

Detector Control Systems in the LHC Experiments.

Joseph E. Rothberg

Physics Department, University of Washington, Seattle, WA, 98195
rothberg@cern.ch

Abstract

Architectures chosen for their Controls systems by experiments, front-end interfaces, and features of the control system will be described. The control system chosen for the LHC experiments is a powerful commercial product, PVSS, which supports distributed processing and includes tools for object-like device description, archiving, alarms, user-created panels and scripts. The Joint Controls project in the ITCO group at CERN is developing many commonly needed tools: a Framework, interfaces to power supplies, configuration tools, etc. The DCS group within ATLAS has developed a Local Monitor Board (ELMB) for data acquisition of temperatures and voltages, and for transmission of digital signals for front-end hardware. Examples of control systems and devices will be given.

The Joint Controls Project (JCOP) is part of the CERN IT/CO group and is a collaboration between the experiments and CERN support groups such as Gas, Cryogenics, DSS, LHC, etc. One of its purposes is to provide components of DCS systems and in particular a “Framework” for DCS. The JCOP project leader is Wayne Salter. Information on the projects of the IT/CO group headed by David Myers can be found in [1].

B. SCADA

A Market Survey and evaluation of Supervisory Control and Data Acquisition Systems (SCADA) was carried out at CERN in 1999. As a result of this process PVSS-II was chosen as the SCADA system for the LHC experiments. PVSS is produced by an Austrian company, ETM. [2]. Information on PVSS can be found at [3].

I. INTRODUCTION

Detector control systems involve control, configuration, readout, and monitoring of hardware devices as well as monitoring of external systems such as LHC, the electrical system, gas system, etc. An important aspect is communication with the data acquisition system (DAQ) for those experiments in which DCS and DAQ are separate systems. DCS must also provide logging of data and storage in a database. DCS systems will implement Finite State Machine (FSM) behaviour and have partitioning capability.

The DCS system may be viewed as a collection of **components** such as front-end devices, drivers, communications packages, FSM, a Framework, and a user interface. An alternative view is in terms of **functions** to be performed: readout, control, front-end configuration, monitoring, alarm reporting, partitioning, and data storage.

A. DCS support at CERN

In each of the LHC experiments a DCS group has been established. The group leaders are listed in Table 1.

Table 1: DCS groups

Experiment	
ATLAS	Helfried Burckhart
ALICE	Lennart Jirden
CMS	Joao Varela
LHCb	Clara Gaspar

II. DCS ARCHITECTURE

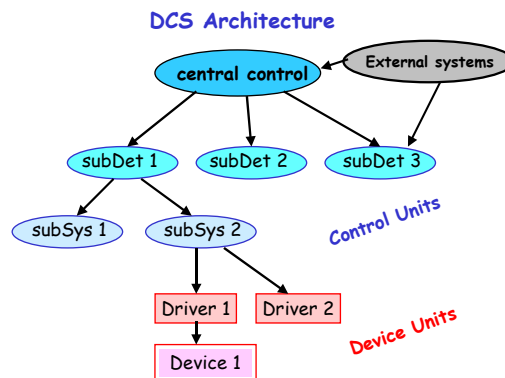


Figure 1: DCS Architecture

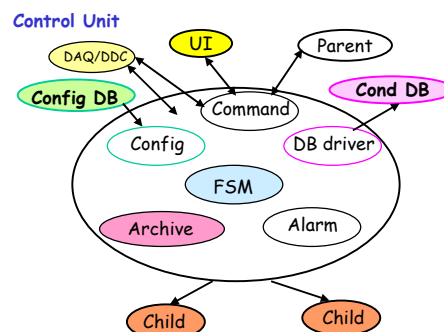


Figure 2: Control

A. Example Architectures

The DCS architecture for an experiment can be viewed as being composed of Control units and Device units linked in a tree-like structure under a main controller as shown in Figure 1. The Control units implement FSM behaviour and can be configured, archive data, issue alarms, and communicate with DAQ as in Figure 2. The Device units have more limited functionality (no communication with DAQ or with the Conditions database) but include a driver for hardware devices.

In the case of LHCb both the DCS and DAQ are part of the same Experiment Control System (ECS) as seen in Figure 3; the same software tools and components are used throughout and the overall control system is more homogeneous and easier to develop and maintain. Normally status information and alarms are sent upward in the tree and commands are sent downward toward the device units.

LHCb Generic SW Architecture

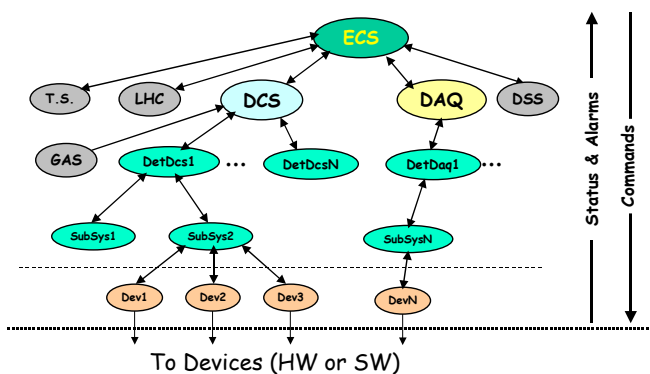


Figure 3: The LHCb Experiment Control System

The control system for the CMS ECAL sub-detector is shown in Figure 4.

In the ATLAS control system seen in Figure 5 there is a hierarchy of control stations for subsystems and for entire subdetectors. External systems and the DAQ communicate with DCS via data, commands, and message transfer protocols. These systems communicate both with the global control station and with subdetector control stations. There are connections at several levels with the configuration and conditions databases.

The control systems will be implemented in PVSS with some components (Framework, FSM tools, drivers) being developed at CERN. PVSS provides an object-like data structure with “datapoint types” created by the user to model the complexity of devices and data structures in the experiments. “Datapoints” are instances of these objects and can easily be created in large number as needed by the application. These structures will be described in more detail

in Section IV. The usual features of SCADA systems are also available: alarms, archiving, GUI’s, and a scripting language.

CMS ECAL

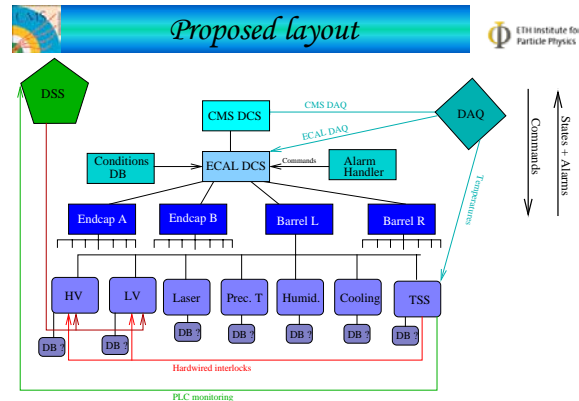


Figure 4: The CMS ECAL control system

Finite state machines will be operational in control and device units. The FSM will ensure that, for example, the HV state of a sub-detector is appropriate depending on external conditions. Figure 6 shows that the running state depends on inputs from the LHC, the gas system, HV, and the run control of the DAQ. FSM capabilities will be part of the PVSS Framework using the SMI++ tool kit [4]. Both DAQ and DCS will support partitioning so that individual subsystems can be included or excluded from the control hierarchy and readout. More complex partitioning options are also possible. Partitioning will be essential for the commissioning phase of detector operation.

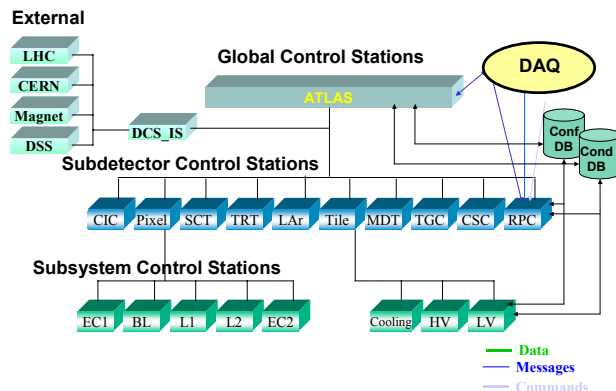


Figure 5: The ATLAS DCS system showing a hierarchy of control stations and communication with external systems and DAQ.

FSM with External Inputs

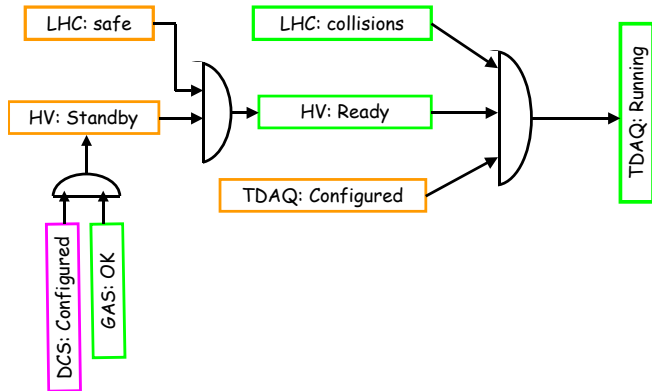


Figure 6: Example of Finite State Machine with external inputs.

B. Communications

Several communications protocols are already in use and others are planned for the LHC experiments.

1) DAQ to DCS (DDC)

Communications in ATLAS between DAQ and DCS is handled via the DDC system sketched in Figure 7. This has been used in test beams for several years; it permits exchange of data, the sending of commands from DAQ to DCS, and the reporting of messages (for example, error conditions) from DCS to DAQ. On the DCS side the communications package is interfaced directly to PVSS while on the DAQ side various components of the on-line system are used.

Complete documentation is available as part of ATLAS on-line software. In LHCb the DAQ and DCS are integrated so that some communications can be handled by PVSS itself.

2) Distributed Information Management (DIM)

The DIM system was originally developed for the DELPHI experiment at LEP and has been heavily used in the ATLAS Muon system test beam and in LHCb. Figure 8 shows that it allows inter-process communication between processes running on Windows systems and on Linux; interfaces in both C and C++ are available. Data structures in very flexible formats can be exchanged between servers and clients; data can be sent on request or by subscription. A PVSS “manager” is available which permits exchange of data by a very simple mechanism between datapoints in PVSS and other processes running on different platforms. A nameserver process establishes the communication channels between clients and servers. A very complete manual with examples is available on the web [5].

3) Data Interchange Protocol (DIP)

Data exchange will be needed between the control systems of the experiments and various external systems including LHC, the electrical distribution system, cooling, magnet systems, cryogenics, Technical Control, etc. Requirements for this have been defined and at present several implementations are being reviewed for selection in the near future. The review is being carried out under the Joint Controls Project.

4) PVSS Distributed Systems

Distributed PVSS systems provide a mechanism for exchange of data between DCS processes running on different platforms. The values of datapoint elements set in one system are visible on other systems that are part of the distributed network.

(DAQ to DCS Communications)

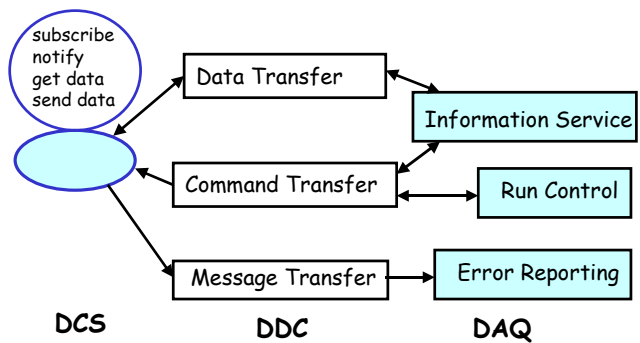


Figure 7: DAQ to DCS communication (DDC)

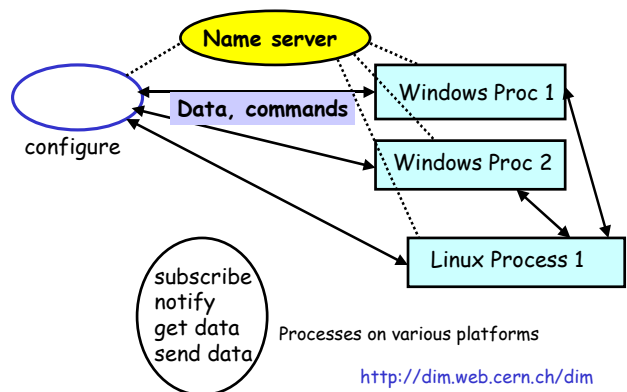


Figure 8: DIM communication

III. FRONT-END DEVICES

The front-end devices that are centrally supported include CAEN power supplies SY1527 and SY2527, the ISEG power supply, Wiener power supplies and fan trays, and the Extended (or embedded) Local Monitor Board (ELMB) and its CAN network.

Figure 9 shows some of the front-end devices that are being used in part of the ATLAS Muon test beam setup in 2003. Other components in use but not shown include high voltage power supplies and additional digital connections to front-end electronics cards. Analogue parameters such as temperatures and outputs of gas flow and pressure measuring equipment are recorded with the ELMB developed in ATLAS but widely used. Alignment devices are read by two different mechanisms: in one system a frame grabber is used while another system uses specially designed circuits on VME boards to read out and digitise CCD's and to illuminate them.

Details on the use of front-end devices in DCS systems will be presented in several talks in sessions A3 and B3.

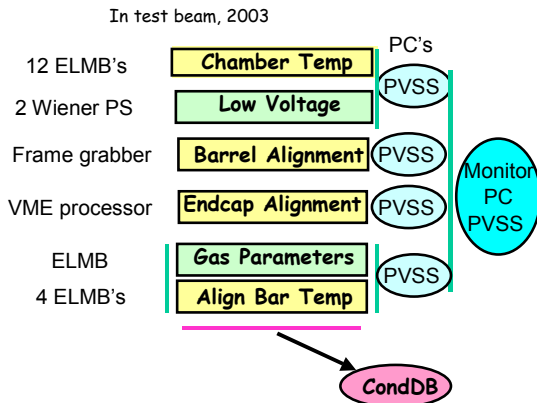


Figure 9: Front-end devices for ATLAS Muon Test Beam, 2003

A. The ELMB

The ELMB will be described in detail in a presentation by H. Boterenbrood in session A3. Here some of the features and applications are summarised briefly. The ELMB is based on an ATMEL microcontroller ATmega128 (4 MHz). The CAN controller is based on a SAE81C91. The ELMB board holds a multiplexed 16 bit ADC with 64 channels. In the CERN version of the packaging the input voltages may be conditioned with plug-in attenuators and other signal adapters. The readout can be periodic, polled, “on-change”, and “on-leaving-window”. In addition to the general purpose ADC, readout of two magnetic field probes (three components of the fields) is provided in the NIKHEF implementation. The ELMB also provides digital input and output lines which will be used for commands and status communication with the front-end electronics cards. An optional DAC can be added. The ELMB features low power consumption, opto-isolation, and radiation-tolerance.

Links to ELMB hardware and software can be found on web sites [6], [7].

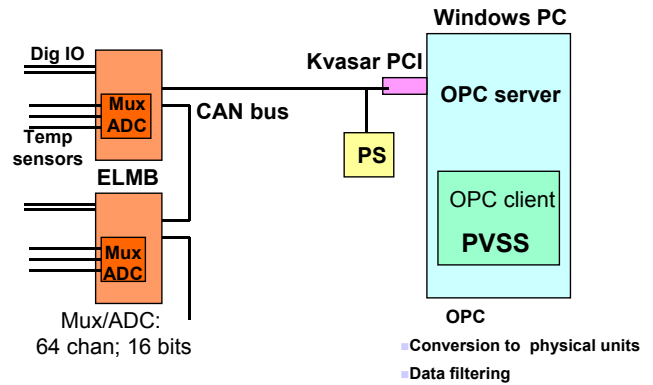


Figure 10: ELMB configuration

A simplified diagram of ELMB's on a CANbus is shown in Figure 10. The Kvasar PCI CAN card [8] has been selected as a standard and an OPC client/server makes the data available to PVSS. Figure 11 shows the temperature readout arrangement that is being used in a test beam. Twelve ELMB's are being read on two 50 meter CAN buses going to ports of a Kvasar card. Each ELMB (one per MDT chamber) reads about 20 temperature sensors. Data are stored in PVSS datapoints and are directly sent to a Conditions Database implemented in MySQL on a Linux computer.

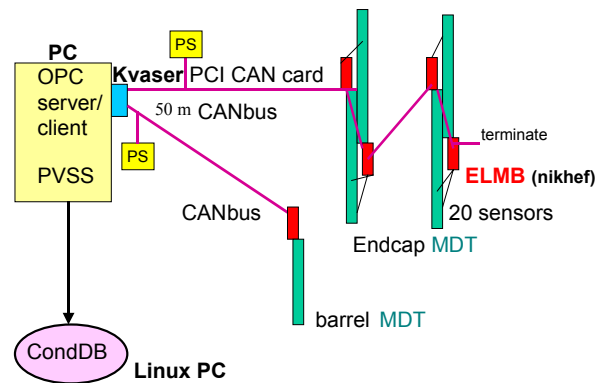


Figure 11: ELMB configuration for Temperature readout

Typical performance figures for ELMB's on a CAN bus based on recent tests are shown in Table 2

Table 2: ELMB performance

CANbus rate	125 kb/second
ADC conversion rate	15 Hz
Single ELMB readout (64 channels)	4 seconds
Multiple ELMB's per CANbus	>~ 4 seconds
32 ELMB's/bus; 6 buses	30 seconds

The readout rate of a single ELMB is determined by the ADC conversion rate but for many ELMB's on a single bus channels from different ELMB's are interleaved so that the overall readout time is not increased by a large amount. For the case of 32 ELMB's per bus it appears that the readout speed is limited by the speed of PVSS "managers" in PVSS version 2.12. In the test that measured this all the buses produced values simultaneously which is a "worst case" [9]. It is also expected that PVSS version 3 will have improved performance. In the ATLAS Muon MDT system now being designed for ATLAS scale it is expected that approximately 32 CAN buses with 32 ELMB's per bus will be used but this arrangement is still in the testing stage. Data will be recorded "on-change" and on leaving a window but the limits and expected data rates are not yet known. A photograph of the NIKHEF implementation of the ELMB is shown in Figure 12. A diagram of connections between the ELMB and the front-end electronics card used in the ATLAS muon system is given in Figure 13 and a more detailed sketch by H. Boterenbrood of connections to the front-end is shown in Figure 14. The most important function of the ELMB in this application will be to download thresholds and other configuration parameters to the chamber services module (CSM) via JTAG and thence to the 18 on-chamber readout cards (mezzanine cards). The same ELMB will also record voltages, currents, and temperatures at the front-end via an ADC on-board the CSM. This ADC is identical to the one on the ELMB board.

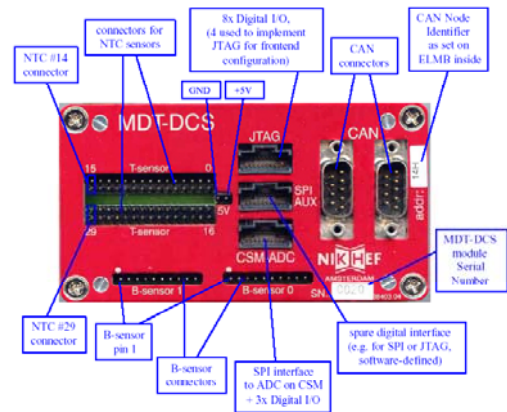


Figure 12: ELMB (NIKHEF)

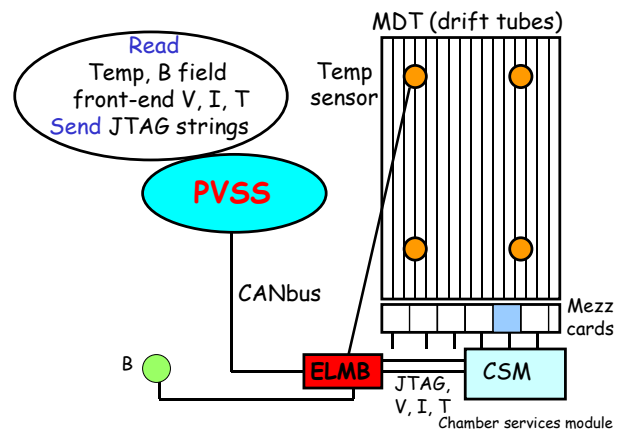


Figure 13: ELMB connected to MDT Chamber Services Module

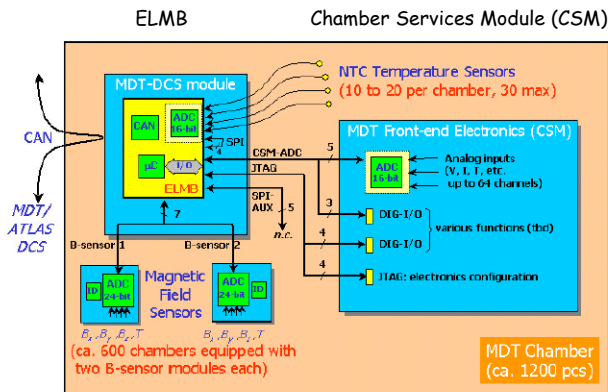


Figure 14: Details of ELMB to CSM connection.

In another interesting application the ELMB has been programmed to record and histogram charge collected on the TGC detector wires. The advantages are that all of the ELMB's are doing the processing in parallel and the bus traffic is reduced. Such histograms of chamber charge taken during test beam running are shown in Figure 15.

A detailed report was presented one year ago [10].

Histogram of analog chamber charge - H8

- One wire analog channel per chamber is read, integrated on DCS-PS and histogrammed on the ELMB

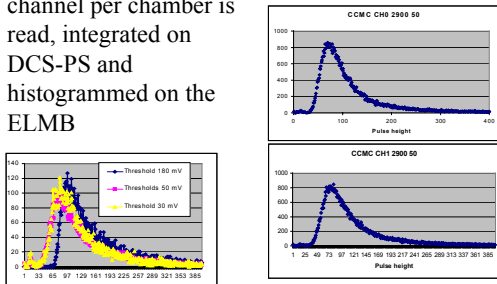


Figure 15: ELMB used for Histogram collection.

B. SPECS

The Serial Protocol for Experiment Control System [11] is used by LHCb for communication with front-end hardware. An overview is given in Figure 16

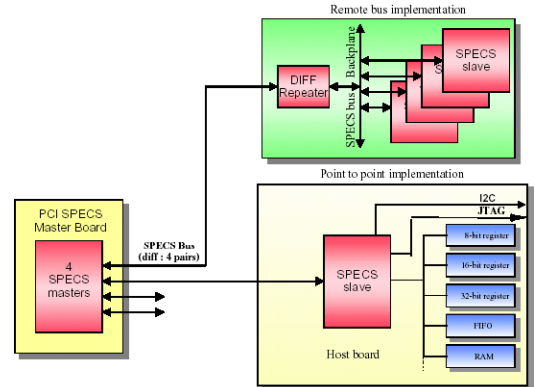


Figure 16: SPECS

C. Wiener Power Supplies

Figure 17 shows a configuration of two Wiener power supplies on a CANbus connected to the same four port Kvaser card as ELMB's

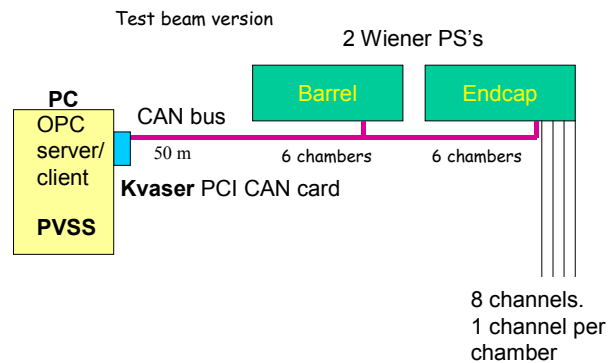


Figure 17: CANbus for Wiener Power Supplies

D. Alignment Systems

Several different alignment systems used in ATLAS provide examples of non-standard front-end components which are interfaced to the DCS system for configuration, control, and monitoring.

In the RASNIK system used in the Muon endcap and barrel regions a coded mask with a checkerboard pattern is

viewed by a CCD camera. The coded regions give a coarse location on the mask and the boundaries of the squares are used to determine a very precise position. Figure 18 shows a reconstructed mask image and the optical system; the magnification is normally close to 1.0 and the square sizes are between 120 and 340 microns with a pixel size of 10 microns giving an intrinsic precision below a micron. The optical devices are mounted on the drift chambers and on nearby structures and permit correction of chamber positions for tracking to the very high precision needed for momentum measurement. The design single MDT tube resolution is 80 microns and the error from the alignment should not exceed 30 or 40 microns.

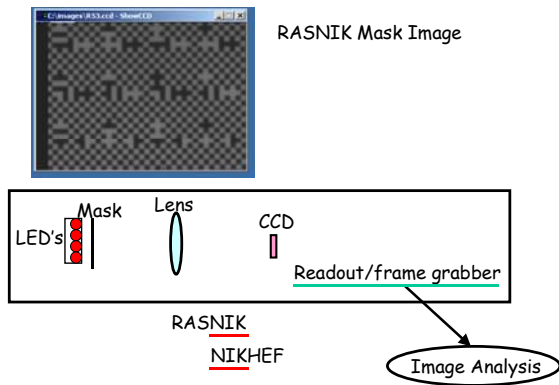


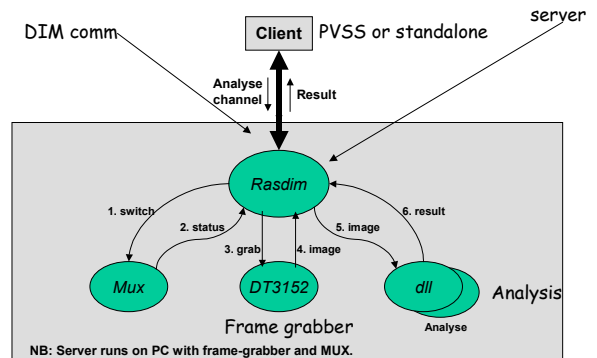
Figure 18: RASNIK components for Alignment

The pattern is acquired with a framegrabber (for the barrel devices) and by specially built CCD readout electronics (for the endcap region) [12].

points to detailed notes on the endcap system. Additional position information is obtained from patterns of light spots read out in a similar way. The readout architecture for the barrel devices is shown in Figure 19. It is based on a framegrabber card in a Windows PC. The server component interacts with the hardware devices via a multiplexer and sends the image to an analysis task which may run on another platform. The upper level client program is based on PVSS and all communications between tasks is via the DIM system.

ATLAS Muon Barrel Alignment

PVSS and client/server model
Architecture is similar to Endcap



Robert Hart, NIKHEF

Figure 19: Alignment readout architecture (Muon Barrel)

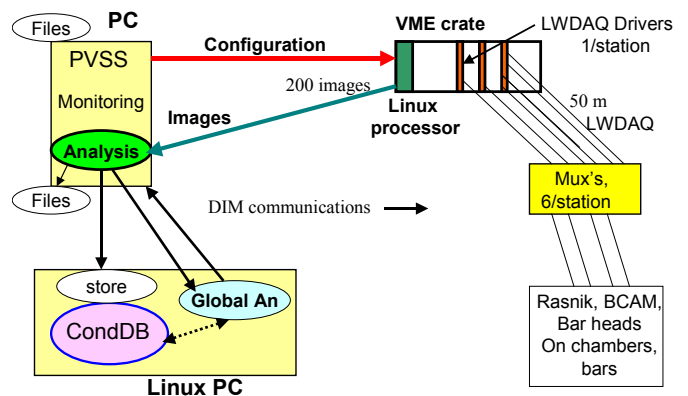


Figure 20: Alignment Readout architecture (Muon Endcap)

The endcap hardware organization [12] which is different is sketched in Figure 20.

points to detailed notes on the endcap system. Here the CCD readout boards (LWDAQ Drivers) and multiplexers are controlled by a VME Linux processor while a PVSS process on a Windows PC is responsible for configuration and monitoring. The image analysis program runs as a separate thread on the PC and receives the images (83 kB) via DIM from the embedded VME processor at a rate of about one image per second. Analysis results are sent via DIM to the PVSS monitoring task, to a conditions database storage process, and to a global reconstruction program which determines chamber positions in three dimensions in real time. In the test beam setup about 200 different optical images are acquired and analysed every four minutes while in the ATLAS endcap about 10000 images will have to be processed on a time scale fast enough to respond to temperature variations.

A typical PVSS monitoring panel is shown in Figure 21; this includes histograms of analysis failures, tables of results, push buttons to obtain more detailed information, and status “LED’s”. Such a panel was adequate for a system of 200 optical images but for LHC scale summary and hierarchical panels will be needed. Designing appropriate panels for the eventual device multiplicity needed will take some thought.

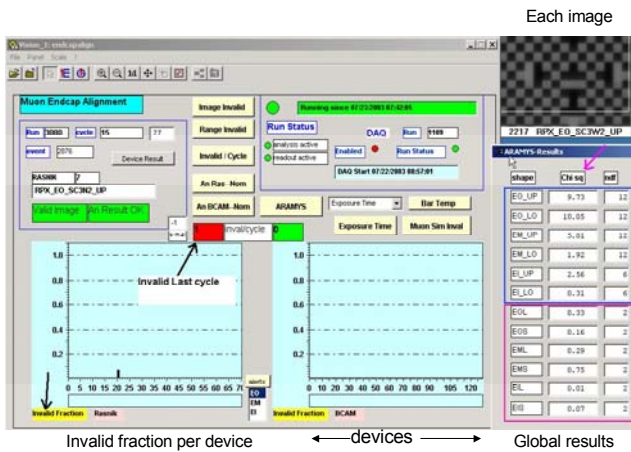


Figure 21: A PVSS monitoring panel (Endcap Alignment)

IV. PVSS

A. PVSS Distributed Systems

A crucial feature of PVSS is the ability to create distributed systems of PVSS projects. A “beta” version 3.0 of PVSS was released in August 2003. One of its most important features is the ability to set up distributed systems in a simple way. Once a system is configured as being distributed all of its datapoints are visible to other systems in the same cluster. However there is no traffic across the network unless a specific datapoint is displayed, modified, or a request is made to signal a change in a datapoint value (dpConnect in PVSS terminology). At LHC very large distributed systems will be needed. An example of a portion of a distributed system is shown in Figure 22. Data from lower levels can be monitored at higher levels in the tree. Version 3 also allows panels to be displayed in a web browser without installation of PVSS. When connecting the panels and required parts of PVSS are automatically downloaded.

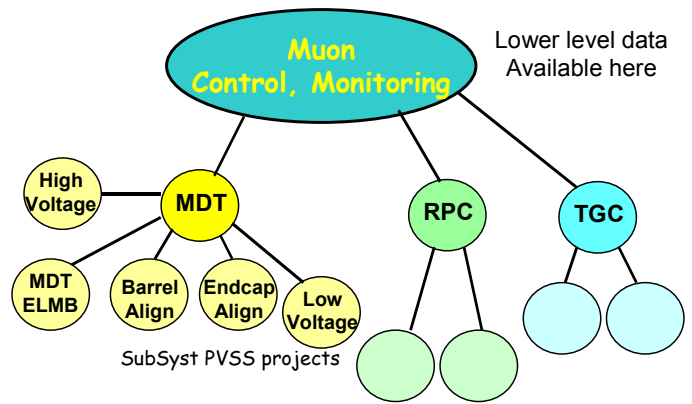


Figure 22: An example of a PVSS distributed system

B. Data Archive

In the present version of PVSS data are archived in a proprietary database which is not readable from outside PVSS. In PVSS version 3.0 to be released at the end of November 2003 there will be an option to archive data in Oracle. This will permit non-DCS applications to access the data. The relationship between this archive and the Conditions database has yet to be designed. The Conditions Data Base will contain all parameters needed for data analysis and diagnostics. DCS will provide to it most of the information that is read and in addition a condensed summary of the status of each system (for example, HV, gas, LV).

C. Aliases and Views

The hardware view of a controlled/read item might be Wiener Power Supply 2, channel 3. However the non-expert user should be able to refer to that channel by its logical or geographical name, for example, MDT chamber EML3. This possibility will be available in the framework and uses the PVSS feature called datapoint aliases.

D. Device Modelling

One of the strong points of the PVSS system is the ability to model devices and data structures with complex "Datapoint types". Instances of these objects then represent the actual devices. Functions provided by the system allow the user to access these instances in scripts and to be notified (via a call-back to a user function) on change of a value. In Figure 23 an example datapoint type “Rasnik” for a Rasnik image is shown along with two instances, R1 and R2. The Rasnik datapoint type includes configuration information, analysis output and statistics on the analysis results; these are themselves datapoint types whose structure is shown in Figure 24. When a datapoint type is changed (for example by adding more nodes) the instances automatically inherit the changes.

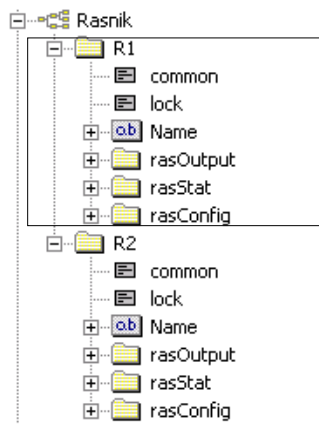


Figure 23: Data point type

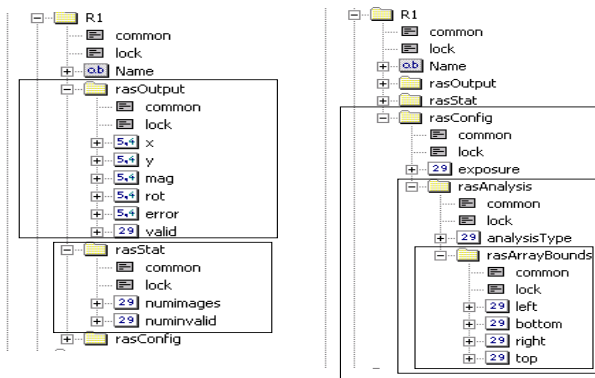


Figure 24: Data point type details

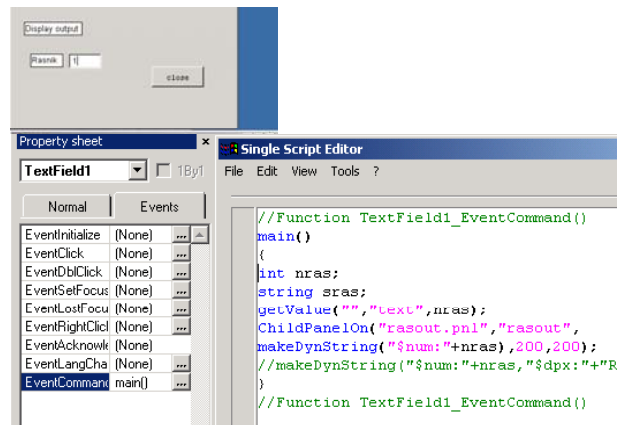


Figure 25: An input panel and its script

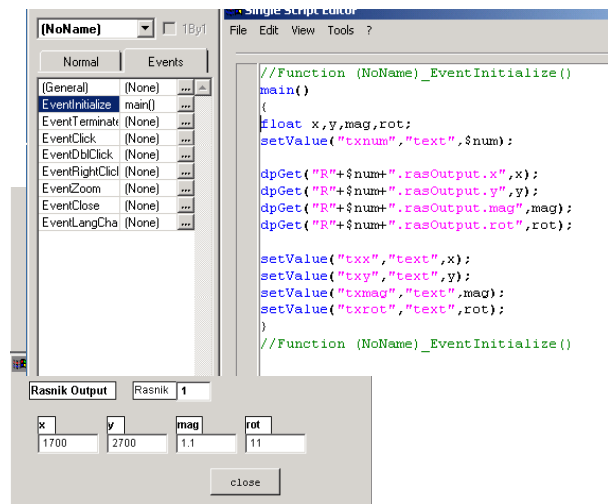


Figure 26: An output child panel and its script

V. ACKNOWLEDGEMENTS

Two very simple panels using these datapoints are shown in Figures 25 and 26. Also shown are the complete scripts needed to receive user input on the first panel and to display the analysis output for the corresponding device on the second “child” panel. The scripting language is C but usage depends heavily on PVSS functions such as “dpGet”, “setValue”, “getValue”, etc. Parameters may be transmitted to generic child panels and objects with the “\$” syntax used in this example for the device number. The datapoint name is then constructed using the device number. Scripts may be attached to all available GUI actions and events such as “EventClick”, “EventInitialise”, etc. An adequate on-line help is available that describes all functions.

The author would like to thank: Wayne Salter (CERN-JCOP), Clara Gaspar (CERN-LHCb), Fernando Varela (CERN-ATLAS DCS), James Cook (CERN-ATLAS DCS), Shlomit Tarem (Technion-ATLAS), Henk Boterenbrood (NIKHEF), Robert Hart (NIKHEF) for very helpful discussions and explanations and for the use of diagrams.

VI. REFERENCES

- [1] The ITCO group. <http://itco.web.cern.ch/itco>
- [2] ETM. <http://www.etm.at>.
- [3] PVSS. <http://www.pvss.com>.
- [4] SMI Tool kit. B. Franek and C. Gaspar, "SMI++ - An Object Oriented Framework for Designing Distributed Control Systems," in IEEE Trans. Nucl. Sci., Vol 45, 4, p.1946, August 1998.
- [5] DIM. <http://dim.web.cern.ch/>.
- [6]ELMB.
<http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DCS/ELMB/ELMBhome.html>

<http://elmb.web.cern.ch/ELMB/elmb128.html>
- [7] ELMB
<http://www.nikhef.nl/pub/departments/ct/po/html/ELMB/ELMB16.pdf>

<http://www.nikhef.nl/pub/departments/ct/po/html/ELMB/ELMBresources.html>
- [8] Kvaser. <http://www.kvaser.com>
- [9] ELMB Bus test. F. Varela Rodriguez, Systems of ELMB Buses using the Kvaser PCI CAN card,
ATLAS DCS Internal Note, DCS-IWN17 (2002).
- [10] TGC <http://lhc-electronics-workshop.web.cern.ch/LHC-electronics-workshop/2002/Poster/tarem.pdf>
- [11] SPECS D. Breton and D. Charlet, "The Serial Protocol for the Experiment Control System of LHCb," LHCb Technical Note 2003-004.
- [12] EndcapAlignment.
<http://alignment.hep.brandeis.edu/atlas/>