IDEAS ON A GENERIC CONTROL SYSTEMS BASED ON THE EXPERIENCE ON THE 4 LEP EXPERIMENTS CONTROL SYSTEM

R. BARILLERE, J. M. LE GOFF, H. MILCENT, R. STAMPFLI
CERN, European Organization for Nuclear Research, 1211 Geneva 23, Switzerland

Most of the large slow control systems in the LEP collider experiments are distributed heterogeneous and multi-standard. But in spite of the appearances, they have a lot in common. From our direct experience on the L-3 slow control system and from the informations we obtained on the 3 other LEP experiments control systems we have come to the conclusion that it should be possible to build a Generic Control Package from which any control system could be derived. This software package is entirely based on relational databases and is intended to provide all the necessary tools to build a modular, coherent, easy to update and to maintain control system. Among other things this package should include user friendly interfaces, expert systems, and powerful graphic monitoring and control tools. This paper will present our general ideas about the realization of such a package.

1. Introduction

The need for large and dedicated Slow Control Systems for the High Energy Physics Experiments at CERN only became clear with the construction of the LEP collider. The large number of parameters involved in the controls as well as their wide distribution over the sites forced physicists to reconsider the entire organization of such systems. In this paper, the 4 LEP experiments control systems will be presented in some details followed by a summary of their main characteristics. Out of this study some general ideas which could be used for the elaboration of generic tools for future Slow Control systems will be presented.

2. The 4 LEP experiments Slow Controls

The 4 experiments are composed of many sub-detectors and several thousand operational parameters which have to be set and monitored in order to insure that the experiments are in a correct state for data-taking. These include quantities such as high and low voltages, temperatures, pressures and gas mixtures. The role of these Slow Control Systems is to monitor the experimental conditions locally and generate alarms in the main control room when faults are detected. Operators alerted by the alarm signals correct for the fault either manually or remotely via a computer. The systems are also used for the routine setup of operating conditions for example high voltages, general monitoring and logging of operating conditions.
Since the quantities to be monitored vary only on a time scale of seconds the slow control systems need not to have very fast responses. However, they must be very reliable and able to handle a large number of very different parameters.

2.1 The ALEPH Slow Control [1]

The ALEPH system is implemented in a number of distinct parts which are listed below:

1) Independent control and monitoring programs for each sub-detector run on VAX™ computers configured in a VAX™ cluster. These main displays of the currently monitored equipments allow operators to control the detectors and log the information to storage media.

2) All the sub-detectors slow control programs communicate with their equipment via a single server program running on a dedicated microVAX™ 3100 workstation in the cluster (fig. 1). The slow control server is the only program which communicates directly with the microprocessor system outlined below, which monitors and control the equipment. An interactive graphical status display showing the layout and status of the system is also implemented on this station.

3) Low-level monitoring and control functions are performed by programs running in a distributed system of microprocessors attached to the sub-detector hardware. They are capable of understanding, executing and replying to simple commands sent to them from the VAX server program.

4) Networks are used for communication between the slow control server and the microprocessors. The processors are connected to a number of UTI-NET

local area networks and the VAX cluster to Ethernet. Ethernet-UTI-NET gateways have been installed to allow communications between the two different networks.

5) A large relational database describing the correspondence between the physical detector and the slow control system has been implemented on the VAX™ cluster. It is used extensively by the server to translate requests coming from the application programs given in terms of the detector high level commands into commands which will be understood by the microprocessor programs (low-level commands). The connectivity and the essential elements of the slow control systems are illustrated in figure 1.

The gateway software is written in C and runs on a Motorola 68010™ processor under the OS-9 operating system. Its role is to synchronize the transfer of messages between the two data networks which have very different physical characteristics. The program is loaded on resetting the system from files residing on the VAX™ cluster Network disk.

The central element of the ALEPH slow control system, as indicated before is a relational database which describes all the aspects of the system. It is one sub-section of the online database which also contains the descriptions of the Fastbus system and the readout configuration. The entity-relationship model is used in the design of all the databases [1]. In such databases, elements in a system are represented by tables. A particular instance of an element occupies a row in a table and each attribute which describes the elements occupies a column. Furthermore, the description of each element may be supplemented by specifying its relationship with other tables in the database or in other databases. The advantage of this type of structure is to avoid the need of repeating information unnecessarily and thus to reduce the possibility of conflicting entries and enhancing the data integrity. Entity-relationship diagrams have been used during the design phase of the database and provide a convenient means of displaying the database structure.

A graphic status color display showing the current status of the system is running on a colour workstation in the VAX™ cluster. In addition it allows inconsistencies in the database to be spotted more easily. The display has been made interactive and layered to cope with the large amount of informations involved.

The display program is driven by a graphical database derived from the slow control one using the ALEPH graphics package [1].

2.2 The DELPHI Slow Control [2]

The DELPHI slow control system is composed of three major logical components. The G-64 microprocessor systems, very similar to the one of ALEPH, provide the interface to the hardware to be controlled or monitored. The intermediate level software running on microVAXes and VAX™ stations, called the equipment computers, which handles all the low level processors belonging to a single detector. And finally the high level software which performs the overall control is running on the central control computers (VAX™8700 and VAX™4000).

1) The G-64 Systems

Each system is equipped with a Motorola 6809™ processor, RAM/ROM memory and commercial cards like ADCs, DACs and HV controllers. The communication with the intermediate layer is performed by a G-64-cheapernet interface board which has been developed at CERN. The main functions of the standardized software running in these systems are the following: at startup, the software sets the value of all physical channels to preselected ones which are stored in the VAX™ and downloaded to the G-64. During normal running operation, it monitors the values of all the channels in a tight program loop and reports the significant anomalies and the alarms to the program running in the VAX™ equipment computer. Also during normal operation it receives and carries out instructions coming from the equipment computer like, ramping up or down the high voltage of a given device.

2) VAX equipment computer software

The software running in each sub-detector equipment computer handles all the aspects of the communication between the G-64 layer and the master central layer. It receives actions from the top layer and distributes them to the interested G-64 system. It also collects all the alarm messages and anomalies from the bottom layers and forwards them to the top layer. Finally it updates the central DELPHI database with the values of the parameters and their status. At startup this program also performs the downloading of the preset conditions of each parameters from the database to the G-64 targets. Standard software has been developed by DELPHI to perform these functions. Its form was determined by the choice of the automated control system based

on a CERN package called the State Management Interface (SMI) of the Model software [3].

3) Central Control and Interfacing Software

This layer also uses the SMI package but applied at a higher hierarchical level than before. This SMI single program interacts with all the different sub-detector control systems. No fortran processes are involved. A central operator's display allows the overall state of the experiment to be viewed from a VAX™ station 3100 using the Motif™ Graphics system. On the basis of the information displayed the operator can decide to start or to stop the run. The messages generated by the sub-detector control systems are also displayed on this graphic interface. A general view of the system is presented in figure 2.

2.3 The L-3 Slow Control [3]

The L-3 system is also modular and is composed of four different layers (fig. 3), each of them performing very distinct functions:

1) The BBL3 Layer

This a non-computer controlled hardwired layer dedicated to the high level alarms whose consequences would be of major importance for the experiment if the necessary actions were not taken on time. Each of these hardware box offers a maximum of 48 isolated inputs and 32 outputs specially designed to perform important actions as to power off a control room or a piece of equipment which is not operating well. A manually programmable matrix allows any input combination to perform any number of distributed actions. The all system is being independently read out by the computer of the detector which owns the box and by the second layer of the slow control system described after.

2) The local Slow Control Layer

This is a typical local control system which reads out the parameters of a given piece of hardware, performs a local analysis of its behavior, reports to the central layer for alarms or new data and takes local actions which are within its range. Depending on the equipment this software is running either on a VAX™ computer or on a dedicated PC or a VME OS-9 system. It has been designed to be as little equipment dependent as possible and is reasonably data driven.

3) The central Slow Control Layer

For safety reasons, this very important layer is running on a standalone microVAX™3200. It houses the database which describes all the parameters the L-3 slow control is responsible for. Its main functions are to collect from the network all the informations related to all the equipments like alarms or digital monitoring data blocks. After every update of the shared memory which holds the present status of the experiment, an expert system is triggered to analyse the situation. It takes actions when necessary. These actions will be broadcasted to the lower level computers where they will be physically executed. In case of time out due to network problems or to other reasons the expert system may take higher level actions if it has been foreseen in the database. Otherwise, the serious situation in which the system is, will be immediately reported to the operator. In case of hardware alarms where the actions are taken independently of the software, the system will report the new status of the equipment of concern and will not proceed any further. At each modification of the content of the shared memory, a trace of what happened is written on a daily file. It will also appear to the operator as a structured message which has been preprocessed from the information contained in the database. A graphic display system running on various Macintosh™ computers located in the main control room and in various places at CERN will immediately be informed of any changes.

4) The Color Graphic Display System [4]

This sophisticated graphic display has been entirely written in Object Oriented C making use of the THINK C™ library of the SYMANTEC company. It allows an operator to view the status of the entire experiment just by clicking into sensitive boxes whose colour represents the worse status of all the parameters or windows they give access to. The all window tree mechanism is housed in the database of the third layer as well as the description and location of all the parameters to be displayed. At startup the application inquires to the central slow control the date and time of the last modifications in the database. If they differ from the ones corresponding to its local copy, it will request an update over the network. It will then build the application objects from these new static data. It is important to mention that this graphic system is entirely data driven and does not need any maintenance. The application software is itself available on a Macintosh™ server if needed and does not have to be resident in the macintosh™.

2.4 The OPAL Slow Control [4]

Unlike the three other systems, the OPAL slow control system is not a layered hardware system. A series of VME OS-9 creates all connected via Ethernet constitutes the backbone of the system. Each of them is dedicated to various tasks regarding a part of the experiment. A Macintosh™ using the Hypercard™ software of the Apple Company is directly connected to the VMEBus and allows a local graphic display of the parameters under the control of a given crate. The software which performs these tasks is organized in three layers presented in figure 4 and described below.

1) Human Interface

There is a variety of possible interactions with the control system, such as sending a command and retrieving an answer message. A comprehensive and powerful human interface to handle the whole OPAL detector has been implemented on an Apple Macintosh™IIx, using the Hypercard™ tool. It uses graphics to interact with the operator. SC_MENU is an alphanumeric menu-driven interface which runs on a terminal. Run control is a dedicated data acquisition program, while SC_ALARM displays warnings and alarms for the whole system. Any computer on the network can be used as a sending or receiving node; however, a clear separation is made between "harmless" commands (e.g. to get a status) and actions on the detector. The later are normally not enable from remote sources.

2) Skeleton

At boot time, SC_PROC starts all the programs in layer 2 and supervises them. They are all written in C language. Commands are sent to the program SC_FILTER, which checks their validity, writes them in a log-file if required and passes them to the appropriate application in layer 3. This program executes the command and may them send an answer message back to SC_ROUTER which has already been notified where the answer should be sent to and which passes it on. Unsolicited output of the application programs, such as errors, is accepted by SC_ERROR, which logs it in a file and, if required, passes it on. Automatic corrective actions are handled in the same way and sent to SC_FILTER.

3) Applications

An important consequence of this clean separation into three layers is that the same application programs can run in both stand-alone mode (e.g., for debugging) and in an integrated mode with the entire system. They can be written in any language. The application programs are also supervised by SC_PROC, which starts and stops them on request. A typical example is the program which reads the ADC of common controls. This program is driven by a table which gives the characteristics of individual channels such as their description, warning and alarms limits, actions to take, etc. It compares all parameters with their nominal values every 5 seconds. Asynchronously, it also accepts commands to show the status, change default values, disable actions, etc.

3. Main characteristics of the 4 systems

Due to the large number of parameters to be controlled and their wide distribution, the 4 slow controls have adopted the layered and distributed system principle. For the same reasons dedicated and unspecialized slow control programs have been written to reduce redundancy in the code. In most cases special efforts have been made on the design to provide systems as close as possible to an ideal data driven slow control. But unfortunately, in most cases it has not been possible to achieve this goal all the way down to the lowest control layers. The importance of databases in such systems have been clearly understood and will be the key part of future slow control systems. These four systems have been running since the LEP collider started in July 1989 and are expected to operate until late in the decade. In the four cases major upgrades of the hardware of the detectors in the experiments are expected. These will force every slow control system to follow up closely. But due to the limitations of some of the software and hardware, and also due to the lack of modern software engineering tools when these systems have been created, it is to be anticipated that major maintenance efforts will have to be done.

4. A possible Generic Control System

To satisfy fully the data driven constraints, such a system should entirely be based on large relational databases where tools like CASE tools and entity relationship models would insure a full coherency between the different tables.

As illustrated in figure 5, a set of external software engineering tools like a Configuration Manager and a Coherency Manager could be implemented to help the user to perform the design and the upgrades of his slow control system by following strict and formalized rules. Furthermore, with no extra efforts an up to date formatted documentation would be readily available. The data blocks and their definition would then be extracted from the database and distributed over the network to the target computers where the necessary updates would be accomplished. At every level of the system, tools would be provided to allow an automatic implementation of the newly received data blocks. An acknowledgement mechanism in relation with the Configuration Manager would also have to be designed to reduce the system instability during these transition phases.

The internal organization of the database and the corresponding software layers are presented in figure 6. Five main layers can be identified.

1) The Local Acquisition and Command Layer

This layer is dedicated to perform the typical and basic operations of any local control systems. Namely, read the data from the hardware, interprete them according to the local Acquisition and Command database content and take the actions which could be requested by the Hardware Control layer.

2) The Hardware Control Interface.

This layer takes locally the decisions which are necessary to keep the lowest layer in operation according to the Hardware Control database, for trivial decisions, and to the Software Control layer for more complicated ones.

3) The Software Control Interface

All the actions which can be internally taken and which are independent of the other control systems will be handled at this level. Furthermore this level is responsible for the synchronization between the communication level and the three bottom layers during the updates of the system.

These three layers are completely autonomous and can perform a full local control of a piece of equipment in a stand alone mode. A optional expert system could very well be included in the third layer when needed by the designer.

4) The communication layer

All the data and alarm transfers to the central control system is taken care by this layer. It is also responsible for the temporary storage of the new data blocks issued by the database until the full update mechanism has been completed.

5) The Central Control Layer

This layer holds a copy of the values of all the parameters controlled by the experiment. It will forward the actions requested by the operator to all the sub-systems involved and later report for success or failure. No actions issued at this level will take effect at the level of a local system, if it has not previously been implemented in its Software Control Interface layer. Since this can only be done by the person in charge of the equipment, this avoid spurious and not commissioned actions to take place.

A distributed color graphic system will display either the global status of the experiment or the local status of a piece of equipment by interrogating the central control Layer. The same format and philosophy should be provided for both local and global displays. The upgrades of this system should be handled through the configuration manager.

5. Conclusions

In this paper, it is not possible to go into more details into the organization of this generic systems. But we believe that with the support of industrial software products like CASE tools and others it would be possible to provide the slow control designers with a set of tools which could help them greatly in the setup and maintenance of their systems all along the years of operation.

References

[1] The ALEPH Slow Control System, S. Wheeler et al., ALEPH 90-67 DATAACQ 90-08.

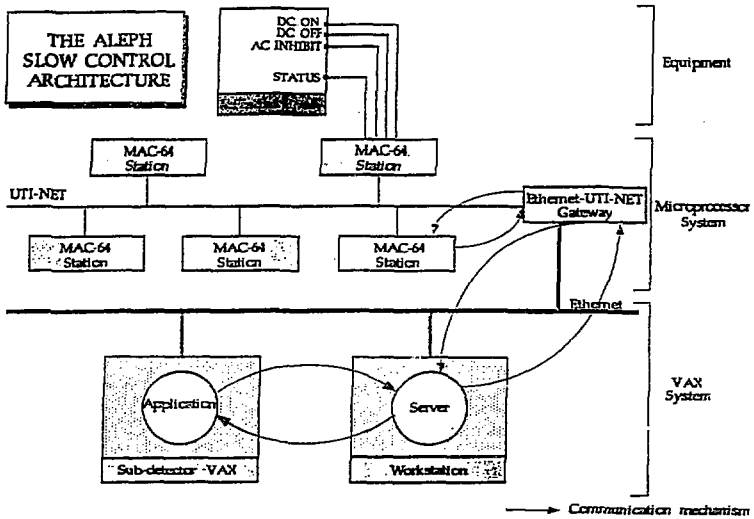
[2] The DELPHI Slow Control System, R. Sekulin at al., DELPHI Internal Note.

[3] The L-3 Slow Control System, J. M. Le Goff et al. Paper presented at the Conference on Software Engineering Artificial Intelligence applications in High Energy Physics, LYON March 1989.

[4] The ARGUS graphic system, J. M. Le Goff et al. Paper presented at the CHEP91 conference, TSUKUBA March 1991.

[5] The Control system of the OPAL detector at LEP A.K. Amundsen et al. NIM A293(1990)145-147.

Figure 1: The ALEPH SLOW CONTROL ARCHITECTURE



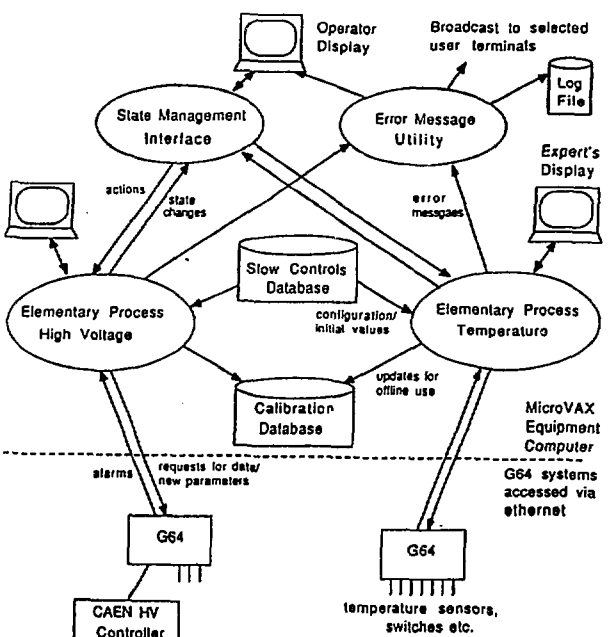
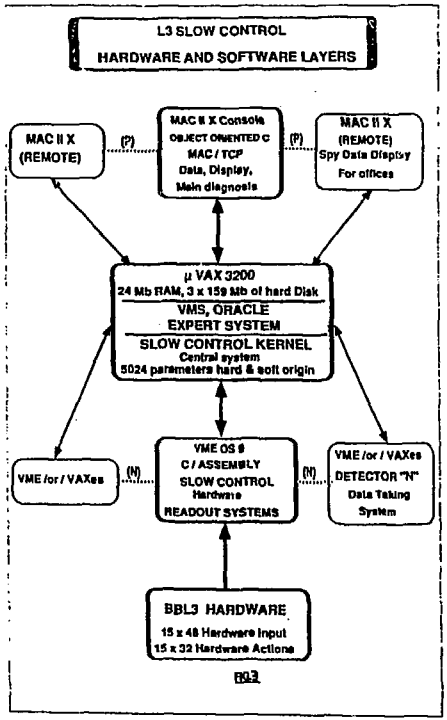


FIG.2: DELPHI SLOW CONTROLS SYSTEM



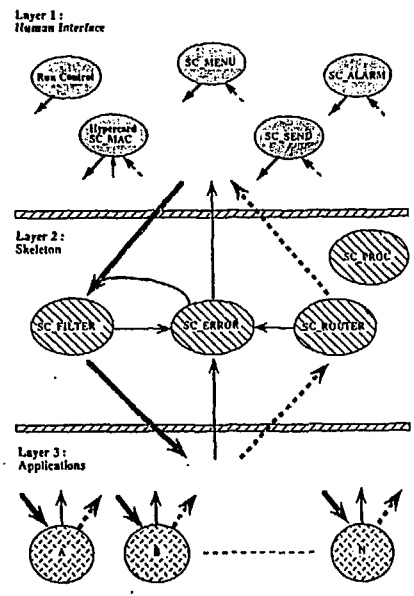


Figure 4: The OPAL SLOW CONTROL SOFTWARE LAYERS

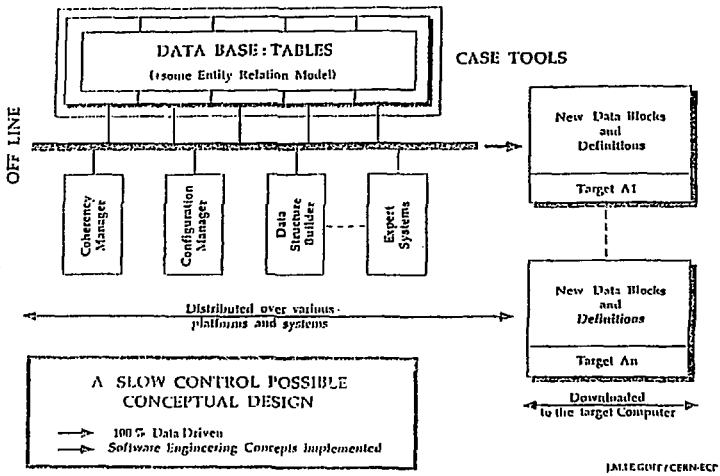


FIGURE 5: The GENERIC CONTROL OFFLINE Configuration