Jinst

PUBLISHED BY INSTITUTE OF PHYSICS PUBLISHING AND SISSA

RECEIVED: *January 22, 2008*
REVISED: *April 17, 2008*
ACCEPTED: *May 8, 2008*
PUBLISHED: *May 27, 2008*

The detector control system of the ATLAS experiment

A. Barriuso Poy,[a] H. Boterenbrood,[b] H.J. Burckhart,[a] J. Cook,[a] V. Filimonov,[c] S. Franz,[a] O. Gutzwiller,[a] B. Hallgren,[a] V. Khomutnikov,[c] S. Schlenker[a]* and F. Varela[a]

[a] *CERN, CH-1211 Genève, Switzerland*
[b] *NIKHEF, Kruislaan 409, 1098 SJ Amsterdam, The Netherlands*
[c] *PNPI, Gatchina, Leningrad reg., 188300, Russia*
E-mail: stefan.schlenker@cern.ch

ABSTRACT: The ATLAS experiment is one of the experiments at the Large Hadron Collider, constructed to study elementary particle interactions in collisions of high-energy proton beams. The individual detector components as well as the common experimental infrastructure are supervised by the Detector Control System (DCS). The DCS enables equipment supervision using operator commands, reads, processes and archives the operational parameters of the detector, allows for error recognition and handling, manages the communication with external control systems, and provides a synchronization mechanism with the physics data acquisition system. Given the enormous size and complexity of ATLAS, special emphasis was put on the use of standardized hardware and software components enabling efficient development and long-term maintainability of the DCS over the lifetime of the experiment. Currently, the DCS is being used successfully during the experiment commissioning phase.

KEYWORDS: Detector control systems (detector and experiment monitoring and slow-control systems, architecture, hardware, algorithms, databases); Software architectures (event data models, frameworks and databases).

2008 JINST 3 P05006

*Corresponding author

Contents

1. Introduction 2

2. Requirements 3

3. Overall system design 4

3.1 ATLAS operation model 4

3.2 System architecture 4

4. Hardware components 4

4.1 ELMB 6

4.1.1 ELMB Hardware 6

4.1.2 ELMB software 7

4.2 Standardized commercial systems 9

4.2.1 VME crates 9

4.2.2 Power supplies 9

4.2.3 DCS Control Station PC 10

5. Software components 10

5.1 PVSS 10

5.2 JCOP Framework 12

5.3 CANopen OPC server 13

5.4 The Finite State Machine toolkit 14

5.4.1 State Machine logic, structuring, and object orientation 14

5.4.2 Inter-Process Communication and Partitioning 15

5.4.3 Object persistency and user interfaces 16

5.5 Access control 16

5.6 Configuration database 17

5.7 Conditions database 17

5.8 Data visualization 18

5.8.1 PVSS trending 19

5.8.2 ROOT viewer 19

5.9 Connection to external control systems 20

5.10 DCS-TDAQ communication 21

5.11 Software management 21

6. Read-Out chain 23

7. Operations layer **23**
7.1 Operations model 23
7.1.1 The detector as Finite State Machine hierarchy 23
7.1.2 Event and error handling 26
7.2 Operator interfaces 28
7.2.1 Finite State Machine operator interface 28
7.2.2 Alarm screen 30
7.2.3 Remote access 31
7.2.4 Data visualization 32
7.3 DCS operation 33
7.3.1 Procedures for operation 34
7.3.2 Alarm response 34
7.3.3 FSM operation 34
7.3.4 Run Control operation 34

8. Subsystem control **35**
8.1 Sub-detectors 35
8.2 Infrastructure 36
8.2.1 Cavern and counting room environment 36
8.2.2 Infrastructure services 38
8.2.3 LHC accelerator 38
8.2.4 DSS 38

9. Conclusions **39**

1. Introduction

ATLAS is a general purpose High Energy Physics (HEP) experiment, scheduled to start data taking in 2008 at the Large Hadron Collider (LHC) at CERN, Geneva, Switzerland. Its scale is unprecedented in HEP, in both terms of size — the detector elements are distributed over a cylindrical volume of 25 m diameter and 50 m length — and organization — more than 2000 people of 167 institutions in 37 countries contribute to the project. The ATLAS experiment [1] consists of three detectors, the inner tracker, the calorimeter, and the muon system, each of which is composed of several sub-detectors. The complexity of these detectors, containing a huge number of individual detector elements — the Pixel sub-detector alone already accounts for 60 million electronic channels — presented new challenges for the design and the implementation of the Detector Control System (DCS) for ATLAS.

The DCS has the task to permit coherent and safe operation of ATLAS and to serve as a homogeneous interface to all sub-detectors and to the technical infrastructure of the experiment. The DCS must be able to bring the detector into any desired operational state, to continuously monitor and archive the operational parameters, to signal any abnormal behavior to the operator, and to allow manual or automatic actions to be taken. In order to synchronize the state of the detector with the operation of the physics data acquisition system, bi-directional communication between DCS and run control must be provided. Finally, the DCS has to handle the communication between the ATLAS sub-detectors and other systems which are controlled independently, such as the LHC accelerator, the CERN technical services, the ATLAS magnets and the Detector Safety System (DSS).

The Joint Controls Project (JCOP) [2] was founded in order to maximize synergy effects for the DCS of the four experiments at the LHC by using common DCS components. Within JCOP, standards for the use of DCS hardware were established and a commercial controls software product has been selected to serve as the basis for all DCS applications. This software package was substantially extended by a comprehensive framework of software components and implementation policies.

This paper focuses on the DCS architecture, common hardware components and standards, the design and implementation of the DCS back-end software, as well as operational aspects. A detailed description of the specific sub-detector control system hardware can be found in [1] and references therein. The paper is structured as follows: after a short summary of the requirements, an overview of the system architecture will be given in section 3. The hardware and software components will be presented in detail in section 4 and 5 respectively, and the description of the data flow will follow in section 6. The overall operations layer is discussed in section 7, followed in section 8 by an overview of the individual controls applications with emphasis on the common infrastructure of the experiment.

2. Requirements

The ATLAS experiment is hierarchically organized in a tree-like structure of detectors, sub-detectors, sub-systems, etc. with each element having a certain level of operational independence. This must be reflected in design and implementation of the DCS. The timescale of more than three decades for the R&D, construction and exploitation phases requires the control system to be able to evolve from small, very flexible stand-alone systems for prototyping and component assembly up to the final size for integrated operation of the experiment as a whole. The DCS has also to support different modes of operation, ranging from stand-alone running of components, e.g. for maintenance or calibration, up to coherent physics data taking. Given the large variety of equipment to be controlled, the standardization of the hardware and of the software interfaces is of primary importance for the homogeneous control of all different detector components, enabling the development of a uniform operator interface as well as minimizing implementation and maintenance efforts. Due to the enormous number of individual channels to be supervised by the DCS — several 100.000 parameters in total — the design and implementation must provide a high level of scalability. As the DCS needs to interact with various control systems outside of ATLAS, flexible and platform-independent interfaces must be provided.

The experiment itself is located in a cavern 100 m underground and is not accessible during operation because of the presence of ionizing radiation. Therefore, the control system must be fault-tolerant and allow remote diagnostics. In particular, the dependence of low-level control procedures on external services such as computer networks should be kept as small as possible.

3. Overall system design

3.1 ATLAS operation model

The ATLAS experiment is operated by two collaborating systems: the DCS and the Trigger and Data-Acquisition (TDAQ) [3] system. The former constantly supervises the hardware of the experiment and its infrastructure. The latter performs the read out of detector data generated by proton collisions — the “physics events” — and directs the data streams from the digitizers to mass storage. Thus TDAQ operates only during physics data taking periods for which it also takes control of the DCS. The DCS is responsible for continuous monitoring and control of the detector equipment and is supervised by a human operator in the control room.

3.2 System architecture

For the implementation of the DCS, commercial solutions are used wherever possible. Within the frame of JCOP, a comprehensive market survey of industrial Supervisory Controls And Data Acquisition (SCADA) systems has been carried out and the product ProzessVisualisierungs- and SteuerungsSystem (PVSS II) [4] from the company ETM has been selected. PVSS is a device-oriented and event-driven control system which can be distributed over a large number of PCs running Windows or Linux as operating system.

Figure 1 shows the DCS architecture which is organized in Front-End (FE) equipment and a Back-End (BE) system. For the connection of the BE system to the FE, the industrial fieldbus CAN [5] has been chosen in most of the cases, while some devices are connected using Ethernet or Modbus. Whenever possible, the industry standard OPC [6] is used as the software protocol for communications. Otherwise, application-specific protocols have been developed.

The FE equipment consists of purpose-built electronics and their associated services such as power supplies or cooling circuits. In addition, a general purpose I/O and processing unit called Embedded Local Monitor Board (ELMB) [7] was developed, which can operate in the high magnetic field and under ionizing radiation in the cavern. It can either be embedded in the electronics of detector elements or directly read out sensors in a stand-alone manner. The BE is organized in three layers (see figure 1): the Local Control Stations (LCS) for process control of subsystems, the Sub-detector Control Stations (SCS) for high-level control of a sub-detector allowing stand-alone operation, and the Global Control Stations (GCS) with server applications and human interfaces in the ATLAS control room for the overall operation. In total, the BE consists of more than 150 PCs, connected as a distributed system for which PVSS handles inter-process communication via the local area network. The full BE hierarchy from the operator interface down to the level of individual devices is represented by a distributed Finite State Machine (FSM) mechanism allowing for standardized operation and error handling in each functional layer.

4. Hardware components

The ATLAS DCS uses the CAN industrial fieldbus and the CANopen [5] protocol for its FE I/O, where possible and appropriate. CAN was selected by the CERN fieldbus working group [8] due to it’s low cost, flexibility and especially due to its insensitivity to magnetic fields, in contrast to e.g. Ethernet which is rendered unusable by strong magnetic fields. Examples for the usage of CAN are

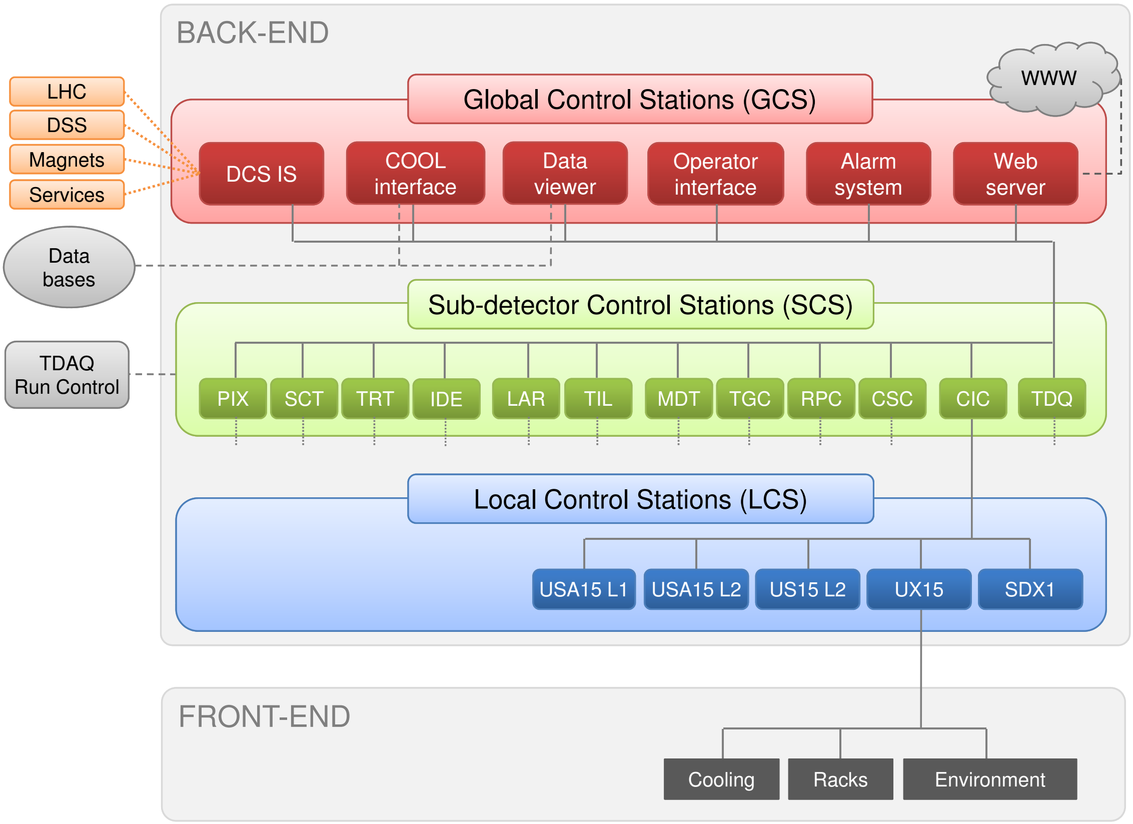


Figure 1. ATLAS DCS architecture.

the monitoring of environmental parameters such as temperature and humidity, the configuration and monitoring of detector FE electronics, and the control and monitoring of power supplies. Many of the electronics systems in and around the detector are custom designs, done by various groups from all over the world.

In general, important requirements for the DCS FE I/O are:

- low cost (i.e. use of commercial components),
- low power consumption,
- high I/O channel density.

If the FE I/O electronics is located in the detector cavern there are additional requirements to be met:

- remote firmware upgrades must be possible,
- insensitivity to magnetic fields,
- tolerance to the radiation levels present at that location, integrated over the lifetime of the experiment.

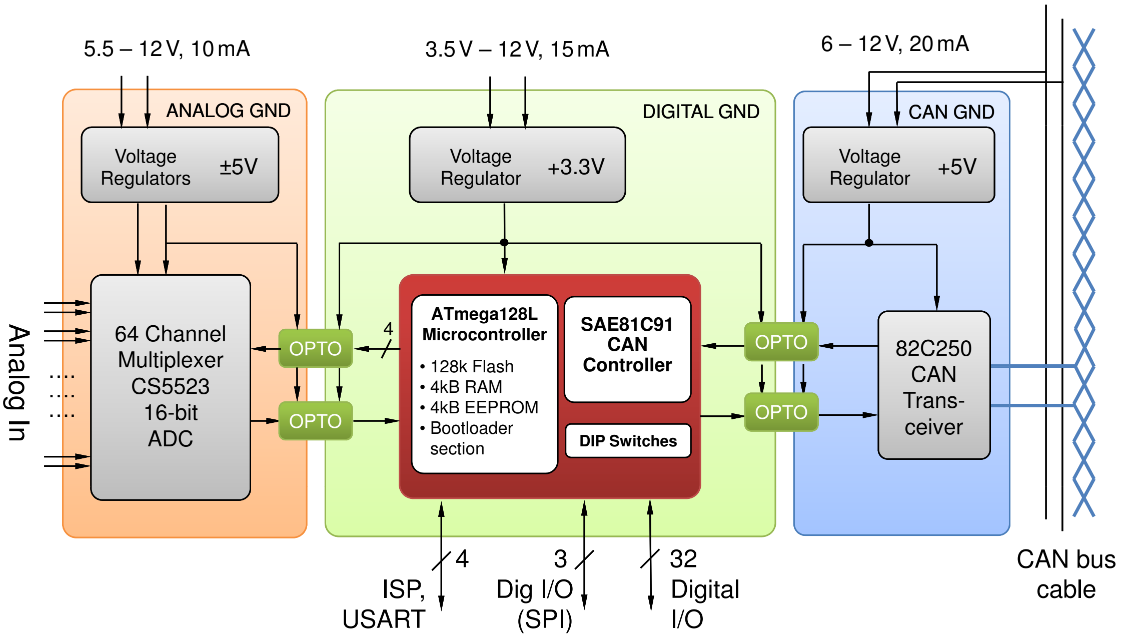


Figure 2. Block diagram of the ELMB, showing the separately powered board sections. The indicated currents are typical for no readout activity.

4.1 ELMB

In order to avoid redundant R&D and maintenance efforts for FE I/O solutions and to promote a common solution for interfacing custom designs to the DCS, the ELMB was conceived and developed since no commercial solution existed which satisfied all requirements.

The ELMB features a 8-bit microcontroller (ATmega128 from ATMEL [9]) with single clock-cycle instructions running at 4 MHz, analog and digital I/O, and a CAN bus interface. Care was taken to select components that comply with the magnetic field and radiation requirements. Further, the ELMB can be embedded within custom designs, connecting electronics and/or sensors with the higher levels of the DCS hierarchy using the CAN bus interface.

4.1.1 ELMB Hardware

In figure 2, a block diagram of the ELMB is shown. The ELMB is divided into three sections which are separately grounded and powered, and are interconnected by opto-couplers. Each section has a voltage regulator with current and thermal protection, safe-guarding against Single-Event-Latch-up. The typical current consumption of each section is shown.

The ELMB provides 64 differential analog inputs, an analog multiplexer and an Analogue to Digital Converter (ADC) with a resolution of 16 bits, as well as 32 digital I/O lines. The ADC can be configured for 6 different voltage ranges between -2 V and $+5$ V, unipolar or bipolar. In addition, a 3-wire Serial Peripheral Interface (SPI), used on-board for accessing the CAN-controller, allows to control additional components. Finally, In-System-Programming (ISP) and Universal Synchronous Asynchronous Receiver Transmitter (USART) serial interfaces are available. The analog section with ADC and multiplexer circuitry for the analog input channels may be omitted if only digital I/O is needed. The size of the ELMB board is $50 \times 67\,\mathrm{mm}^2$.

In order to provide standardized connectivity for stand-alone applications, a general purpose motherboard was developed with connectors for digital I/O, analog inputs, CAN and power [10]. It also contains sockets for signal conditioning adapters for all analog input channels. Using different adapters, several sensor types such as NTC10k, 2-wire PT1000 or 4-wire PT100 can be directly connected to the motherboard. Further, a dedicated adapter allows the use of an external Digital to Analog Converter (DAC), e.g. in order to control high voltage power supplies using analog signals.

The radiation levels within the experiment cavern outside of the inner detector were estimated in [11] to range from 0.4 Gy per year in the hadronic calorimeter and up to 900 Gy per year in the forward direction. These dose limits contain a safety factor of 5. Since the ELMB contains off-the-shelf components for which the radiation tolerance was a priori unknown, a series of extensive radiation tests [7] was performed with the ELMB using a dedicated CERN test facility [12]. The ELMB was qualified for use at locations in the experiment with radiation levels not exceeding 1 Gy per year including a safety factor of 10 compared to the tolerance determined by these tests. All ELMBs used for the DCS have been placed in locations shielded by the calorimeters such that the specified dose will not be exceeded during the foreseen lifetime of ATLAS. Special components available for space applications were several orders of magnitudes more expensive at the time of the ELMB design phase and thus not considered for use for the ELMB in order to meet the low-cost requirement.

Finally, the ELMB had to be insensitive to magnetic field strengths exceeding 1 T present within the ATLAS toroidal fields. During a dedicated test, for which an ELMB prototype was exposed to a magnetic field of up to 1.4 T, no influence of the magnetic field on operational parameters and data acquisition could be found [13].

4.1.2 ELMB software

The embedded software for the microcontroller on the ELMB is written in the C programming language.

General purpose software. The microcontroller flash memory is programmed with two applications:

- a bootloader,
- a general purpose CANopen I/O application [14].

The bootloader enables in-situ upgrades of the user application firmware remotely via the CAN bus. At power-up the bootloader takes control over the ELMB, allowing to upload new application software into the flash memory. This upgrade process via the CAN bus, which also uses the CANopen protocol, can even be performed while other nodes on the same bus continue normal operation. The bootloader cannot be erased from the memory remotely.

The general purpose CANopen application supports the analog and digital ports of the ELMB allowing its usage without need for additional firmware development. At production stage, this same application is used to test the ELMB hardware, as well as calibrating the 6 voltage ranges of the ADC. The calibration constants are stored on-board in the microcontroller EEPROM.

The general purpose application supports, by means of the on-board ADC and multiplexers, 64 analog input channels, up to 16 digital inputs and up to 16 digital outputs. In addition it provides support for an external DAC module with up to 64 analog outputs. The application conforms to the CANopen DS-401 Device Profile for I/O modules [5]. Many features of the application are configurable and can be saved in the EEPROM. Such settings are typically read or written using CANopen SDO (Service Data Object) messages.

The so-called “process data” — the analog and digital inputs and outputs — can efficiently be read out or written to using CANopen PDO (Process Data Object) messages. A PDO message is a non-confirmed CAN message with one sender and one or more receivers, containing no additional protocol overhead and up to 8 Bytes of data. The PDOs can be configured for asynchronous transmission. In this case digital inputs are transmitted on change, and analog inputs are transmitted whenever a value exceeds a predefined threshold.

Special attention has been given to prevent faulty program behavior due to Single-Event-Effect due to radiation, resulting from bit-flips in on-board memory or registers. This affects mainly RAM and device registers, and to a much lesser extent flash memory and EEPROM. These software measures taken include:

- use of the microcontroller Watchdog Timer,
- periodic refresh of microcontroller, CAN-controller and ADC register where possible (values are copied from flash or EEPROM memory),
- rarely changing configuration parameters are (re)read from EEPROM prior to each use,
- a majority voting scheme for CAN buffer management variables,
- full CAN-controller reset when no message has been received for a certain period of time,
- Cyclic Redundancy Check (CRC) on parameter blocks in EEPROM memory and on program code in flash memory,
- mask off unused bits in variables (e.g. a boolean uses one Byte but is only 1-bit significant).

A special version of the application supports a remotely configurable CAN node identifier or address by means of standard CANopen messages. This address is subsequently stored in the EEPROM, overriding the on-board switch address setting.

Finally, the application supports two additional protection mechanisms, “Node Guarding” and “Life Guarding”. Node Guarding in CANopen is a mechanism whereby a Network Management (NMT) master checks the state of nodes on the bus at regular intervals in two different ways:

- The master sends a Remote Transmission Request (RTR) for a Node Guard message to each node on the bus; a node that receives the RTR, sends the Node Guard message which contains one data byte indicating the (CANopen) state of the node, as well as a toggle bit. If a node does not reply, the master signals this to the higher-level software and/or takes an appropriate action.

- Each node on the bus sends a heartbeat message at regular intervals. The NMT master monitors the messages and signals inactive nodes which did not send a heartbeat message within a certain timeout period to the higher-level software and/or takes an appropriate action.

Life Guarding in CANopen is a mechanism whereby a node checks the communication with its host or master, by applying a timeout on messages received. If a Life Guarding timeout occurs, the node should take the appropriate action. The general purpose application resets and re-initializes the CAN-controller and tries to resume normal operation, after sending a CANopen Emergency message.

User-specific software. The source code of the general purpose CANopen application for the ELMB is available for users who want to customize this application software to fit their specific needs. Alternatively, a CANopen firmware framework for the ELMB is available for users who want to develop custom I/O and control, but want to benefit from a ready-to-use framework that handles all CAN communications transparently. The only requirement for user application software is that the interface to the outside world has to be compliant with the CANopen protocol, thus facilitating integration with the ATLAS DCS.

4.2 Standardized commercial systems

The ATLAS DCS uses as much standardized equipment as possible in order to reduce the amount of development and maintenance efforts required. Due to the complexity and specialization of the ATLAS detector, some of the FE equipment needs to be purpose built. However, associated services were standardized in many cases. This section gives details on the standardized commercial electronics that have been selected for use by the experiment.

4.2.1 VME crates

The industrial standard VME has been selected to house electronics. VME crates are placed in the underground electronics rooms, which are not effected by radiation or magnetic field. A common project was set up for the procurement of complete powered and cooled crates for the four LHC experiments.

The crates provide monitoring of the cooling fans (speed, status etc.), of temperature at various points and of general status information (for example, power on/off, error etc.), and allow for commands to be given such as switch on/off and crate reset.

4.2.2 Power supplies

The DCS makes large use of the ELMB for reading sensors which need to be powered. Due to the distribution of these modules over the full volume of the detector, the power required is distributed via the CAN bus cables. For this purpose, a Power Supply Unit (PSU) [15] was designed and produced. This CAN PSU supplies 12 V and a maximum of 3 A, which is sufficient to power up to a maximum of 63 ELMBs per bus. Up to 16 of these buses may be powered by a single CAN PSU, for which the current and voltage can be monitored for each bus independently. Additionally, with the PSU it is possible to switch the power for each bus individually.

The DCS must further monitor and control the power supplies for the detector electronics, for which many common requirements have to be met, such as setting and reading of voltages and currents, over-voltage and over-current protection for the equipment, and thermal supervision. As for the ELMBs and ELMB PSUs, OPC has been chosen to be the mandatory controls interface to the DCS (see also section 5.3).

The standard approach was to define a modular system where crates and controllers are common and different power supply modules are selected as needed. There are two types of implementation possible. One solution is to have the complete power supply situated in the underground electronics rooms, where the environment is safe. This is however not practical for low voltages with high currents because of large voltage drops in the 200 m long cables. This was avoided by separating control unit and power module, leaving the former in the counting room and placing the latter in the cavern. The cable between the two can then carry a higher voltage with less current. Following this approach, the power supply modules must be sufficiently radiation tolerant and must be able to operate in a strong magnetic field. The choice of individual power supply models varies between the different sub-detectors for which details can be found in [1] and references therein.

4.2.3 DCS Control Station PC

The hardware platform for the BE system is an industrial, rack-mounted PC. The applications in the LCS require good I/O capability, whereas the applications in the GCS are mainly processing-intensive. As a consequence, two different PC models have been chosen as the standard. A common requirement for both is stability of operation and robustness against hardware failures. Therefore, both models feature redundant, hot-swappable power supplies and disk shadowing. The system configuration of the LCS machines includes two Intel Xeon processors (3 GHz), 2 disks (250 GB each), 2 GB RAM, and an Intelligent Platform Management Interface (IPMI) for remote control. They have 3 PCI slots available, e.g. to house CAN interface boards. The GCS machines feature two Intel Xeon quad-core processors (2.33 GHz), 2 disks (500 GB each), 8 GB RAM, and also an IPMI interface. For the SCS the model is selected, which fits best the needs.

As operating system either Windows or Linux is used, the former mainly for the LCS because of the OPC protocol, the latter for the operator interfaces and servers in the GCS layer. The operating system is downloaded from central servers via the network.

5. Software components

5.1 PVSS

The commercial SCADA package PVSS is the main framework for the ATLAS BE applications. As the ATLAS DCS will read out and process data from more than 200.000 sensors, it was necessary to find a solution which provides sufficient scalability with respect to distributed computing and different controls application tasks. Four main concepts of PVSS make it suitable for a large scale control system implementation such as the ATLAS DCS:

- A control station (PC) runs a so-called “Project” which contains a number of processes, called “Managers”. Different types of Managers may be used depending upon the type of application the Project is being used for, therefore avoiding unnecessary overhead.

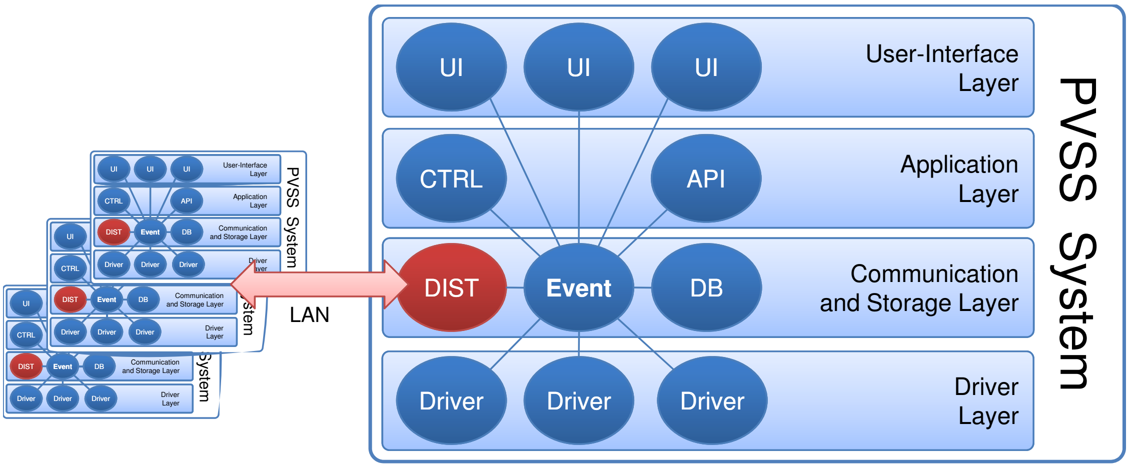


Figure 3. PVSS Manager structure showing the respective functional layers. Several Projects can be connected via LAN to form a Distributed System.

- Each PVSS Project uses a central database for all current data values, stored in objects called "Datapoints". All Managers have full database access for which PVSS provides transparent synchronization. Data processing is performed in an event-based approach using multi-threaded callback routines upon value changes.

- Different Projects can be connected via LAN to form a "Distributed System" allowing to remotely access the databases and events of all connected Projects. This provides scalability up to the full size of the ATLAS DCS with in excess of 150 different control stations.

- A generic Application Programming Interface (API) allows to extend the functionality of control applications using additional software components.

Figure 3 shows the different process layers of a PVSS Project. In the following, the PVSS functionality is briefly described with respect to these layers and the relevance for the DCS is discussed.

In order to interface with the FE hardware, PVSS offers a number of Managers — "Drivers" — that allow for industry standard protocols to be used. In particular, Drivers for data read-out via Modbus, for communication with Programmable Logic Controllers (PLC), and OPC are available.

Within the communication and storage layer, data is processed by a central "Event Manager" which broadcasts value changes to other Managers which may also be located within remote Projects connected via the "Distribution Manager". The data can be made persistent using an "Archive Manager" that stores data into a relational database and allows for the information to be read back into PVSS for the purpose of diagnostics, e.g. using trend plots as described in section 5.8. More importantly, the use of a relational database permits data access from other applications completely independent of PVSS such as detector calibration and physics data analysis programs (see also section 5.7).

Control procedures in the application layer are programmed in a C-like scripting language called "Control". With this interpreted language, PVSS provides full access to the Project database, dynamic type conversion, and the possibility to develop and change the application at run-time while at the same time protecting the low level data acquisition and processing. Further, Control scripts can run as stand-alone processes for which PVSS ensures continuous running without the necessity of human interaction. For more application-specific requirements beyond the intrinsic Control functionality, PVSS allows for the development of custom Managers and shared libraries through a C++ API.

The User Interface (UI) layer consists of dedicated Managers which interpret user-built panels accessing the Datapoints via Control scripts and display control system conditions and processes to an operator. Any UI can be protected by an access control mechanism restricting the interaction with all other Managers according to predefined privileges.

PVSS is available for the Windows and Linux operating systems and provides a process monitoring mechanism which ensures the continuous availability of all Managers for every Project running on the respective control station.

5.2 JCOP Framework

The JCOP Framework (FW) [2] consists of a set of guidelines, components and tools designed to facilitate the implementation of homogeneous controls applications for the LHC experiments using PVSS. The FW guidelines define a naming convention for PVSS Datapoints, functions and files, and also cover the standard look and feel of the graphical interfaces of the control system. The FW components ease the integration of standard hardware devices such as the ELMB or commercial power supplies and provide all required functionality to monitor and control the detector electronics racks and cooling systems. The FW also provides a component to retrieve information from external systems such as the ATLAS magnets, the LHC accelerator and the CERN technical infrastructure. The list of FW components comprises:

- a Finite State Machine (FSM) toolkit that allows to model the hierarchical organization of the control system and to sequence the operation of the detector according to a well-defined set of states and possible transitions,

- a configuration database component, which may be used for homogeneous storing and access of configuration parameters in a relational database external to PVSS,

- a generic interface between the FSM and the Configuration Database designed to automate the handling of the configuration data required at run-time,

- tools to configure and display trends of PVSS data,

- a multi-platform middleware for inter-process communication called Distributed Information Management (DIM) [16], together with a simplified version specialized for reliable data transfer — the Data Interchange Protocol (DIP),

- generic libraries for development of PVSS applications that access an external relational database,

- a System Configuration Database which holds the overall layout of the control system and the relations between its constituents, e.g. arrangements of computers, PVSS Projects, and software revisions,
- a user interface to the PVSS alert system tailored to the needs of the LHC experiments,
- an access control component providing a detailed authorization scheme assigning sets of operator privileges to operator roles with respect to detector subsystems,
- a Farm Monitoring and Control component for the computers used in the control system,
- a System Overview Tool, which monitors the integrity of the different Managers for the complete Distributed System of PVSS stations,
- a centralized software management tool, which allows the remote administration of FW components in all DCS computers from a central console.

The modular approach followed by the FW promotes standardization within the control applications and allows for the reutilization of code, significantly reducing development and maintenance efforts.

5.3 CANopen OPC server

A CANopen OPC server has been developed in order to control data sources connected to CAN bus and to acquire their data. This data is then made available to the PVSS system using the interface OPC which is a standard providing a mechanism for communicating with numerous data sources. It is based on the COM/DCOM [17] technology of Microsoft. The specification describes the OPC COM objects and their interfaces, implemented by an OPC server. The server allows for concurrent communication of different applications with the same hardware, supports all interfaces and requirements of the OPC Data Access Specification, and can work both with local and remote clients.

The OPC server transfers data to the devices using the high-level protocol CANopen. This protocol defines how to read and write data using asynchronous and synchronous modes, as well as supervising the CAN bus itself and any connected CAN bus devices.

A CAN bus consists of several fieldbus nodes containing a microcontroller with firmware. The nodes send and receive messages of up to 8 Bytes in length and a header (integer number) called the “Communication Object Identifier” (COB_ID). In the case of the ELMB, data from several sensors use a single COB_ID which allows the number of devices in the system to be reduced. The CANopen OPC server supports this multiplexing of messages and can translate the values into separate objects used by PVSS.

The CANopen OPC Server consists of two components, with the first implementing all OPC interfaces to which any application such as the PVSS OPC Driver may be connected. The second part represents a hardware-dependent implementation of the interaction mechanism between the CAN bus driver and CANopen devices. It was developed as a COM component, which allows the porting to new hardware interfaces with minimum effort. There are implementations of this component for KVASER [18] and National Instruments [19] CAN cards.

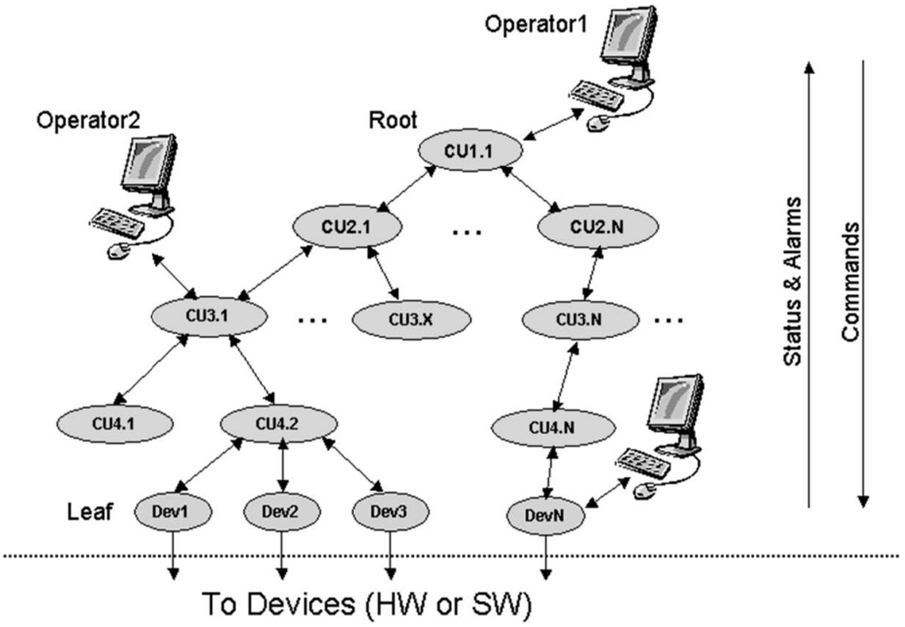


Figure 4. FSM toolkit concept (taken from [20]). Device-oriented objects (Dev) can be grouped into domains represented by Control Units (CU) arranged in several layers. State changes are propagated upwards, and commands downwards in the hierarchy. Operator interfaces can obtain control over any object within the individual layers.

The CANopen OPC server imports all information needed to define its address space (i.e. the named data available to the OPC client) from a configuration file. The configuration file defines the names of all data sources which the OPC client can use to read from and write data to.

5.4 The Finite State Machine toolkit

The representation of all detector control elements by a Finite State Machine (FSM) is a key element in the homogeneous control of the heterogeneous collection of detector FE systems. The JCOP FSM [20, 21] provides a generic, platform-independent, and object-oriented implementation of a state machine toolkit for a highly distributed environment, interfaced to a PVSS control application.

5.4.1 State Machine logic, structuring, and object orientation

The FSM toolkit allows for the design and implementation of dedicated state machines of the detector FE systems and of software procedures. The collection of these device-oriented state machines can be structured by layers of higher-level state machines. These derive their own state from the levels below via programmable logic, allowing the complete detector control system to be supervised by a single state machine unit (see figure 4).

Following an object-oriented approach, the programmable behavior of the FSM units are defined in FSM object types. The device-oriented types implement their state and state transition definitions within the associated PVSS application with access to the full PVSS functionality. The

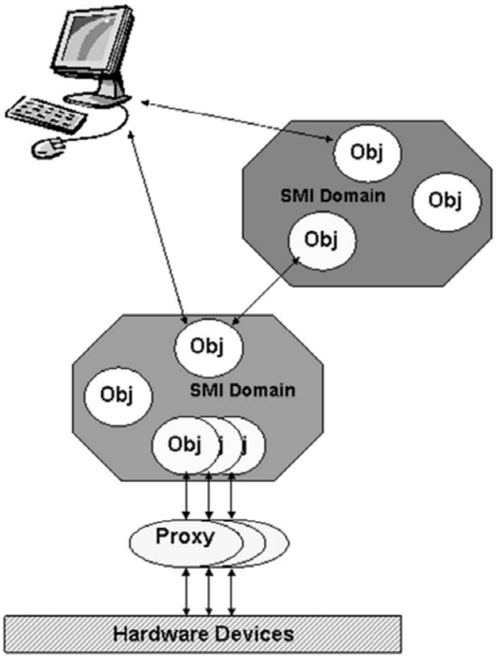


Figure 5. SMI inter-process communication (taken from [20]).

logical object types are programmed using the rule-based State Manager Language (SML) [21] which allows the use of type-oriented state and command logic in interaction with other FSM units. The actual FSM hierarchy is formed by object instantiation and the definition of object relationship.

5.4.2 Inter-Process Communication and Partitioning

The FSM objects are grouped into State Manager Interface (SMI) domain processes. The individual processes communicate via the DIM protocol and thus can be distributed within a LAN. The device-oriented objects are interfaced via a proxy process represented by the PVSS application (see figure 5).

In order to be able to operate different detector parts independently, individual SMI domains can be separated from the control hierarchy. Further, the partitioning capabilities of the FSM toolkit allow operating parts of the hierarchy in distinct modes. Device-oriented FSM objects can be detached from the tree (“Disabled”) such that they do not propagate their state nor receive commands. FSM domain objects called “Control Unit” (CU) can be put into the following modes:

- Included – the state and command propagation is enabled,
- Excluded – the state and command propagation is disabled,
- Ignored – the state propagation is disabled but commands are accepted,
- Manual – the state propagation is functional but commands are ignored.

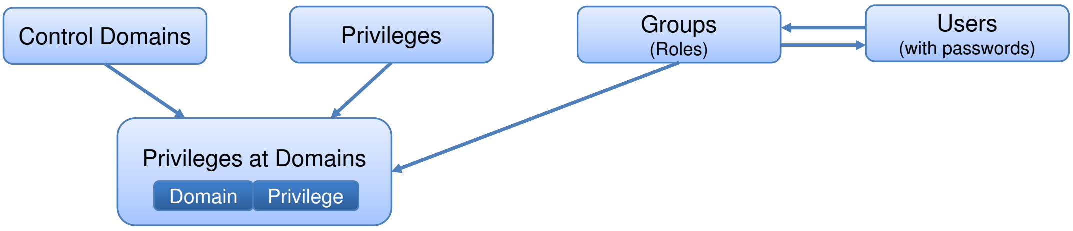


Figure 6. JCOP FW Access Control data model for user authorization.

5.4.3 Object persistency and user interfaces

The attributes of any instantiated FSM object inside a SMI domain are made persistent within Datapoints of the associated PVSS project. This allows for archiving of the FSM states and transitions and provides a framework for monitoring and control of the FSM functionality using PVSS user interfaces. In particular, the FSM toolkit contains operator interface software elements providing control over FSM object attributes, transitions, and ownership. An operator can obtain the ownership over a CU and its associated objects via the user interface in two different modes: The “Exclusive Mode” for which only the user interface of the owner can alter object attributes and send commands, and the “Shared Mode” allowing access via any interfaces, however without permitting to change the object ownership.

5.5 Access control

Access Control (AC) for different control aspects within ATLAS DCS is provided by a set of tools restricting access to the control interfaces to authorized users. In addition to the protection of the FSM operator interface, AC may be applied to the sub-detector’s expert panels. The DCS AC mechanism is based on the JCOP FW Access Control component while the administration of the access privileges is performed using a Lightweight Directory Access Protocol (LDAP) [22] repository.

The data model of the access control defined by the JCOP FW AC package is shown in figure 6.

A control domain is a part of the control system, which has certain sets of control privileges defined, e.g. the possibility to access or modify data elements of a particular PVSS Project. Any DCS subsystem, such as an individual sub-detector, can be declared as a control domain. A set of pairs <domain, privilege> called “Role” corresponds to a group of users with certain rights.

The component comprises a graphical user interface for the AC administration and an API for applying access control in the project panels. A common access control policy for a whole distributed system of PVSS stations is provided by means of an AC server implementation which ensures the synchronization of authorization data. The user administration and the authentication mechanism are outsourced to a LDAP repository.

As an extension of the functionality of the FW AC component, the authorization data consisting of the list of users and associated Roles are downloaded to the DCS AC server periodically or on request, while the user authentication is performed at run-time using standardized LDAP routines. Availability of the access control mechanism in case of network problems is provided by caching the user authentication and authorization data in both the AC server and the local DCS workstations.

5.6 Configuration database

The ATLAS DCS configuration database (ConfDB) is designed to manage the setting of the detector and DCS parameters (such as calibration constants, voltage settings, and alarm thresholds) depending upon the operation mode of the experiment. Furthermore, it stores the configuration of DCS equipment. The ConfDB is arranged as a set of Oracle databases (one per sub-detector) available for all DCS applications.

ConfDB access is based on the Configuration DB FW component which defines the data model comprising two main entities. A “Configuration” contains sets of devices with their static properties, for example HW addresses, archiving settings, etc. The “Recipe” is a set of values that are run-time specific, such as set points for output channels and alert thresholds, which may change depending on the detector operation mode.

The Configuration DB component includes a graphical user interface, which allows an expert to manage the database connection, and to store/retrieve Configurations and Recipes for selected sets of devices in an interactive mode. Advanced facilities include editing recipes, viewing the database tables, and executing queries using the Structured Query Language (SQL).

The generic API provides database access and Configuration/Recipe handling routines for PVSS applications which allow for event driven detector configuration depending on specific run conditions. The typical use case for Recipes at run-time is a Control script which continuously analyzes the current detector subsystem condition, selects and retrieves the proper Recipe on the base of the Recipe name and/or version, and applies them to the PVSS application. The main use cases for Configurations is restoring a set of devices interactively during a possible system recovery. The configuration data can be used within offline reconstruction and analysis routines independent from PVSS.

Within the ATLAS DCS, the ConfDB FW component has been extended providing the subdetector control system developers with:

- a user interface template comprising, in particular, two levels of access rights (operator and expert) for storing and applying Configurations and Recipes,
- a set of generic functions, allowing the storage and retrieval of the configuration DB entities, taking into account the ATLAS DCS naming convention.

Altogether, the ConfDB software components guarantee that the configuration data is stored homogeneously at a single centrally supported location, enabling efficient run-time access and providing reliability and maintainability.

5.7 Conditions database

The ATLAS detector conditions data are stored in a relational database (CondDB) for efficient access by any offline calibration, reconstruction, and analysis applications. The database is accessed by a dedicated API, called COOL [23], and was highly optimized for the offline applications only instead of being used for diagnosing detector problems. The amount of data stored in the CondDB must be kept to a minimum because of the two following main reasons:

- the conditions data must be replicated to external sites for analysis and reconstruction all over the world

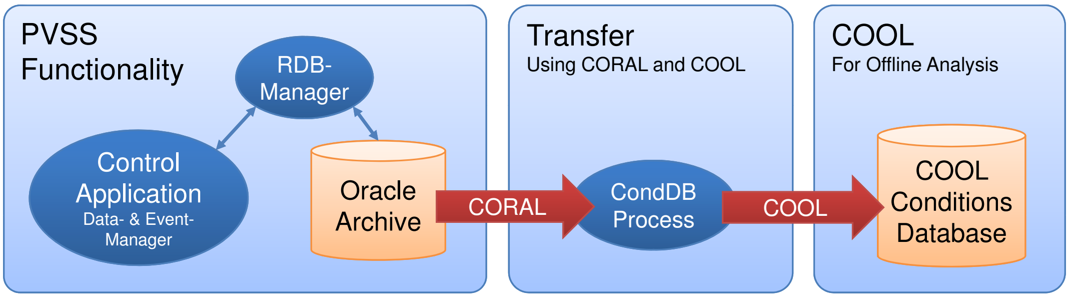


Figure 7. Flow of data through the archiving chain from the PVSS application to the COOL database.

- extensive data processing is required for analysis, therefore searching large amounts of conditions data for the required information is an unnecessary overhead.

As a consequence, only selected conditions data acquired by DCS using the PVSS archiving facility, which was optimized for writing data rather than reading, must be transferred into the CondDB using COOL.

COOL stores logically grouped data into "folders". Similarly, PVSS has the concept of Datapoints that hold real-time data from FE equipment, grouped in a structure for a given device. Therefore, a dedicated process — PVSS2COOL — reads selected PVSS data from the PVSS archive database using a generic access API (CORAL) [24], maps the data from PVSS Datapoints into COOL folders of a corresponding structure, and writes it into the CondDB (see figure 7). Datapoint types, which represent devices, are associated with corresponding folders, which will then allow data from one or more of each type of device to be stored in the folder.

The configuration for the COOL folders and the data they should contain is defined within PVSS, allowing the sub-detector DCS experts to structure the data for offline analysis. Once the configuration is completed, PVSS2COOL is used to create the folders in the COOL database. After the folders have been created, the transfer process runs at predefined time intervals, replicating and transposing the data into the format required for COOL. For performance reasons, the time intervals for bulk data transport must be optimized. Short time intervals result in performance losses due to increased transaction overhead. Large time intervals will not only delay the availability of data, but may cause the amount of data to be transferred exceeding a tolerable limit.

5.8 Data visualization

All values received from PVSS may be stored in a database in order to allow for diagnostics of problems and to check the evolution of values over time. In order to be able to study the historical data in an efficient and convenient way, e.g. by comparing values of different data sources against each other, dedicated data visualization tools are required.

The main use cases are:

- the display of trends for individual DCS parameters by the operator in the ATLAS Control room with the option to show data for a selected time interval in the past,

- more sophisticated visualization plots such as histograms, profiles, and scatter plots to allow for more detailed analysis,

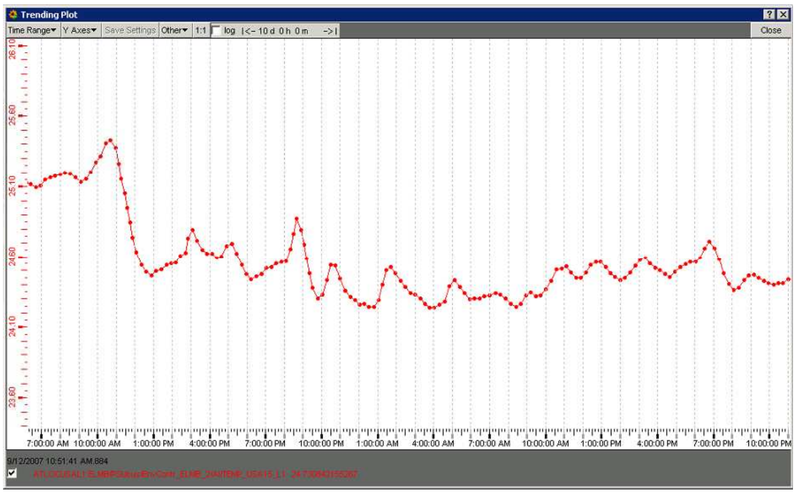


Figure 8. PVSS trend showing a temperature value over time.

- a web-based display to view data and to diagnose incidents.

Common requirements for the different display methods are such that the selection of the required data must be intuitive to use, and that the load on the server supplying the data must be minimal and must not influence the stability and availability of the DCS BE.

The two examples of visualization tools described below are available and more advanced tools are being developed.

5.8.1 PVSS trending

The PVSS trending allows the viewing of historical data from within PVSS applications. Tools are available for the selection of what data should be shown and a trend plot may then be displayed, as shown in figure 8.

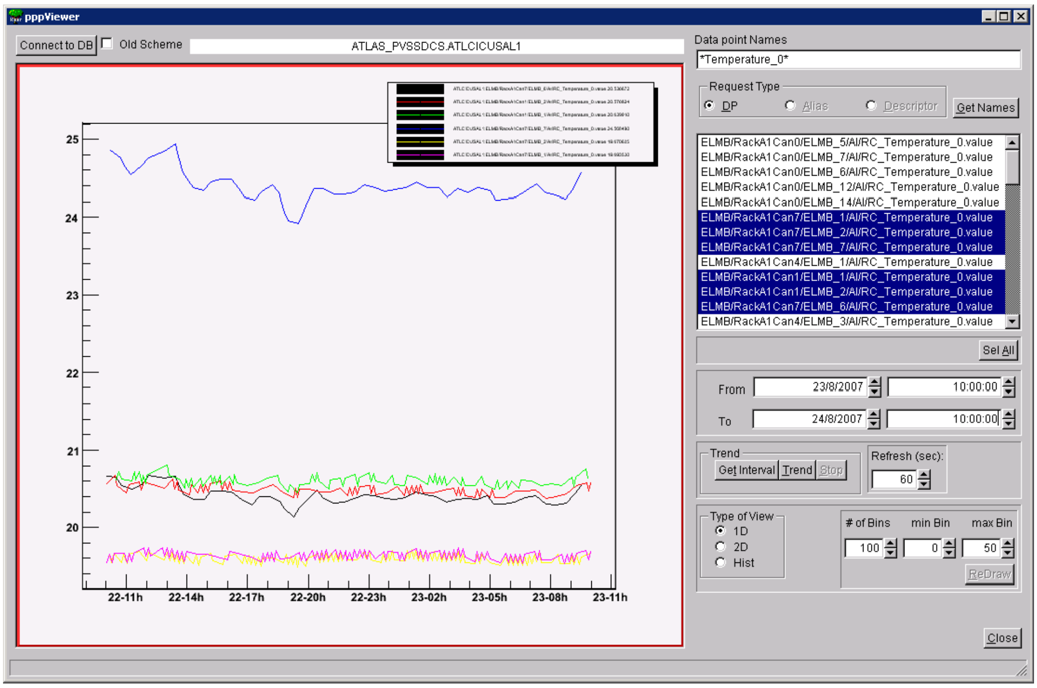
The tool features standard graphics capabilities including changing scales and multiple data display formats. In addition, data which are marked as invalid by the hardware, are displayed by PVSS in a different style.

5.8.2 ROOT viewer

Another data viewer is based on the graphics and analysis package ROOT [25] and has full access to all data within the PVSS Oracle archive. The implementation is completely independent from PVSS.

Data can be selected by data source identifiers allowing the usage of pattern matches using wildcards, and be displayed for a chosen time interval with the option of real-time refreshing of the current values. The data may be represented as a trend (as shown in figure 9), as a 2D surface plot or as a histogram.

Any selected data can be stored using the native ROOT format thus allowing more complex analysis exploiting the advanced functionality of ROOT.



ATLAS_PVSSDCS.ATLCICUSAL1

ATLCICUSAL1:ELMB/RackA1Can7/ELMB_6/AI/RC_Temperature_0.value 20.530672
ATLCICUSAL1:ELMB/RackA1Can1/ELMB_2/AI/RC_Temperature_0.value 20.576624
ATLCICUSAL1:ELMB/RackA1Can1/ELMB_1/AI/RC_Temperature_0.value 20.639910
ATLCICUSAL1:ELMB/RackA1Can7/ELMB_7/AI/RC_Temperature_0.value 24.500490
ATLCICUSAL1:ELMB/RackA1Can7/ELMB_2/AI/RC_Temperature_0.value 19.670625
ATLCICUSAL1:ELMB/RackA1Can7/ELMB_1/AI/RC_Temperature_0.value 19.692530

Data point Names: *Temperature_0*

ELMB/RackA1Can0/ELMB_5/AI/RC_Temperature_0.value
ELMB/RackA1Can0/ELMB_7/AI/RC_Temperature_0.value
ELMB/RackA1Can0/ELMB_6/AI/RC_Temperature_0.value
ELMB/RackA1Can0/ELMB_12/AI/RC_Temperature_0.value
ELMB/RackA1Can0/ELMB_14/AI/RC_Temperature_0.value
ELMB/RackA1Can7/ELMB_1/AI/RC_Temperature_0.value
ELMB/RackA1Can7/ELMB_2/AI/RC_Temperature_0.value
ELMB/RackA1Can7/ELMB_7/AI/RC_Temperature_0.value
ELMB/RackA1Can4/ELMB_1/AI/RC_Temperature_0.value
ELMB/RackA1Can1/ELMB_1/AI/RC_Temperature_0.value
ELMB/RackA1Can1/ELMB_2/AI/RC_Temperature_0.value
ELMB/RackA1Can7/ELMB_6/AI/RC_Temperature_0.value
ELMB/RackA1Can4/ELMB_3/AI/RC_Temperature_0.value

From 23/8/2007 10:00:00

To 24/8/2007 10:00:00

Refresh (sec): 60

# of Bins 100, min Bin 0, max Bin 50

Figure 9. Root viewer showing multiple parameter values over time.

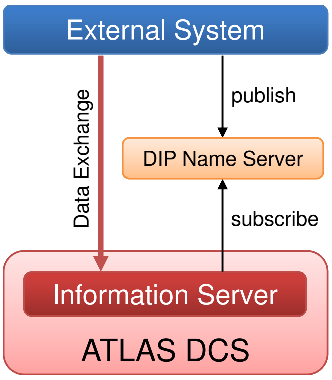


Figure 10. Data exchange between ATLAS DCS and external control systems via DIP.

5.9 Connection to external control systems

The data exchange between the ATLAS DCS and external control systems is handled via DIP. This protocol is a thin layer on top of the DIM process communication interface designed for highly reliable event-based data transfer. As illustrated in figure 10, a DIP server publishes data items to a dedicated name server. A client process fetches the server publication information from the name server and subscribes at the DIP server to selected data items, resulting in an event-triggered, pushed data transfer from the server to the client. A JCOP FW component provides a DIP server and client implementation for PVSS — a DIP Manager — including user interfaces for user-friendly publication and subscription handling.

All external control systems are interfaced to the ATLAS DCS using a dedicated DCS Information Server (DCS IS) in the GCS layer (see figure 1). Information from the external systems is transferred via DIP into the IS PVSS Project, thus made available to all DCS stations within the Distributed System, and stored in the PVSS Oracle Archive. A generic error handling mechanism using the DIP quality monitoring facilities has been implemented for all subscriptions on the DCS IS signaling any error condition related to the DIP communication via the PVSS alarm system.

5.10 DCS-TDAQ communication

The DCS-TDAQ Communication (DDC) package has been developed for the information exchange between the DCS and the TDAQ system and for coherent control of the experiment during physics data taking periods. DDC provides the following functionality:

- the execution of DCS control commands triggered by the TDAQ system, which may or may not be synchronized to FSM transitions, reporting success or failure of the command back to TDAQ,
- the bi-directional data exchange between TDAQ and DCS,
- the transfer of DCS alarms and messages to TDAQ.

The DDC package is integrated into the ATLAS TDAQ software [26] and uses several of its facilities. The command transfer facility is implemented as a custom application (DDC Controller) on the base of the TDAQ Run Control (RC) package. For the execution of "Non-transition Commands" (see below) the TDAQ Information Service (TDAQ IS) is used. The data transfer is performed via the TDAQ IS, while the DCS messages are passed via the TDAQ Message Reporting System. All DDC facilities use the DIM package for inter-process communication.

The TDAQ RC and the DCS implement different state models requiring a state synchronization logic which defines a correspondence between both. The synchronization is provided by the DDC controller that triggers the necessary DCS action as a result of a TDAQ RC FSM transition according to its configuration. Alternatively, the Non-transition Command facility allows DCS actions to be triggered asynchronously with TDAQ RC transitions. These commands could be the reset of an individual device or the execution of a calibration sequence triggered by an operator, or an automatic action initiated by a TDAQ application.

5.11 Software management

Due to the size of the control system, the large diversity of software elements and the long lifetime of the experiments, a centralized software management strategy was adopted for the ATLAS DCS. This approach addresses the following points:

- multi-user/multi-location development of the controls applications,
- integration of the different software elements of the control system,
- visualization of the current organization of the system,
- remote upgrading of the software configuration of PVSS Systems,

- version handling of the software components,
- minimization of the downtime by permitting to restore a system to the last known configuration in the event of system failures.

The software management approach distinguishes two main aspects:

- the set-up of the basic infrastructure of the computers,
- the installation and configuration of the PVSS Projects.

The software infrastructure management, such as the installation of the operating system and security patches, and the deployment of a set of common applications, such as PVSS, OPC servers, etc., is handled with the tools recommended by the CERN Computer Network Infrastructure and Controls [27] working group. In particular, the Computer Management Framework (CMF) is used to handle computers operated by Microsoft Windows, and Linux for Controls (L4C) is used for machines running the Linux operating system. Although CMF and L4C share the same design principles, they have been implemented as two separate schemes: while in CMF the desired software configuration of the computers is defined using a graphical interface and is stored in a central database, the L4C scheme uses a template-based approach using a tree structure of editable text files. In both cases the configuration information is accessed by software daemons running on the nodes in order to carry out the installation or un-installation of applications according to the contents of the database or templates.

The PVSS Project management is covered by the usage of the FW Component Installation Tool and a central software repository containing the application components specific to the individual PVSS Project. The functionality of the FW Component Installation Tool is two-fold. On one hand, the tool allows developers to package the PVSS-based applications as individual components that can be installed in different Projects. On the other hand, the tool permits to manage the list of PVSS-based components installed in a set of computers. This is achieved by defining the desired configuration of the nodes in a central configuration database. The database provides versioning of different software configurations, enabling a DCS station to be restored to a previous configuration in the event of failure. The FW Component Installation Tool provides graphical interfaces to manage the contents of the database and to give an overview of the configuration in the database. The database is accessed by the software agents of the FW Component Installation Tool running in the remote nodes that perform the installation or un-installation of the PVSS components according to the contents of the database.

PVSS application elements such as user interfaces and Control scripts and libraries, as well as FW components are stored in a central repository located on a shared network drive. All these components can thus be consistently accessed from LCS, SCS or GCS Systems as well as from outside of the DCS BE, avoiding additional management efforts to keep track of differences in the local software installations. In order to keep track of development, the contents of the central repository is tagged into the Concurrent Versions System (CVS) [28]. The repository is cached transparently on each DCS station to ensure its availability in case of network disconnections.

6. Read-Out chain

Figure 11 shows a typical example of the readout chain based on the ELMB. The ELMB are placed in the ATLAS cavern and gather data from sensors that are distributed over the whole detector volume. Any ELMB node can be configured to transmit the calibrated sensor data either at regular time intervals or on-change. In the latter mode, which compares the readings against pre-defined thresholds, a first level of data reduction is achieved. All ELMB are interfaced to an associated LCS in the underground counting rooms via CAN. These CAN buses are operated at 125 kbaud allowing a maximum bus length of about 500 m. In total, 63 ELMB can be daisy chained on a single CAN bus and are powered over the bus cable by dedicated power supplies. These power supplies are located in the counting room thus enabling to power cycle all nodes in a bus in case of errors. They also monitor the current consumption of the buses in order to detect aging effects of the ELMB due to radiation.

Several buses can be connected to a single computer running an OPC server which decodes the received CAN frames, may also perform conversion from raw to physical values, and sets the relevant items in its address-space. This triggers the communication with PVSS resulting in an update of the respective Datapoints. PVSS permits further calibration procedures to be applied to the values, and to display and plot the results. Alarms can be generated by comparing the updated values against pre-defined thresholds. These sensor values may also be used to trigger state changes of the device-oriented FSM objects. In some cases, automatic actions may be initiated by means of a Control script within the local PVSS system, or by any associated FSM object within the FSM hierarchy. The values read are stored in the PVSS Oracle Archive. Finally, this database is accessed asynchronously by the PVSS2COOL process which replicates the subset of the data needed for offline analysis into the ATLAS Conditions Database.

7. Operations layer

The DCS operations layer provides the top level interface between the DCS BE systems and both, human operators and the TDAQ run control system which allows automated DCS operation during data acquisition runs. Further, the operations layer provides tools for efficient problem diagnostics and handling by detector experts. The two main components of the operations layer are

- a Finite State Machine hierarchy representing the operational condition of each part of the detector by a set of well defined states and distinct transitions between these states
- an Alarm System indicating problems on the level of individual detector devices.

Both components are represented by dedicated user interfaces, accessible at the DCS operator station in the ATLAS control room and remotely through local and wide area networks.

7.1 Operations model

7.1.1 The detector as Finite State Machine hierarchy

The complete DCS BE is mapped onto a hierarchy of Finite State Machine (FSM) elements implemented using the FSM toolkit. State changes are propagated upwards and commands downwards

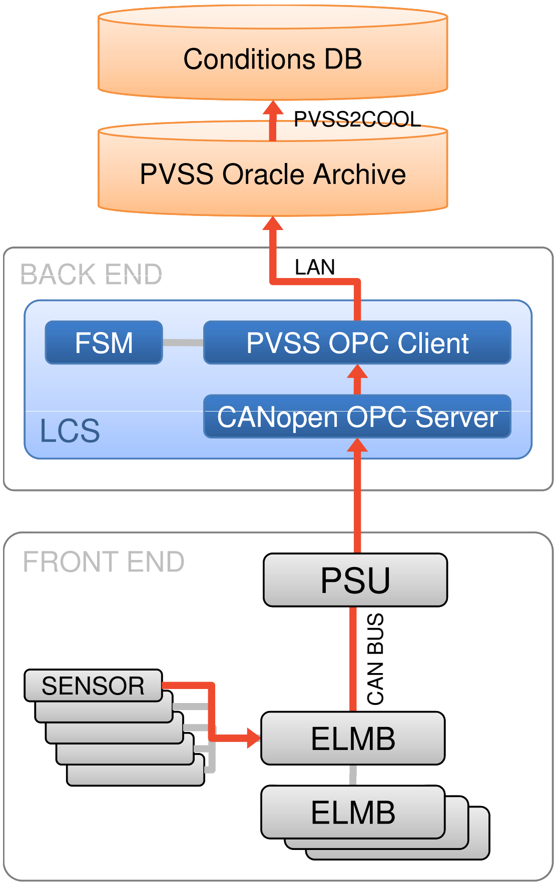


Figure 11. The ELMB readout chain.

in the hierarchy allowing for the operation of the complete detector by means of a single FSM object at the top of the hierarchy.

Hierarchy structure. Figure 12 illustrates the ATLAS FSM structure on the basis of example parts of the tree. The ATLAS sub-detectors are divided into partitions corresponding to the so-called Trigger, Timing and Control (TTC) partitions [3] of the data acquisition system which can be read out independently. The partitions themselves are further subdivided into subsystems and infrastructure services, which in turn may be structured geographically. The lowest level elements are formed by individual or groups of devices.

The chosen hierarchy granularity is a trade-off between the desired level of detail, interdependence of FSM elements with other branches of the hierarchy, and FSM software performance constraints. The distribution of FSM objects within the network of control stations is done accordingly: a GCS contains the top node, sub-detectors and their partitions are located on the respective SCS, and all lower level objects are situated on their dedicated LCS connected to the FE devices.

State model. The individual detector FE devices and logical elements such as device groups and geographical detector segments are represented by device-oriented FSM objects. The present condition of each object is reflected by a limited set of states. A state transition of such a

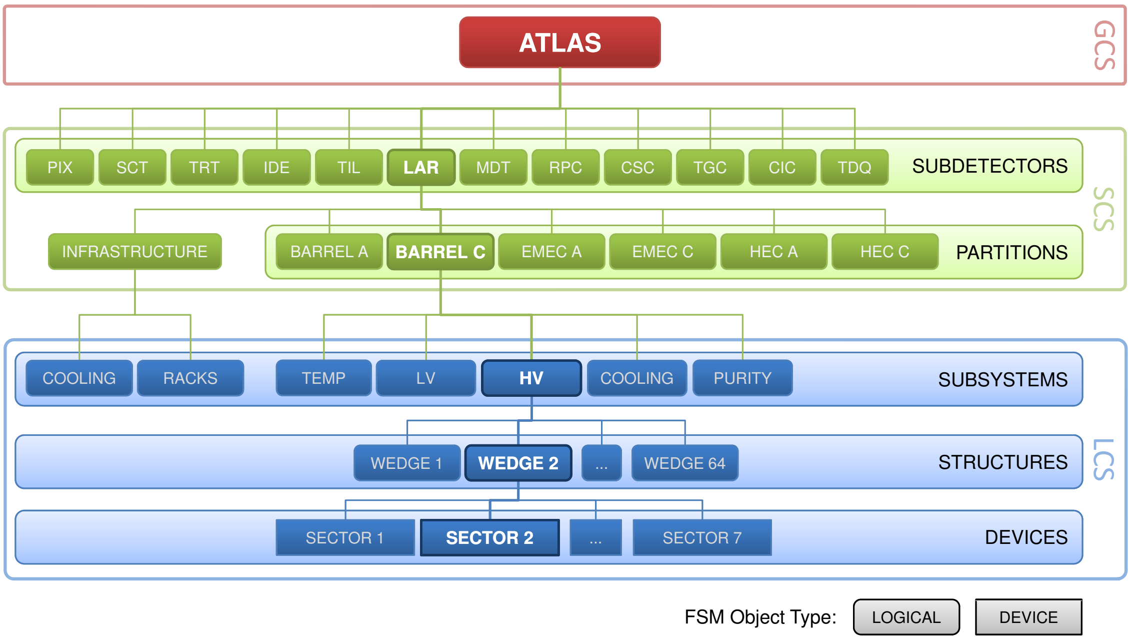


Figure 12. ATLAS DCS Finite State Machine hierarchy showing selected structure elements.

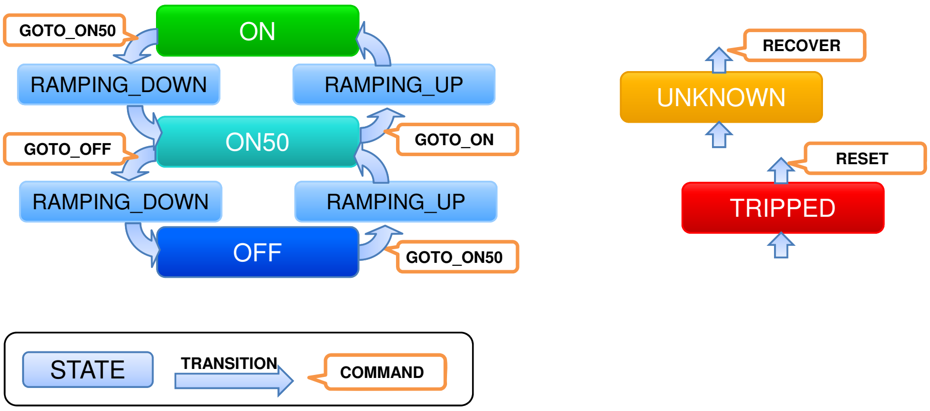


Figure 13. Device object state model.

device-oriented object is triggered either by a spontaneous condition change or by means of a dedicated command. Figure 13 shows the ATLAS device state model, allowing a device to change from its ground state OFF to its operational state ON via optional stages. In case the current condition is undefined, e.g. due to a lack of communication with a sensor, the state is set to UNKNOWN. Other possible states include device or procedure specific conditions such as ongoing calibration or power supply trips.

For logical objects, the mandatory states are READY or NOT_READY, reflecting conditions for which data taking is possible or impossible (see figure 14). The state UNKNOWN is used in case the condition cannot be verified. The actual state of these logical objects is determined by the

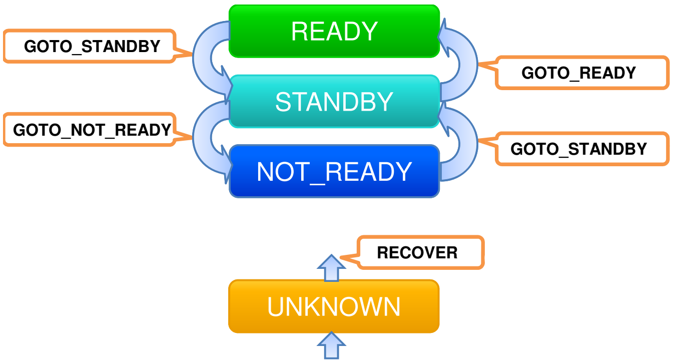


Figure 14. Logical object state model.

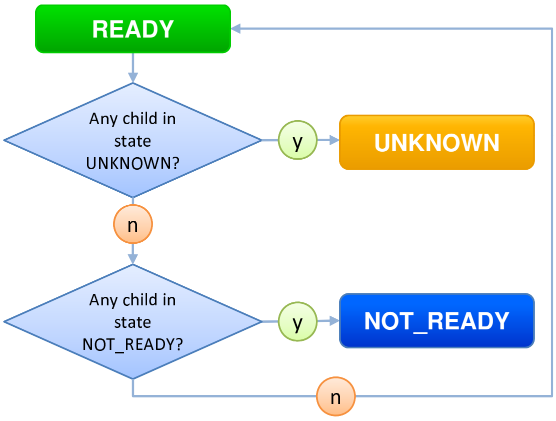


Figure 15. State rule example for a logical object.

states of the associated lower level objects (children) via state rules implemented using SML (see figure 15).

State transitions. State transitions can be initiated for every FSM object in the hierarchy by commands with the optional use of command parameters. Logical objects propagate the commands to their children according to SML rules again using optional parameters such as a run type which determines the set of configuration data to be used.

7.1.2 Event and error handling

During detector operation, any problem occurring must be detected, signaled and possibly automatically recovered from. On the individual device or channel level, the PVSS alarm mechanism is used to report any abnormal value of a single monitored parameter. Due to the vast number of these parameters, the individual alarms are coupled to an alarm signalization mechanism within the FSM on the lowest hierarchy level and propagated upwards in the hierarchy. This represents an effective alarm reduction mechanism and allows for automatic error recovery at each hierarchy level. Finally, all control application specific events, FSM state changes, command response, and automatic actions are processed centrally for immediate operator feedback and are written to a database for offline problem diagnostics.

Device alarms. For each critical parameter on a LCS, an alarm is raised if the parameter value is out of the range specified for normal operation. The alarm thresholds are attributed to one of the following severities:

- Warning: normal operation possible, but problem should be investigated in order to avoid forthcoming malfunctions,
- Error: normal operation not possible or will become impossible very soon,
- Fatal: normal operation not possible and eventual implications for other systems, immediate reaction required.

The alarms can be configured to require an acknowledgment by an operator in order to be removed from the alarm system once the parameter value has left the problematic range. To avoid the accumulation of a large number of alarms on the user interface, the individual device alarms are grouped into summary alarms which are organized hierarchically in correspondence with the FSM tree structure.

Higher level error signalization. Error signalization within the FSM hierarchy is performed using a parallel tree of "Status" objects. For each FSM object in the hierarchy, a dedicated Status object instance is assigned with the states OK, WARNING, ERROR, and FATAL, signaling a problem in the corresponding part of the detector with the same severity definitions as specified above for the device alarms. Any Status change is propagated upwards in the Status tree and thus allows for error detection within the upper layers of the detector tree and permits to identify problematic devices by following the propagation path downwards.

Automatic actions and error recovery. In addition to the fast detection of problems, the Status allows the execution of commands in reaction to the problem or automatic error recovery at any level in the hierarchy. For example, in case the ramp up (NOT_READY → READY) of a calorimeter segment low voltage failed due to power supply trips, the resulting Status change to ERROR can trigger a repetition of the state transition command (GOTO_READY). In case the trip was accompanied by a FATAL Status, e.g. due to overheating of equipment, the corresponding action in response to FATAL would be to ramp down all devices within the segment (GOTO_NOT_READY).

Error masking. In the cases where problematic equipment cannot be repaired or replaced immediately, problems with individual FSM objects or sub-trees can be temporarily masked by changing the partitioning mode of the corresponding object, i.e. disabling a device or excluding a sub-tree. Similarly, sets of or individual alarms may be masked at the level of operator interfaces.

Process logging. Each DCS control application process provides a detailed event log. In order to classify the events, all messages use predefined message types such as Information, Warning, Error, Fatal, Action. The types of reported events include:

- PVSS application-specific messages such as process status or errors (PVSS internal message types),
- changes of FSM object State (Information) and Status (message class corresponding to Status severity),

- FSM command received (Action) and execution progress (Information or chosen severity),
- control application error messages with respect to data and event handling (chosen severity).

A central logging process on a GCS collects all application logs and archives them on a central database server.

7.2 Operator interfaces

The DCS is operated from a dedicated station within the ATLAS control room. The station provides several screens which host the two primary user interfaces — the FSM Screen for operation of the detector Finite State Machine hierarchy and the Alarm Screen for alarm recognition and acknowledgment. Further interfaces used on demand are tools for data visualization, process log viewer, and operator log book. Full remote access to all user interfaces is possible by a terminal server for expert operators only. Static status monitoring is provided by web pages on a dedicated web server world-wide.

7.2.1 Finite State Machine operator interface

The ATLAS FSM Screen is the primary operator interface for the ATLAS DCS. It provides the operator with a view of any selected FSM object and allows free navigation within the complete detector FSM hierarchy. Each FSM object has an associated panel supplying the operator with detailed information such as parameter values, trends, synoptic views within the context of the selected object. Actual control is performed by means of FSM configuration and commands.

Figure 16 shows an example of the FSM Screen with its individual modules:

1. FSM module:
   shows the currently active FSM object and its children. For each object, its name, State, Status, and partitioning mode are displayed and various operations such as hierarchy navigation, sending of commands, and partitioning actions can be performed.

2. Main module:
   shows the main panel associated to the currently active FSM object. On navigation to another FSM object within the hierarchy, the panel displayed is changed accordingly.

3. Secondary module with two possibilities:

   - shows the secondary panel associated to a chosen FSM object – decoupled from the main navigation — which allows displaying information for a second FSM object in addition to the currently selected one.
   - shows a 3-dimensional representation of all objects contained within the current FSM subtree. The object shapes are loaded from a database and its color reflects the current State or Status. The canvas can be freely rotated and zoomed. Further, the object shapes can be used for navigation such that the operator, triggered by color changes, can navigate with a single click to any problematic object for investigation.

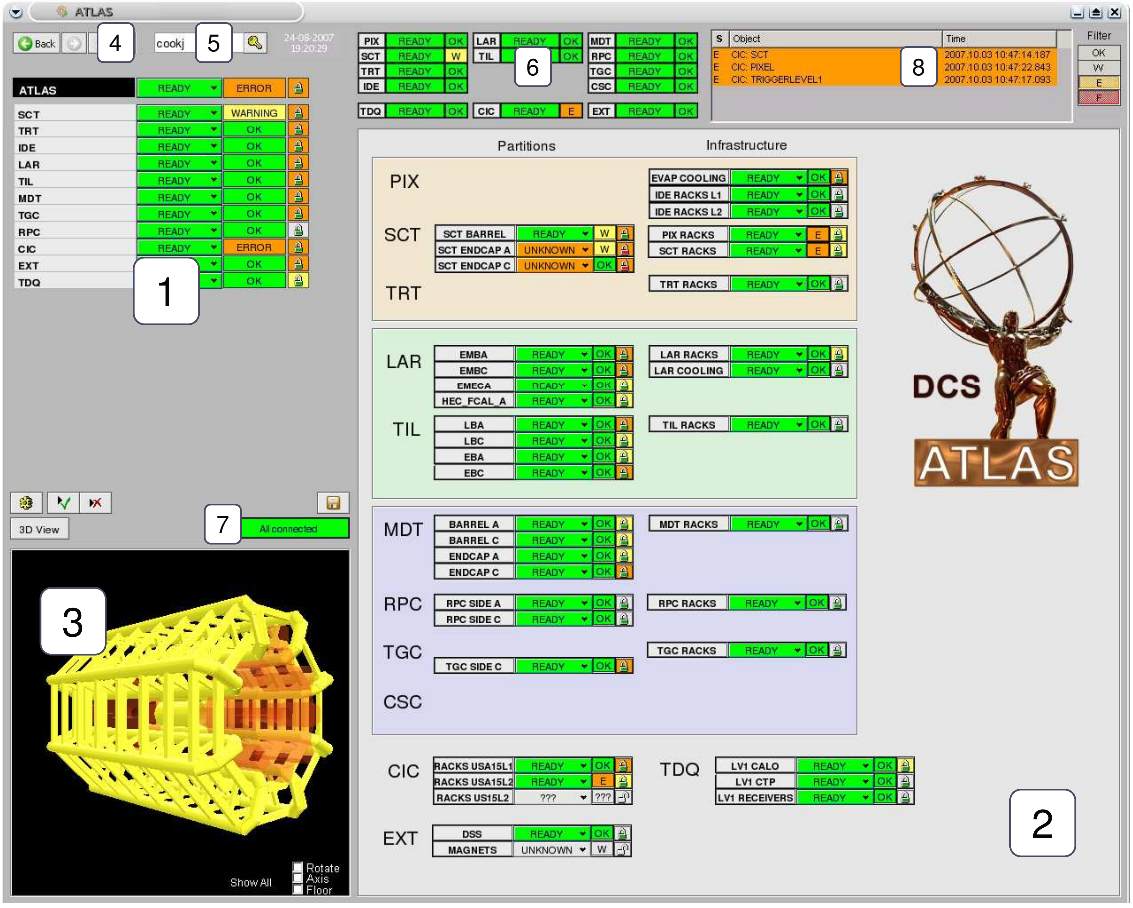


Figure 16. FSM operator interface.

4. Navigator module:
   navigation buttons, similar to those of a web browser, which allow accessing the navigation history, i.e. back, forward, up, and home.

5. Access control module:
   shows the user currently logged in and provides access control information such as the list of users.

6. Overview module:
   shows an overview of the State and Status of all ATLAS sub-detectors.

7. Connection module:
   informs about the connection state of the complete distributed system of control stations with respect to the operator station. The number of missing connections if any and the names of disconnected systems are displayed as a tool-tip.

8. Error list module:
   displays the Status of all controlled FSM objects with device object children. The list can be

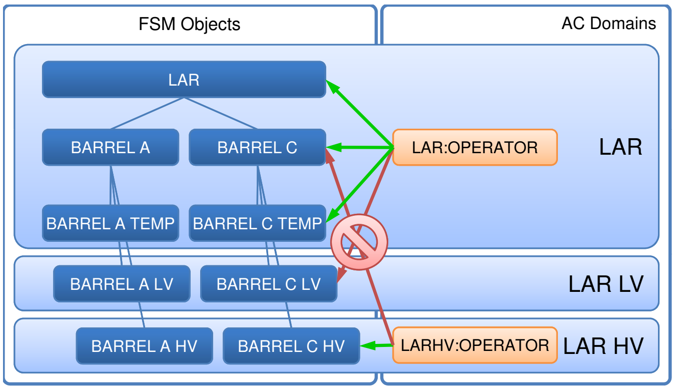


Figure 17. FSM access control scheme. A user with the role LAR:Operator can control all FSM nodes assigned to the LAR domain. Only a LARHV:Operator can operate the high voltage subsystem of a partition exclusively.

filtered according to the Status severity and used for the direct navigation to the respective level of the hierarchy thus allowing for quick error recognition and diagnostics.

The FSM Screen is protected by the Access Control (AC) mechanism using predefined privileges. The “Monitor” privilege is always granted by default such that every user is able to observe the status of the system. Two levels of control access are distinguished for each AC domain, the “Operator” and the “Expert” role.

An Operator is allowed to perform FSM-related actions such as changing partitioning modes and sending commands to all FSM objects within a particular AC domain. Each sub-detector has an AC domain associated to its FSM subtree allowing a user with the sub-detector Operator role to control the whole sub-detector as such. Additional sub-detector AC domains can be defined for particular sub-systems such as a high voltage system. Only Operators within these domains can control FSM objects within the corresponding subtree. Accordingly, as illustrated in figure 17, a calorimeter Operator can send a complete partition to the READY state, but is neither allowed to exclude a particular high voltage segment from the tree nor sending that segment to READY.

A domain Expert inherits the Operator role for the respective domain and can perform in addition FSM commands which are declared as expert-only. Additionally, an Expert is allowed to use control elements within FSM panels, can operate panels of the LCS control application in this domain, and directly manipulate control application parameters.

7.2.2 Alarm screen

The ATLAS Alarm Screen (see figure 18), based on the JCOP alert screen implementation, displays the list of currently active alarms within the whole DCS, allowing an operator to identify problematic detector elements, take corrective measures, and document the actions taken. Further it can be used to diagnose problems at a later stage even after the alarms have been cleared. The individual user interface elements are:

1. Alarm table
   For each alarm, the following is displayed:

- the severity,
- the description of the corresponding detector element,
- a short problem description,
- the parameter value at the current time and at the time when it was raised,
- the acknowledgment state if applicable,
- the time at which the alarm was raised.

Further options are available on a context menu for each line, such as a possibility to enter an operator comment for the alarm or open a trend plot for the corresponding parameter value.

2. Filter settings
   The list of alarms in the table can be filtered according to the following parameters:

   - originating control station (Project),
   - severity,
   - element identifier (pattern match),
   - acknowledgment type.

   The filter settings can be pre-defined in presets.

3. Mode selection
   Allows toggling between two modes:

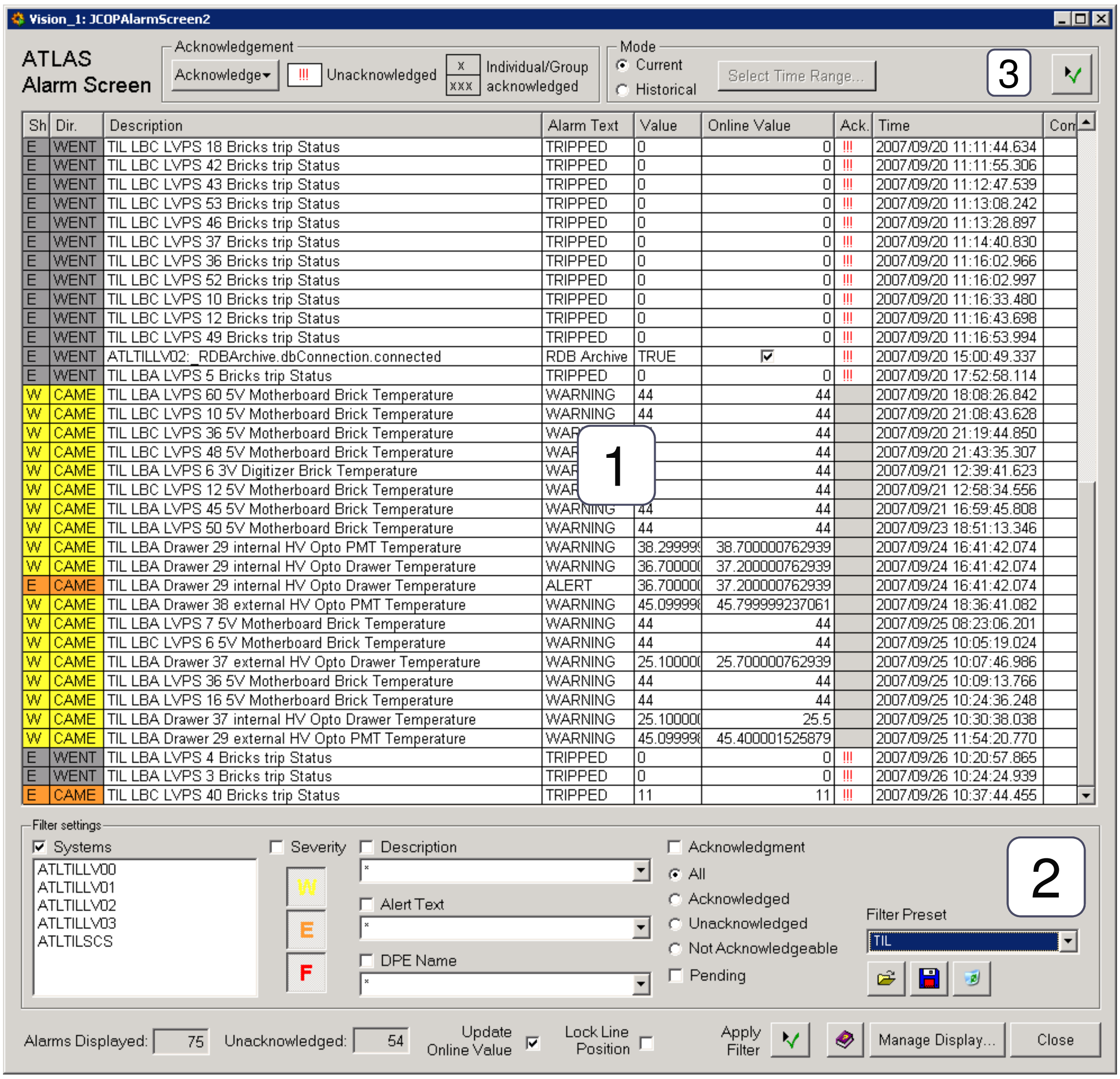
   - Current: Shows the list of currently active alarms.
   - Historical: Shows alarms which were active during a specified time period.

In order to keep the number of active alarms to a minimal and manageable level, two alarm reduction mechanisms exist. Firstly, summary alarms can be configured such that the alarm screen suppresses the corresponding individual alarms if their number exceeds a predefined threshold. In this case, only the summary alarm is visible to the operator. Secondly, active alarms can be temporarily masked in case the problem is understood and cannot be solved on a short timescale.

7.2.3 Remote access

In addition to the user interfaces in the main control room, all user interfaces are available to satellite control rooms allowing the respective operators to gain control of the DCS FSM provided they own the proper access control privileges.

Terminal server. Full operator access to the control interfaces outside of the ATLAS local area network is granted remotely only to registered detector experts using a Windows Terminal Server (WTS) which is accessible from wide area networks. The user interfaces are started on this server connecting only to relevant PVSS processes on the respective control station within the ATLAS local area network, thus minimizing the work load put onto DCS machines. The use of conflicting FSM commands is prevented by the mechanisms of the FSM toolkit which allows direct command access only from the main operator interface unless the operator explicitly allowed shared control by other users. All interfaces on the WTS are protected by the access control mechanism.



Sh	Dir.	Description	Alarm Text	Value	Online Value	Ack.	Time	Com
E	WENT	TIL LBC LVPS 18 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:11:44.634	
E	WENT	TIL LBC LVPS 42 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:11:55.306	
E	WENT	TIL LBC LVPS 43 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:12:47.539	
E	WENT	TIL LBC LVPS 53 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:13:08.242	
E	WENT	TIL LBC LVPS 46 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:13:28.897	
E	WENT	TIL LBC LVPS 37 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:14:40.830	
E	WENT	TIL LBC LVPS 36 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:16:02.966	
E	WENT	TIL LBC LVPS 52 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:16:02.997	
E	WENT	TIL LBC LVPS 10 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:16:33.480	
E	WENT	TIL LBC LVPS 12 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:16:43.698	
E	WENT	TIL LBC LVPS 49 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 11:16:53.994	
E	WENT	ATLTILLV02:_RDBArchive.dbConnection.connected	RDB Archive	TRUE	☑	!!!	2007/09/20 15:00:49.337	
E	WENT	TIL LBA LVPS 5 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/20 17:52:58.114	
W	CAME	TIL LBA LVPS 60 5V Motherboard Brick Temperature	WARNING	44	44		2007/09/20 18:08:26.842	
W	CAME	TIL LBC LVPS 10 5V Motherboard Brick Temperature	WARNING	44	44		2007/09/20 21:08:43.628	
W	CAME	TIL LBC LVPS 36 5V Motherboard Brick Temperature	WAR		44		2007/09/20 21:19:44.850	
W	CAME	TIL LBC LVPS 48 5V Motherboard Brick Temperature	WAR		44		2007/09/20 21:43:35.307	
W	CAME	TIL LBA LVPS 6 3V Digitizer Brick Temperature	WAR		44		2007/09/21 12:39:41.623	
W	CAME	TIL LBC LVPS 12 5V Motherboard Brick Temperature	WAR		44		2007/09/21 12:58:34.556	
W	CAME	TIL LBA LVPS 45 5V Motherboard Brick Temperature	WARNING	44	44		2007/09/21 16:59:45.808	
W	CAME	TIL LBA LVPS 50 5V Motherboard Brick Temperature	WARNING	44	44		2007/09/23 18:51:13.346	
W	CAME	TIL LBA Drawer 29 internal HV Opto PMT Temperature	WARNING	38.29999	38.700000762939		2007/09/24 16:41:42.074	
W	CAME	TIL LBA Drawer 29 internal HV Opto Drawer Temperature	WARNING	36.70000	37.200000762939		2007/09/24 16:41:42.074	
E	CAME	TIL LBA Drawer 29 internal HV Opto Drawer Temperature	ALERT	36.70000	37.200000762939		2007/09/24 16:41:42.074	
W	CAME	TIL LBA Drawer 38 external HV Opto PMT Temperature	WARNING	45.09999	45.799999237061		2007/09/24 18:36:41.082	
W	CAME	TIL LBA LVPS 7 5V Motherboard Brick Temperature	WARNING	44	44		2007/09/25 08:23:06.201	
W	CAME	TIL LBC LVPS 6 5V Motherboard Brick Temperature	WARNING	44	44		2007/09/25 10:05:19.024	
W	CAME	TIL LBA Drawer 37 external HV Opto Drawer Temperature	WARNING	25.10000	25.700000762939		2007/09/25 10:07:46.986	
W	CAME	TIL LBA LVPS 36 5V Motherboard Brick Temperature	WARNING	44	44		2007/09/25 10:09:13.766	
W	CAME	TIL LBA LVPS 16 5V Motherboard Brick Temperature	WARNING	44	44		2007/09/25 10:24:36.248	
W	CAME	TIL LBA Drawer 37 internal HV Opto Drawer Temperature	WARNING	25.10000	25.5		2007/09/25 10:30:38.038	
W	CAME	TIL LBA Drawer 29 external HV Opto PMT Temperature	WARNING	45.09999	45.400001525879		2007/09/25 11:54:20.770	
E	WENT	TIL LBA LVPS 4 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/26 10:20:57.865	
E	WENT	TIL LBA LVPS 3 Bricks trip Status	TRIPPED	0	0	!!!	2007/09/26 10:24:24.939	
E	CAME	TIL LBC LVPS 40 Bricks trip Status	TRIPPED	11	11	!!!	2007/09/26 10:37:44.455	

Filter settings

☑ Systems: ATLTILLV00, ATLTILLV01, ATLTILLV02, ATLTILLV03, ATLTILSCS

□ Severity: W, E, F

□ Description: *

□ Alert Text: *

□ DPE Name: *

□ Acknowledgment: ● All ○ Acknowledged ○ Unacknowledged ○ Not Acknowledgeable

□ Pending

Filter Preset: TIL

Alarms Displayed: 75 Unacknowledged: 54 Update Online Value ☑ Lock Line Position □

Figure 18. Alarm system operator interface.

WWW monitoring. The information provided by the FSM Screen is exported to a web server in order to allow static monitoring of the ATLAS status. All FSM object information and their corresponding main panels up to the level of sub-detector partitions are collected from persistent user interface instances within a time interval of less than one minute. Thus, the user interface is effectively mapped to a web page without imposing any additional load to the network of BE control stations. The information is accessible world-wide using a common web-browser as shown in figure 19.

7.2.4 Data visualization

Conditions data written to the database archive can be accessed in several ways. The FSM panels displaying operational parameters either contain or allow to open PVSS trend plots required to diagnose error conditions occurring during operation. In order to use more sophisticated tools such

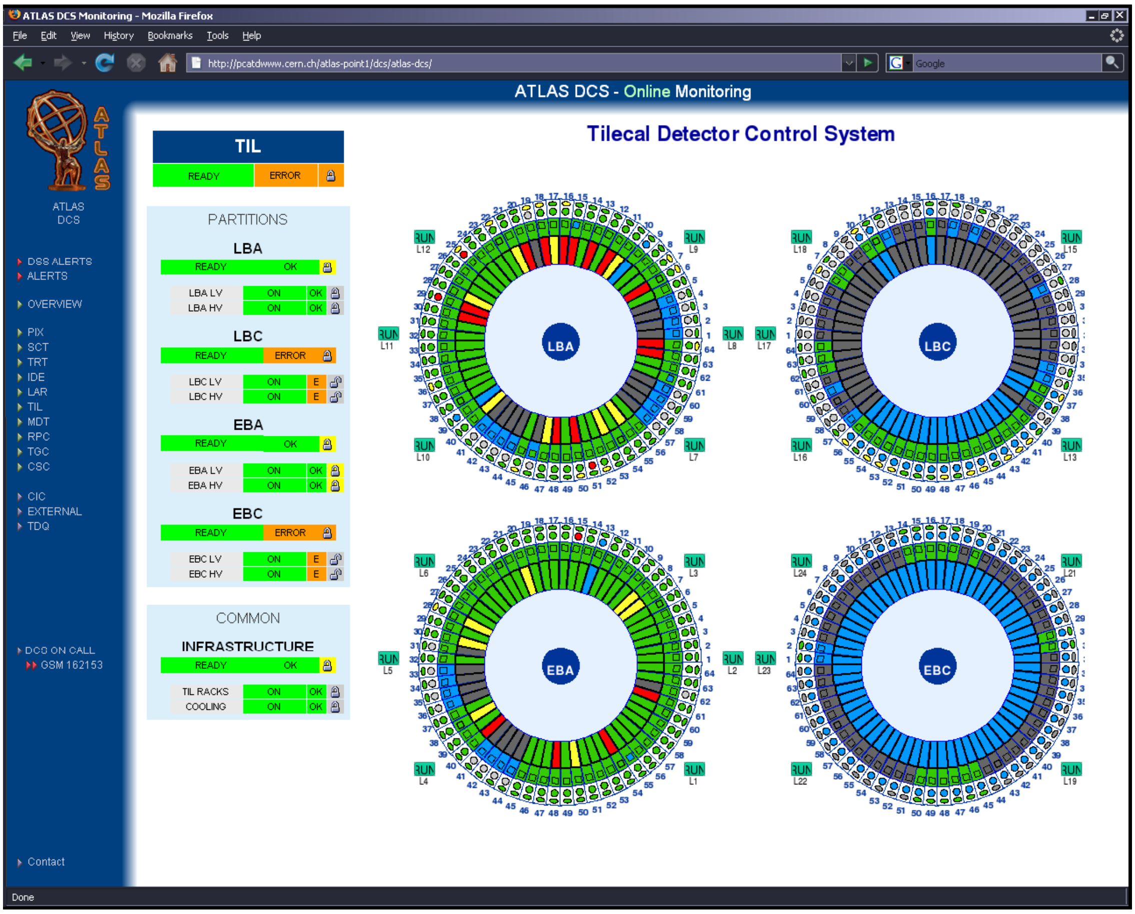


Figure 19. WWW monitoring page showing the online status of the Tilecal sub-detector.

as the ROOT data viewer (see section 5.8.2) in a user friendly way, a coherent data element identification scheme is used, in contrast to the element naming within the control application which is usually chosen according to hardware installation or technological constraints. An individual data element is characterized by a description containing sub-detector identifier, partition name, and subsystem name followed by a logical identification string, for example “TIL LBC LVPS 45 5V Motherboard Brick Temperature”. The element description usually indicates the element position with respect to the FSM hierarchy and can be used by offline analysis data queries without control system expert knowledge.

7.3 DCS operation

The ATLAS DCS has to supervise all detector components at all times. A dedicated operator in the ATLAS control room reacts on any non-desired change of detector conditions using the operator interfaces introduced above. Any problem recovery requires close coordination with the ATLAS subsystem operator concerned within the control room. Further, the operator performs any change in detector condition or configuration which may be required using FSM commands on the respective parts of the detector hierarchy.

For actual physics data taking, the TDAQ RC is allowed to send FSM commands to the partition level of the FSM hierarchy using the DDC interface (see section 5.10). During ongoing runs, any error condition of a DCS partition can be signaled to RC which may react - depending on the problem severity - by either stopping the data acquisition process or detaching the problematic partition and trying to recover the problem by sending DCS FSM commands to the respective partition.

7.3.1 Procedures for operation

Overall detector operation is organized in shifts, for which the operator has to complete a check list of tasks to be performed at the shift start and prior to initiation of a data acquisition run. In a first step, all currently active alarms have to be resolved or understood and masked. Subsequently, the FSM hierarchy has to be prepared by including all detector partitions participating in the run and checking the availability of all subtrees. Finally, the partitions have to be moved to the desired State (for example READY) and Status OK. Thereafter, the operator keeps on monitoring the system and reacts to problems emerging.

7.3.2 Alarm response

Whenever alarms are raised within the system, they are followed up according to their severity WARNING, ERROR, and FATAL, with possible implications ranging from preventive actions to the interruption of the data taking process. Normally, detailed problem diagnostics is delegated to the respective sub-detector expert. During this investigation, the corresponding sub-system alarms may be masked.

7.3.3 FSM operation

Operation using the FSM hierarchy requires well defined procedures since misconfigurations or the sending of conflicting commands by several operators represents a considerable risk for the efficient and safe detector operation.

By default, the FSM control is held exclusively by the operator. Whenever it is necessary for a remote operator or expert to gain control over a particular part of the tree, the operator can move the top FSM object of the respective sub-tree to the Shared partitioning mode. After the intervention has been finished, the Exclusive mode is restored. In case of a persistent problem with one or a group of FSM objects, the elements can be Disabled or Excluded from the hierarchy such that the error condition is effectively removed from the system and normal operation can be resumed.

7.3.4 Run Control operation

The ATLAS Run Control operates the data taking with a state machine using a different state model than that of the DCS FSM. In order to synchronize the DCS and RC states, each data taking partition of the respective hierarchy has an associated object which handles state and command synchronization via DDC. A RC command for any TDAQ TTC partition is interpreted by its DDC Controller, translated to the corresponding DCS FSM command and sent to the FSM object of the respective DCS partition. In turn, during data taking the RC must react on any state change of the participating DCS partitions since it may prevent acquiring usable physics data.

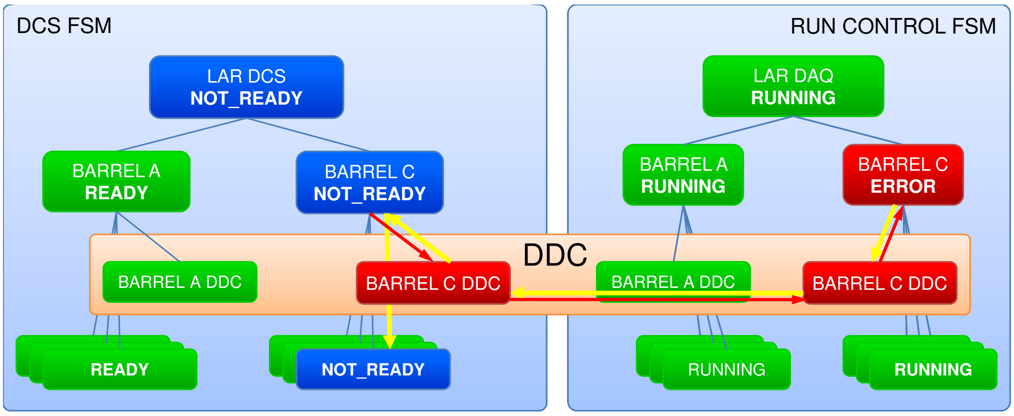


Figure 20. DCS-TDAQ communication for a sub-detector. The DCS FSM partition serves as primary interface for reporting state changes (red arrows) to the corresponding TTC partition within the TDAQ Run Control hierarchy and may receive and execute RC partition commands (yellow arrows).

Figure 20 illustrates a typical example for the communication paths used. During a run, a problem within a DCS partition leads to a state change of the partition from READY to NOT_READY, which is processed by its DCS DDC object and transferred to the corresponding DDC controller within the RC tree (red path). The controller raises its error flag which is propagated to higher RC hierarchy levels, indicating that the data acquisition may be problematic. In order to recover the error, RC can either stop and restart the run by using RC transitions, or send a GOTO_READY command as a Non-transition Command via DDC to the DCS partition, which will in turn execute the action and propagate the command to the problematic subsystem (yellow path). During the RC command execution in the DCS FSM, no interfering commands can be issued from any other operator interface. In the case of an unsuccessful recovery attempt, RC may exclude this TTC partition from the data taking process.

8. Subsystem control

The Sub-detector Control Station has a central role for the supervision of a sub-detector. It allows full stand-alone operation of the entire sub-detector, interacts with the corresponding part in the TDAQ read-out, and provides connection to the global operations layer of ATLAS. The design and implementation of the applications of the SCS and all connected LCS are the responsibility of the sub-detector groups, using the standard software tools and packages described above. A central DCS team in ATLAS guides the sub-detector groups in setting up and using these components. This team also implements the supervision of all parts of ATLAS, which are not under the direct responsibility of a sub-detector. These monitoring and control applications form the Common Infrastructure Control (CIC), grouped within a dedicated hierarchy, represented by an SCS and its associated LCS. External control systems are included in the CIC tree, retrieving the corresponding information from the DCS Information Server (see section 5.9).

8.1 Sub-detectors

Each sub-detector is hierarchically structured and modeled with the FSM as discussed in sec-

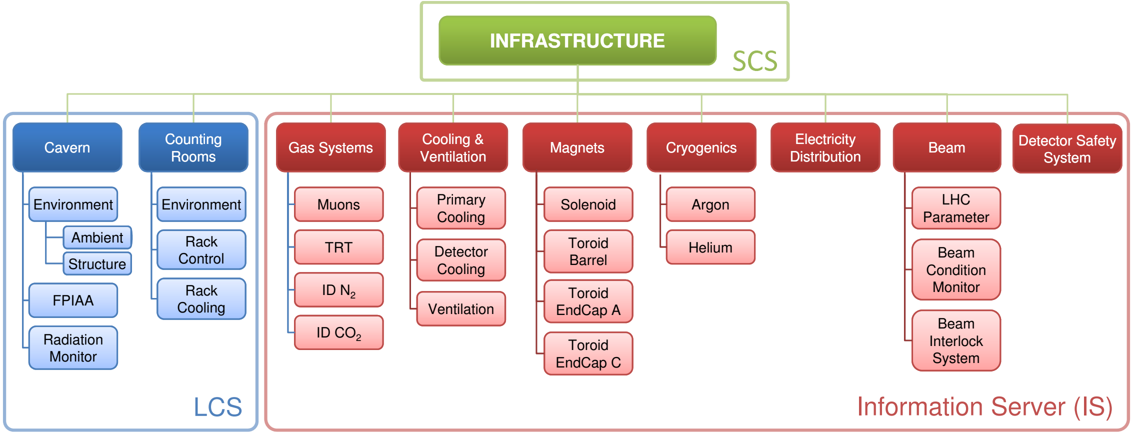


Figure 21. FSM structure of the common infrastructure.

tion 7.1.1. At the topmost level, each sub-detector consists of up to 6 TTC partitions. The structure below this level can be organized in two ways — either “functionally” — in high voltage, low voltage, detector electronics, gas, etc. — or “geometrically” — in barrel, side A, side C, etc. This choice is based on practical aspects. The depth of the hierarchy is defined individually for each sub-detector and can have up to the order of 10 levels. The lowest level is formed by the device objects, such as voltage channels or sensors. The detailed structure of the sub-detector control and the description of associated procedures can be found in [1] and the references therein.

8.2 Infrastructure

All parts of ATLAS, which are common to the detector are supervised by the CIC, which is structured as shown in figure 21.

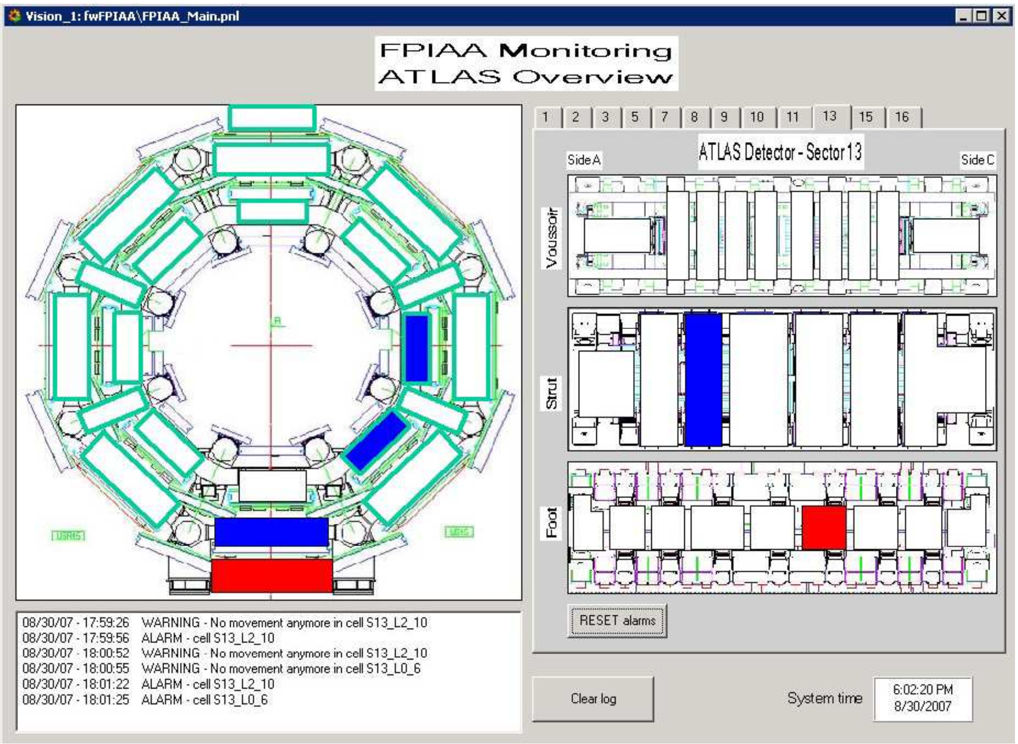
This equipment is geographically distributed over the whole experiment and in total 6 LCS have been set up, two serving the detector cavern UX15, and one for each of the electronics rooms underground USA15L1, USA15L2, and US15L2, and one for the TDAQ computer rooms in SDX1 at the surface. The SDX1 station includes also the supervision of the gas building SGX1. The sensor read-out is mainly based on the ELMB as described below.

8.2.1 Cavern and counting room environment

The general environment in UX15 is supervised using a CAN network with about 90 ELMB nodes, which cover the whole volume of the experiment. About one third of the 8000 channels are presently used, the rest are foreseen for future upgrades. All sensors described in the following are read by these ELMBs.

Instrumentation has been installed to measure ambient parameters such as temperature, humidity, pressure. More than hundred PT100 probes have been placed on the aluminum structures of the detector. These sensors and their connections are designed to operate up to 200 °C in order to be able to localize and assess major accidents, e.g. a fire or a cryogenic leak.

The space of the experiment outside of the calorimeters is accessible in periods without beam. As these volumes are often quite small and confined and interconnected in a very complex way, a



```
08/30/07 - 17:59:26   WARNING - No movement anymore in cell S13_L2_10
08/30/07 - 17:59:56   ALARM - cell S13_L2_10
08/30/07 - 18:00:52   WARNING - No movement anymore in cell S13_L2_10
08/30/07 - 18:00:55   WARNING - No movement anymore in cell S13_L0_6
08/30/07 - 18:01:22   ALARM - cell S13_L2_10
08/30/07 - 18:01:25   ALARM - cell S13_L0_6
```

Figure 22. Synoptic panel showing the presence of personnel in the ATLAS experiment. Blue indicates activity in this cell, red signals an abnormal situation.

dedicated system called Finding Persons Inside ATLAS Area (FPIAA) [29] has been developed in order to be able to track people inside the ATLAS cavern, e.g. during detector maintenance periods. Some 500 infrared sensors detect the movement of people. These signals are analyzed in real time and in a case of an abnormal situation of a person not moving for a long time, an alarm is generated and displayed in the control room. The overview panel of the FPIAA system is shown in figure 22.

The level of radiation inside ATLAS needs to be continuously monitored. About 60 radiation sensors [30] will be installed just outside of the calorimeter and in the Muon detector. Their FE electronics is also read out by the ELMB network in the cavern.

In each of the counting rooms, ELMBs are installed to read temperature, humidity and pressure sensors. Connection to analogue signals of the safety systems such as smoke and environmental radiation levels is envisaged.

The electronics racks are cooled by air flow. A turbine unit forces air through the equipment to be cooled and which is interspersed with water-air heat exchangers. The chilled water is provided by a secondary cooling plant as described below. Each turbine unit includes a monitor board which supervises all operational parameters such as temperature, humidity, air flow, and electricity distribution. A JCOP framework software component allows the supervision of the PLC of the electricity distribution system in order to switch power to individual racks and combines the information for a complete electronics room within an overview panel. Further, the power control and condition monitoring for each rack and is handled by an FSM object implementation.

8.2.2 Infrastructure services

Cooling and ventilation. Cooling and ventilation is an infrastructure service provided by CERN. A primary water cooling plant is installed at the ATLAS site on the surface. It provides cooling to secondary plants underground, which cool a sub-detector or equipment like vacuum pumps or cables. The operational parameters of each of these secondary plants are read out by the respective sub-detector. In addition, it is foreseen that the CIC monitors the overall status of the primary and all secondary cooling plants. The ventilation system for the underground rooms and the cavern is operated autonomously and its status will be transmitted to the DCS IS.

Electricity distribution. The electricity distribution system as part of the CERN infrastructure is supervised by a dedicated control system. It is planned that for the part relevant for ATLAS (e.g. distribution cabinets, switch boards, UPS systems) the status is also monitored by DCS.

Gas systems. Each of the gas systems of the different sub-detectors is controlled by a dedicated PLC. All PLCs are supervised by one PVSS system, which is not part of the ATLAS distributed control system. The operational parameters are published by DIP and the DCS IS transfers them into PVSS Datapoints for further distribution to the sub-detectors. The relevant sub-detector control project sets up a FSM hierarchy for its gas system and includes it in its sub-detector tree.

Magnets and cryogenics. The ATLAS magnets and cryogenics systems are also controlled by dedicated PLCs and monitored by a stand-alone PVSS station and information is retrieved via DIP to the DCS.

Each of the infrastructure systems is represented in the CIC FSM hierarchy including corresponding status panels. All infrastructure parameters are available to the sub-detector applications and are archived to the CondDB.

8.2.3 LHC accelerator

The interaction by software between ATLAS and the LHC is handled by DCS. Dedicated instrumentation on both sides provides detailed information about luminosity and backgrounds via the DIP protocol. The state of the LHC accelerator is presented to the ATLAS operator by the DCS. A Beam Interlock System (BIS) combines signals in ATLAS indicating high backgrounds, and in case of danger for the detector sends a hardware interlock signal to LHC in order to dump the beams. The status of the BIS is presented by the CIC as an FSM unit.

8.2.4 DSS

The Detector Safety System (DSS) [31] has the task to detect possibly dangerous situations for the detector e.g. due to overheating, failure of services, etc. and to shut down the relevant detector automatically. It is based on redundant PLCs and is supervised by a stand-alone PVSS system. All alarms of the DSS are transmitted to the DCS and can be used by the sub-detectors to execute predefined control procedures. DSS actions can be delayed in order to enable DCS to execute shut-down procedures before the DSS switches off the equipment. An example for this interaction is a cooling failure of the counting room racks. As soon as the DSS triggers the corresponding alarm, the alarm signal is propagated via the DIP protocol to the DCS IS. The individual sub-detector applications react on this signal change within the DCS IS by shutting down the equipment in the racks in a controlled way, before the DSS cuts the rack power after the predefined delay of 5 minutes.

However, the interaction between the DSS and DCS is limited to one direction: the DCS has no way of influencing the operation of the DSS.

9. Conclusions

The design and implementation of the ATLAS DCS is based on 4 building blocks:

- Industrial solutions
- CERN-developed components (JCOP)
- ATLAS-wide integration packages
- Sub-detector specific controls procedures

This approach fulfills all requirements for coherent detector control and allows minimizing the implementation and maintenance efforts.

When choosing commercial equipment for the experiment, the interface to the DCS was considered in the selection process. As standard hardware connection either Ethernet or the CAN fieldbus are used and the communication standards are OPC and CANopen. In cases where no commercial devices which fulfilled the specific requirements existed, in particular tolerance to ionizing radiation and operation in a strong magnetic field, a dedicated solution was developed. Most of them include the general purpose I/O system ELMB as controls interface. The general design of the ELMB allowed using it in a wide range of embedded applications and also as a stand-alone unit reading sensors directly. As the ELMB is used in all LHC experiments a large volume of more than 10 000 units has been produced, making this project very cost-effective.

Concerning software, the selection of the SCADA software product PVSS proved to be a good choice. The readout interfaces for various types of FE equipment, the possibility to store conditions data persistently in relational databases, the open design allowing for custom software extensions, and the good scalability within a highly distributed system made PVSS suitable for all ATLAS DCS applications. The use of PVSS has provided a common skeleton for the development and implementation of the individual applications thus facilitating long-term maintainability, though at the expense of dependence on external maintenance of PVSS itself. As the four LHC experiments are similarly structured and hence have many common requirements, further synergy effects in software development were achieved successfully in the JCOP collaboration. The resulting common software solutions were mostly engineered in the experiment controls group IT/CO at CERN with additional contributions from individual LHC experiments.

In order to achieve a homogeneous control system for the whole of ATLAS, the DCS BE applications, distributed within a network of dedicated control stations, are hierarchically structured following the natural segmentation of the detectors with several functional layers. Strong emphasis was put on guidelines and development conventions and on the implementation of common application components. The monitoring and control of each part of the control hierarchy is provided by a finite state machine mechanism, which effectively reduces the complex set of FE component states of the different ATLAS sub-detectors to a single overall state. Efficient error recognition and handling is provided by a centralized alarm system which raises alarms at the granularity of the individual FE devices and propagates these alarms within the finite state machine hierarchy.

The individual DCS applications for the respective sub-detector components read and archive conditions data, and implement dedicated control procedures for the associated equipment. Thus, expert knowledge contained within these control procedures becomes an integrated part of the DCS itself and evolves as operational experience is gained with time. This is particularly important as the experiment is scheduled to run for more than a decade and personnel will change during this period. Accordingly, an effort has been made to manage software centrally and to track all development stages. In addition to the sub-detector control applications, supervision of the common infrastructure within ATLAS such as the racks in the counting rooms and environment monitoring was implemented and is routinely running. Furthermore, external systems such as gas and cryogenics services as well as the condition of the LHC accelerator are interfaced coherently using a multi-platform communication protocol and are integrated within the common control mechanisms.

At the time of writing this paper, the control of about 50% of the ATLAS experiment is integrated within the overall DCS. During the ongoing commissioning of ATLAS, it was proven that the DCS scales up to the level needed and is able to continuously provide stable detector operation.

Acknowledgments

We are thankful to W. Heubers and Y. Ryabov for many valuable discussions about concept and implementation of the DCS and for their continuous support. We are also much indebted to our colleagues in the JCOP from the CERN controls group IT/CO, from the LHC experiments ALICE, CMS, and LHCb, and from the electronics support group PH/ESS for their contributions and the good collaborative spirit. We would also like to thank H. Kvedalen, R. Mattsson, and B. Varnai, who participated at the preparation of the DCS in the frame of their studies. Finally, the contributions of the group from the Fachhochschule Wiener Neustadt, which recently joined us, are gratefully acknowledged.

References

[1] The ATLAS collaboration, *The ATLAS experiment at the CERN Large Hadron Collider*, to be published in JINST (2008).

[2] O. Holme, M. Gonzalez Berges, P. Golonka and S. Schmeling, *The JCOP framework*, proceedings of the *10th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS 2005)*, Geneva, Switzerland (2005).

[3] ATLAS TDAQ collaboration, *Integration of the Trigger and Data Acquisition Systems in ATLAS*, reported at *International Conference on Computing in High Energy Physics (CHEP '07)*, Victoria BC, Canada, to be published in *J. Phys. CS* (2007).

[4] ETM professional control GmbH, PVSS II SCADA Product, http://www.pvss.com.

[5] CAN in Automation, CAN, http://www.can-cia.org/.

[6] OPC Foundation, http://www.opcfoundation.org.

[7] 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*, proceedings of the 7th *Workshop on Electronics for LHC Experiments*, Stockholm, Sweden (2001).

[8] CERN Fieldbus Working Group, *CERN Fieldbus Working Group report*, http://ab-div-co-is.web.cern.ch/ab-div-co-is/IndCtrl/FB/Welcome.html.

[9] ATMEL, *8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash*, ATmega128, ATMEL product datasheet http://www.atmel.com/products/AVR/.

[10] B. Hallgren, *Design specification for the ATLAS DCS motherboard*, ATLAS internal note, EDMS document 348202, https://edms.cern.ch/document/348202 (2004).

[11] M. Dentan, *ATLAS policy on radiation tolerant electronics revision 2*, ATLAS internal note, EDMS ATC-TE-QA-0001 (2000).

[12] R. Rausch and M. Tavlet, *On-line radiation test facility for industrial equipment needed for the Large Hadron Collider at CERN*, Prepared for 7th *International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS 99)*, Trieste, Italy (1999).

[13] H. Hufnagel, *Analyse einer Interlock-Box*, Studienarbeit, Institue fur Technische Informatik, Lübeck (2002).

[14] H. Boterenbrood, *CANopen Application Software for the ELMB128*, NIKHEF, Amsterdam, http://www.nikhef.nl/pub/departments/ct/po/html/ELMB128/ELMB23.pdf, Version 2.3.

[15] J. Cook, *ELMB PSU Introduction*, ATLAS internal note, EDMS document 771964, https://edms.cern.ch/document/771964/1 (2004).

[16] C. Gaspar, M. Dönszelmann and Ph. Charpentier, *DIM, a portable, light weight package for information publishing, data transfer and inter-process communication, Comput. Phys. Commun.* **140** (2001) 102.

[17] Microsoft Corporation, The Component Object Model Specification, http://www.microsoft.com/Com/resources/comdocs.asp (1995).

[18] KVASER AB, Kvaser PCIcan, http://www.kvaser.com/prod/hardware/pcican.htm.

[19] National Instruments Corporation, NI USB-8476 LIN Interfaces, http://sine.ni.com/nips/cds/view/p/lang/en/nid/203383.

[20] C. Gaspar and B. Franek, *Tools for the automation of large distributed control systems, IEEE Trans. Nucl. Sci.* **53** (2006) 974.

[21] B. Franek and C. Gaspar, *SMI++ — Object Oriented Framework for Designing and Implementing Distributed Control Systems, IEEE Trans. Nucl. Sci.* **51** (2004) 513.

[22] OpenLDAP Foundation, *OpenLDAP — community developed LDAP software*, http://www.openldap.org/.

[23] R. Hawkings, F. Viegas and G. Dimitrov, *Relational databases for conditions data and event selection in ATLAS*, reported at *International Conference on Computing in High Energy Physics (CHEP '07)*, Victoria BC, Canada, to be published in *J. Phys. CS* (2007).

[24] R. Chytracek et al., *POOL development status and production experience, IEEE Trans. Nucl. Sci.* **52** (2005) 2827.

[25] R. Brun and F. Rademakers, *ROOT: an object oriented data analysis framework, Nucl. Instrum. Meth.* **A 389** (1997) 81.

[26] I. Alexandrov et al., *Online software for the ATLAS test beam data acquisition system*, *IEEE Trans. Nucl. Sci.* **51** (2004) 578.

[27] S. Lüders, *Update on the CERN Computing and Network Infrastructure for Controls (CNIC)*, CERN-IT-Note-2007-015 (2007).

[28] Ximbiot LLC, CVS, http://ximbiot.com/cvs/.

[29] C. Cardeira et al., *Communication architecture of a system to find persons inside ATLAS*, Proceedings of the 6th *IEEE International Workshop on Factory Communication Systems*, Torino (2006).

[30] M. Mikuz et al., *Diamond pad detector telescope for beam conditions and luminosity monitoring in ATLAS*, *Nucl. Instrum. Meth.* **A 579** (2007) 788.

[31] O. Beltramello et al., *The detector safety system of the ATLAS experiment*, submitted to *JINST*.