SOFTWARE MANAGEMENT OF THE LHC DETECTOR CONTROL SYSTEMS

F. Varela, CERN, Geneva, Switzerland

Abstract

The control systems of each of the four Large Hadron Collider (LHC) experiments will contain of the order of 150 computers running the back-end applications. These applications will have to be maintained and eventually upgraded during the lifetime of the experiments, ~20 years. This paper presents the centralized software management strategy adopted by the Joint COntrols Project (JCOP) [1], which is based on a central database that holds the overall system configuration. The approach facilitates the integration of different parts of a control system and provides versioning of its various software components. The information stored in the configuration database can eventually be used to restore a computer in the event of failure.

INTRODUCTION

The back-end software of the control systems of the LHC experiments is based on the commercial Supervisory Control And Data Acquisition (SCADA) package Prozeßvisualisierungs- und Steuerungsystem (PVSS) [2] and the JCOP Framework (FW) [3]. It consists of a large variety of inter-cooperating control applications that are distributed over the back-end computers running either Linux or Microsoft Windows. These control applications have been developed by different groups all around the world and are presently being integrated by the central controls teams at the experiments facilities.

Due to the size of the systems, the large diversity of software elements and the long lifetime of the experiments, a centralized software management approach is of vital importance. In the following the strategy developed by JCOP is presented.

JCOP SOFTWARE MANAGEMENT STRATEGY

The JCOP centralized software management approach tackles the following points:

- Installation of the basics software infrastructure of the computer, e.g. operating systems, security patches, and deployment of common applications like PVSS.
- Multi-user/multi-location development of the PVSS-based applications.
- Integration of the different software elements of the control system.
- Providing a coherent overview of the current organization of the system.
- Remote incremental upgrades of the configuration of the nodes.
- Versioning of the software components.
- Minimization of the downtime by permitting to restore a system to the last known configuration.

In order to address these use cases, JCOP utilizes two separated set of tools:

- The CERN Computer Network Infrastructure and Controls (CNIC) [4] Working Group frameworks, Computer Management Framework (CMF) and Linux for Controls (L4C) for the installation of Microsoft Windows and Linux computers respectively.
- The Component Installation Tool and the System Configuration Database of the JCOP FW for the management and configuration of the PVSS based applications.

CMF AND LINUXFC

Both L4C and CMF allow controlling the installation, configuration, and management of software applications (e.g. the operating system and the corresponding patches, office or controls software suites) on a defined set of computers. Although CMF and L4C share the same design principles, they have been implemented as two separate schemes: While in CMF the desired software configuration of the Microsoft Windows computers is defined using a graphical interface and is stored in a central database, the L4C scheme uses a template-based approach using a tree structure of editable text files to define the configuration of the Linux nodes. In both cases the configuration information is accessed by software daemons running on the nodes in order to carry out the installation or un-installation of applications according to the contents of the database/templates. It is important to mention that both CMF and L4C are computer-oriented and therefore they do not allow addressing individual PVSS applications on a computer. Hence, the installation and configuration of PVSS applications, is beyond their scope and it is addressed using the JCOP FW Component Installation Tool.

FW COMPONENT INSTALLATION TOOL

The FW Component Installation Tool is a software package that can be included into a PVSS application and that allows system experts to modify interactively the list of FW components installed. Installing a new component implies to modify the real-time database of the target PVSS application in order to add the data-points required for the operation of the components and to copy the

component files (e.g. libraries, graphical interfaces, help) into a place that can be accessed by the application. Typically the components are installed on a network drive so that these files can be shared and accessed by multiple applications.

The functionality of the Component Installation Tool was recently extended in order to allow developers to create their own software packages that can later be installed using the same tool in a different PVSS application. This feature eases the multi-user and multi-location development of control applications as developers can work separately on different parts of the system and these can afterwards be merged into the final system. The packaging functionality of the Component Installation Tool also allows to define dependencies across specific versions of components that are checked and resolved at installation time. A versioning system, like CVS, is frequently used to keep track of the different versions of these components.

The FW Component Installation Tool only permits to address individual PVSS applications interactively. In order to overcome these limitations, the functionality of the FW Component Installation Tool was enhanced to allow for remote installation of components in multiple PVSS applications from a central console. The approach followed to provide this functionality is similar to that of CMF, where the required configuration of the nodes in the control system is defined in a central database called the System Configuration DB, and the software daemons running in the remote nodes, called database agents, execute the component installation requests.

FW SYSTEM CONFIGURATION DATABASE

The main motivation for the FW System Configuration Database was to complement the functionality of the CMF master database by addressing all information of the control system that is beyond the scope these software management frameworks, i.e. the overall layout of the distributed control system and the relationships between its different integrating elements, e.g. hierarchy of machines, assignment of PVSS applications and user-specific applications to computers, connectivity between PVSS applications, FW components installed in a PVSS applications, etc. The FW System Configuration DB is integrated in the JCOP FW as a sub-component of the FW Configuration DB [5]. A crucial difference with respect to the CMF database is that the FW System Configuration Database is accessed by PVSS systems running under both Microsoft Windows and Linux and hence, it provides an integrated view of the software configuration of all computers in the control system.

The information in the FW System Configuration DB represents the desired configuration of the control system and may eventually be used to restore a system in the event of a failure. This database also serves as the master source of configuration data for other tools of the FW, like the System Overview Tool [6].

FW Installation Tool DB Agent

The FW System Configuration Database is originally populated directly from the running PVSS applications by means of the DB Agents of the FW Component Installation Tool. These agents are implemented as scripts running in the background and there is one instance per PVSS application. The first time the DB agent is started-up an initial set of information of the current application is exported to the database, namely:

- Computer configuration, e.g. name, IP addresses, etc.
- PVSS application information, e.g. application name, source directory, list of PVSS executables used, etc.
- PVSS version and patch level applied.

The list of FW components and sub-detector specific components installed in a PVSS application is treated in a special manner. The DB agents distinguish two management modes of PVSS applications: central and local. The application management mode can be selected by the user. In the local management mode, the running PVSS application is the source of the configuration and the information is exported from the application to the database. The FW System Configuration Database works as master for the configuration for centrally managed applications. Should the list of components installed in a centrally managed application be modified, the DB agents will update the component list in the application accordingly, i.e. install or uninstall components until the configuration of the application reflects that defined in the database. In this case, the DB agents differentiate between two different types of requests: silent and interactive. Silent requests force the installation/un-installation of components without any user intervention whereas an interactive request requires the user's acknowledgement.

The DB agents check the consistency between the information in the database and the PVSS applications at regular time intervals and synchronize the contents of either the database or of the PVSS application dependent on its management mode. The results of the consistency check and of the synchronization process are written to the database by the DB agents as well as the time when these two steps were performed. This allows to supervise the running state of the DB agents of the different PVSS applications and to detect configuration mismatches. In the event of an inconsistency between the contents of a PVSS application and of the System Configuration DB being detected, an alarm is raised in the central alarm screen of the experiment. In such a case, expert graphical interfaces are available to help identify the cause of the inconsistency.

FW System Configuration DB Editor

An overview of the whole configuration of the control system is provided by the FW System Configuration DB Editor. This application supports two different views, the so-called project setup and the PVSS system hierarchy views. The former view enables the hierarchical navigation throughout the physical setup of the control system whereas the latter displays the hierarchical

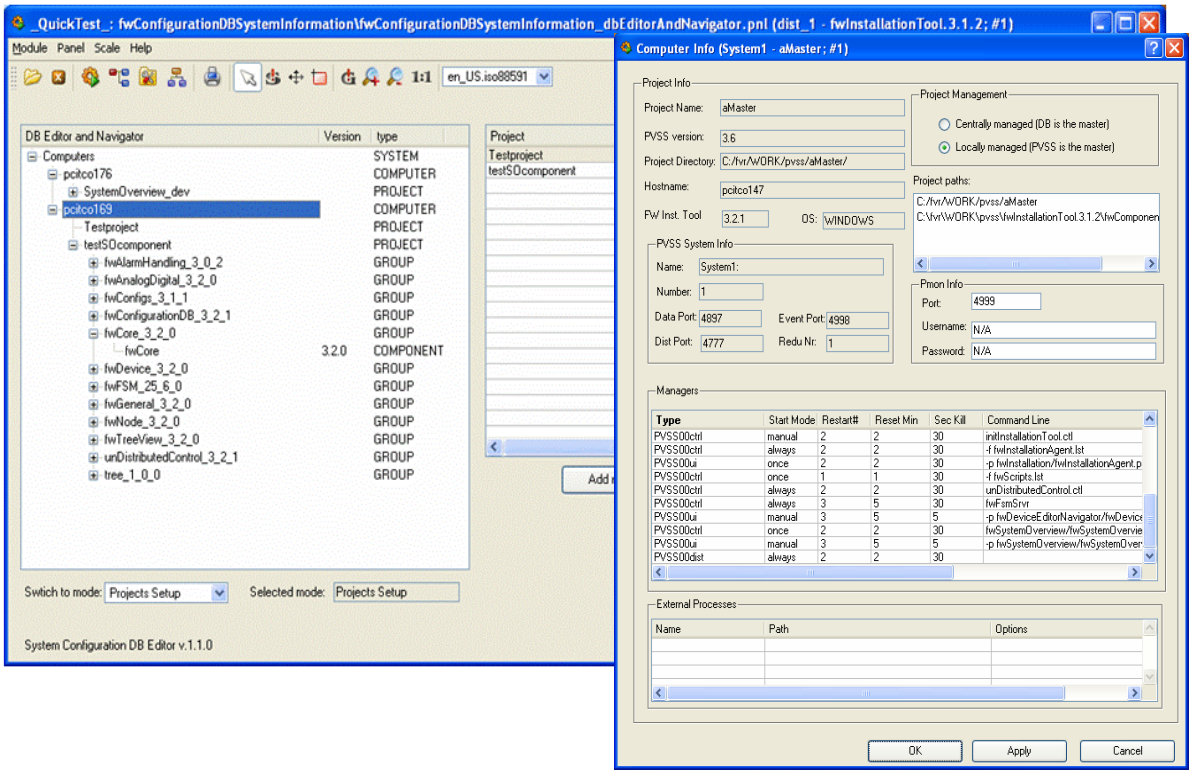

Figure 1: FW System Configuration DB Editor.

arrangement of PVSS applications. The selection of any of the items in the view triggers the communication with the database in order to obtain more detailed information, which is displayed on the right-hand side of the graphical interface. The consistency of the configuration between the running applications and the contents of the database, as well as the timestamp when the last check was performed, are also accessible via the user interface.

The System Configuration DB editor also allows to modify the contents of database provided that access permissions are granted. This functionality enables the remote and simultaneous management of multiple PVSS applications of the control system, e.g. to install/uninstall FW components in centrally managed applications.

CONCLUSIONS

The JCOP software management strategy eases the integration of the different parts of the control systems and allows for incremental upgrades of the software components thus enabling a smooth evolution of the control systems over the lifetime of the experiments. Presently, this approach is successfully being followed by CMS and the other experiments are expected to adopt it soon. It is envisaged to enhance the functionality of the various tools by providing access control, which will enable various users to upgrade concurrently different parts of a single PVSS application without conflicts.

ACKNOWLEDGEMENTS

I would like to express my gratitude to M. Sliwinski and M. Gonzalez who started the work on the FW Component Installation Tool. Special thanks go to S. M. Schmeling who greatly extended the functionality of the tool and participated in the design of the overall software management approach for LHC experiments' control systems.

REFERENCES

[1] D. R. Myers *et al.*, "The LHC Experiments' Joint COntrols Project, JCOP", ICALEPCS 1999, Trieste, Italy.

[2] http://www.pvss.com

[3] O. Holme, M. Gonzalez Berges, P. Golonka, S. Schmeling, "The JCOP Framework", ICALEPCS 2005, Geneva, Switzerland.

[4] S. Lueders, "Update on the Computing and Network Infrastructure for Controls (CNIC)", ICALEPCS 2007, Knoxville, USA.

[5] L. del Cano *et al.*, "Extending the capabilities of SCADA device modelling for the LHC experiments", ICALEPCS 2003, Gyeongju, Korea.

[6] M. Gonzalez, K. Joshi, F. Varela, "The System Overview Tool of the Joint Controls Project (JCOP) Framework", ICALEPCS 2007, Knoxville, USA.