# The LHCb CONFIGURATION DATABASE

L. Abadie, CERN, Geneva, Switzerland and Samovar/GET/INT
C. Gaspar, E. van Herwijnen, R. Jacobsson, B. Jost, N. Neufeld, CERN, Geneva, Switzerland
LHCb 2004-087

*Abstract*

The aim of the LHCb configuration database [1] is to store information about all the controllable devices of the detector. The experiment's control system (that uses PVSS [2]) will configure, start up and monitor the detector from the information in the configuration database. The database will contain devices with their properties, connectivity and hierarchy. The ability to store and rapidly retrieve huge amounts of data, and the navigability between devices are important requirements. We have collected use cases to ensure the completeness of the design. Using the entity relationship modelling technique we describe the use cases as classes with attributes and links. We designed the schema for the tables using relational diagrams. This methodology has been applied to the TFC (switches) and DAQ system. Other parts of the detector will follow later. The database has been implemented using Oracle to benefit from central CERN database support. The project also foresees the creation of tools to populate, maintain, and configure the configuration database. To communicate between the control system and the database we have developed a system which sends queries to the database and displays the results in PVSS. This database will be used in conjunction with the configuration database developed by the CERN JCOP (Joint Controls Project) [3] project for PVSS.

## INTRODUCTION

LHCb is one of the four particle detectors at the CERN LHC (Large Hadron Collider). The monitoring, configuration and operation of experimental equipment will be handled by the ECS (Experiment Control System) [4].

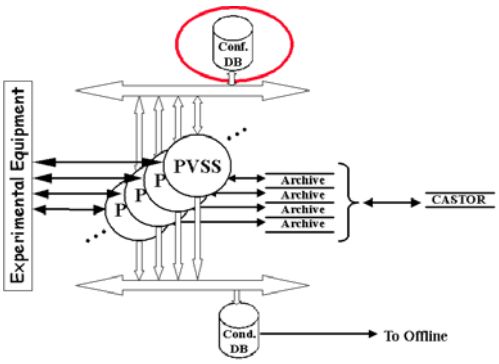


Figure 1: External data handling architecture

Fig. 1 shows the relationship between ECS and other systems. PVSS, a SCADA (Supervisory Control and Data Acquisition) system provides the interface to the experiment's equipment. All information regarding the equipment will reside in a configuration database.

In order to define a common architecture between the four experiments, JCOP offers a common framework and tools for PVSS. One of these tools provides part of a configuration database.

## THE SCHEMA

In this section we describe the objectives and requirements of the LHCb configuration database, and the methodology used to design its schema.

### *Objectives*

The database should contain the data that the ECS needs to configure the experiment's equipment:

- **Static properties**. These correspond to properties that don't change or change very infrequently, e.g. the hardware addresses, the geographical location, device structure. Changes in this data involve a hardware intervention.
- **Dynamic properties**. These correspond to properties that change frequently, e.g. alarm settings, hardware settings such as ramping speed.
- **Connectivities**. This describes how a device can be connected to other devices, e.g. a switch can be connected through one of its input ports to a supervisor and through one of its output ports to a card somewhere in the detector. This allows the creation of routing tables on the fly for dynamic programming of switches.
- **Hierarchy**. This describes how components are nested, e.g. a chip sits on a board that can be found in a crate which is placed in a rack.

### *Requirements*

The following requirements should be taken into account:

- The schema should cater for all detector subsystems.
- The database should be complete so that the ECS finds all the required information.

- The design should guarantee a reasonable response time when a query resulting in a large volume of data is loaded from the database into the ECS.
- Easy maintenance of the tables.

### *Methodology*

The following methodology has been applied for each subsystem:

- Consider its dataflow. Fig. 2 and Fig. 3 show respectively the LHCb online readout system architecture and the TFC (Timing and Fast Control) system dataflow. From this we can determine:
  - The list of all devices in the subsystem, e.g. readout supervisor, TFC switch, different fan-out types such as TTCtx, TTCrx belong to the TFC system.
  - The connections between devices, e.g. the TFC switch is connected to the readout supervisor on input and to a TTCtx fan-out on output.

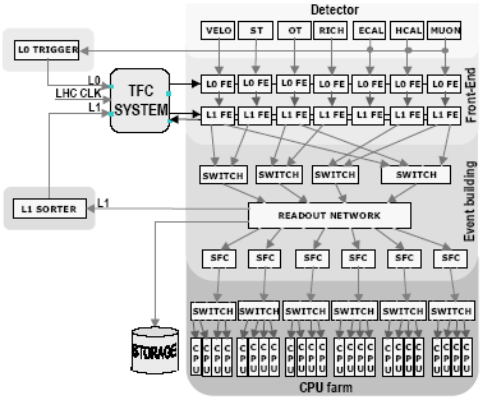


Figure 2: LHCb online dataflow

- Collect use cases. The following is an example from the TFC system (see the section "Concrete example: the TFC system" for definitions). For the **activity** "physics", using a **partition** consisting of the velo and rich subdetectors, determine the appropriate readout supervisor and the internal connectivity of the TFC switch. This use case uses the concepts of links and paths (cf Fig. 4).
- Use the entity relationship model to design the schema of the tables. Fig. 5 and Fig. 6 show an extract of the table design.

## IMPLEMENTATION

This section describes the components and tools of the configuration database.

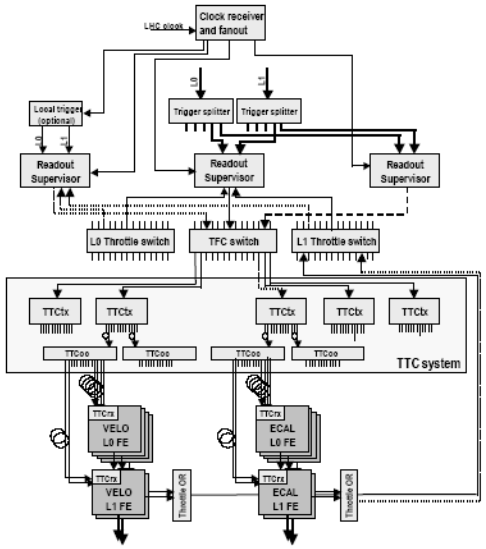


Figure 3: TFC system dataflow

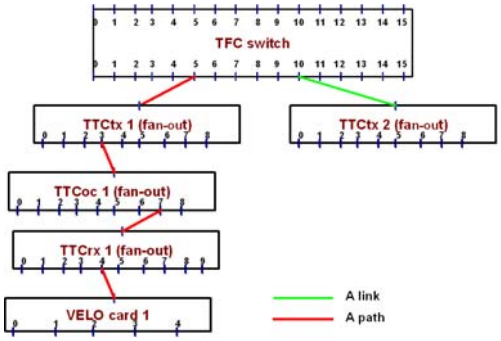


Figure 4: Link and path concepts: a path between a VELO card and the TFC switch

### *Integration of the JCOP configuration database tool*

An important requirement was to fully exploit the work done by JCOP. This tool provides tables that contain static and dynamic device properties. They are created automatically in the specified database from PVSS panels. It does not store information about links or hierarchy which are stored in tables specific to LHCb. Some work had to be done to provide a unified interface to both sets of tables; the LHCb tables point to information stored in the JCOP part to avoid duplication of data. Fig. 7 shows the 2 kinds of tables inside the configuration database. PL/SQL programs ensure the coherence between the JCOP tables and the LHCb tables. We also need to ensure that the device names chosen

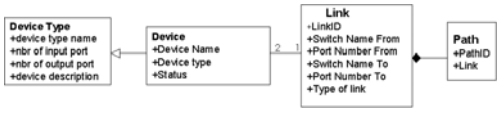


Figure 5: Entity relationship model

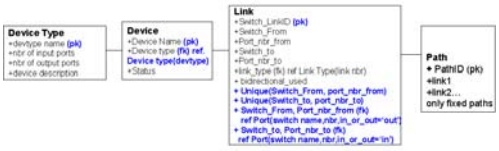


Figure 6: Table design and constraints

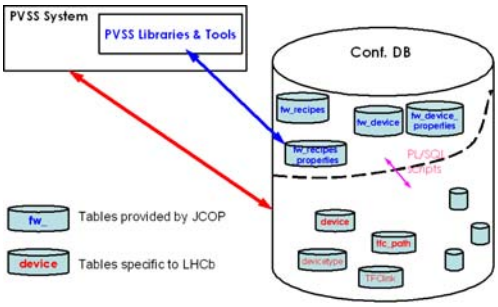


Figure 7: PVSS and the Configuration Database

by subdetector applications are the same throughout both table sets.

### *Implementation details*

The configuration database has been implemented using Oracle technology because of the CERN Central support team and the wide range of tools. For future extensions of the database to other subsystems the JCOP Configuration Database tool will need to be adapted; we are providing the JCOP team with feedback. We also communicate between PVSS and the LHCb tables via ProC/C++ (Oracle tool) to access the database and C/C++ to encapsulate the SQL and PL/SQL statements.
In parallel, we have implemented in Python a tool to edit and navigate through the database (see next subsection for details). We keep the different versions of PVSS projects and other software in CVS [5].

### *Database editor*

Standard database editors are good at making the table structure explicit. We required a tool to inspect the links and hierarchy of the devices in the database graphically. It should also allow for convenient mass insertion of data via ”rubber banding” (copy-paste). A Python prototype (cdbVIS [6]) allows viewing of the connectivities (see Fig. 8 for links and Fig. 9 for paths).

## CONCRETE EXAMPLE: THE TFC SYSTEM

### *TFC system overview*

Fig. 10 illustrates the TFC system [7].
The main components of the TFC system are:

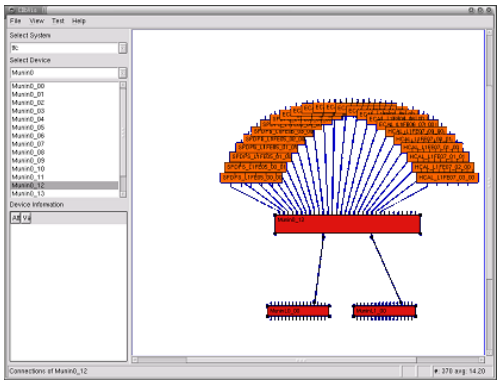

Figure 8: Possible connectivities: a Throttle OR is connected to a Throttle switch on its output and to L1FE boards on its output

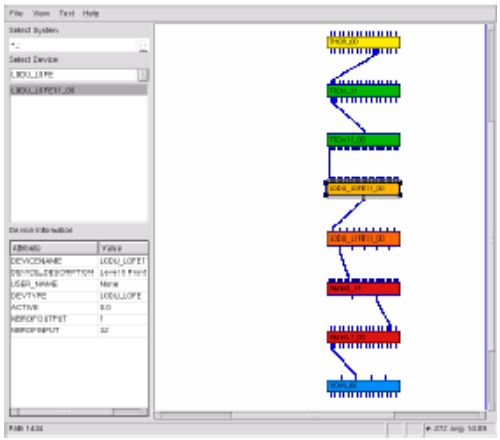

Figure 9: View of the path from the readout supervisor to the Throttle OR via a FE board and the TFC switch

- **The readout supervisors**: handle all timing, triggering and control of the FE electronics.
- **The TFC switch**: associates a number of elements (e.g. subdetectors) to a readout supervisor.
- **The Throttles**: feed back throttle information to the appropriate readout supervisor.

### *Requirements*

To test the configuration database it was integrated into an operational control system. The requirements were the following:

- obtain the connectivities between a particular selection of subdetectors (partition) and the TFC switch from the database (LHCb part).
- get from the database(LHCb part) the list of free readout supervisors corresponding to a selected activity (e.g. physics, testing). This allows us to determine the TFC switch routing table.

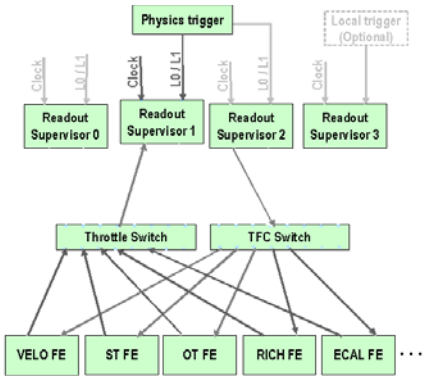


Figure 10: TFC simplified architecture

- save (load) into (resp. from) the database (JCOP part), the readout supervisor activities (recipes).

### *Implementation*

A C++ program receives queries from PVSS and sends back the result. We have used and customized panels from the JCOP configuration database tool. The following figures show different screenshots from the project.
Fig. 11 shows the panel to select the subdetector(s) and an activity for the readout supervisor.
Fig. 12 shows the connectivities between the selected subdetector(s) and the TFC switch. It also suggests a readout supervisor to the user.
Fig. 13 enables the user to save new activities for the readout supervisor and to monitor the selected readout supervisor. It also displays statistics and the current settings corresponding to the activity.

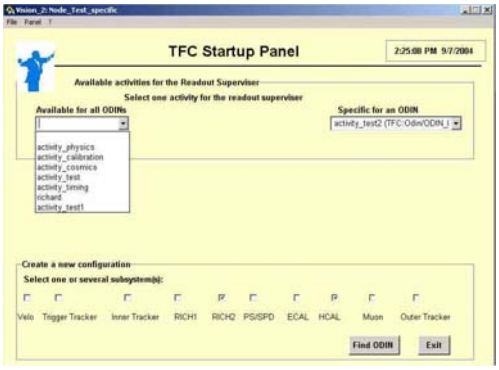


Figure 11: Selection of a running mode and subsystems

## CONCLUSION

The design schema has been done for the DAQ and TFC system. We will apply it for the other LHCb subdetectors. We have also integrated PVSS and the configuration database (JCOP and LHCb parts). An API is required to enable the clients to interact with the database in a standard way. The cdbVis editor will be extended to permit mass insertion of data, and integration with PVSS for graphic fault identification.

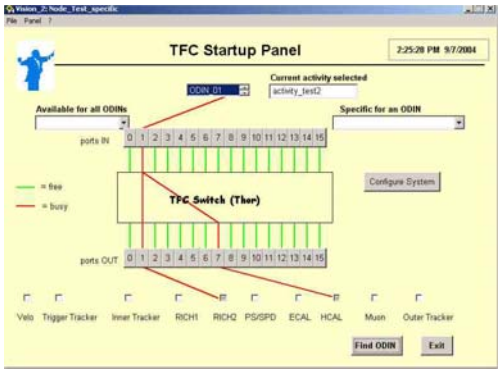


Figure 12: Connectivities between subsystems and the TFC switch.

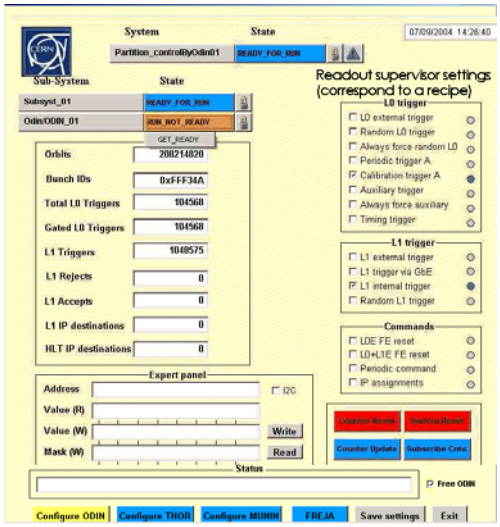


Figure 13: Readout supervisor: control and monitoring.

## ACKNOWLEDGEMENTS

We would like to thank Laura Del Cano for her help with the Configuration Database Framework tools.

## REFERENCES

[1] http://lhcb-comp.web.cern.ch/lhcb-comp/ECS/configurationdb/default.htm

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

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

[4] http://lhcb-comp.web.cern.ch/lhcb-comp/ECS/default.htm

[5] http://isscvs.cern.ch/cgi-bin/cvsweb.cgi/TFC/?cvsroot=lhcb

[6] http://lhcb-comp.web.cern.ch/lhcb-comp/ECS/configurationdb/cdbVis.htm

[7] http://lhcb-comp.web.cern.ch/lhcb-comp/TFC/default.html