THE ALICE EXPERIMENT CONTROL SYSTEM

F. Carena[1], W. Carena[1], S. Chapeland[1], R. Divià[1], J-C. Marin[1], K. Schossmaier[1], C. Soós[1], P. Vande Vyvre[1], A. Vascotto[1]
[1]*CERN, Geneva, Switzerland*

ABSTRACT

ALICE (A Large Ion Collider Experiment) [1] is the heavy-ion experiment being prepared for the Large Hadron Collider (LHC) at CERN. Running the experiment implies performing a set of activities on several particle detectors. In ALICE these activities are grouped into four domains: Detector Control System (DCS), Data Acquisition (DAQ), Trigger (TRG) and High Level Trigger (HLT).

The Experiment Control System (ECS) is the top control level of the ALICE experiment.

In October 2004 the ECS has been used to control the combined beam tests of the ALICE Inner Tracking System (ITS) with an experimental setup made of three particle detectors. All the different combinations of detectors were successfully tested: three detectors in standalone mode, any combination of two detectors running together while the third one was in standalone mode, and the three detectors together.

INTRODUCTION

The control of the ALICE experiment [2] is based on several independent 'online systems'. Every 'online system' controls operations of a different type and 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 present phase, however, detectors are prototyped, 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 [3]. 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.

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 assigned detectors lists for different partitions may overlap: the same detector can appear in different assigned detectors lists. Assigned detectors lists 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, at any time it can be active at most in one. 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, interacting with the Run Control (RC) process that steers the data acquisition for the whole partition, and sending 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 RC process that steers the data acquisition 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 RC process that steers the data acquisition 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.

ECS ARCHITECTURE AND COMPONENTS

Every detector in standalone mode or assigned to a partition is controlled by a process called Detector Control Agent (DCA) and every partition is controlled by a process called 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 at can be used so 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 1. For each detector, there is a DCS object (xxx-DCS), an LTU control object (xxx-LTU), and a DAQ RC object (xxx-RC).

The major components of the ECS 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. This function requires the coordination and the synchronization of the detector hardware controlled by the DCS, of the RC process steering the data acquisition for the given detector, and of the LTU associated to this detector. This function is implemented in the same way for all the detectors.
- 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.

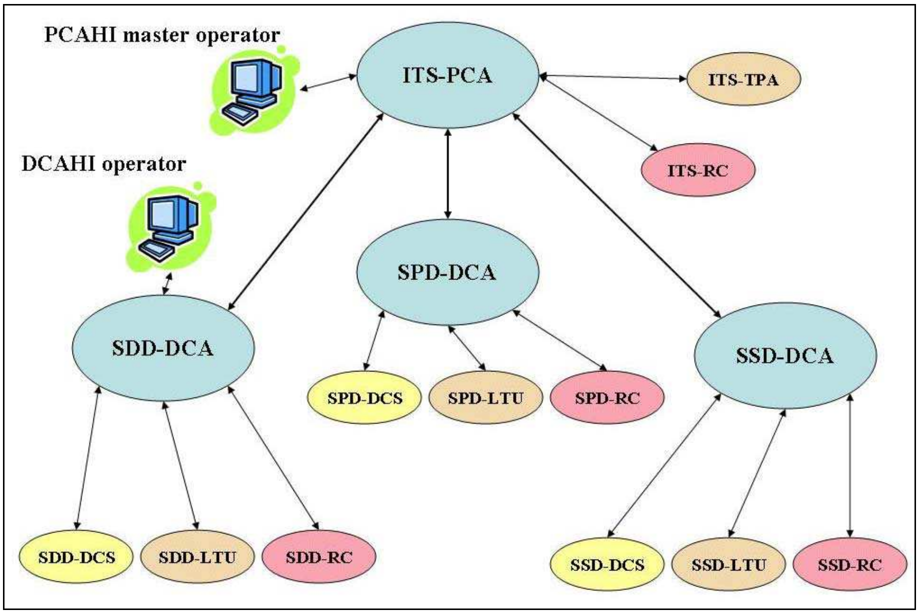

Figure 1: ECS architecture.

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. This function requires the coordination of the DCS of all the active detectors, of the RC process steering the data acquisition for the whole partition, and of the TPA associated to the partition. This function is implemented in the same way for all the partitions.
- 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. He/she can send commands to the DCA, change the rights granted to the DCA, and send commands directly to objects in the DCS, DAQ, 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. He/she 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, and TRG 'online systems'. Without the mastership of the PCA, the PCAHI can only get information and cannot issue active commands.

INTERFACES BETWEEN ECS AND ‘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.

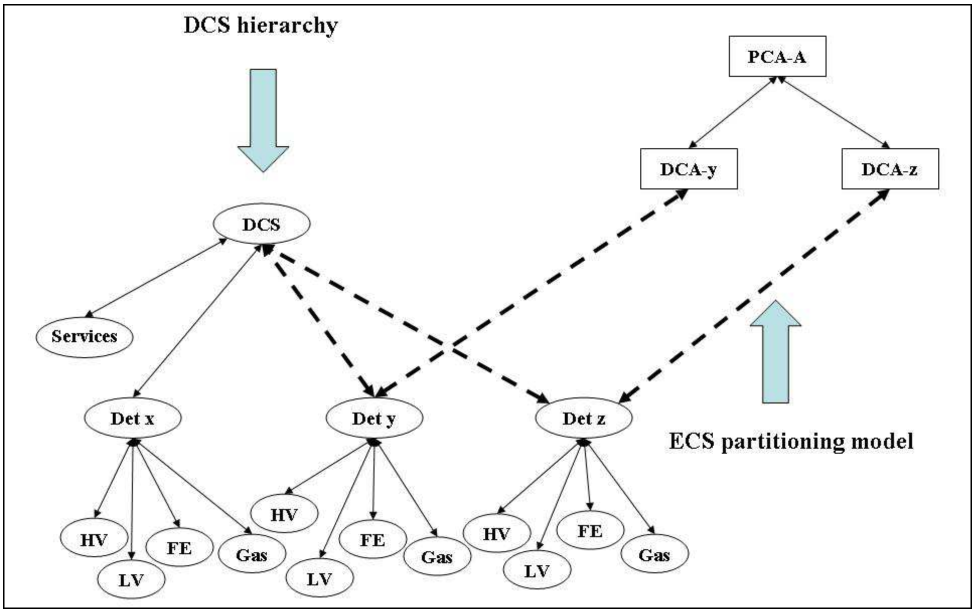

Figure 2: ECS/DCS interface

ECS/DCS interface

The DCS describes the ALICE experiment as a hierarchy of particle detectors and of infrastructure services. Its model of ALICE is based on FSMs and is implemented as a tree structured set of SMI++ domains and objects. Within this tree every detector is represented by a different sub-tree. The status of the roots of these sub-trees is the status of the different detectors as seen from the DCS point of view.

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 2 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 3 shows the ECS/TRG interface. An 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.

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.

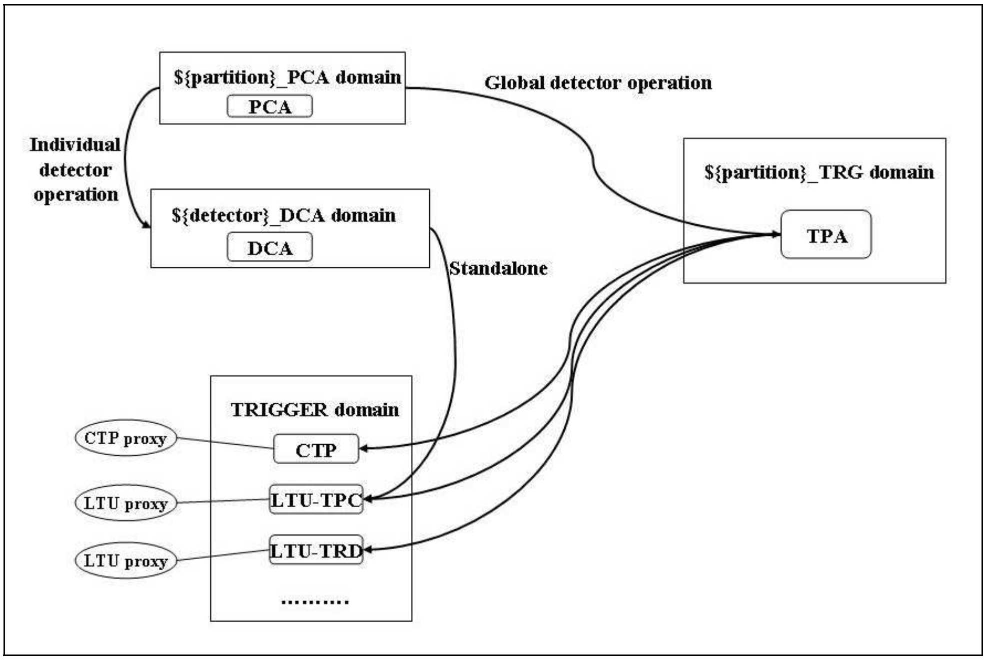

Figure 3: ECS/TRG interface

ECS/DAQ interface

The interface between the ECS and the DAQ is made of SMI++ objects representing RC processes. These are:

- An RC process per detector. Every RC process steers the data acquisition for a given detector and for that detector only.
- An 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 ECS gets status information about the HLT through a single SMI++ object representing the HLT 'online system' as a whole.

ALICE INNER TRACKING SYSTEM TEST

In October 2004, the ECS has been used to control the combined beam test of the ALICE Inner Tracking System (ITS) consisting of three particle detectors: the Silicon Drift Detector (SDD), the Silicon Pixel Detector (SPD), and the Silicon Strip Detector (SSD). The overall experimental hardware and software setup controlled by the ECS was made of the following elements:

- A trigger system based on three LTUs associated to the three detectors and a special version of the TPA. The special version of the TPA was created in view of the ITS test to hide the absence of a CTP (the CTP will only be available at the beginning of 2006) and to allow the emulation of the CTP basic functions.
- A complete DAQ system allowing multiple, parallel data acquisitions for standalone detectors or partitions. Part of this system were the Detector Data Links (DDLs) connecting the detectors to the DAQ PCs, the Read Out Receiving Cards (RORCs) allowing the reception of data through the DDLs, the DAQ PCs running the DAQ software, and the DAQ software itself. DATE V5 [4] was used as DAQ software
- Dummy versions of the DCS for the three detectors: the DCS for the three detectors was not sufficiently developed to be compatible with the ECS/DCS interface.

From the ECS point of view, this experimental setup was made of three DAQ RC processes handling the data acquisition for the three detectors when working alone, a fourth RC process handling the data acquisition for the whole ITS partition, three LTUs, a TPA, and three dummy DCS objects. Three DCAs and one PCA were required to control the overall setup. Figure 1 shows the ECS architecture and components applied to the ITS test.

All the possible modes of operation were successfully tested: three detectors in standalone mode, any combination of two detectors running together and the third one in standalone mode, and the three detectors running together. Switching between modes of operation was fast (a matter of seconds), performed by ECS operators using PCAHIs and DCAHIs, and totally transparent for the DAQ and the TRG 'online systems'.

CONCLUSION

The ALICE ECS has been intensively developed in the last 18 months. The architecture and the interfaces defined in the TDR [2] have been implemented and successfully tested during a combined test beam of the 3 ITS detectors. Some developments are still needed before the experiment start-up: Several detectors have not yet implement a DCS based on FSM, therefore the DCS states of these detectors are not included yet in the ECS. Some detectors have not yet developed their calibration procedures. Moreover some information on the configuration of the 'online systems', such as the definition of the Trigger classes, is not available yet and therefore the ECS uses some temporary definition.

While the above issues will be included as soon as available, the ECS architecture has been tested during beam tests, and proved to be solid and flexible enough to include all the future extensions.

REFERENCES

[1] ALICE Collaboration, Technical Proposal, CERN/LHCC/95-71

[2] ALICE Technical Design Report on Trigger, Data Acquisition, High-Level Trigger and Control System, CERN-LHCC-2003-062.

[3] B.Franek and C.Gaspar, SMI++, An object oriented framework for designing distributed control systems, IEEE Trans. Nucl. Sci. 45 (1998) 1946-1950.

[4] ALICE DAQ and ECS User's Guide, Internal Note ALICE-INT-2005-015.