; this might take into account, for example, the need for RAID disks, disk shadowing or other fault tolerance features.

In general, the design philosophy adopted will be to allocate one Control System Server per function, so that the loss of one server does not disable a significant number of Control System facilities. The detailed design phase will further consider ways of minimising any period when the Control System has to run with reduced facilities due to loss of any one Control System Support Server.

Control System Support Servers will include :

- IOC Boot Servers, containing boot images and database text files for a each IOC.
- Development Servers (Section 2.5.1), containing the Application Development Environment.
- Relational Database Server, used to hold the Control System Relational Database Management System (Section 2.10).
- Log File Server, used to hold files for the Alarm Log and Operations Log windows of the EPICS ALH package and for files collected by the EPICS CMLOG program.
- Archive Server, used to hold data files for all history activity, from short-term back history through to long-term archive data.
- Accelerator Model File Server, containing configuration files and data files relating to the Accelerator Model.
- External Access Gateway Server, containing configuration files and data files relating to Control System access from outside of the trusted Control System network.

Requirements for any additional Control System Support Servers will be defined during the detailed design phase.

2.9 NAMING CONVENTION

2.9.1 Introduction

A binding device naming convention [19][20][21] is required not only to allow control system software to reference precisely and unambiguously accelerator components, but also to facilitate communication between the various technical groups and specialists responsible for design, construction and operation of the accelerator complex.

It is anticipated that the names will be used extensively on engineering drawings, in design documents, in control system software, in informal and formal discussions, in databases and on device labels.

It is hoped that the naming convention will provide a concise and consistent way of identifying an individual component. Also, through the use of a hierarchical structure, the name should provide some guidance as to the physical location of the device.

Ideally, device names should not be unduly restricted by the underlying software and systems, but in reality careful consideration must be given to ensuring compatibility with EPICS which has been selected as the control system toolkit for DIAMOND.

The device name will consist of a maximum of 20 alphanumeric characters. It can be broken down into 4 or 5 sections. The 'dash' character "-" is used to delimit the sections and is the only non-alphanumeric character allowed.

2.9.2 Device Name Format

The format is as follows:

DDD-[SSS]-TTT-CCCCC-NN

where [...] indicates an optional section.

2.9.3 Domain (DDD)

Domain identifies a major area of the accelerator complex. It consists of a minimum of one and a maximum of three characters. The first character must be alphabetic; the remaining characters must be alphanumeric.

This section is mandatory.

Table 2.9-1

DDD	Domain
LIN	Linac
L2B	Linac – Booster transfer line
BOO	Booster ring
B2S	Booster – Storage ring transfer line
SR	Storage ring
FE	Front Ends
BL	Experimental area
CS	Control system
SRV	Services
TST	Test systems

2.9.4 Sub-Domain (SSS)

Sub-domain is used to further sub-divide complex domains such as SR (Storage Ring). Each domain has its own separately defined list of sub-domains. It consists of a minimum of one and a maximum of three characters. The first character must be alphabetic, the remaining characters must be alphanumeric.

This section will be omitted for most simple domains but is mandatory within domains for which sub-domains have been defined.

For the Storage ring domain (SR):

Table 2.9-2

SSS	Sub-domain
Axx	Arc xx
Sxx	Straight xx

For the Front End (FE) and Beamline (BL) domains:

Table 2.9-3

SSS	Sub-domain
Ixx	Insertion device xx
Axx	Upstream dipole xx
Bxx	Downstream dipole xx
Cxx	Upstream dipole xx IR
Dxx	Downstream dipole xx IR

2.9.5 Technical Area (TTT)

Technical area defines the technical sub-system to which the device belongs. It can also be used to help identify the group or person responsible for the device. It consists of a minimum of one and a maximum of three characters. The first character must be alphabetic, the remaining characters must be alphanumeric.

This section is mandatory.

Table 2.9-4

TTT	Technical Area
MAG	Magnets
VAC	Vacuum
PSU	Power supplies
RF	RF systems
DIA	Diagnostics
AL	Alignment
PSS	Personal Safety
RAD	Radiation Safety
TIM	Timing system
H20	Water
MON	Monochromator
MIR	Mirrors
CRY	Cryogenic systems
AIR	Compressed air
GAS	Gases
MOV	Movements (motors)
EA	Experimental apparatus

2.9.6 Component (CCCCC)

This section defines the particular type or style of device. Each Technical Area will have a separately defined list of valid component names. It consists of a minimum of one and a maximum of five characters. The first character must be alphabetic, the remaining characters must be alphanumeric.

It is anticipated that each technical group will define their own list of names in agreement with the Controls group.

This section is mandatory.

2.9.7 Identifier (NN)

A two-digit number identifying a particular instance of a device. A leading zero (0) must be used where necessary.

Although there is no need to specify devices in numeric order, this practice is to be encouraged whenever possible.

This section is mandatory

2.9.8 EPICS Record Names

EPICS record names are constructed from device names by appending a record specifier (and optionally, a sub-system name) as follows:

DEVICE_NAME[:SUB_SYSTEM]:RECORD

where [...] indicates an optional component.

sub_system is intended to allow the use of hierarchy within the EPICS IOC database (i.e. CapFast hierarchy). Only one level will be allowed.

Both ***sub_system*** and ***record*** are no more that three characters in length. However, if the ***sub_system*** component is not used then ***record*** may be up to seven characters in length.

2.9.9 CDEV Name Space

CDEV has its own device name space that is defined in a DDL file. This name space is independent of the EPICS name space and can, in principle, use a completely separate naming convention. However, this temptation should be avoided at all costs and CDEV device names must use this naming convention without exception. This has the extra advantage that the substitution operator (<>) can be used in the DDL file to generate automatically an EPICS PV name from a device name.

It is anticipated that DDL files will be automatically generated from the system Relational Database.

2.9.10 Examples

Table 2.9-5

SR-A01-VAC-IONP-01	Storage Ring Arc 1 Ion Pump controller No. 1
LIN-PSU-FQUAD-02	Linac F-Quadrupole 2 power supply
FE-I01-VAC-VALVE-01	Insertion device 1 front end Vacuum valve No. 1
BL-I01-DIA-TVM-02	Insertion device 1 beamline TVM No. 2
TST-VAC-PIRG-03:SMON	Test system Pirani vacuum gauge No 3. Status monitor EPICS record

2.10 Relational Database

2.10.1 Introduction

The EPICS toolkit is based around the concept of a distributed, real-time database. In other words, information about the current state of the accelerator is collected from many IOCs situated around the accelerator complex. On start-up these IOCs must be provided with a local database file that defines the devices to be controlled, initialisation data, alarm limits, calibration data etc. This database is a simple ASCII text file that is downloaded to the IOC from a server on boot-up. Each IOC requires its own unique database.

Similarly, each IOC database file is usually generated using information from several different sources such as CapFast or VisualDCT schematic drawings or simple text data files.

There is also a large pool of information that is needed by the users and designers of the control system that does not reside within the distributed database provided by the IOCs. This information includes such things as magnet calibration data, lattice element survey offsets, device serial numbers, configuration data for application software, and so on. The standard EPICS distribution provides no mechanism for managing or organising this large amount of data.

Most large accelerator projects over the last 10 years that use an EPICS control system have attempted to solve this problem by storing all the information in a relational database [22][23][24]. However, because of the many intricate relationships between the different types of information to be stored, the design of these databases is very complex. The use of relational databases within the EPICS community has not yet matured enough for there to be a "standard" solution available so some original design work is involved.

2.10.2 User requirements

The following user requirements have been identified:

- The database will need to be a central repository, storing all the IOC database configuration data.
- Configuration scripts should be able to be retrieved and modified, then changes saved in the RDB.
- Physical elements, such as devices and logical elements in the control system will need to be stored and modified in the RDB.
- EPICS data (records/fields) will need to be stored and modified in the database.
- Alarm limits will need to be stored and modified in the database. These can be switched on and off in the database and the severity can be changed.
- A user should be able to find all records common to a particular device type.
- Devices should have data identifying who installed/checked them and when.
- RDB security will be dealt with in the database itself.
- Users will need user access rights to restrict them from accessing certain RDB areas. Access categories should include read-only, read-write and no access.
- Information and results from accelerator physics experiments will need to be stored for reference and future experiment design. The tables storing the results will have no direct effect on, or link to, the configuration data.
- The database should be made available over the Internet; this could allow remote users to control the RDB or be restricted to read-only access.
- It should be possible to add, delete or edit data in the RDB.
- The RDB should provide information to all users, with access to certain areas for only those actors who require it.
- Data about alarms and signals should be archived in the database when there is a change or it is time for an update.

- The database should be straightforward to use, with simple ways of entering and extracting data, for example using forms and barcodes.

- A user should be able to query the database to find out any information that is stored, provided he or she has appropriate access rights.

- The database should be able to produce reports, e.g. exception reports when something extraordinary occurs.

2.10.3 Implementation Details

(a) Choice of Relational Database Management System

The estimated size of the database and the anticipated transaction rates are moderate compared to many database systems in use today. Most commercial and many Open Source RDBMSs will easily be able to handle the requirements. However, ORACLE [25] has been chosen for DIAMOND for the following reasons:

- ORACLE is the market-leader RDBMS

- It is already in use at several EPICS-based accelerator laboratories

- ORACLE expertise is widely available

- ORACLE includes a very comprehensive suite of management, application design and web-publishing tools.

(b) Storage Requirements

Preliminary estimates show that the total database size will be in the region of 2 – 3 GBytes. This is based on the assumption that the database will contain all IOC, record, field and device information as well as physics parameters and application configuration information. It does not include any allowance for storing historical archive data, experimental data sets or graphical information.

(c) Data Integrity Checks

The database design will include many internal data integrity checks. This will help to ensure that the data is both valid and consistent. Careful thought will be given to the choice of data type for each field and to the specification of valid ranges of values. Data validation should be handled within the database whenever possible. Validation by database application software will also be required in more complex cases.

(d) Database Security

Security will be built into the database. It will be possible to restrict access to each table to certain users or groups of users. The user and group lists will be defined within the database itself and access will be managed by the RDBMS.

(e) Backup and Archival

Adequate provision must be made for online backup and long-term archival of the database. Given the anticipated database size, DLT drives will be perfectly adequate for online backup. For long-term archival database snapshots can be stored to an optical storage media such as DVD-RAM.

2.10.4 Application Integration

It is important to provide a simple, co-ordinated mechanism that will allow application software to access both dynamic data from the IOCs and static data from the relational database. Subsection 2.6.8 adresses this through the adoption of CDEV as the standard API for control system applications. CDEV provides a common API that can be used to retrieve either dynamic or static information from an IOC or the RDB. It was developed to work with EPICS and has been adopted by several other accelerator labs. It also provides an object-oriented view of devices.

2.10.5 User Interface tools

A suite of interface tools will be required to perform the following tasks:

- Import CapFast/VisualDCT schematics into the RDB
- Import IOC ASCII database files into the RDB
- Create IOC ASCII database files from the RDB
- Create CDEV DDL files from the RDB
- Import Microsoft© Excel spreadsheets into the RDB
- Modify/Edit RDB data
- Produce reports from the database
- Execute user-defined queries on the database

These tools can be provided using either conventional programming languages such as C or Java, a scripting language (Tcl), a proprietary database application development tool such as ORACLE Forms, or a web-based interface.

2.11 SECURITY

2.11.1 Philosophy

The design of the Control System will ensure maximum ease of use within the control room and equipment hall. To achieve this, there will be very few security measures in place for users in these locations. Where security is required within the control system bounds the EPICS access control mechanisms will be used. Access to the Control System from outside these locations will be strictly controlled and managed by use of physically separate networks, a firewall mechanism and dial-in modem access.

2.11.2 EPICS Access Security

The EPICS package incorporates a sophisticated, rule-based access security mechanism, which has been designed to have minimal performance effect on database read/write times. Access security can be applied to each individual record field and is configured in an Access Security Configuration File. Access security can be enabled and disabled via an entry in the IOC download files, each of which will be held on a server in a secure, password-protected account.

As an example of what is possible with this mechanism, the following security policy for a Linac is implementable:

A) Anyone can have read access to all fields at any time.

B) Linac engineers, located in the injection room or control room, can have write access to most level 0 fields if the Linac is not in operational mode.

C) Operators, located in the injection room or control room, can have write access to most level 0 fields at any time.

D)The operations supervisor, linac supervisor, and the applications developers can have write access to all fields.

E) Most records use the above rules, but a few (high voltage power supplies etc) are placed under tighter controls. These follow rules (A) and (D), but not (B) and (C).

EPICS access security will remain disabled during the development and commissioning stages of the Control System, to ensure that development and commissioning teams are unhindered by security issues. During this phase of the project, a set of access security rules will be clearly defined, and these will be implemented when the Control System is ready to enter the operational phase.

Access security is on a per-IOC basis. Therefore, if it proves necessary, it will be possible to run the Control System with commissioned IOC's under access control and non-commissioned IOC's free of access control to prevent delays to development and commissioning teams.

EPICS access security does not apply to IOC test commands. Therefore, support staff can still use these commands for database reads or writes as part of any diagnostic or debugging investigations, even on an IOC that is under access control. Separate security measures (e.g. Firewall, dial-up security, password-protected accounts) will be used to ensure that only authorised staff can gain direct access to IOC's at the VxWorks level to use these test commands.

2.11.3 Firewall

All DIAMOND Control System components will reside on a dedicated network – in security terms, this is the trusted network. All other networks are collectively referred to as the untrusted network. The trusted network will be separated from the untrusted network by a Stateful Multi-layer Inspection firewall installed on a standalone PC. This type of firewall currently offers the best combination of security and minimal performance effect.

Key aspects of this configuration include:

- The firewall PC should run a dedicated minimal operating system, sufficient only for supporting the firewall itself. This minimises the danger of the firewall being compromised.

- It will be possible to read Control System parameters from the untrusted network via web interfaces or similar mechanisms.

- Software (running on the untrusted network) obtaining data from the Control System must communicate through a known tcp port, so that a security policy can be allocated to that port on the firewall. In practice, this will mean that data requests will be channelled through a custom-written sender/receiver software combination; the sender program will reside on the trusted network side of the firewall and the receiver program will reside on the untrusted side. Transactions between the sender and receiver programs will consist of well-defined commands requesting data from the Control System. Such commands will be parsed by the sender program, and any illegal commands rejected. Commands will be defined such that no command can request a volume of data sufficient to cause any overload on the DIAMOND Control System itself.

- Ideally, the firewall should be configured to allow office-based support staff with suitable authorisation to have password-protected access to control room facilities via an X-Windows emulator such as Exceed. If provided, this type of access must be configured with a time-out, such that an auto logout occurs after a period of inactivity.

External Internet access will not be permitted from the DIAMOND Control system. Instead, a Control System Intranet will provide operators with access to documentation and other content relevant to control room staff.

2.11.4 Dial-in modem access

Support staff will require dial-in access to the DIAMOND Control System, which will bypass any firewall. Only trusted users (i.e. those authorised to access the trusted network) will be allowed this type of access.

Either of the two following strategies could be used to provide secure dial-in access:

- Dial-back modem scheme. When a call is placed to a dial-up, the modem responds by hanging up and dialling back to a pre-programmed telephone number.

- Dial-in Authenticator scheme. With this security solution, a dial-in connection can only be established by successfully entering the security code for an authenticator device. The authenticator is a hardware device attached to the dial-in modem. Various authenticator mechanisms exist; some rely on a security code being entered via the initiating dialler, others rely on a separate hardware "key" being used. In the latter case, the hardware "key" matches up to one authenticator only

The latter solution has some cost implication, but offers improved security.

2.12 PHYSICAL STRUCTURE

2.12.1 Control Instrumentation Areas

The control system will interface to the technical system at 48 CIAs. The CIA will house the control system interface layer, the IOCs and the instrumentation of the technical system. The identified CIAs, racking and space required are shown in Table 2.12-1.

Table 2.12-1 Proposed Control Instrumentation Areas

Ref No	Control Instrumentation Area	Number of	Racks Identified	Rack Space Required
1	Linac and LTB	1	18	20
2	Booster Synchrotron	6	32	60
3	Booster RF	1	5	10
4	Booster Power supplies	1	3	10
5	BTS	1	5	10
6	Storage Ring Instrumentation	24	394	480
7	Storage Ring Dipole Power Supply	1	2	5
8	Storage Ring RF	1	11	20
9	Technical Service Plant Room	1	3	5
10	Cryogenic Services	1	5	10
11	Physics Room	1	7	10
12	Main control room	1	1	
13	Control System Computer room	1	4	10
14	Beamlines	7	48	70
	Total	48	519	721

2.12.2 CIA Physical requirements

The CIAs will be located in close proximity to the equipment they are controlling. For the SR this is around the inner shield wall. Each CIA will be contained within a partition-type enclosure with a false floor and air conditioning. This ensures effective cable management, the local cooling necessary for computer systems and maintenance of a clean environment, and reduces noise to the overall building from fans within the equipment.

2.12.3 Main Control Room

The MCR will be where the DIAMOND facility is controlled. The operator interface will be through workstations on the DIAMOND control system. There will be a limited requirement for hardwired interfaces to technical system such as personnel safety lockout keys and wide band connection to monitor RF signals. The room will support the operations team, staff from the physics groups and staff from technical groups.

2.12.4 Control System Computer Room

The CSCR will house the central computer systems that make up the DIAMOND control system. In this area all signals connecting the central computers to the 48 CIAs will be terminated. Also located in the CSCR will be central systems for generating timing and event information and managing global machine protection. The computer systems in the control room will include the computer servers for the operational control system and servers and workstation for control system development.

2.12.5 Physics Room

The PR will make key signals available to enable experiments to be set up to monitor and characterise aspects of the accelerators. To realise this, access will be provided to the DIAMOND control system and to wide-band (RF) signals from around the accelerators.

2.13 NETWORK

2.13.1 Requirements

The network infrastructure will support the control system to operate the DIAMOND accelerators and beamlines, and provide network connectivity for other computers located in or in close proximity to the accelerators and beamlines e.g. RGA monitors.

The control system network needs to support real-time access to the embedded servers and IOCs, and booting of the IOCs. To preserve real-time performance under varying load condition, the network will feature low utilisation and high network bandwidth.

Reliability of the network will have a direct impact on the operational reliability of DIAMOND. Experiences from other light sources have shown that the control system network should be structured to include an element of redundancy [26]. This will be realised in the fibre optic cables and at the server level. Analysis of network topologies has shown that a star configuration from a switched hub with redundancy has superior reliability to that of a bus or ring structure [27].

2.13.2 Network Services

There will be two physically separate computer networks provided. Each will be distributed to the accelerators and beamlines. The control system network will be used exclusively for control system connections with limited routes to the site network or Internet. A second computer network will provide connections for other computer systems required around the accelerators, which are not integral to the control system. This network will be connected to the site network and hence the Internet.

The network structure will also support the distribution of timing system event information, machine protection signals and monitoring of the IOC crates using the CAN Bus network standard.

2.13.3 Network Physical structure

The network will be structured in star configuration running from the CSCR to each of the 48 CIAs. The CIAs will contain the network services, the control system IOCs and the plant instrumentation. The connection from each CIA to the control system computer room is likely to consist of one 12core multi-mode FO cable and one 12 core single-mode FO cable.

Multi-mode FO cables support current network standards with single-mode providing an upgrade path to multi Giga bit standards. The final mix of multi-mode fibres and single-mode will be decided nearer the installation date, taking into account the then-current state of technology.

The hub of the star structure will be in the CSCR and for the computer networks will be two network switches. All cables, fibre optic and copper, will be terminated in patch panels in racks to provide effective management and hence flexibility in connecting to the network switches and other hardware. This will also contain management of the network, the IOC crate monitoring, event distribution and machine protection.

At each CIA, a rack will contain the network distribution components and termination. The distribution will consist of two network switches, for the control network and secondary network, fan-out of the event signals and translation of the CAN bus from FO to copper signals. To manage distribution, copper and FO, cables from each of the CIA racks will be terminated in patch panels.

To manage the network load, from multiple IOCs booting concurrently, and to give redundancy in the boot process, each network rack will contain a boot server for the IOCs local to that area.

To enable remote access to each of the IOCs through the console port, each network rack will contain a terminal server. The terminal server will be accessed through the secondary (non control system) network. This ensures access to the IOC console ports in the event of a faulty IOC saturating the controls network with traffic.

Local connections will be provided to Ten Base T wall sockets around the accelerator enclosures.

Connecting each control system instrumentation rack to the network distribution rack will be four cores of multi-mode fibres and four Ten Base T copper connections. This arrangement provides for a maximum of two IOCs per rack. Each IOC will be provided with a network connection, console connection and event signal.

2.14 EQUIPMENT PROTECTION

2.14.1 Interlocking Structure

The control system will provide interlocking and protection of equipment where required. ***This excludes personnel protection, which is a separate system that is monitored by the control system.*** Protection will be classified as High Integrity Interlocks, Routine Interlocks or Prudent Limits.

(a) High Integrity Interlocks

High Integrity Interlocking will be used to prevent equipment from behaving in a manner that could lead to substantial damage of property or loss of operation of the facility. It will be based on hard-wired logic with no software component. For example, High Integrity Interlocking will be used to protect the vacuum vessel and dipole magnets, on the SR and possibly the Booster, from loss of cooling water.

(b) Routine Interlocks

Routine Interlocking will be used to protect equipment from damage, for example to prevent a vacuum valve being opened without good vacuum on both sides. This will be based on commercial PLCs and will use software to define the logic. Realising the interlocking in PLCs, as opposed to the IOCs, removes the need to protect IOC outputs in the event of an IOC crash, so improving reliability. Through the use of warm rebooting, it also enables an IOC to be restarted while maintaining its outputs so as not to disturb the operation of the equipment. It further provides a level of isolation from the operational parameters of the CS, ensuring that changes are not overly easy to make, but maintains flexibility to enable technical group to make changes where required.

(c) Prudent Limits

The third level of protection, Prudent Limits, will define operational limits and bounds for equipment. For example, it would be used to ensure that a power supply is only switched on at minimum current, to define the maximum operating current and rate of change of current. This will be provided by the logic in the IOC. Changes to these limits may be required when operating the accelerators in a non-routine mode. This requires that changes to prudent limits can be made through control applications.

2.14.2 Global interlocking

From around the accelerators there is a requirement to receive interlock signals and bring them together to protect equipment. Two global interlock signals have been identified for each of the SR and Booster. These are RF permit (RFP) and dipole permit (DPP).

RFP will be produced from the water flow interlock on the storage ring vessel and front ends, the temperatures from radiation-sensitive components, and electron beam positions through IDs. The RFP will operate by removing the RF to dump the beam in the event of a fault condition.

The DPP will be produced from dipole magnet water flows and temperatures. The DPP will operate by switching off the dipole power supply.

A RFP and DPP will be produced in each cell, most probably in custom VME modules using hardwired logic, as they will be a class of High Integrity Interlocks. The modules will enable monitoring of input interlock through the IOC. All interlock transitions to a fault condition will be latched to catch momentary disturbances.

The outputs of the cell modules will be pulse streams, which will be sent to a central area, the CSCR, over fibre optic cables where they will be concatenated to produce the global permits. The overall management being carried out by VME modules provides monitoring and latching of the states of the inputs from each cell. Figure 2.14-1 shows a block diagram of the system. Such a scheme is used on CEBAF for fast protection [28], in which a 5MHz pulse stream is used to transmit interlocks. This gives a response time of 40µsec.

The use of a simple pulse stream provides for a known response characteristic, good electrical immunity and fail safe operation. The presence of the pulse stream indicates a good interlock, while absence of the pulse stream, either as a steady high or low, indicates a failure of an interlock or a fault in the system. The use of the pulse stream makes the signal suitable for distribution using balanced line differential signals, coaxial cables but most likely fibre optics. The intention is to use the network fibre optic infrastructure from each control instrumentation area to and from the CSCR.

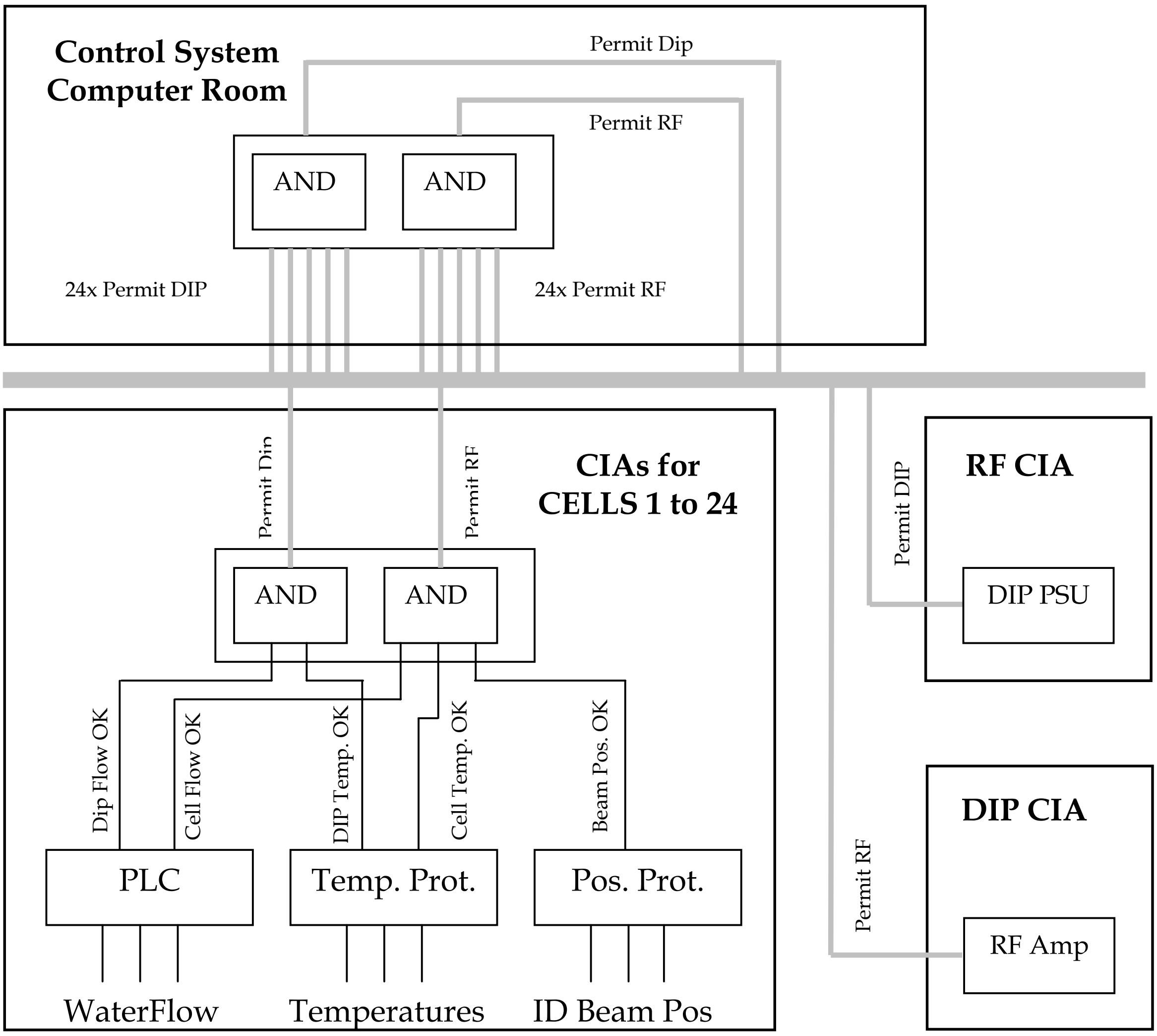

Figure 2.14-1 Block Diagram of Global Interlocking

2.15 TIMING SYSTEM

2.15.1 Introduction

The Timing System is required to provide control of the beam transfer from the electron source to the storage ring and provide diagnostic equipment and beamline equipment with synchronisation signals. The most recent light sources [29] have made use of commercial equipment and built upon equipment designed by other light sources, often in collaboration with industry; it is envisaged that the same approach will be adopted for this project.

2.15.2 Fast Timing

The task of a timing system is to synchronise all the relevant components in an accelerator complex. One part of this task is to control the injection by triggering the particle source and firing the transfer line components like injection and extraction pulsed magnets at the correct times. Also, beam diagnostic components such as beam position monitors and current transformers have to be synchronised to the passage of the beam. This has to happen with fine time resolution, to RF frequency clock precision and low jitter, and is termed Fast Timing.

2.15.3 Event System

Other tasks for the timing system are related to synchronising components where the resolution is more relaxed, for example triggering the magnets for an acceleration ramp, triggering operational sequences such as the filling of the storage ring, BPM acquisition, feedback timing, insertion device control, and supplying the distributed control system with time synchronisation for control and correlation of data. The time resolution for these tasks is less demanding; these tasks are often termed Events. Event Signals will be produced to storage revolution frequency precision with predictable jitter.

2.15.4 Timing System Components

To realise the accelerator timing system it is important to consider what has been used at other recently constructed sources and the integration into the EPICS control system. The time-stamp system already exists within the EPICS infrastructure and can be used in conjunction with the Event System which was developed at the APS [30] and enhanced by the SLS. This APS/SLS Event System can be used to realise all slow timing requirements. The Event System is fully EPICS compatible and the required VME modules are available.

Table 2.15-1 SLS version of the APS Event System Specification

Events	8 Bit code – 255 events
Resolution	20nS
Event TX Trigger	Hardware Input, Software, Event Ram Clock.
Event RX Output	Hardware Output, Software (EPICS record process)
Transmission Medium	Gigabit Ethernet

The requirements for fast timing are more specific to a particular accelerator dimensions and operation. Two options are available for the hardware for fast timing, the KEK TD4V as a VME module delay generator and the Stanford Research DG535 as a self-contained instrument. Each is available with EPICS drivers to provide the controlled delays.

Table 2.15-2 Fast Timing hardware options

	KEK TD4V	Stanford Research DG535
Form	VME 6U	Bench / Rack mounting
Delay	16 Bit / RF clock	0 to 1000 S – 5pS steps
EPICS Support	Yes	Yes via GPIB
Channels	1	4
Jitter	4.5pS at 508MHz	<60pS

2.15.5 System Structure

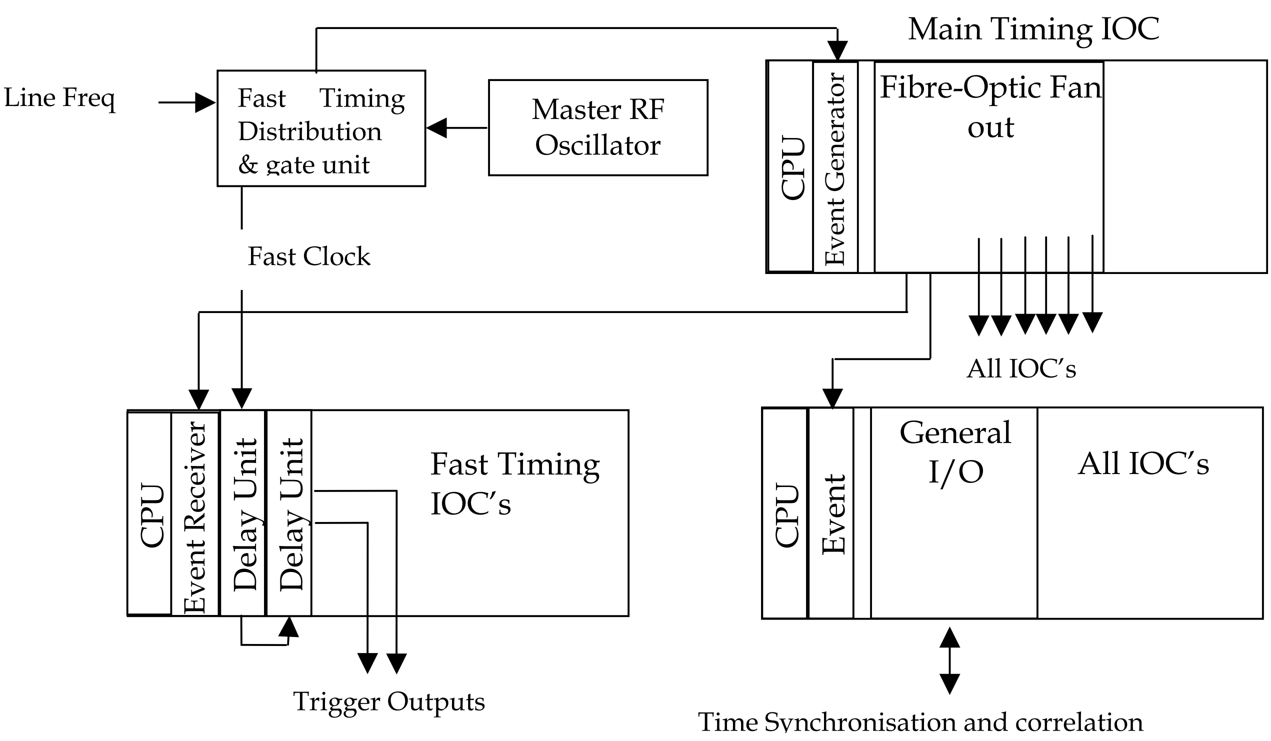

Figure 2.15-1 Block Diagram of the Event System and Fast Timing

Figure 2.15-1 gives an overview of the Event System and Fast Timing control. The Event Generator receives a start signal from the RF clock gated with a line frequency component. Events are then sent to all IOCs' Event Receivers for time stamp synchronisation and to output relevant event signals or process EPICS records. The fast timing IOCs will require a fast clock and trigger derived from the RF source but fast sequences can be initiated upon receipt of an event.

2.15.6 Signal Distribution

The Fast Timing and Event signals will be distributed over fibre-optic cable for reasons of noise immunity and distance capabilities. Further investigation is needed into the delay drifts that could be introduced by the fibre installation from temperature differentials and the standardisation of length-induced delays in a facility the size of DIAMOND.

2.16 BEAM POSITION FEEDBACK

2.16.1 Introduction

A beam position stabilising system is required on a 3^{rd} generation light source to maintain orbit stability to within 10% of beam dimension and to provide dynamic correction of low-frequency orbit disturbances. The proposals presented here are very much based on the work on the APS [31], ALS [32], and SLS [33].

2.16.2 Global Feed Back

The feed back system will use measurements of the position of the electron beam in the storage ring and the photon beams in the beamline front-ends. This information will be compared against a reference orbit and the error used to calculate desired corrections to be applied to corrector magnets in the storage ring.

The response matrix relates the effect of small changes in corrector magnet fields to the resulting changes in the particle beam orbit as measured at chosen BPMs. By inverting the response matrix the relationship that maps orbit perturbations to changes in corrector magnet fields is obtained. For small orbit errors, this relationship is assumed to be linear and time-invariant. Different objectives, such as correcting the orbit in an rms sense or correcting specific locations, can be achieved by choice of BPM and corrector locations and by applying different weights to correctors or BPMs when computing the inverse response matrix.

2.16.3 Performance

Two global feedback systems, operating in parallel, are proposed to correct orbit errors on DIAMOND, namely a Slow system correcting DC drift and a Fast system correcting beam disturbances to 100Hz. These systems would use data from both the electron and photon BPMs and operate on either or both of the steering magnet or fast correctors.

Table 2.16-1 Feedback system comparisons

Feed Back	Correcting	Update Rate
Slow	DC drift	0.1Hz
Fast	0.2mHz -100Hz	2KHz

For both systems the BPMs need to be sampled synchronously, which will be achieved using Events distributed to the IOCs.

2.16.4 Slow Correction

The Slow correction will correct the orbit at ten-second intervals, using the desired correctors and BPMs to compensate for slow changes in the orbit. This will maintain the user-steered orbit applied at the beginning of each fill. Communication to the BPM and Steering IOCs will use the EPICS CA communication mechanism.

The slow correction will be realised as a dedicated application running on either a console or a computer server.

2.16.5 Fast Correction

Fast correction is not possible through EPICS CA mechanisms because of insufficient bandwidth. It will be realised at the IOC level on separate feedback processor boards dedicated to this function. This involves partitioning the correction calculation across the 24 Steering IOCs to calculate the correction values for the steering elements local to that IOC. Each steering IOC requires access to all the BPM values, to give flexibility in the correction algorithm. This requires a high speed connection to share data between the 24 BPM IOCs and 24 Steering IOCs. Two potential solutions for this are to use either reflective memory or network broadcasts.

EPICS process variables will be used to control the feedback process, by down loading algorithms to the feedback processors, setting coefficients and update rates.

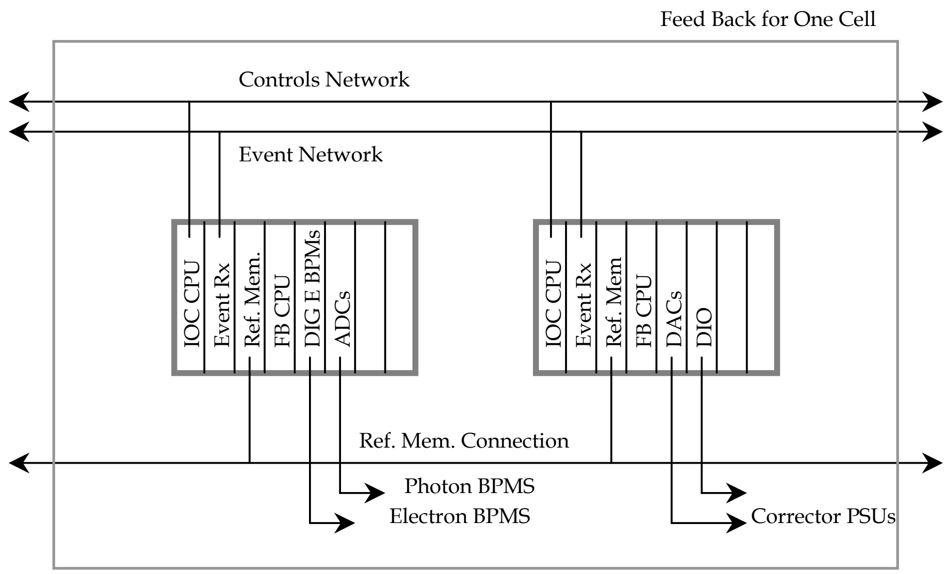

Figure 2.16-1 Reflective memory structure for one cell

(a) Reflective Memory

Reflective memory is an off-the-shelf solution to distribute information across multiple computer systems without requiring processor time. It enables BPM data to be written to the reflective memory module in each of the BPM IOCs and appear in memory in all the Steering IOCs.

In the system shown in Figure 2.16-1, an event received by all BPM IOCs would cause the photon and electron BPM values to be read by the feedback processor and written to the reflective memory board for each of the 24 IOCs. The data would propagate to all the steering IOCs and when all values are received, the feedback calculation would be carried out on the Steering IOC to produce the new steering settings. These values would then be written to the steering elements in conjunction with the slow system values received through EPICS process variables.

Commercially available reflective memory sizes and update rates provide for moving multi-megabytes per second across tens of boards, and so should easily meet the requirements of this application.

(b) Network Broadcast

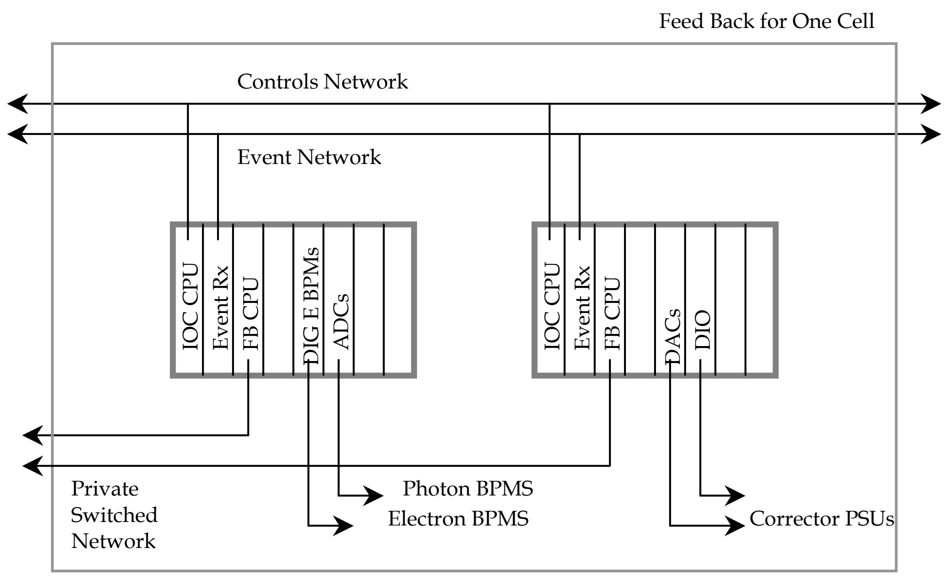

Figure 2.16-2 Network broadcast structure for one cell

In the Network Broadcast system each of 48 feedback processors, in the BPM and Steering IOCs, is connected to a private network with a central switch in a star configuration. The feedback processor in each of the BPM IOCs reads the BPM values and broadcasts them over the network to be received by each steering IOC. The broadcasts take place simultaneously, but do not collide, as the switch buffers each packet as it receives it. The switch then forwards the packets to all the Steering IOCs.

In the system shown in Figure 2.16-2 an event received by all BPM IOCs would cause the photon and electron BPM values to be read by the feedback processor and broadcast over the private network. When each of the 24 broadcasts has been received by all of the Steering IOCs, the calculation would be carried out on each Steering IOC to produce the new steering settings. These values are then written to the steering elements in conjunction with the slow system values received through EPICS process variables.

This option is cheaper in terms of hardware because it alleviates the need for the reflective memory boards, but incurs a development overhead to produce software for the broadcasting. The performance achievable using broadcasts also needs to be determined to establish whether it would meet the requirements of this application.

2.16.6 Feedback Processor

On the Steering IOC, the calculation to produce the new steering values from BPM values and the inverse response matrix needs to be carried out. The time available to carry out this calculation is dependent on the desired update rate, the time to acquire and distribute the data and the time for the correction to propagate through the steering power supply, magnet and vessel to the beam.

While the current generation of PPC processors offer similar performance as DSP processors, in terms of operations per second they do not have the architectural feature of DSP processors for signal processing intensive calculations. However, the performance available from current PPC makes them suitable to carry out the feedback calculations at a price advantage over DSP processors.

2.17 WIDE BAND MULTIPLEXING

In order to perform diagnostics and control it is necessary to monitor various RF signals. As far as possible this will be done locally in each CIA using oscilloscopes and spectrum analysers under the control of an IOC. The IOC will provide the control and data read back of the analysis equipment through various EPICS records.

However, some functions, such as bunch statistical analysis, will require the analysis equipment to be present in the MCR. This requires a wide band network. Using high quality co-axial cable to connect each of the RF signals to a RF network switch (or multiplexer) in the Physics Room, the required signal will then be transferred to the analysis equipment in the MCR, on the control desk.

The RF switch in the PR will be controlled by an IOC allowing the selection of between eight and sixteen signals to be transferred to the MCR. This means that the operators can select which signals to analyse from the control system

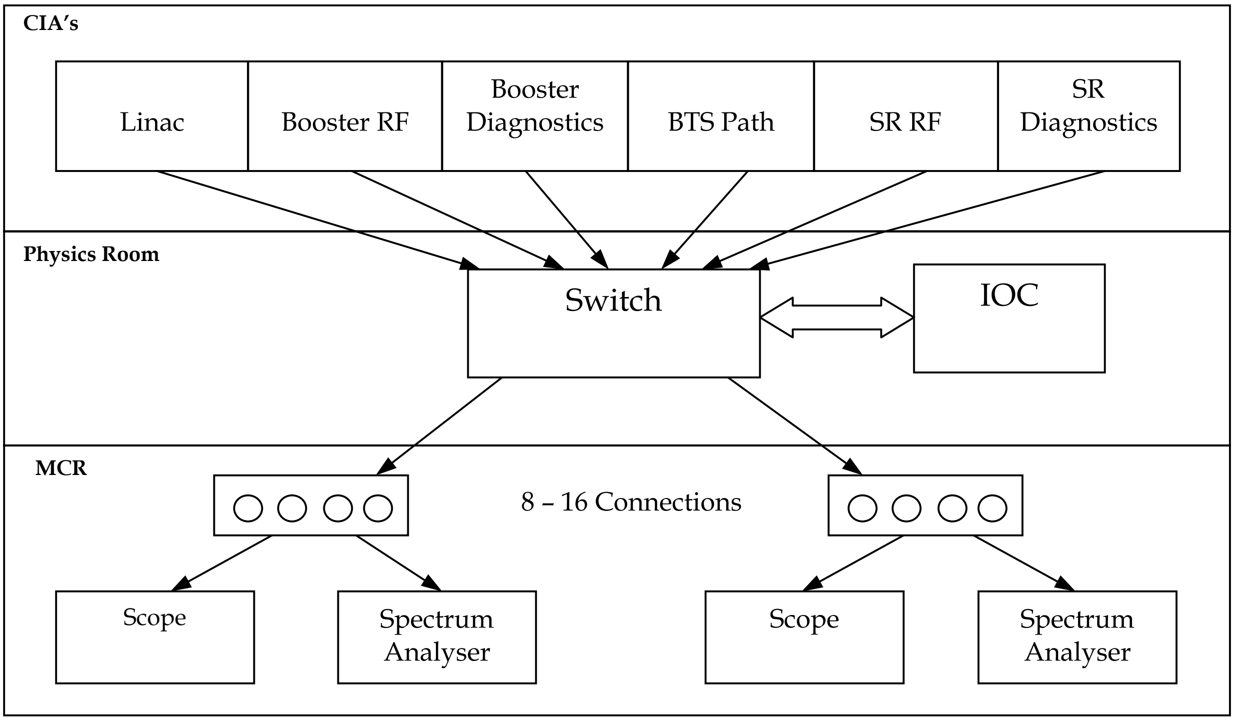

Figure 2.17-1 Block Diagram of Wide Band Multiplexing System

2.18 GLOSSARY

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

ADC	Analogue to Digital Converter
ADT	Array Display Tool
ALH	Alarm Handler
API	Application Programmers Interface
ALS	Advanced Light Source
APS	Advanced Photon Source
CA	Channel Access
CDEV	Common Device
CIA	Control Instrumentation Area
CLIP	CDEV Linear Access Protocol
CORBA	Common Object Request Broker Architecture
CS	Control System
CSCR	Control System Computer Room
CVS	Concurrent Versions System
DAC	Digital to Analogue Converter

DCT	Database Configuration Tool
DDL	Device Definition Language
DSP	Digital Signal Processor
DP	Display Page
DPP	Dipole Permit
EDM	Extensible Display Manager
EPICS	Experimental Physics and Industrial Control System
FO	Fibre Optic
GPIB	General Purpose Interface Bus
HEP	High Energy Physics
IF	Interface
IL	Interlock
IOC	Input Output Controller
IP	Industry Pack
KM	Knob Manager
LAN	Local Area Network
MCR	Main Control Room
MEDM	Motif Editor Display Manager
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
OPI	Operator Interface
PLC	Programmable Logic Controller
PR	Physics Room
RDB	Relational Data Base
RDB	Relational Data Base Management System
RFP	RF Permit
SCADA	Supervisor Control and Data Acquisition
SDDS	Self Describing Data Sets
SNC	State Notation Compiler
SLS	Swiss Light Source
SNL	State Notation Language
TACO	Telescope and Accelerator Control System
TANGO	TACO and New Generation Objects
VME	Versa Module Eurocard (The meaning is now obsolete)

REFERENCES

[1] M. T. Heron, B. G. Martlew, "A Review of options for the DIAMOND Control System", Proceedings of PAC99

[2] EPICS, http://www.aps.anl.gov/epics

[3] TANGO, http://www.esrf.fr/computing/cs/tango/tango.html

[4] V System, http://www.vista-control.com/

[5] M. T. Heron, B. G. Martlew, A. Oates and W. R. Rawlinson, "An analysis of the SRS Control System upgrade", Proceedings of PCaPAC 2000

[6] CVS, www.cvshome.org

[7] http://www.windriver.com/

[8] CapFast, www.phase3.com/epics.html

[9] VisualDCT, http://kgb.ijs.si/VisualDCT

[10] VisualSNL, http://www.sls.psi.ch/controls/software/VisualSNL/

[11] J. Chen et al, "CDEV: An Object-Oriented Class Library for Developing Device Control Applications", Proceedings of ICALEPCS 95

[12] W. Ackers, C. Watson, J. Chen, "The CDEV Linear Internet Protocol Definition", http://www.jlab.org/cdev/doc/doc_1.5/cdevProtocol.html

[13] W. Ackers, C. Watson, J. Chen, "CDEV Gateway", http://www.jlab.org/cdev/doc/doc_1.5/cdevGateway.html

[14] M. Borland, "Getting Started with SDDS", http://www.aps.anl.gov/asd/oag/manuals/GettingStartedWithSDDS/HTML/GettingStartedWithSDDS.html

[15] VxWorks, http://www.windriver.com/

[16] SBS Green Spring, www.sbs.com/computer/products/mio.shtml

[17] Tews, www.tews.com

[18] Hytec Electronics, www.hytec-electronics.co.uk

[19] "Systems Requirement Document for Equipment, Device and Signal Naming", SNS102000000-SR0001-R00,

REFERENCES, Continued

http://www.sns.gov/projectinfo/ics/119/ctrlStdsHandbook/operInterfaceStds/srdNaming.pdf

[20] "APS Control System Naming Convention", http://epics.aps.anl.gov/asd/controls/nomenclature/

[21] A. Streun, "SLS Functional Device Naming Convention", SLS-TME-TA-1998-0001, http://slsbd.psi.ch/pub/slsnotes/naming/slsnaming.htm

[22] K. Kasemir, "SNS EPICS Config. Database", Presentation at EPICS Collaboration meeting, May 2001, http://lansce.lanl.gov/lansce8/epics/Epics2001SL/kay/EPICS_RDB.ppt

[23] T. Pal, "SLS Oracle Database Documentation", http://www.sls.psi.ch/controls/database/index.html

[24] Oracle Database@Bessy II, http://oracle.bessy.de/

[25] Oracle Corporation, http://www.oracle.com

[26] W. P. McDowell and K. V. Sidorowicz, "An Accelerator Controls Network Designed for Reliability and Flexibility", Proceedings of ICALEPCS97

[27] E. W. Kamen and S. S. Singh, "A Comparative Reliability Study of Computer Network Configurations for Large-Scale Accelerator Systems", Proceedings of ICALEPCS97

[28] CEBAF-TN-0193, Description and Interface Guide for the CEBAF Fast Shutdown System, CEBAF Technical Notes.

[29]T. Korhonen, M. Heiniger, "Timing System of the Swiss Light Source", Proceedings of ICALEPCS2001

[30] F. Lenkszus, R. Laird, "The APS Event System", Proceedings of ICALEPCS95

[31] J. A. Carwardine and F. R. Lenkszus, "Real-Time Orbit Feedback at the APS", Proceedings of BIW98

[32] C. Steier, A. Biocca, E. Domning, S. Jacobson, G. Portmann, Y. Wu, "Design of a Fast Global Orbit Feedback System for the Advanced Light Source", Proceeding of PAC 2001

[33] M. Böge, M. Dehler, T. Schilcher, V. Schlott, R. Ursic, "Global Orbit Feedback System for the SLS Storage Ring", Proceedings of ICALEPCS 99