AUTOMATING THE CONFIGURATION OF THE CONTROL SYSTEMS OF THE LHC EXPERIMENTS

F. Calheiros, P. Golonka, F. Varela, CERN, Geneva, Switzerland.

Abstract

The supervisory layer of the Large Hadron Collider (LHC) experiments is based on the Prozeßvisualisierungs- und Steuerungsystem (PVSS) [1] and the Joint COntrols Project (JCOP) Framework (FW) [2]. This controls framework includes a Finite State Machine (FSM) toolkit, which allows to operate the control systems according to a well-defined set of states and commands. During the FSM transitions of the detectors, it is required to re-configure parts of the control systems. All configuration parameters of the devices integrated into the control system are stored in the so-called configuration database. In this paper the JCOP FW FSM-Configuration database tool is presented. This tool ensures the availability of all required configuration data, for a given type of run of the experiment, in the PVSS sub-detector control applications. The chosen implementation strategy is discussed in the paper. The approach enables the standalone operation of different partitions of the detectors simultaneously while ensuring independent data handling.

INTRODUCTION

The different modes of operation of the LHC experiments impose strong requirements on the flexibility and re-configurability of their control systems. The control systems must allow for partitioning in order to exclude, from a given run, those parts of the detector that are not ready for data-taking, or to allow for the stand-alone operation of different sub-detectors as required for calibration. Each of these control systems comprises a large variety of hardware devices and of inter-cooperating software applications, which are distributed over ~150 computers. The configuration of these devices and applications from a configuration database must be synchronized in order to allow for the coherent operation of the experiment for a given run type. Due to the size of the systems and the large number of people involved in their development, it is of crucial importance to standardize the interface to the configuration database for all LHC experiments. The FSM-Configuration DB tool is a generic utility developed centrally at CERN that automates the handling of configuration data required by the controls systems at run time.

THE FW FSM

Among the many tools provided by the JCOP FW, the FSM toolkit [3] is of special relevance for the operation of the experiments. The FSM package allows to model in PVSS the different parts of the control systems by means of decision units characterized by a set of well defined states and possible transitions between them. The FSM units are arranged in a tree-like structure that represents the hierarchical organization of the experiments' control systems. Typically, the leaf units of an experiment FSM tree, called *device units*, are directly connected to the detector equipment. The states of these child units are used by the FSM engine to resolve the states of the parent nodes, called *control-units*. Hence this permits to summarize the overall operational state of the detector control system from the states of its multiple integrating parts. Commands are propagated downwards in the FSM hierarchy and their execution triggers the state transitions of the FSM nodes and therefore, it sequences the operation of the control system. An important feature of the control units is that they allow to partition out a sub-tree of the FSM hierarchy for stand-alone operation.

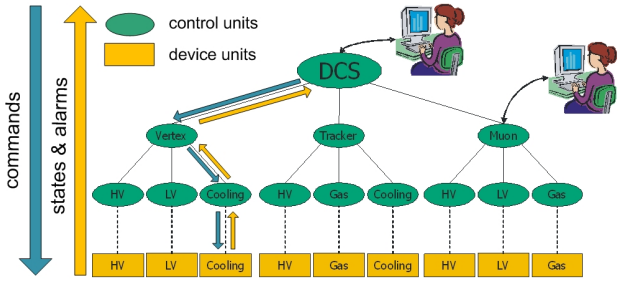

Figure 1: FSM control system architecture.

THE CONFIGURATION DATABASE

In the span of 20 years of lifetime of the LHC experiments, the configuration parameters of the detector control systems will vary as a result of changes in the mode the detectors are operated, sensor re-calibrations due to equipment ageing, etc. To ensure consistent management and safe permanent storage of these parameters, an approach based on a configuration database is envisaged.

In terms of PVSS applications, the following categories of configuration data may be distinguished:

- *System Configuration*, which describes the overall layout of the distributed control system, hierarchy of computers, hardware configuration, network connectivity, software versions, etc.
- *Static Configuration*, which covers these aspects of the PVSS data-points that change rarely such as hardware addresses, value smoothing or conversion formulas.
- *Dynamic Configuration*, which contains the configuration data directly related to runtime control and which are subject to frequent changes during the experiment operation like set-points and alert-threshold configuration.

To address the common needs of the LHC experiments for management of configurations of PVSS applications

using a database, the Configuration Database Tool [4] was developed as a component of the JCOP FW. The tool, based on the widely used Oracle® relational database technology, manages static and dynamic configurations. The management of system configuration is explained in [5]. In the context of this article, only dynamic configuration data are of relevance.

Dynamic configuration data are grouped into so called *recipes*. A recipe is a collection of settings (values or/and alert parameters) corresponding to a number of data-points in PVSS, that is referred to as a whole, and identified by a unique name. Applying a recipe to a control system may be seen as executing a complex command that involves setting a large number of related parameters and reconfiguring alert thresholds. Evolving versions of recipes may be saved to the configuration database, where the changes are tracked as subsequent versions of dynamic configuration data. The Configuration DB Tool allows to pre-load, on request, the recipe data from the database into a local storage in the PVSS application, called *recipe caches*, to optimize the runtime performance. Once the recipe data are loaded into the PVSS application, the caches may be accessed directly regardless of the state of the connection to the database.

THE FSM-CONFIGURATION DB TOOL

The FSM-Configuration DB Tool automates the handling of configuration data required at run-time by the detector control systems. The tool bridges the FSM of the control systems and the configuration database together to ensure the availability of all configuration data for a given type of run, during the FSM transitions.

The FSM-Configuration DB Tool integrates in the FSM tree specific device units called, in the terminology of the tool, *configurators*, which act as links between the FSM and the Configuration DB, as shown in Fig. 2. The configurators make use of the functionality of the Configuration DB Tool to access the database and pre-load the configuration data into the recipe caches. Following a FSM command, these caches can be accessed by the configurators to apply the recipes to the devices in the hierarchy tree.

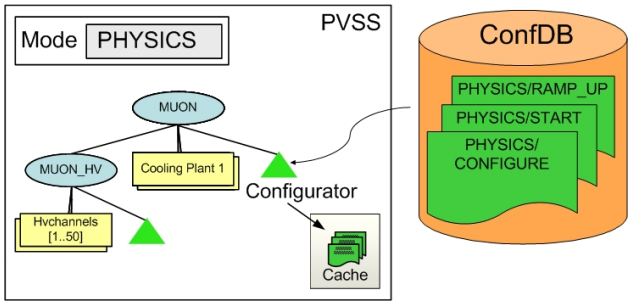

Figure 2: Configurator loading the recipes.

In order to cope with the eventual partitioning of the tree, the tool adds one configurator per FSM control domain to manage the configuration of all devices in the sub-tree. This guarantees the availability of the configuration data during stand-alone operation of a sub-system or throughout the parallel operation of multiple parts of the detector according to different run modes.

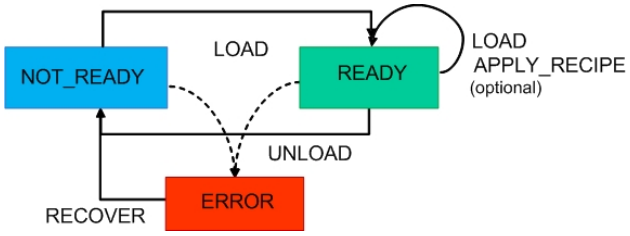

Figure 3: Configurator state diagram.

Fig. 3 shows the state diagram of the configurator. Upon start up of the FSM hierarchy, the configurator establishes connection with the configuration database and determines the list of devices to be handled in its sub-tree and their FSM transitions. If these operations are successful, the state of the configurator is set to NOT_READY. In this state, the operator may define the type of data-taking run to be performed, e.g. PHYSICS, COSMICS. Following the reception of the LOAD command, the configurator accesses the database and loads all recipes required for the given run type and stores them in recipe caches. At this stage, the FSM-Configuration DB Tool assumes a relation, based on a naming convention, which links the recipes and the FSM commands for a run mode, e.g. PHYSICS/RAMP_UP. The FSM-Configuration DB Tool permits to handle multiple recipes associated with a single FSM command. In response to the users' requests, the functionality of the tool was extended to pre-load recipes not linked to FSM commands that may, however, be required at run time. These so-called *user recipes* do not follow a naming convention. The list of user recipes to be handled by each configurator in the FSM hierarchy can be modified by means of graphical interfaces.

Consequently to the successful caching of the complete set of recipes related to a given run mode, the state of the configurator is set to READY. It is important to mention that although a recipe in the database may contain data corresponding to a larger set of devices, each configurator only handles the information strictly required by the devices in its sub-tree. In order to minimize the time required for the reconfiguration of the detector for a different type of run, the configurator may pre-load configuration sets for other run modes, also in the READY state.

The data cached into PVSS can be used either directly by the scripts of the applications or by the configurators upon the reception of the APPLY_RECIPE command in the READY state (Fig. 4). The former case maximizes flexibility as different nodes may access the data independently from each other, allowing delays, value-based conditions and other specific criteria to be included in the scripts. The latter scenario improves the performance as the configurator groups the request for configuration from all active devices and then applies the recipes to the whole set simultaneously.

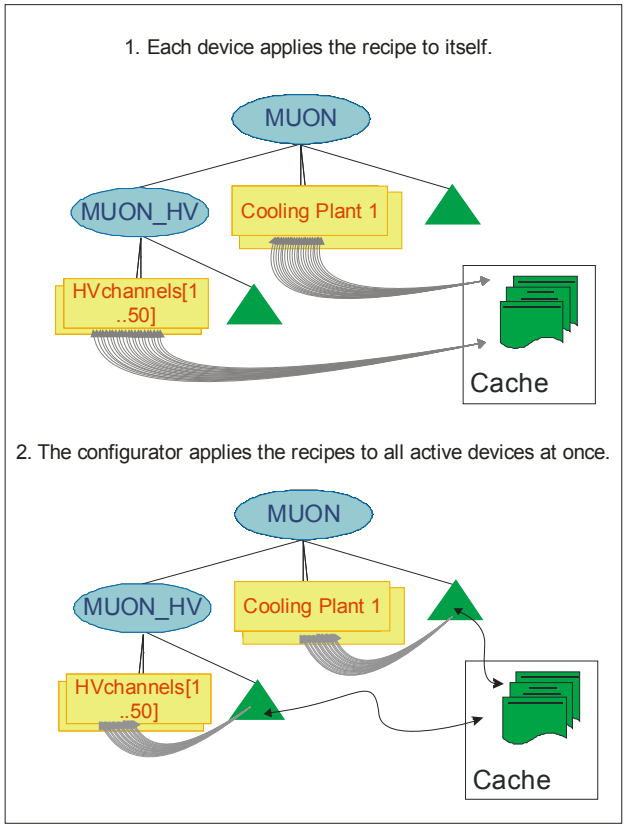

Figure 4: Applying a recipe.

In the event of errors, like database connection loss or inconsistency between the list of devices in the database recipes and in the running PVSS application, the state of the configurator is set to ERROR. In this state, the configurator responds to the RECOVER command by trying different recovery actions depending on the nature of the problem. At the end of the run, the UNLOAD command allows the operator to clear the configuration data loaded in PVSS.

The FSM-Configuration DB Tool is presently being used by several sub-detectors. An example is the MUON HV LHCb project [6], in an advanced phase of development, where all the equipment operated via FSM is configured from the configuration database using FSM-Configuration DB Tool. In this application, the FSM hierarchy tree is composed of more than 2400 devices. The configuration of these devices during the operation throughout different FSM states involves the management of more than 9000 data-point values at each stage that are handled in parallel by 10 configurators.

CONCLUSIONS

The FSM-Configuration DB Tool represents a common solution for the four LHC experiments to automate the handling of configuration data required by the control systems during operation. The tool standardizes the access to the configuration database and hence reduces significantly the development and maintenance efforts. The adopted implementation model respects the flexible partitioning schema of the experiments. The FSM-Configuration DB tool is in advanced stage of development and its functionality is presently being enhanced based on the experience gathered by several sub-detectors.

REFERENCES

[1] PVSS made by ETM professional control AG, Eisenstadt, Austria, http://www.pvss.com.

[2] O. Holme and M. Gonzalez Berges and P. Golonka and S. Schmeling, “The JCOP Framework”, ICALEPCS 2005, Geneva, October 2005.

[3] C. Gaspar and B. Franek, “Tools for the Automation of Large Distributed Control Systems”, IEEE Trans. Nucl. Sci., vol.53, no. 3, pp. 974-979, June 2006.

[4] P. Golonka, “The JCOP Framework Configuration Database Tool”, in preparation.

[5] F. Varela, “Software Management of the LHC detector Control systems”, ICALEPCS 2007, Knoxville, October 2007.

[6] M. Lenzi, “High Voltage Control System for Muon Detector”, LHCb Internal Working Note in preparation.