Implementation

Collaboration

The SLS control and data acquisition system needs to be delivered on time, at minimum cost and meeting the present and future needs of the users of the system. In order to meet these requirements, a conservative design is proposed, closely mirroring the best practices and designs used at other laboratories. The SLS control system will be built using Software components from the EPICS collaboration and from industry. By collaborating on the development of the control system, the Epics community ensures that developments and improvements made can be of common benefit, and a wider pool of expertise and experience is available.

The requirements for the control of the SLS are no more demanding than those for existing machines such as APS, and ESRF. Thus, basing the control system design on existing machines minimizes risk, and allows early deployment of control system components, which is vital due to the very short time available for installing and commissioning of the SLS. Early delivery of controls to the equipment groups for laboratory development allows application programs to be written and tested in a timely manner. It also reduces the considerable effort that would be needed in building temporary, ad-hoc solutions in the laboratory, and the re-writing or integration that would then be needed in the final system.

Features

Advances in both hardware and software technology should be incorporated in the SLS control system where appropriate. Areas where new developments are of interest to the SLS are: the use of Personal Computers, the incorporation of field-bus for control of slow devices, and the use of object-oriented technology.

Some key features of the SLS Control system include:

- Use of the **EPICS** toolkit.
- **Java & Tcl** used for applications and tools.
- An Object Oriented API (**CDEV**).
- **VME** process level running **VxWorks**
- **Field-bus** extensively used.
- the use of common technology between machine control, beamline control, and experimental data acquisition.

Supervisory level

Supervisory level will consist of general purpose computers with high resolution displays. In the main control room these console computers with be equipped with three display monitors each. Similar computers at main equipment locations will allow local control, but will typically have a single display. Consoles on beam-lines for experimental control may require one or two displays. These computers will provide the environment for running control and monitoring application programs and for data analysis. Standard tools will provide the ability to monitor and display any parameter in any equipment in the control system. Archiving and backup/restore of machine state will be accomplished at this level. While most applications can be built using the standard graphical tools provided, the ability to rapidly build special applications will be met using TCL-Tk for building new operator interfaces and for scripting.

Applications will be written in Java, to allow a powerful development environment which is simple to use and platform independent.

<table>
<tr><td>JAVA</td><td>TCL</td><td>LABVIEW?</td><td>PvWave?</td></tr>
<tr><td colspan="4">CDEV</td></tr>
<tr><td colspan="3">EPICS</td><td>ORACLE</td></tr>
</table>

PC technology has an apparent price performance advantage over the more traditional UNIX workstations for operator display. However the rapidly changing world of desktop PC technology means that today’s hardware and software systems may be outdated quickly. By developing applications using portable environments and languages it is possible to delay the choice of final operator interface hardware and operating system. The use of Java, and TCL-Tk means that the control system can use whatever hardware and operating system technology is most cost effective and stable. Initial Development will use PC's with both Unix (Linux) and NT operating systems. The final decision on what should be installed should be made as late as possible, probably six months before final control room installation. For this reason the use of Motif, X-windows, MFC (Microsoft foundation class libraries - used by visual basic, visual C++ etc.), and other non portable environments will not be supported.

Display tools - including GUI (Graphical User Interface) builders, plotting, correlation, and 'super knob', will use the standard tools from the Epics collaboration and commercial tools running on top of CDEV. Some commercial widgets can also be used, as long as they have good cross-platform support. Initially Spec-Tcl/Java from Sun microsystems we be used to build user screens, but JDM, the Java Display Manager, from Jefferson Lab should be considered.

Sequencing - including startup and shutdown scripts will be accomplished using TCL. This will allow the rapid generation, and easy modification of an event driven sequencing interface.

Alarm monitoring - the monitoring of control system components will be accomplished using Epics facilities, possibly integrated into a standard commercial alarm monitoring package using SNMP (Simple Network Management Protocol). Alarms or exception conditions will also optionally trigger events within the control system, for instance for the orderly shutdown of equipment. It will be possible to filter alarms, provide a hierarchy of alarm conditions, and to identify first fault. All devices will include a history buffer which can be frozen on an alarm condition, in order to provide post-mortem analysis. HP Open-View would be a good candidate as an alarm package, and it is already extensively used at PSI.

Database - support will be provided to store persistent data, archive data and configuration data in a standard database. Persistent data includes installation and maintenance history of equipment. Archive data will allow long term cataloging and storage of settings and measurement. Configuration data includes sets of parameters that can be recalled to operate the machines under a given condition. Oracle will be the initial database product used, but developments in object-oriented database technology will be followed. New developments in the Epics community for archiving will also be closely monitored. Equipment groups will be responsible for the maintenance of the database information, such as engineering unit conversions. Appropriate tools will be made available to access and modify the central database information from standard packages such as Excel spreadsheets.

Processing level

The Processing level will consist of VME systems running the VxWorks real-time operating system. The use of VME provides a proven high performance bus with a vast range of processor and input/output cards available. Other high performance bus architectures such as PCI or compact-PCI are worth considering. However it is not yet clear if these synchronous bus architectures will have as long a useful lifetime as (asynchronous) VME bus, and generally PC technology tends to become rapidly outdated. Communication between real-time crates and Operator Interface is over standard local area network connections. This will initially be 10 and 100 Mbps switched Ethernet over fiber optic links. Devices requiring high performance input output such as beam position monitors will be directly interfaced to real-time crates, other slow devices will be interfaced using a field-bus.

Hardware **timing** signals will be distributed to each rack over fiber optic lines (by the RF group). These signals can then be distributed to each VME crate (with co-ax), where a VME counter card will be able to generate additional hardware timing to trigger equipment, and to generate software interrupts. These signals will consist of the 3Hz Linac cycle clock, the (~1us) Booster revolution clock, and the (~1us) Storage ring revolution clock. The (~500 MHz) RF clock will also be needed at some locations, but this will require a different counter card. The provision of this hardware timing, will allow all measurements made to be accurately time-stamped. This will permit the correlation of measurements across all systems, and meaningful archiving of data. The APS timing system, is a good starting point or the selection of hardware timing components.

Device interface level

The Device Interface level consists of a field-bus interface to equipment. Field-Busses supported will be CAN and LONWORKS. The use of a direct Field-bus interface to equipment will remove the need for a front end crate based bus system such as G64. Standard device controllers including field bus interfaces will be provided for Power supplies, vacuum pumps and stepper motors. Standardized daughter boards and chip-set designs will be made available for inclusion in equipment, as well as general purpose interfaces including DAC/ADC and digital input output. The device interface level will also provide circular buffers for post mortem analysis. Examples of suitable front-end controllers include the ADA card developed for the BessyII project. This card includes an 16bit ADC, DAC and opto-isolated digital IO, controlled by a 386 processor.

Network

Each of the 2,500 devices in the SLS control system, will consist on average of ten attributes (channels). In order to estimate worst case load, it can be assumed that each of these send all data at 10Hz to a supervisory program. Channel access sends, on average, 24 Bytes per channel, and TCP overhead should be less than 20% due to channel access efficiently packing many channels in a TCP message. In this (worst) case the total system throughput would be: 2500 devices x 10 channels x 10Hz x 24 bytes x 1.2 (TCP overhead) = 7.2 M Bytes per second. However this bandwidth is split among the 60 VME crates in the system. The average bandwidth per crate is therefore: 7.2 M Bytes / 60 = 120 K Bytes per second which is < 10% of a 10 M bit/second Ethernet connection. Using switched Ethernet, each crate does not share the network bandwidth with others. A more critical point could be the throughput of the application servers (workstations), even though they would use faster (100 M bit per second) switched Ethernet networking. Assuming 10 servers receiving all data, the average network utilization of each would only be <6% of available bandwidth.

A. In addition the use of call-backs (monitors) to only send changes in values, will decrease average traffic even further, by at least one order of magnitude. The transmission of many waveform buffers, at high rates, can saturate any network. SLS does not require transmitting all BPM raw data to Workstations. The most required is ten simultaneous 1000 point buffers at a rate of 10 Hz. This requires 10 (BPM readings) x 1000 (points) x 2 (bytes) x 10 (Hz) = 200 bytes per second = 20% of a 10 Mbit Ethernet, even if all signals originate from the same VME crate.