An Integrated Experiment Control System, Architecture, and Benefits: The LHCb Approach

C. Gaspar, B. Franek, R. Jacobsson, B. Jost, S. Morlini, N. Neufeld, and P. Vannerem

***Abstract*—LHCb's Experiment Control System will handle the configuration, monitoring, and operation of all experimental equipment involved in the various activities of the experiment. A control framework (based on an industrial SCADA system) allowing the integration of the various devices into a coherent hierarchical system is being developed in common for the four Large Hadron Collider (LHC) experiments. The aim of this paper is to demonstrate that the same architecture and tools can be used to control and monitor all the different types of devices, from front-end electronics boards to temperature sensors to algorithms in an event filter farm, thus providing LHCb with a homogeneous control system and a coherent interface to all parts of the experiment.**

***Index Terms*—Automation, control systems, finite state machines, knowledge-based systems, large-scale systems, SCADA systems.**

I. Introduction

LHCb [1] is one of the four particle detectors in preparation for the Large Hadron Collider (LHC) at CERN, which will start operation in 2007.

LHCb's Experiment Control System (ECS) will handle the configuration, monitoring, and operation of all experimental equipment involved in the different activities of the experiment:

- data acquisition and trigger (DAQ): timing, front-end electronics, readout network, Event Filter Farm (EFF), etc.;
- detector operations (detector contol systems—DCS): gases, high voltages, low voltages, temperatures, etc.;
- experimental infrastructure: magnet, cooling, ventilation, electricity distribution, detector safety, etc.;
- interaction with the outside world: LHC accelerator, CERN safety system, CERN technical services, etc.

The relationship between the ECS and other components of the experiment is shown schematically in Fig. 1. This shows that the ECS provides a unique interface between the users and all experimental equipment.

The ECS will provide for the integration of the different activities in the experiment, such that rules can be defined, for example: stop the DAQ when the high voltages trip or start taking data when the LHC machine goes into colliding mode. Even though the different activities will be integrated and operated as a whole during physics data taking, during other periods, like commissioning, test, subdetector calibration, etc., the different parts of the experiment will allow for independent and concurrent operation, in a stand-alone manner.

In order to avoid operator mistakes and to speed up standard procedures, the system will be as automated as possible, i.e., there should be no need for operator intervention for all standard running procedures, including, when possible, the recovery from error situations.

Whenever complete automation is not possible, the system shall be intuitive and easy to use, since the operators, 2 to 3, will not be experts in the control system.

In order to fulfill these requirements, a common approach was taken in the design of the complete system and the same tools and components are being used for the implementation of the various parts of the system. A uniform, homogeneous control system brings benefits in several areas.

1) The integration and automation of the different activities is facilitated by the use of the same tools and protocols for the implementation of all components.
2) The operation of the system is made simpler: the user will recognize standard features throughout the system, for example, the same partitioning rules. A common look and feel is also easier to achieve if the same tools are used to build the different user interfaces.
3) Less manpower is necessary to design and implement the system if the available expertise is concentrated on producing a small set of tools. The same applies to any needed upgrades and to the maintenance of the system.

A common project: the Joint Controls Project (JCOP) [2] was setup between the four LHC experiments to define a common architecture and a framework to be used by the experiments in order to build their control systems. LHCb will use these tools for the implementation of all areas of control in the experiment.

II. Architecture

From the software point of view, JCOP adopted a hierarchical, tree-like, structure to represent the structure of subdetectors, subsystems, and hardware components. This hierarchy should allow a high degree of independence between components, for concurrent use during integration, test or calibration phases, but it should also allow integrated control, both automated and user-driven, during physics datataking.

This tree is composed of two types of nodes: "Device Units" (Devs) which are capable of "driving" the equipment to which they correspond and "Control Units" (CUs) which can monitor and control the subtree below them, i.e., they model the behavior

Manuscript received June 5, 2003; revised October 1, 2003.

C. Gaspar, R. Jacobsson, B. Jost, S. Morlini, N. Neufeld, and P. Vannerem are with CERN, CH-1211 Geneva 23, Switzerland (e-mail: clara.gaspar@cern.ch).

B. Franek is with Rutherford Appleton Laboratory, Chilton, Didcot OX11 0QX, U.K.

Digital Object Identifier 10.1109/TNS.2004.828878

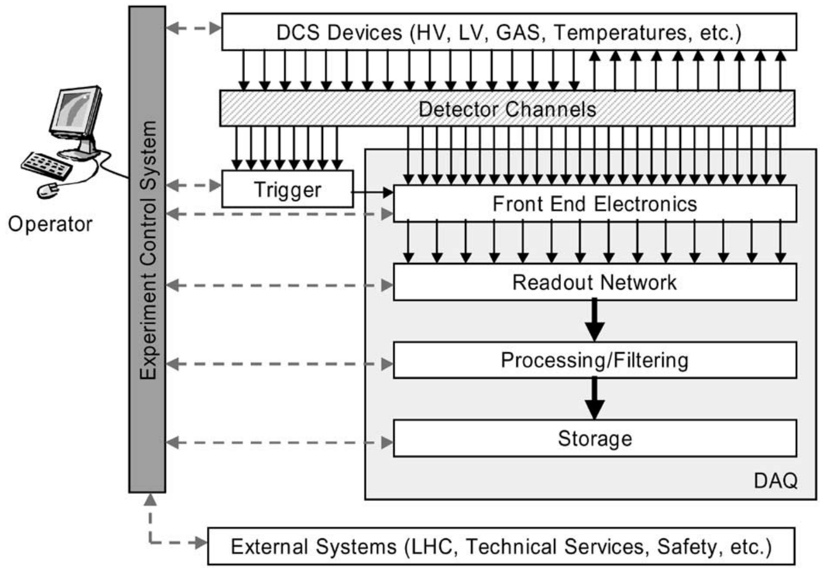

Fig. 1. Scope of the ECS.

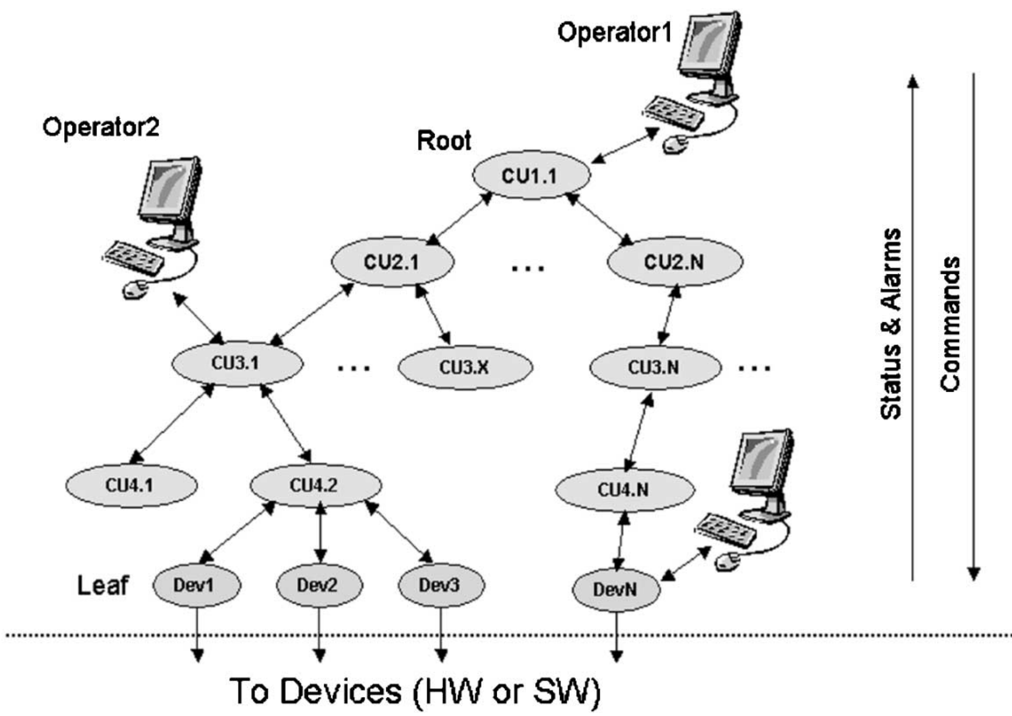

Fig. 2. JCOP software architecture.

and the interactions between components. Fig. 2 shows the hierarchical architecture defined by JCOP.

The architecture defined by JCOP is the basis for the development of the common framework. Each LHC experiment can than adopt this architecture and use the framework tools wherever they find it suitable. While the other LHC experiments chose to use the JCOP tools for the implementation of the Detector Control System (DCS), LHCb decided to use them for the control of the complete experiment. Fig. 3 shows the architecture of LHCb's experiment control system. At the bottom of the tree there are the devices to be controlled, these are grouped into subsystems, then onto subdetectors. Subdetectors are grouped by area of activity, DAQ or DCS and their states are combined with information received from external systems (the LHC machine, the CERN Technical Services—TS, the Gas Systems, and the Detector Safety System) in order to arrive to a combined, decision making, top-level entity.

From the hardware point of view, the control system will consist of a small number of PCs (high-end servers) on the surface connected to large disk servers (containing databases, archives, etc.). These will supervise other PCs (in the order of one hundred) that will be installed in the underground experimental area and provide the interface to the experimental equipment. Fig. 4 shows a generic view of LHCb's hardware architecture.

There is an enormous number and variety of devices to be supervised by the underground control PCs. As such, the control

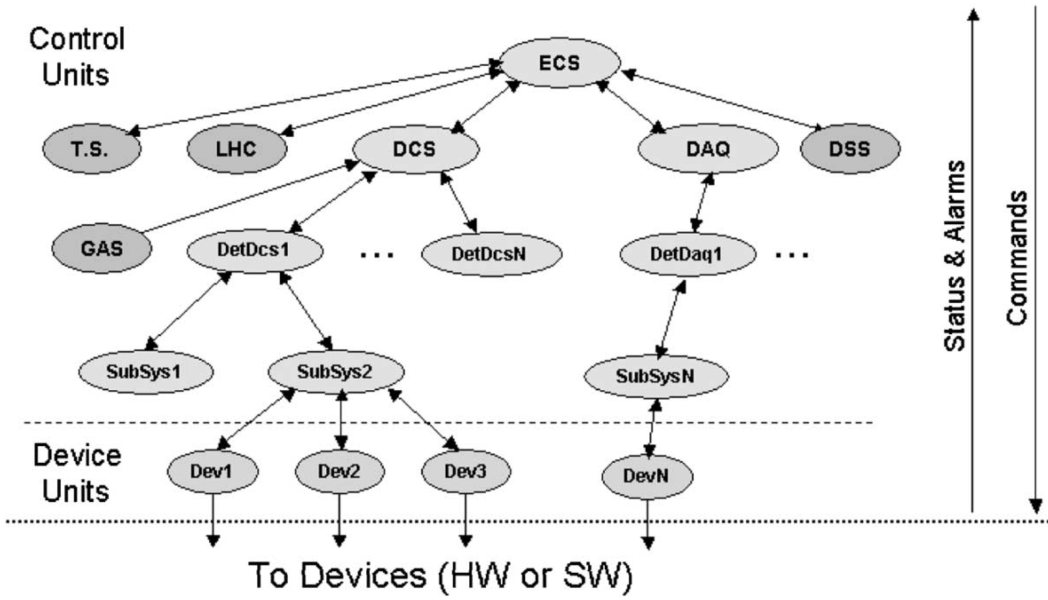

Fig. 3. LHCb's ECS software architecture.

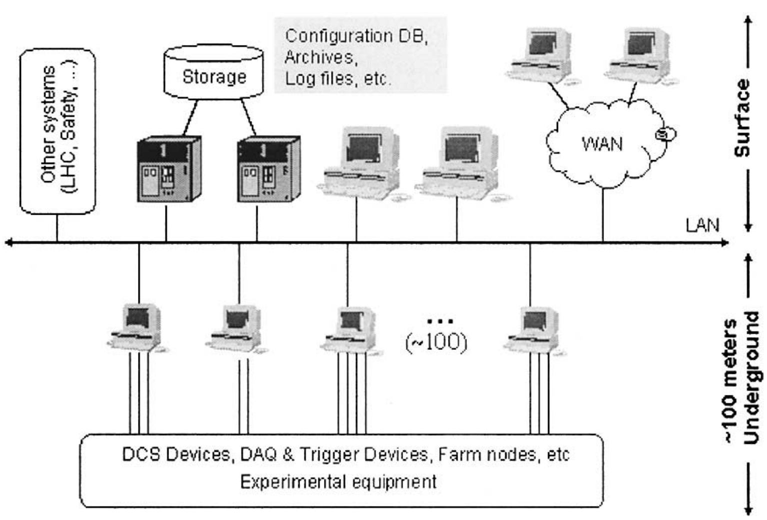

Fig. 4. LHCB's ECS hardware architecture.

system has to provide standard interfaces to the different types of devices and a framework for the integration of these various devices into a coherent complete system. In the following paragraphs, we will first describe the control framework and then the interfaces proposed for the different types of equipment.

III. The Framework

The LHCb Control Framework will be based on the JCOP Framework [3]. It will provide for the integration of the various components (devices) in a coherent and uniform manner. JCOP defines the framework as:

> "An integrated set of guidelines and software tools used by detector developers to realize their specific control system application. The framework will include, as far as possible all templates, standard elements, and functions required to achieve a homogeneous control system and to reduce the development effort as much as possible for the developers."

The architectural design of the software framework is an important issue. The framework has to be flexible and allow for the simple integration of components developed separately by different teams and it has to be scalable to allow a very large numbers of channels.

Some of the components of this framework include the following.

- Guidelines imposing rules necessary to build components that can be easily integrated (naming conventions, user interface look and feel, etc.).
- Drivers for different types of hardware, such as fieldbuses, and PLCs.
- Ready-made components for commonly used devices configurable for particular applications, such as high voltage power supplies, etc.

- Many other utilities, such as data archiving and trending, alarm configuration, and reporting, etc.

The JCOP framework is based on the PVSS II SCADA system [4] and addresses, among others, the following issues.

A. Hierarchical Control

The framework offers tools to implement a hierarchical control system. The hierarchical control tree is composed of two types of nodes: "Device Units" which are capable of monitoring and controlling the equipment to which they correspond and "Control Units" which can model and control the subtree below them. In this hierarchy "commands" flow down and "status and alarm information" flow up.

Control units are typically implemented using Finite State Machines (FSM), which is a technique for modeling the behavior of a component using the states that it can occupy and the transitions that can take place between those states.

PVSS II does not provide for FSM modeling and, therefore, another tool—SMI++ [5] has been integrated with PVSS for this purpose. SMI++ allows for the design and implementation of hierarchies of Finite State Machines working in parallel. SMI++ also provides for rule-based automation and error-recovery.

B. Distributed Systems

Due to the large scale of the system in terms of I/O channels, in the order of millions, and also to guarantee operating independence between the different subdetectors, the Control System of the LHC experiments will have to be distributed across many machines.

Both PVSSII and SMI++ allow for the implementation of large distributed and decentralized systems. There is no rule for the mapping of Control Units and Device Units into machines, i.e., there can be one or more of these units per machine depending on their complexity, or other factors such as development teams they "belong" to. The framework will allow users to describe their system and run it transparently across several computers. Since both PVSSII and SMI++ can run on mixed environments comprising Linux and Windows machines, the user can also choose the best platform for each specific task.

C. Partitioning

Partitioning is the capability of monitoring and/or controlling a part of the system, a subsystem, independently and concurrently with the others in order to allow for tests, calibration, etc.

Each Control Unit knows how to partition "out" or "in" its children. Excluding a child from the hierarchy implies that its state is not taken into account any more by the parent in its decision process, that the parent will not send commands to it and that the owner operator releases ownership so that another operator can work with it.

It was felt that excluding completely a part of the tree was not flexible enough, so the following partitioning modes were defined and implemented in the Framework.

- **Included**—A component is included in the control hierarchy; it receives commands from and sends its state to its parent.
- **Excluded**—A component is excluded from the hierarchy, it does not receive commands and its state is not taken into account by its parent. This mode can be used when the component is either faulty or ready to work in stand-alone mode.
- **Manual**—A component is partially excluded from the hierarchy in that it does not receive commands but its state is still taken into account by its parent. This mode can be used to make sure the system will not send commands to a component while an expert is working on it. Since the component's state is still being taken into account, as soon as the component is fixed the operations will proceed.
- **Ignored**—A component can be ignored, meaning that its state is not taken into account by the parent but it still receives commands. This mode can be useful if a component is reporting the wrong state or if it is only partially faulty and the operator wants to proceed nevertheless.

The partitioning mechanism has also been implemented using PVSSII and SMI++ integrated tools.

D. Error Handling

Error handling is the capability of the control system to detect errors and to attempt recovery from them. It should also inform and guide the operators and to record/archive the information about problems for maintaining statistics and for further analysis offline.

Since SMI++ is also a rule-based system, errors can be handled and recovered using the same mechanism used for "standard" system behavior. There is no basic difference between implementing rules like "when system configured start run" and "when system in error reset it." The recovery from known error conditions can be automated using the hierarchical control tools based on subsystem's states. In conjunction with the error recovery provided by SMI++ full use will be made of the powerful alarm handling tools provided by PVSS II for allowing equipment to generate alarms (possibly using the same conditions that generate states), for archiving, filtering, summarizing and displaying alarms to users and to allow users to mask and/or acknowledge alarms.

E. System Operation and Run Control

The framework will provide configurable operation panels. These panels will have predefined areas showing the states of the hierarchical components, their partitioning modes, their alarm states, etc. and user defined areas that are specific to the task of that particular component. The user can navigate through the hierarchy by clicking on the different components.

The panel showing the component at the top of the hierarchy provides a high-level view of the complete underlying system.

In LHCb, the top of the hierarchy corresponds to the full experiment, allowing the user to have an integrated view of the experiment status and to interact with the different subsystems, the DCS, the DAQ, etc. The main interface to a physics experiment is normally called the "Run Control," unlike the other LHC experiments, LHCb's Run Control will be exclusively based on JCOP tools. The first prototype of LHCb's Run Control panel is shown in Fig. 5.

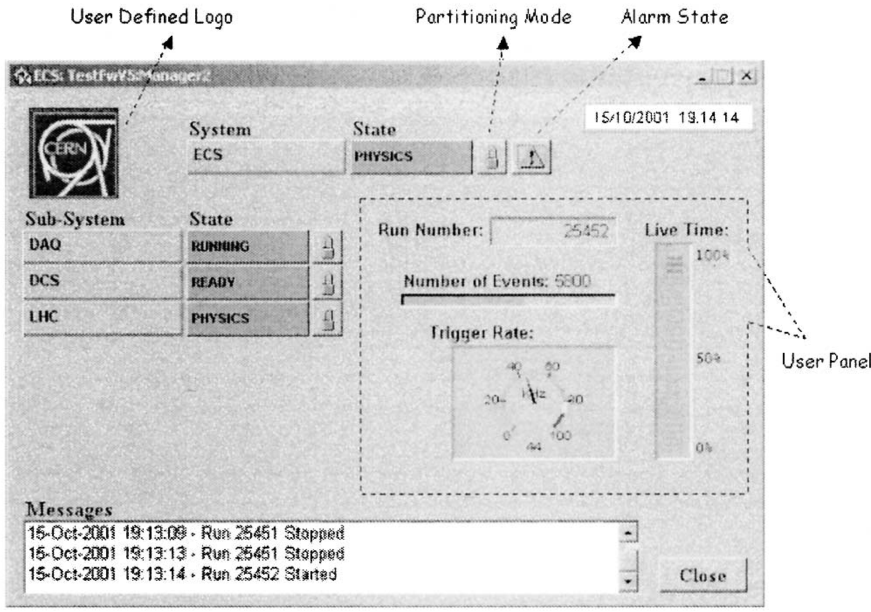

Fig. 5. Prototype Run Control interface.

The operation of the different subsystems, or complete subdetectors when working in stand-alone mode, is based on the same tools and will provide similar interfaces.

F. LHCb and the Control Framework

LHCb is not only a user of the framework, it is also a major contributor, in particular, the FSM integration and the implementation of the FSM-based functionality, like the control's hierarchy and the partitioning rules, are LHCb's responsibility.

LHCb will distribute to its subdetector developers a specialized version of the framework, tailored for the needs of the LHCb experiment. This specialized version will include LHCb's naming conventions, color codes, etc. and will also extend the framework with components designed to control LHCb specific devices like the CC-PCs described below.

IV. DAQ AND TRIGGER CONTROL

LHCb's DAQ system [6], including the timing and trigger systems, the front-end electronics, the readout chain, and the event-building network, will be composed of thousands of electronics boards or chips. These electronics have to be initialized, configured, monitored, and operated. There are two basic categories of electronics.

- Electronics boards or chips close to the detector in the radiation area. This electronics has been designed with the radiation constraints in mind and require only the I2C and JTAG protocols to access chips.
- Boards in counting rooms (no radiation), these boards can make use of large memory chips or processors and they require I2C, JTAG, and a simple parallel bus to access the board components.

The architecture devised for the control of electronics is represented in Fig. 6. All electronics equipment will contain a slave interface (S) providing the necessary protocols: I2C, JTAG, and a simple parallel bus. When there is a need to control electronics located directly on the detectors, where radiation levels can be high, I2C and JTAG are driven over approximately 10 m, from the board containing the slave interface to the chips on the detector. This avoids the necessity of radiation-hard slave interfaces, since they only have to be radiation tolerant. The slave interfaces are then connected via a master PCI board (M) to a PC. Depending on the protocol there might be the need for an Intermediate (I) board to transform the long-distance protocol into the short-distance protocol.

One important requirement for the slave interface is that resetting the slave part on the board should not perturb datataking activities, i.e., it should not induce signal variations that might disturb the rest of the board's components.

Three solutions have been agreed by the collaboration for interfacing electronics to the control system, the SPECS or the ATLAS ELMB for the radiation areas and credit card-sized PCs for nonradiation areas.

A. The SPECS

The Serial Protocol for Experiment Control System (SPECS) [7] is an evolution of the ATLAS Serial Protocol for the ATLAS Calorimeter (SPAC). The SPECS slave has been improved for radiation tolerance and the SPECS Master for increased functionality. The SPECS protocol can transfer data at rates up to 10 Mbit/s. The SPECS slave is made radiation tolerant and single event upset (SEU) tolerant by using an antifuse FPGA and implementing triple voting on all necessary registers. The SPECS Master card is a PCI card implementing four SPECS interfaces (i.e., it can drive four SPECS buses). The SPECS specifies the use of an intermediate board to translate the long-distance protocol (~100 m, from the counting room where the PC is to the other side of the wall) into the short-distance protocol (a few meters) to the SPECS slaves.

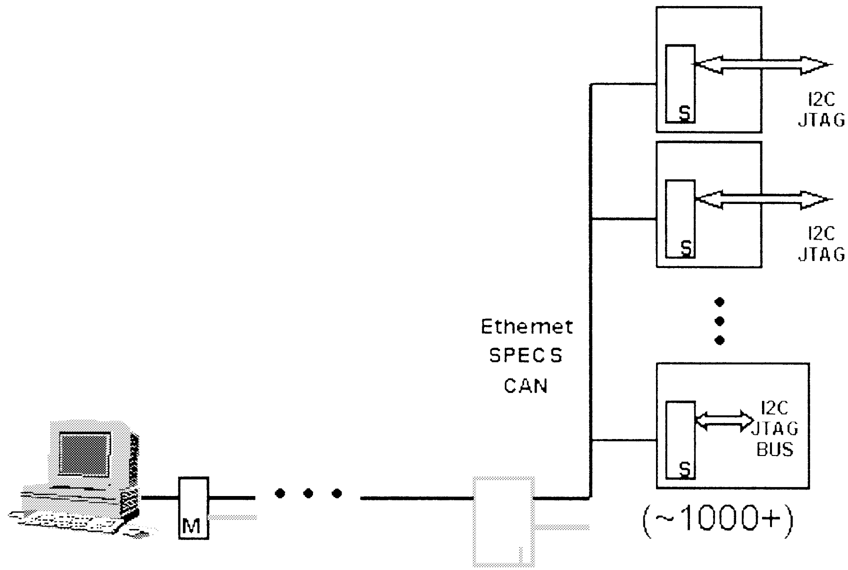

Fig. 6. Schematic view of the control path into electronics boards.

B. The ATLAS Embedded Local Monitoring Box (ELMB)

The ATLAS ELMB [8] is based on microcontrollers and uses the CAN bus as an interface. The ELMB contains 64 multiplexed ADC channels and was originally designed as an I/O device for analog and digital values. Since it outputs I2C and JTAG it can also be used to control electronics. The CAN bus has a bandwidth of 500 kbit/s for the envisaged length of the bus ($\sim$100 m). The ELMBs mechanism for coping with small doses of radiation is to have two microcontrollers, which can reset each other in case of problems. Any commercial CAN Master PCI card can be used to control the CAN branch. The ELMB has some degree of intelligence. Its microcontroller can be programmed to execute user code, for example to monitor FPGA code against SEUs. This feature will be used with moderation for two reasons: the development environment is complex and the microcontroller program can suffer itself from SEUs.

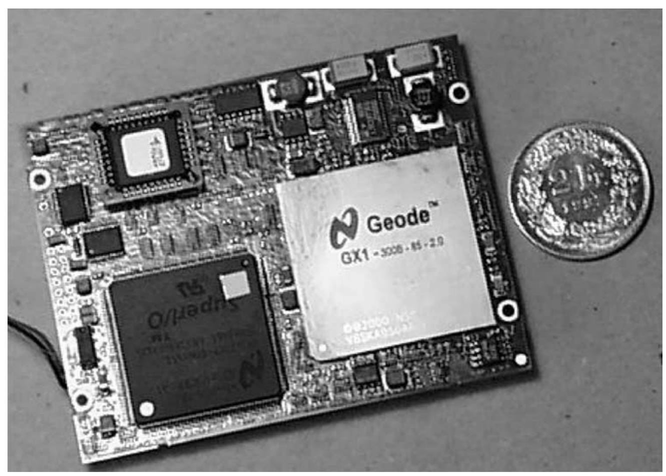

Fig. 7. Photograph of a CC-PC. A two-franc coin is shown for size comparison.

C. Credit Card PCs (CC-PCs)

CC-PCs [9] will be used to control electronics in counting rooms. The electronics in the counting rooms are normally VME sized boards (9U $\times$ 400 mm). The solution adopted is to have point-to-point links to each board via Ethernet and to install on each board a commercial credit card-sized ($66 \times 85 \times 12$ mm^3) PC. The CC-PC (Fig. 7) contains an Intel Pentium compatible CPU and up to 64 MB of memory. It interfaces to I2C, JTAG and a local parallel bus using a simple adapter card. These CC-PCs will run Linux and will be booted remotely via the network.

V. Event Filter Farm (EFF) Control

The EFF, implementing the high level triggers, will make use of commodity items, it comprises a few thousands standard PCs and its control does not need dedicated hardware developments. The EFF is organized in branches, called subfarms. The Sub-Farm Controllers (SFC) will receive event fragments from the event builder switch, assemble them into complete physics events and dispatch them to a free CPU for processing.

Each CPU in the farm, including the SFCs will have an independent Ethernet connection for control purposes separated from the data path. The architecture of the EFF control is represented in Fig. 8.

The Control PCs (at the bottom of the picture) connected to one or more branches of the EFF will be responsible for downloading the correct software into each CPU and for monitoring their operation, including the monitoring and control of the physics/trigger algorithms.

For this purpose, LHCb's offline software framework (GAUDI) has been interfaced to the control system, so that counters, errors, histograms, etc. produced by the physics algorithms can be visualized by the operators using the tools provided by the control framework.

The control of the EFF will be completely integrated in the Experiment Control System.

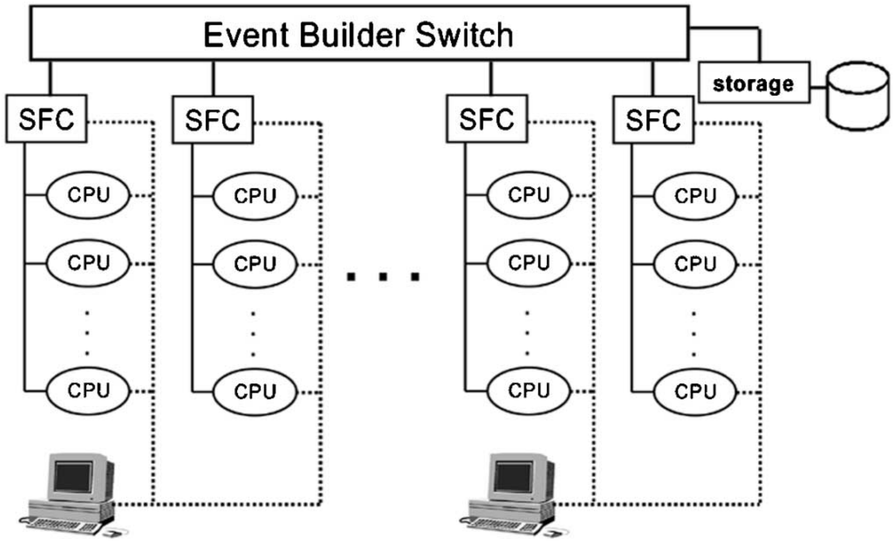

Fig. 8. Schematic view of the control of the EFF.

VI. Detector Control

Another large part of LHCb's control system is the interface to all the equipment involved in the DCS. These include high voltage and low voltage power supplies, temperature and humidity sensors, and many other I/O devices used for calibration, alignment, mechanics, etc.

These devices are integrated into the control system via a PCI card on a PC. Either directly if they come with a dedicated PCI card, using a fieldbus (for simple analog or digital I/O nodes or more complex like the ELMB) or using a Programmable Logic Controller (PLC).

The generic architectural options for the control of detector equipment are shown in Fig. 9.

The choice of this equipment is largely the responsibility of the subdetector teams due to their specific requirements, but aiming for standardization, the following guidelines have been adopted by all LHCb detector groups for the control of this type of equipment.

- Commercial equipment will be used as much as possible.
- ELMBs will also be used for large number of I/O channels or when radiation tolerance is required.
- The hardware interface to the equipment should be one of the CERN recommended fieldbuses: Profibus, CAN, WorldFip, or Ethernet. Devices should be accessible via a PCI card on a PC, not via VME.
- The software interface to the equipment should be an OLE for Process Control (OPC) server, preferably delivered by the hardware manufacturer.
- PLCs will be used whenever fast control loops are needed or whenever the safety of the system requires it. The CERN recommended manufacturers are Schneider and Siemens.

In anticipation of the choice of the subdetectors some equipment is already being integrated in the framework as ready-to-use components: this is the case of CAEN high-voltage power supplies, ISEG, and WIENER low-voltage supplies and the ATLAS ELMB for analog and digital I/O.

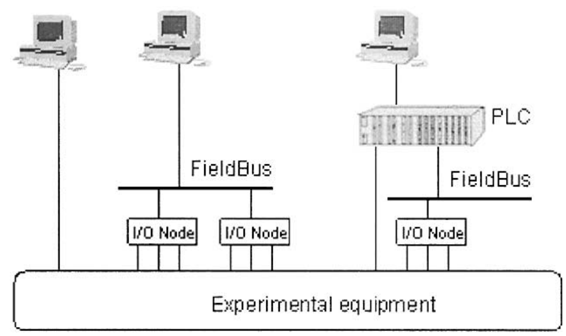

Fig. 9. Schematic view of the connection to DCS type devices.

The DCS is the area that will make the largest use of JCOP common developments.

VII. Infrastructure Control

The experimental infrastructure and environment has also to be monitored and when possible controlled, this includes:

- monitoring environmental parameters in the counting rooms and experimental halls (temperatures, humidity, radiation levels, etc.);
- monitoring and controlling the racks and the crates containing the electronics;
- monitoring and controlling the cooling and ventilation both centrally (for example for the racks) and inside the subdetectors;
- monitoring the electricity distribution;
- monitoring the LHCb magnet;
- monitoring LHCb's gas systems;
- configuring and monitoring the Detector Safety System.

Most of these subsystems are being developed by separate teams at CERN, also in common for the four experiments.

The philosophy adopted is similar for all external systems: the implementation, support, and maintenance of the service will be assured by the team providing the service (this includes a 24-h intervention team in case of problems) but the high-level operation (start, stop, change parameters) is to be provided by the experiment's control systems. All external systems will be interfaced to the ECS via a common, network based, protocol.

The information gathered by the Infrastructure and Environment control subsystem has to be stored and will be used to take decisions in case of problems, for example cutting the power to crates or racks (in an orderly and organized manner) if the temperature increases or the cooling stops, etc.

VIII. Conclusion

In this paper, we present the architecture and tools used by the LHCb experiment in order to implement a homogeneous, integrated control system. LHCb took a common, generic approach to the control of all types of devices from physical devices like high-voltage power supplies or electronics boards to software entities like physics algorithms. From a logical point of view all devices need to be configured, controlled and monitored, and integrated into higher level deciding entities which are responsible for the correlation of events and for the overall coordination, automation, and operation of the full experiment in its different running modes.

References

[1] *LHCb Tech. Proposal*, 1999. LHCb collaboration.

[2] A. Daneels and W. Salter, "The LHC experiments joint controls project, JCOP," in *Proc. Int. Conf. Accelerators and Large Experimental Physics Control Systems*, Trieste, Italy, Oct. 1999, pp. 633–636.

[3] S. Schmeling *et al.*. Controls framework for LHC experiments. presented at Proc. 13th IEEE-NPSS Real Time Conf.. [CD-ROM]ISBN 0-7803-7986-1

[4] PVSS-II [Online]. Available: http://www.pvss.com

[5] B. Franek and C. Gaspar, "SMI++—An object oriented framework for designing distributed control systems," *IEEE Trans. Nucl. Sci.*, vol. 45, pp. 1946–1950, Aug. 1998.

[6] J.-P. Dufey *et al.*, "The LHCb trigger and data acquisition system," *IEEE Trans. Nucl. Sci.*, vol. 47, pp. 86–90, Apr. 2000.

[7] D. Breton and D. Charlet, "SPECS, the serial protocol for the experiment control system of LHCb,", LHCb Tech. Note 2003–004, 2003.

[8] B. Hallgren, H. Boterenbrood, H. J. Burckhart, and H. Kvedalen, "The embedded local monitor board (ELMB) in the LHC front-end I/O control system," in *Proc. 7th Workshop Electronics for LHC Experiments*, Stockholm, Sweden, Sept. 10– 14, 2001, pp. 325–330.

[9] C. Gaspar, B. Jost, N. Neufeld, and S. Schmeling, "The Use of Credit Card Sized PC's for Interfacing Electronics Boards to the LHCb ECS,", LHCb Tech. Note 2001-147, 2001.