Chapter 7

Control System

7.1 Detector Control System (DCS)

7.1.1 Introduction

The primary task of the ALICE Control System is to ensure safe and correct operation of the ALICE experiment [17]. It provides remote control and monitoring of all experimental equipment in such a way that the ALICE experiment can be operated from a single workplace, the ALICE Control Room (ACR) at LHC point 2, through a unique set of operator panels. The system provides the optimal operational conditions so that the data taken with the experiment is of the highest quality. The ALICE control system was designed to reduce the downtime of the experiment and hence contribute to a high running efficiency. It also maximises the number of channels operational at any time, and measures and stores all parameters necessary for efficient analysis of the physics data.

7.1.2 Design strategy and system architecture

The ALICE control system is coordinated and built by the ALICE Control Coordination (ACC) team in collaboration with the sub-detector groups, the online systems groups and the various CERN services groups.

7.1.2.1 Requirements

Although covering a wide variety of components, and being developed by various groups in parallel, the DCS is a coherent and homogeneous system, and allows for easy integration of components. The DCS is flexible and scalable, to accommodate the changes of the experiment during its lifetime. As the control system is supposed to be operational throughout all operational phases of the experiment (data taking, shutdown etc.), it is able to cope with different operational modes, and allows independent and concurrent operation of each sub-detector, or any part of it.

Since the shift crew is not necessarily expert in controls or in the operation of a detector, special attention was given to the presentation of the system to the operator. Remote access to the DCS is needed and a strict access control mechanism was put in place, based on the origin of the access and the user profile. All data required to configure the experiment equipment is stored in databases. All relevant data for the operation of the experiment is archived and is available at

2008 JINST 3 S08002

any time. All data relevant for the analysis of the physics data is stored in a database and is easily accessible from the offline system.

Ensuring the integrity of the detector equipment is largely the task of the control system. The control system is reliable and allows for both hardwired and software actions in case of hazardous situations. It is easily maintainable, also in the long term when some of the original expertise might have disappeared.

7.1.2.2 Methods

Given the large number of sub-detectors and sub-systems that need to be controlled the only way to come to an integrated control system for the ALICE experiment is to use common solutions for common requirements. The ALICE Controls Coordination (ACC) was set up to coordinate all the controls activities within the experiment. The Joint COntrols Project (JCOP, [262]), a collaboration between CERN and the four LHC experiments, provided a discussion forum and developed a set of tools and components used to implement the control system. The ACC team systematically collected the controls requirements for each sub-system involved and established light-weight User Requirements Documents (URDs) in order to enable the identification of communalities. Each URD is accompanied by an 'overview drawing' that depicts the hardware structure of the control system of each sub-detector.

7.1.2.3 Architecture

The ALICE control system is responsible for configuring, monitoring and controlling the equipment of the experiment. These are commercial hardware devices such as power supplies and crates, as well as sub-detector specific equipment like front-end chips, etc. It also covers computing devices such as PCs and PLCs as well as the software processes running on them. This task is accomplished mainly by sending commands and settings to and reading information back from the equipment.

The control system is able to take pre-programmed decisions and automatic actions without operator intervention such as recovering from errors. The operator is able to interact with the control system through a user interface that presents the information and allows issuing of commands.

Hardware architecture

The hardware architecture of the control system can be divided into three layers; a supervisory, control and field layer as shown in figure 7.1. The supervisory layer consists of a number of computers (Operator Nodes, ON) that provide the user interfaces to the operators. The supervisory level interfaces to the control layer, where computers (Worker Nodes, WN) and PLC or PLC-like devices interface to the experimental equipment. These devices collect and process information from the lower, so called field layer, and make it available for the supervisory layer. At the same time it receives information from the supervisory layer to be processed and distributed to the field layer. The field layer comprises all field devices such as power supplies and fieldbus nodes, sensors, actuators, etc.

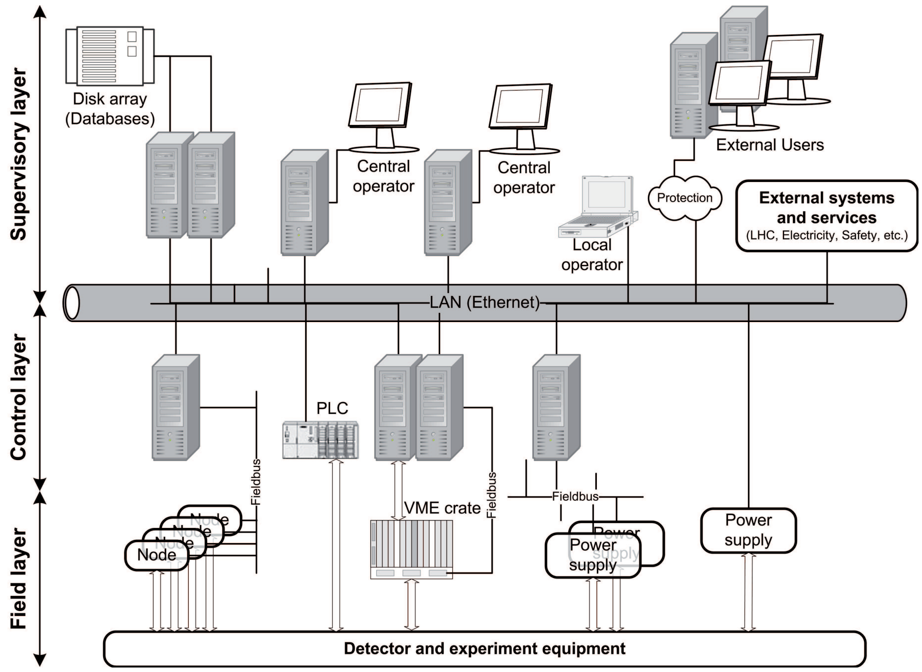

Figure 7.1: ALICE DCS hardware arcitecture.

Software architecture

The software architecture (figure 7.2) is a tree-like structure that represents the structure of sub-detectors, their sub-systems and devices. The structure is composed of nodes which each have a single 'parent' except for the top node called the 'root node' that has no parent. Nodes may have zero, one or more children. A node without children is called a 'leaf', and a subset of a tree's nodes is called a 'sub-tree'.

Control, Logical and Device Units

There are three types of nodes, a Control Unit (CU), a Logical Unit (LU) and a Device Unit (DU) that serve as basic building blocks for the entire hierarchical control system. The CU and LU model and control the sub-tree below it and the device unit 'drives' a device. The hierarchy can have an arbitrary number of levels to provide the sub-detectors with as many abstraction layers as required.

Partitioning

The hierarchy also offers a high degree of independence between its components and allows, by means of the concept of 'partitioning', for concurrent use. Partitioning is the capability of independently and concurrently controlling and monitoring parts of the system, typically sub-trees of the hierarchical control tree. This functionality is essential during the installation and commissioning phase, where parts of the control system might not yet be available but sub-detectors need to control the installed equipment. The feature is also essential during operation for debugging purposes or sub-detector test or calibration running. During longer shutdown periods, sub-detectors might want

2008 JINST 3 S08002

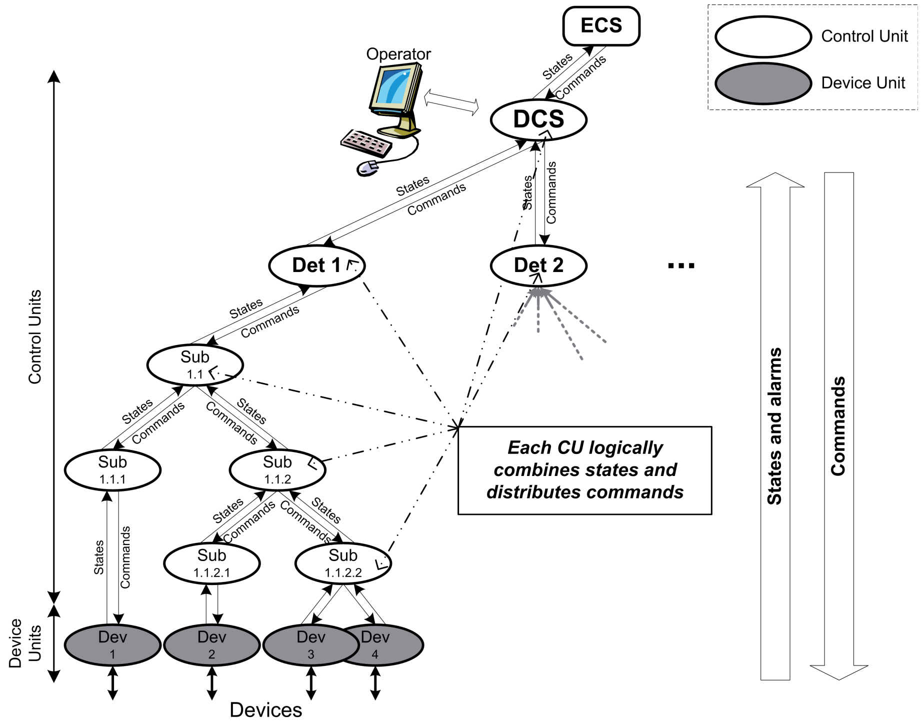

Figure 7.2: DCS hierarchical control arcitecture.

to run their sub-detector control system while other parts of the control system are still switched off. Only the control units in the control tree can become the root node of a partitioned control tree.

Finite-State Machine

The behaviour and functionality of each unit in the control tree is modelled and implemented as a finite-state machine (FSM). The finite-state machine concept is a fundamental component in the control system architecture. It is an intuitive, generic mechanism to model the functionality of a piece of equipment or a sub-system. The object to be modelled is thought of as having a set of stable 'states'. It can transit between these states by executing 'actions' that are triggered either by commands from an operator or another component or by other events such as state changes of other components. Two types of objects can be defined in the FSM concept: abstract objects, represented by a control or logical unit, and physical objects, represented by a device unit in the control tree. This concept allows for distributed and decentralised decision making and actions can be performed autonomously, even when controlled centrally. This will naturally lead to parallelism in automated operations such as error recovery, thus increasing the efficiency of the system.

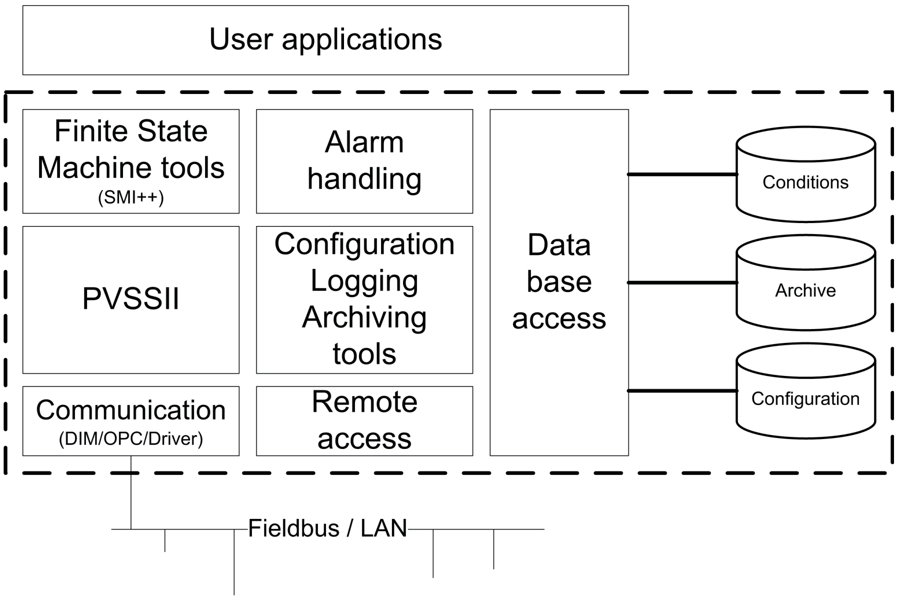

Figure 7.3: Overview of control system components.

7.1.3 System implementation and operation

7.1.3.1 System components

PVSSII

The core software of the control system is a commercial SCADA (Supervisory Controls And Data Acquisition) system: PVSSII. This package was selected after an extensive evaluation performed by the CERN-wide Joint Controls Project (JCOP). PVSSII is used by all four LHC experiments, as well as by several of the services such as gas, safety, etc. It offers many of the basic functionalities needed by the control system.

JCOP and ALICE frameworks

Around PVSSII a framework was built as a joint effort between the four LHC experiments (figure 7.3). This framework provides tools and components for the implementation of all the common tasks that are expected from the control system such as FSM, database access, access control, basic user interfaces, configuration, etc. The JCOP-framework also implemented interfaces to several hardware devices that are commonly used and that hide much of the PVSSII internals from a non-expert end-user. In the same context an ALICE framework was developed to cater for ALICE specific needs.

These tools are used by the sub-detector expert to build their applications. The tools are accompanied by a set of rules and guidelines (naming conventions, user interface look and feel, etc.) to ensure the homogeneity and to simplify the integration of independently developed components.

Operating systems

All PVSSII applications (projects) are running on the Windows operating system on the Worker Nodes (Windows XP) and the Operator Nodes (Windows Server 2003). Specific worker nodes (those interfacing to the Front End Electronics) run the Linux (SLC4) operating system.

Communication and hardware access

Communication with the hardware is restricted to a limited set of communication protocols. Two different protocols cover all the needs. OPC (OLE for Process Control) is a widely accepted and an industry standard to communicate with commercial devices. It implements a client-server mechanism, where the supplier of the device usually delivers the OPC server. PVSSII is a generic OPC client. The second recommended protocol is DIM (Distributed Information Management) that is used to communicate with custom built equipment. DIM is available for many platforms and libraries are available for several computer languages. DIM implements a client-server mechanism over TCP/IP. When a DIM-server is developed for a given piece of equipment, PVSSII can be used to access it through the DIM-client integrated in the framework. In addition PVSSII can also directly communicate with equipment through drivers. PVSSII drivers exist for Modbus devices and some other (mainly industrial) devices.

DIM is also the underlying communication layer for the Data Interchange Protocol (DIP) that is used to exchange data between the control system and LHC as well as with the various services such as gas, electricity, cooling etc.

Finite-State Machine

The hierarchical tree-like control structure, as described earlier, relies heavily on finite-state machines. PVSSII itself does not provide any FSM functionality, but this was added in the framework (SMI++, [263]). There is a very close integration between PVSSII and the FSM tools.

Databases

A large variety of data, needed for the operation of the experiment, has to be stored. The amount and the use of the data vary widely and it is therefore stored in a collection of databases, each optimised for its particular use, all housed in a dedicated Oracle, RAC database server at the experiment site.

PVSSII has its own internal proprietary run-time database which is used to store the values that are read from the devices, information on the configuration of PVSSII itself and any information that is needed for the operation of the PVSSII system. This database is optimised for fast access, as it is an essential part in the operation of the PVSSII system. Data archiving is an integral part of PVSSII and is the mechanism to store the history of any data available in the system that the user decides to archive. PVSSII allows archiving to be done to a file based archive or to a RDB (Oracle). During first commissioning and tuning of the sub-detector a file based system is used, in the final production system data is archived into the database.

The configuration database holds the data needed for the configuration of the whole control system; this includes the configuration of the control system itself, configuration of hardware devices such as power supplies but also chips in the front-end electronics, configuration of processes, etc.

The PVSSII archive described above is an essential tool for system debugging. It contains all information acquired during operation; however retrieval of this data is complicated when attempted from outside of PVSSII. The main reason is that the control system is distributed and accessing of data requires detailed knowledge about its structure. To overcome these limitations the so called Offline Conditions DataBase (OCDB) is used. It mirrors data essential for offline analysis and provides an easy access and retrieval mechanism. The interface between the OCDB and the DCS archive was implemented by the central team as a dedicated PVSSII manager.

7.1.3.2 Applications

The ALICE experiment consists of a fairly large number (~ 130) of sub-systems and controls slices that need to be developed. The aim was to standardise and use common solutions as far as possible. At the device level the sub-detector users were encouraged to use similar types of devices whenever possible, and common specifications were developed for devices to be purchased, such as high and low voltage power supplies, VME crates and VME single board computers. Manufacturers were in this way asked to provide standard control interfaces based on OPC and CERN standard fieldbuses. For the Front-End Electronics (FEE), which is custom made for each sub-detector, a standard software interface for its control and configuration was defined.

Nearly all requirements for High and Low Voltage could be satisfied by three manufacturers (CAEN and Iseg for HV; CAEN and Wiener for LV). This already significantly reduced the development effort.

General purpose monitoring comprises any parameter that is not acquired through any of the other sub-systems. These parameters are typically temperatures measured on the detector. To avoid the dispersion of solutions for this task, a general purpose monitor device was adopted: the ELMB [264]. The ELMB is a general purpose microcontroller (with 64 analogue inputs and 16 digital I/Os) connected via CANbus. An OPC server is available to control the ELMB and access the acquired data. A major feature of the ELMB is that it can work in radiation environments and therefore be used directly on the detector.

Common solutions however not only consist of standard equipment and standard control interfaces, but wherever possible also implement standard logical behaviour of the equipment. A control unit for a class of devices implements a unique FSM state diagram describing the behaviour and commands for that class of devices. This naturally leads to not only uniformity in names of states and commands, but also a 'non-expert' operator is presented with coherent behaviour for similar sub-systems throughout the experiment. The behaviour includes standard operation features such as automatic recovery of anomalies.

Front-End Electronics

The control of Front-End Electronics (FEE) is a complex and delicate task. It involves control of voltage regulators, power switches, error registers, etc., and monitoring of temperatures, voltages, currents, status registers, etc. of the FEE boards. It also involves configuration and initialization of FEE controllers and of all the various custom chips on the detector boards. It requires a very close interaction between the DCS and DAQ, as in the majority of the cases, both systems share the access path to the FEE or use the FEE concurrently. A further challenge is that the architecture and implementation of the FEE is different for each sub-detector and based on custom chips. Many different techniques are deployed to communicate with these chips such as JTAG, Ethernet, GOL, Profibus, CANbus, etc., each requiring a different access strategy. Also the Detector Data Link (DDL) provides functionality for downloading configuration information. In addition, large amounts of data, unusually large compared to other areas of the control system, are involved; especially for configuration of the FEE.

In order to achieve maximum commonality between all different FEE architectures a Front-End Device (FED, [265]) was defined. The FED represents a hardware abstraction layer allowing DCS transparent access to the FEE. It responds to standard commands and performs requested tasks such as loading configuration registers, resetting the chips, etc. If the FEE provides data which needs to be monitored, this is gathered by the FED and made available to the supervising software.

The FED is built as a package of software and hardware with a standardized software interface. A common FED client-server model was adopted for the FED which hides the implementation details to higher software layers. The server communicates with the hardware and publishes data as services. A client can subscribe to services and send commands to the server. Several clients can subscribe to the same server in parallel allowing for distributed monitoring. The DIM protocol was chosen as the underlying communication layer.

Services

The DCS interfaces to various services in order to keep the control system up to date with the experiment operation environment. Some of the services allow active control from DCS, however the majority of interactions are monitoring only. For communication with the services the Data Interchange Protocol (DIP) was defined, which allows a transparent way to exchange information between the systems involved.

Eight sub-detectors in ALICE have a gas system, with their associated control system. The operation and maintenance of the gas system is taken care of by an LHC-wide operations team. However, any information needed by the sub-detectors from the gas system for their operation is made available through DIP. A central application subscribes to the DIP parameters and makes them available in the distributed PVSSII system, for use by the sub-detectors.

Sub-detector cooling systems with their control systems are developed ALICE-wide by Cooling and Ventilation group of the Technical Support Department (TS/CV). These cooling systems (and their controls) are designed, built and installed by the TS/CV team; the control is based on PLC technology. DCS exchanges information and commands through a ‘concentrator PLC’, using the Modbus TCP/IP protocol.

Both gas and cooling systems generate, as backup to the software interface, hardware interlocks allowing sub-detectors to take actions and protect their equipment in case of serious anomalies.

The power distribution racks that deliver power to the individual equipment racks are installed and maintained by Electrical Engineering group (TS/EL), including a control system. This control system, which is PLC based, reads status information (on/off, error, etc.) and provides on/off control for each outlet. The DCS interfaces with these PLCs, via the Modbus TCP/IP protocol.

Each rack is equipped with a monitoring system that allows monitoring of the environment inside the rack and the operational state of the rack. This monitoring system was developed as a common solution for the four LHC experiments and is based on the ELMB. The protection of each rack is ensured by a thermo-switch and a smoke detector; in case of anomaly the power to the rack will be cut.

DCS also exchanges information with the LHC machine, the magnet control system, safety systems and the primary services, and the DIP protocol is used for this. A central service subscribes to the DIP parameters and republishes the information inside the distributed PVSSII system, for use by the sub-detectors.

Detector Safety System and Interlocks

The Detector Safety System (DSS, [266]) is a robust part of the ALICE DCS, designed for high-availability and is based on a redundant PLC system. It is designed to monitor the experiments environment (temperature, presence of cooling, water leaks) and to take automatic protective actions (cut power, close water valves) in case of anomalies.

Hardware interlocks are implemented at several levels. Sub-detectors have implemented various protection mechanisms on their detector equipment; high temperature detected on the electronics and automatically switches off that piece of electronics. Also the DSS is used as an interlock system where independent sensors are available to detect anomalous conditions; the DSS is then programmed to take protective action.

7.1.3.3 Operation

All interaction with the experiment is done through graphical user interfaces (figure 7.4) and this is the only part the end-user will see from the control system. In order to facilitate the operation of the various parts of the experiment, a major effort has been made to achieve the same 'look and feel' for all user interfaces. This is essential for an efficient operation of the experiment, as a small shift crew will operate a large set of different sub-detectors. This crew should therefore be able to rapidly diagnose problems in any of the ALICE sub-detectors; a task that is greatly facilitated by a high level of uniformity across the user interfaces. Therefore a standard ALICE user interface was developed as a framework tool used by all sub-detectors to implement their graphical user interfaces.

As the control system controls the often delicate and unique equipment of the sub-detectors the potential danger of serious and irreversible damage imposes a need for an advanced access control mechanism to regulate interactions of the users with the control system components. The system must be protected against inadvertent mistakes by operators or experts as well as against malicious attacks from outside.

2008 JINST 3 S08002

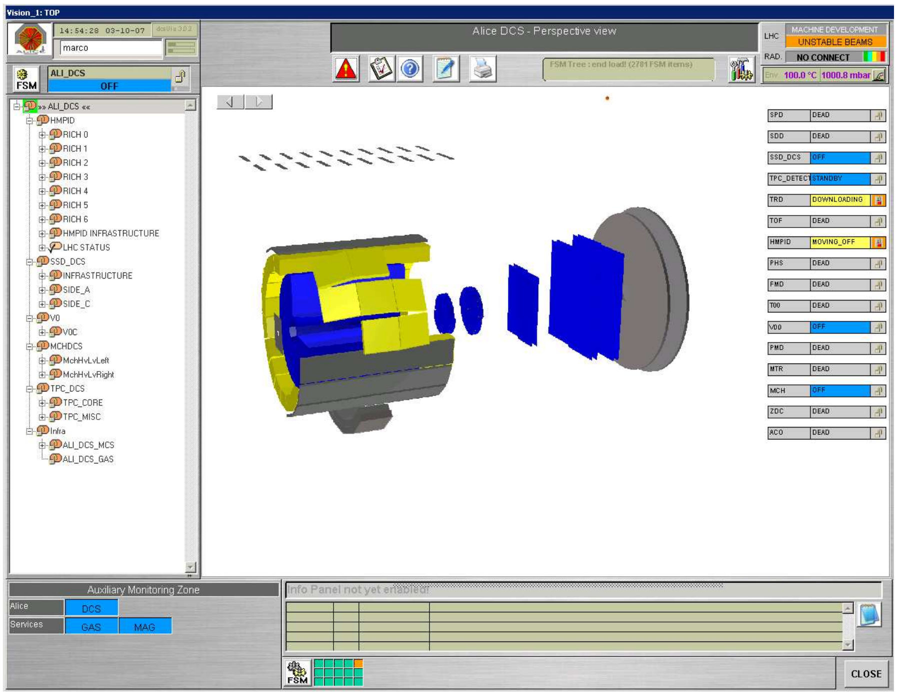

Figure 7.4: ALICE DCS user interface.

7.2 Experiment Control System (ECS)

7.2.1 Requirements

The control of the ALICE experiment is based on several independent 'online systems'. Every 'online system' controls operations of a different type belonging to a different domain of activities: Detector Control System (DCS), Data Acquisition (DAQ), Trigger system (TRG), and High-Level Trigger (HLT). The 'online systems' are independent, may interact with all the particle detectors, and allow partitioning. Partitioning is the capability to concurrently operate groups of ALICE detectors. In the final setup the detectors will mainly work all together to collect physics data. In the commissioning phase, however, detectors are debugged, and tested as independent objects. While this mode, called 'standalone mode', is absolutely vital in the commissioning and testing phase, it will also be required during the data-taking phase to perform calibration procedures on individual detectors. It will therefore remain essential during the whole life cycle of ALICE.

The Experiment Control System (ECS) coordinates the operations controlled by the 'online systems'. It permits independent, concurrent activities on part of the experiment by different operators and coordinates the functions of the 'online systems' for all the detectors and within every partition. The components of the ECS receive status information from the 'online systems' and send commands to them through interfaces based on Finite-State Machines (FSM). The implementation of these interfaces is based on the SMI++ package [263]. The interfaces between the ECS and the 'online systems' contain access control mechanisms that manage the rights granted to the ECS: the 'online systems' can either be under the control of the ECS or be operated as independent systems. In the second case the 'online systems' provide status information to the ECS, but do not receive commands from it.

2008 JINST 3 S08002

7.2.2 System architecture

Partitions and standalone detectors

A partition is a group of particle detectors. From the ECS point of view, a partition is defined by a unique name that makes it different from other partitions and by two lists of detectors: the list of detectors ‘assigned’ to the partition and the list of detectors ‘excluded’ from the partition. The first list, called ‘assigned’ detectors list, contains the names of the ALICE detectors that are members of the partition and can be active within it. This static list represents an upper limit: only the detectors included in the list can be active in the partition, but they are not necessarily active all the time. The lists of detectors assigned to different partitions may overlap: the same detector can appear in different lists. The lists of assigned detectors cannot be empty. The second list, called ‘excluded’ detectors list, contains the names of the ALICE detectors that have been assigned to the partition, but are currently not active in it. This dynamic list is a subset of the assigned detectors list and can be empty. Although a given detector may appear in the assigned detectors list of many partitions, it can only be active in one at any time. The excluded detectors list of a partition contains the names of the detectors that are not active because they are active in another partition, because they are running in standalone mode, or because of an explicit operator request. Explicit operator requests are subject to restrictions: during the data-taking phase the structure of a partition cannot be changed.

Two types of operations can be performed in a partition: those involving all the active detectors, called global operations, and those involving only one active detector, called individual detector operations. The ECS handles the global operations by monitoring the DCS status of all the active detectors. It interacts with the DAQ and HLT processes that steer the DAQ and HLT activities for the whole partition. It sends commands to the Trigger Partition Agent (TPA) that links the partition to the Central Trigger Processor (CTP). When a global operation starts, the ECS inhibits all the individual detector functions. The ECS handles an individual detector operation by monitoring the DCS status of the detector, interacting with the DAQ and HLT processes that steer the DAQ and HLT activities for that particular detector, and sending commands to the Local Trigger Units (LTU) associated to it. When an individual detector task starts, the ECS inhibits the global operations for the partition, but it does not inhibit individual detector tasks on the other detectors. These individual detector operations, such as calibration procedures, can be concurrently performed within the partition.

A standalone detector is a detector operated alone and out of any partition. The tasks performed by a standalone detector are equal to the individual detector operations that are allowed when the detector is active in a partition. The ECS handles these functions watching the DCS status of the detector and interacting with the DAQ and HLT processes that steer the DAQ and HLT activities for that detector and with the LTU associated to it.

The major difference between a standalone detector and a partition with only one single detector is that this last partition is linked to the CTP by a TPA, whereas the standalone detector only interacts with its LTU.

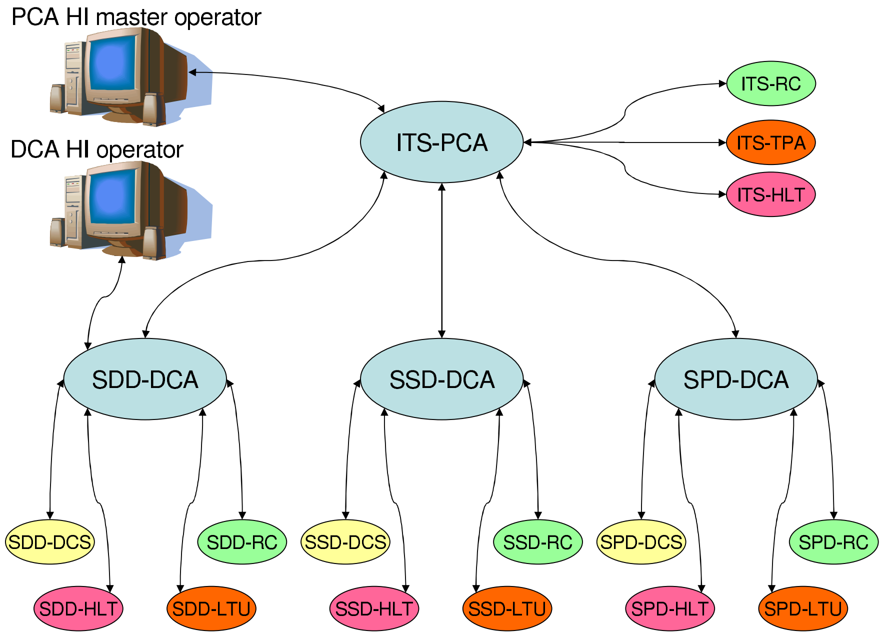

Figure 7.5: The overall architecture of the ALICE ECS applied to the three ITS detectors.

ECS architecture and components

Every detector in standalone mode or assigned to a partition is controlled by a process called a Detector Control Agent (DCA) and every partition is controlled by a process called a Partition Control Agent (PCA). When a detector is in standalone mode, its DCA accepts commands from an operator via a DCA Human Interface (DCAHI). At any time several DCAHIs can coexist for the same DCA, but only one can send active commands: the others can only get status information. When a detector is active in a partition, its DCA accepts commands only from the PCA controlling the partition. Operators can still invoke DCAHIs, but only to get information and not to send active commands. A PCA Human Interface (PCAHI) provides to an operator full control of a partition. At any time, many PCAHIs can be active for the same PCA, but only one has the control of the partition and can be used to send commands. DCAs and PCAs get status information from the 'online systems' and eventually send commands to components of these systems through interfaces based on Finite-State Machines. An example of the ECS architecture with three detectors (SDD, SPD, and SSD), all active in a partition called ITS, is given in figure 7.5. For each detector, there is a DCS object (xxx-DCS), an LTU control object (xxx-LTU), a DAQ RC object (xxx-RC), and a HLT object (xxx-HLT).

The major components of the ECS, described below, are the Detector Control Agent (DCA), the Partition Control Agent (PCA), the Detector Control Agent Human Interface (DCAHI), and the Partition Control Agent Human Interface (PCAHI).

Detector Control Agent (DCA)

There is one DCA for every detector running in standalone mode or assigned to a partition. The main tasks performed by this process are the following:

- It handles standalone data-acquisition runs for the detector working alone.
- It handles electronics setup procedures. This function and its implementation are detector dependent.
- It handles calibration and test procedures. These procedures are by definition detector dependent as well as their implementation.

The DCA accepts commands from one master operator at a time: either a PCA or a DCAHI.

Partition Control Agent (PCA)

There is one PCA per partition. The main tasks performed by this process are the following:

- It handles data-acquisition runs using all the detectors active in the partition.
- It delegates individual detector functions to the DCAs controlling the detectors active in the partition.
- It handles the structure of the partition allowing the inclusion/exclusion of detectors whenever these tasks are compatible with the data-taking runs going on for individual detectors or for the whole partition.

The PCA accepts commands from one PCAHI at a time.

Human Interfaces

An operator can run a detector in standalone mode with a DCAHI having the mastership of a DCA. One can send commands to the DCA, change the rights granted to the DCA, and send commands directly to objects in the DCS, DAQ, HLT, and TRG 'online systems'. Without the mastership of the DCA, the DCAHI can only get information and cannot issue active commands.

An operator can run a partition with a PCA Human Interface having the mastership of a PCA. One can send commands to start global and individual detector tasks, can change the rights granted to the PCA, can change the structure of the partition excluding or including detectors, and can send commands directly to objects in the DCS, DAQ, HLT, and TRG 'online systems'. Without the mastership of the PCA, the PCAHI can only get information and cannot issue active commands.

7.2.3 Interfaces to the online systems

The main components of the ECS receive status information from the 'online systems' and send commands to them through interfaces based on Finite-State Machines. The interfaces between the ECS and the 'online systems' contain access control mechanisms that manage the rights granted to the ECS. The 'online systems' can either be under the control of the ECS or be operated as independent systems where the 'online systems' provide status information to the ECS but do not receive commands from it.

2008 JINST 3 S08002

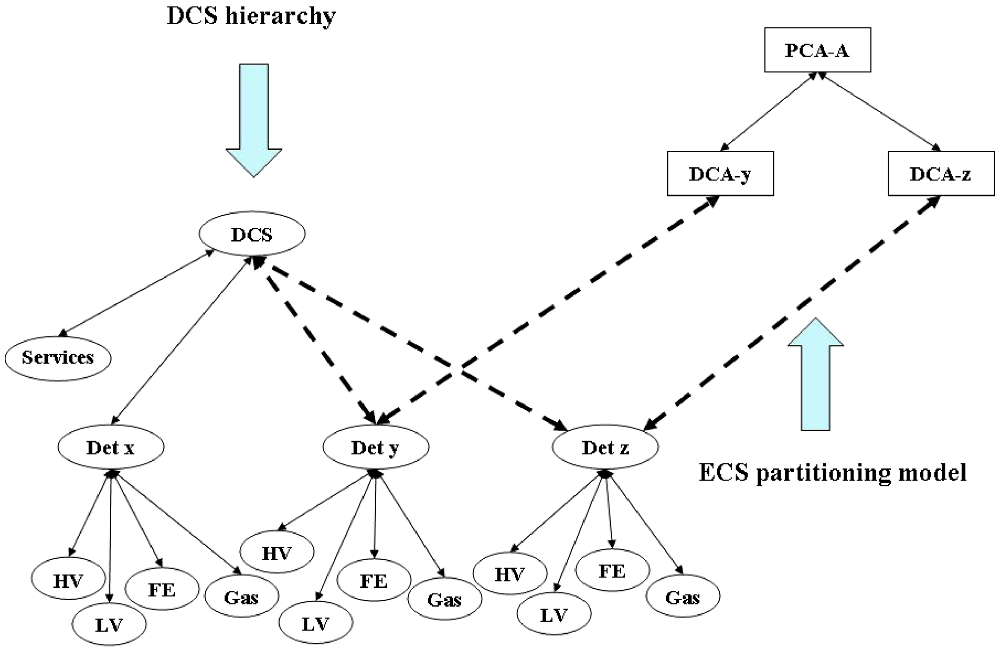

Figure 7.6: The interface of the ECS with the DCS.

ECS/DCS interface

The interface between the ECS and the DCS consists of one object per detector: the roots of the sub-trees described above and representing the detectors within the DCS. These objects can provide status information to the DCS and, at the same time, to the ECS. Every object, depending on the rights granted to the ECS and to the DCS, accepts commands either from the DCS or from the ECS but not from the two systems at the same time.

Figure 7.6 shows an example where two detectors, named 'y' and 'z' are active in an ECS partition named 'A'. The figure shows the double role of the SMI++ objects that provide status information for the two detectors both to the DCS and to the ECS. The figure does not specify if these objects are under the DCS or the ECS control.

ECS/TRG interface

Figure 7.7 shows the ECS/TRG interface. A SMI++ domain named 'TRIGGER' contains the objects describing the basic trigger components: the LTUs associated to the detectors and the CTP. These SMI++ objects are associated to processes, called proxies, which actually drive the LTUs and the CTP. When a global operation is performed, all the detectors active in a partition produce raw data: the generation of raw data by the detectors is performed under the control of their associated LTUs. These LTUs are synchronized by the CTP. There is only one CTP, but many partitions can be operated at the same time and all of them need access to the CTP. The Trigger Partition Agents (TPAs) associated to the different partitions handle the access conflicts. There is one TPA per partition. The TPA interacts with CTP and LTUs.

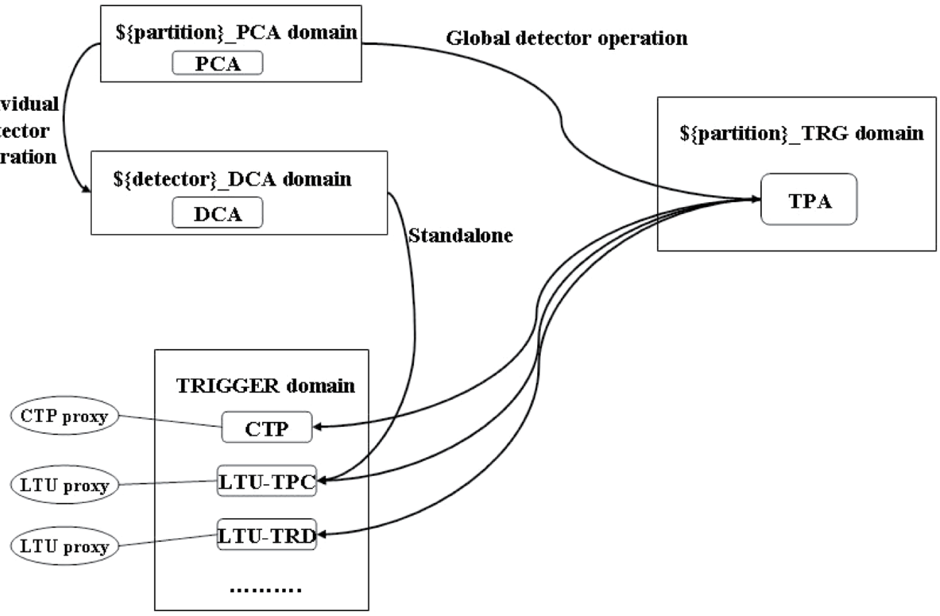

Figure 7.7: The interface of the ECS with the TRG.

When a detector is operated in standalone mode, the DCA controlling it interacts directly with the LTU associated to the detector and the CTP is ignored. When a detector is active in a partition and an individual detector operation is executed on it, the PCA delegates the task to the DCA controlling that detector. The DCA again interacts with the LTU associated with the detector. The CTP is ignored. When a global operation is performed in a partition, the PCA controlling the partition interacts with the TPA that in turn interacts with CTP and LTUs. The PCA has no direct interaction with the CTP and the LTUs.

ECS/DAQ interface

The interface between the ECS and the DAQ is made of SMI++ objects representing RC processes. These are: (a) a RC process per detector, which steers the data acquisition for a given detector and for that detector only. (b) a RC process per partition that steers the data acquisition for the whole partition with data produced by all the active detectors.

ECS/HLT interface

The interface between the ECS and the HLT is made of SMI++ objects representing the state of a part of the HLT as used by a detector or a partition.

7.3 Online Detector Calibration

The condition data including parameters such as detector response calibration, bad channel maps, pedestal values and so on, are evaluated on line from data collected during normal data taking or during special runs. So-called Detector Algorithms (DA) running on DAQ LDCs, GDCs or dedicated Data Monitoring Machines subscribe to the raw data stream and produce the input for the subsequent computing of the desired conditions data. Furthermore, for certain detectors DAs run in DCS and HLT. The DA output is stored on File Exchange Servers (FXS) in the system running the DA, i.e. DAQ, DCS or HLT. Upon completion of a run, the Experiment Control System end-of-run signal triggers a special program called SHUTTLE. This program in turn activates one preprocessor per subdetector which collects the data from the FXSs as well as data from the DCS Archive DB which contains constantly monitored parameters by PVSS, e.g. high voltages and temperatures. From these inputs the preprocessor calculates the conditions, transforms them in "calibration objects" and saves them as ROOT files into a special portion of the AliEn catalogue called Offline Condition DataBase (OCDB). The reconstruction program retrieves the calibration files by querying the OCDB portion of the AliEn catalog.