FINITE STATE MACHINES FOR INTEGRATION AND CONTROL IN ALICE

Giacinto De Cataldo[#(1)] André Augustinus, Marco Boccioli, Peter Chochula, Lennart Stig Jirdén
CERN, Geneva, Switzerland; [(1)] CERN and INFN Bari, Italy.

Abstract

The integration of eighteen sub-detectors in a coherent detector control system in ALICE is one of the major challenging tasks for the full detector operation.

A distributed PVSS SCADA system complemented with the tool SMI++ for modelling the experiment behaviour by finite state machines has been used. A schema of a control hierarchy with interconnecting nodes between each sub-detector and the ALICE DCS is being successfully implemented. The standardization of the sub-detector state diagram provides the user with a common set of commands and states to operate in the same way the experiment and a sub-detector as well. Finally the standard ALICE user interface concentrating both the FSM hierarchy control and the PVSS monitoring is described.

INTRODUCTION

From the control point of view a physics experiment can be seen as a vast hierarchy of systems and subsystems with an experiment control node at the top and single atomic control channels at the bottom. In the case of the ALICE experiment [1] at CERN the many systems and subsystems are being built by many engineers and physicists in different institutes around the world. The integration of the various parts to form a homogeneous system enabling coherent automatic control can therefore be seen as a major challenge. A distributed PVSS SCADA system [2] complemented with a set of tools, the Framework (FW) [3], developed at CERN in the context of the Joint COntrol Project (JCOP) [4], have been then provided to the DCS developer to facilitate the control design and implementation.

This paper focuses on the ALICE FSM hierarchy, built with a FW tool the State Manager Interface (SMI++) [5], and the Alice graphic user interface [6] integrating both the FSM controls hierarchy and the PVSS monitoring feature.

THE ALICE FSM HIERARCHY

A state machine is the oldest known formal model for sequential behavior i.e. behavior that cannot be defined by the knowledge of inputs only, but depends on the history of the inputs.

An introduction to the Finite State Machine (FSM) modelling the detector behaviour can be found in [5]. Here for the basic components provided in the SMI++ as Device Units (DU), Logical Units (LU) and Control Units (CU), used in the ALICE FSM, only a brief description is provided.

The DU is the interface with the hardware and its state mirrors the hardware condition. It behaves according to a state diagram but doesn't contain a control program able to take complex decision.

The LU is an object which can integrate several DU; it can be enabled or disabled in the hierarchy and cannot run in stand-alone mode. It is useful at the bottom levels of a FSM tree for grouping the DUs. It behaves according to a state diagram and each state is calculated on the base of the children's states. A LU can integrate other LUs as well.

The CU is a more complex object which can control one or many LUs or DUs. It can be included, excluded from the hierarchy and operated independently. The CU contains a control program derived from its state diagram and is able to take complex decisions. A CU can integrate other CUs as well.

In fig.1 the main software components of the ALICE FSM structure are shown. Eighteen sub-detectors are integrated with the Infrastructure and the LHC.

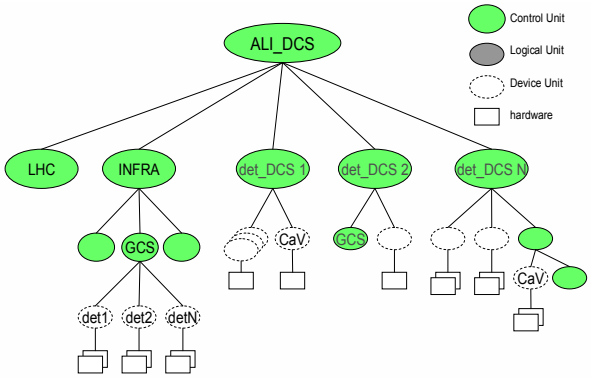

Figure 1: The ALICE FSM hierarchy is shown. In the top node, the logical state from sub-detectors, infrastructure (gas , power, cooling, radiation monitoring…) and LHC (machine state…) are collected while commands from there are send down in the hierarchy.

The FSM structure provides a powerful tool to integrate different sub-detector control systems in a coherent structure. A take/release mechanism allows the partition of the controls hierarchy enabling more than one operator to intervene.

Each sub-detector control system can be excluded from the main hierarchy and taken by the sub-detector expert for debugging purposes.

The FSM Standard state diagram

The sub-detector nodes are based on a standard state diagram shown in fig. 2. Details can be found in [7].

The adoption of this diagram by all the sub-detectors is twofold justified:

[#]giacinto.de.cataldo@cern.ch

- the same set of commands and states are used for all the sub-detectors therefore the experiment shifter can execute standard procedures on any sub-detector (different sequences can therefore be hidden by dummy commands and states);
- simplified Boolean equations can be used to define the global experiment state and no tedious combinatorial Boolean equations have to be written.

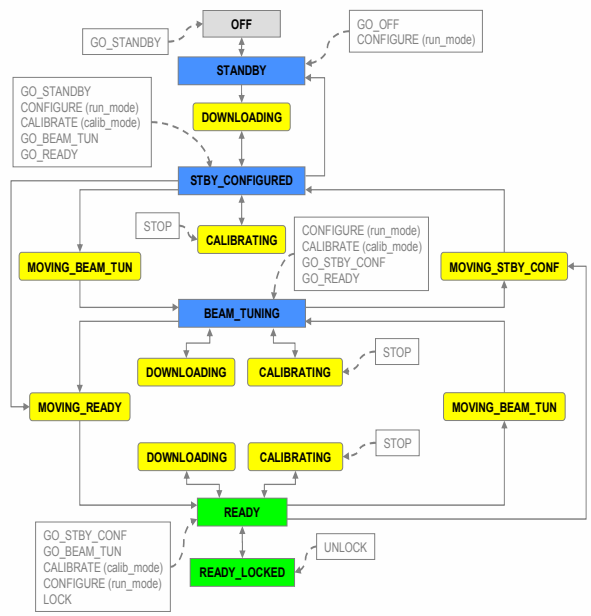

Figure 2: The standard state diagram for the sub-detectors in ALICE. The BEAM_TUNING state set a reduced HV voltage during the beam injection and adjusting phase.

The DCS top node is designed to allow for the sub-detector calibration, configuration (p-p, heavy ions, cosmic rays..) and to face the beam injection and adjusting phase in a safe 'parking' condition (BEAM_TUNING) preventing damages in case of beam loss.

Alarm handling via FSM

Although not shown in the diagram, the ALICE and sub-detector FSM's are complemented with other states dedicated to handle alarm conditions. According to the JCOP prescription, the three increasing alarm severities levels, in ALICE three states named WARNING, ERROR and SYS_FAULT are defined respectively with the colours: yellow, orange and red.

Generally speaking WARNING is not harmful for the device or sub-detector and do not require the operator intervention. ERROR requires the operator intervention but the system can continue to work. SYS_FAULT requires the urgent intervention of the operator and the device is in fault. Except the WARNING the other two states are propagated in the hierarchy up to the top node and managed accordingly by the ALICE DCS and/or the Experiment control system.

One of the FSM benefits is the possibility to program automatic recovering procedures at least for the warning condition, reducing thus whenever possible the operator intervention. In fact small operation crews are supposed to do shifts on the experiment surveillance.

In addition, due to the presence of software interlocks based on the FSM, the network operation is constantly checked and if interruptions are detected then the NO_CONTROL state is set in the relevant FSM. This state is immediately recovered as the connection is established. In case of NO_CONTROL or computer rebooting a dedicated FSM entry state named MIXED allows the FSM to detect the hardware condition and set the corresponding state re-synchronizing software and hardware.

Finally the ALICE FSM is programmed to react with the INTERLOCK state on the hardware interlock conditions generated in systems as gas, cooling, Detector Safety System (DSS) etc..

The INTERLOCK_WENT state informs the DCS operator that the harmful condition is went away and the Acknowledge command can be issued. It will result in the FSM reset passing trough the MIXED state and exiting to set the new one corresponding to the present hardware condition.

The ALICE standard User Interface

The design and implementation of a standard ALICE UI provides the framework where all the different functionalities of the DCS operation are concentrated. It is a flexible tool which can be adapted to the sub-detectors and to the entire experiment as well.

The ALICE UI is shown in fig. 3.

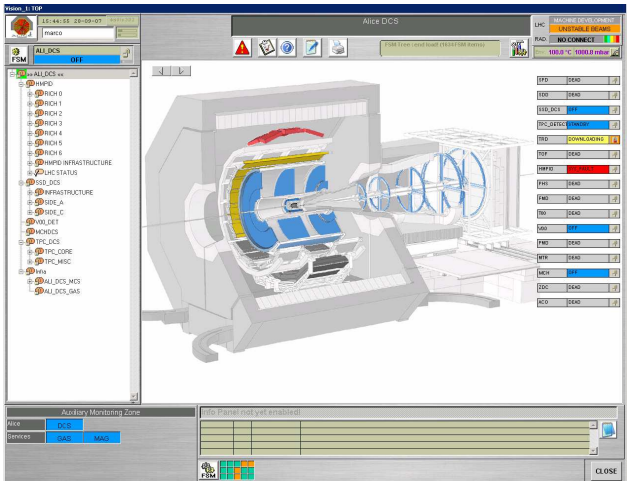

Figure 3: The ALICE standard UI with one of the possible experiment views is shown. Six out of eighteen sub-detectors FSM are already integrated below the ALI_DCS node.

Six sub-detectors are integrated in the ALI_DCS node. An artificially produced SYS_FAULT condition in the HMPID sub-detector is then promptly displayed in the UI via the red colour. The HMPID is located in the magnet in the upper part of the detector. The corresponding state is

shown in the relevant FSM state monitoring in the right side of the UI. In the left top corner of the UI the access control panel can be opened; it allows the user to log in to be granted privileges (observer, operator, expert) that will enables only the corresponding control operations.

On the left side is located the FSM tree browser while on the right part is located the monitoring zone. Here sub-detector specific panels are displayed according to the FSM node selected in the browser. This node is highlighted by a surrounding coloured square.

Provided the relevant privileges are granted, the user can open a FSM control panel (fig. 4) via the button located in the left top corner. From there commands to the FSM hierarchy can be issued and the hierarchy partitioning via the take/release mechanism can be done as well (open/closed coloured locks).

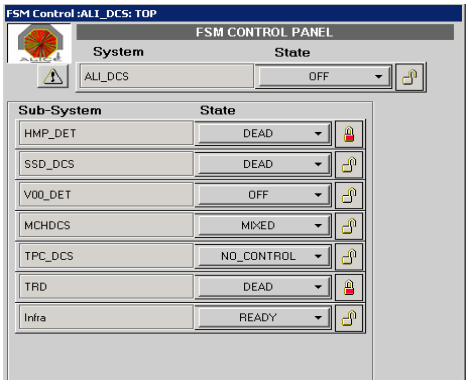

Figure 4: The FSM control panel from where commands can be issued provided the adequate privileges are granted to the operator. The open/closed coloured lock gives information about the take/release condition.

In the bottom left corner is located an Auxiliary Monitoring Zone. There blinking colour coded buttons inform the operator of possible anomalies on important FSM nodes. Pushing the button the corresponding monitor panel is opened.

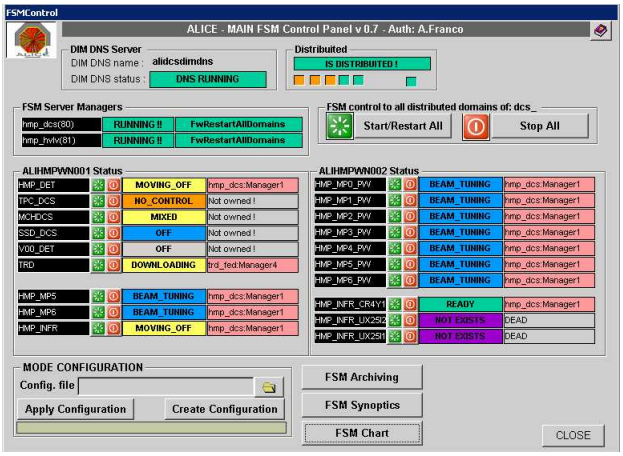

Figure 5: The main FSM control panel. From here start/stop of all or a single FSM process can be done. Only expert operators are allowed to open this panel.

In the centre bottom of the UI is located another FSM button allowing only the expert to access and control the full set of FSM processes (fig. 5) via the relevant start/stop buttons. A set of monitoring squares provides information about the FSM processes in the distributed PVSS project running on tens of different computers.

Finally in the top right part of the UI are located info of general interest as from LHC and environmental pressure and temperature, while in the centre, services of general interest as the PVSS alarm panel, the help page and printer icons are located.

Acknowledgments

We would like to thank Mr. Antonio Franco from INFN Bari, IT for his invaluable work in designing the UI component for the ALICE control coordination team.

CONCLUSION

The ALICE DCS FSM is ready and integrates already six sub-detectors out eighteen plus the infrastructure. The mechanism of taking/release allows the control and the hierarchy partitioning. In this way more operators can simultaneously control different parts of the experiment and/or the single sub-detector.

The adoption of the commercial PVSS SCADA system and the software tool SMI++ developed at CERN have provided an effective development environment to model the experiment and the single sub-detector behaviour with Finite State Machines. The definition of a standard state diagram on top of each sub-detector node provide the ALICE operator with a common set of commands and states to operate in the same way the experiment as well as a single sub-detector.

Finally, the implementation of the standard ALICE User Interface has provided a useful and flexible framework where all the utilities for the FSM operation and PVSS monitoring are concentrated.

REFERENCES

[1] ALICE Collaboration, Technical Design Report of the Trigger, Data Acquisition, High Level Trigger and Control System, CERN/LHCC/2003-062.

[2] http://itcobe.web.cern.ch/itcobe/Services/Pvss/welcome.html

[3] http://itcobe.web.cern.ch/itcobe/Projects/Framework/welcome.html.

[4] http://itco.web.cern.ch/itco/Projects-services/JCOP/welcome.html

[5] http://www.cern.ch/lhcb-online/ecs/fw/FwFsm.html

[6] http://alicedcs.web.cern.ch/AliceDCS/Software/Downloads/AliceDcsUi_v3.0.doc.

[7] http://alicedcs.web.cern.ch/AliceDCS/IntegrationDCS/examples/Alice_DCS_FSM_integration_guidelines_0.4.doc.