Software management of the LHC Detector Control Systems

F. Varela
CERN, Geneva, Switzerland

Introduction

The control systems of each of the four Large Hadron Collider (LHC) experiments will contain of the order of 150 Linux and Microsoft Windows computers running the back-end applications, which are based on the commercial SCADA PVSS [1] and the Joint COntrols Project (JCOP) [2] Framework (FW) [3]. 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 the software elements and the long lifetime of the experiments, ~20 years, a centralized software management strategy is of vital importance.

The JCOP SW Management Approach

The JCOP centralized software management approach tackles the following points:

- Multi-user/multi-location development of the controls applications.
- Integration of the different software elements of the control system.
- Providing a coherent overview of the current organization of the system.
- Incremental upgrades of the software configuration of the remote nodes.
- Versioning of the software components.
- Minimization of the downtime by permitting to restore a system to the last known configuration.

The JCOP strategy distinguishes two main aspects:

- Installation of the basic infrastructure of the computers, e.g. installation of operating systems, security patches, and deployment of common applications, which is addressed with the tools of the CERN Computer Network Infrastructure and Controls (CNIC) [4] working group: Computer Management Framework (CMF) and Linux for Controls (L4C)
- Installation and configuration of the PVSS projects, which is performed with the FW Component Installation Tool, which makes use of the FW System Configuration Database to store persistently the information.

The FW System Configuration DB

Stores the overall configuration of the distributed control system and the relationships between its different elements, like hierarchy of computers, configuration of PVSS projects or the connectivity between PVSS applications.

- This information can be used to restore a system in the event of a failure.
- Configuration source for other tools of the FW, like the System Overview Tool [5].

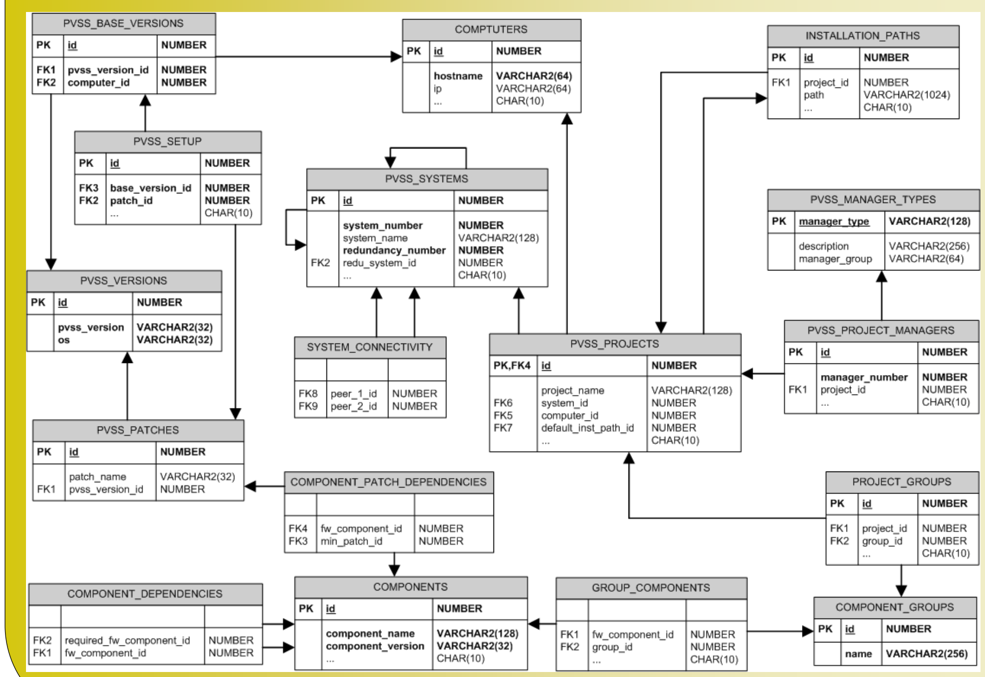

Figure 1: FW System Configuration Database schema

The FW Component Installation Tool

The functionality of the FW Component Installation Tool is two-fold:

- **Packager:**
  - PVSS-based developments can be packed as software components that can later be installed in different PVSS projects.
  - A versioning system, like CVS, is used to keep track of the evolution of these components.
- **Installer:**
  - System experts can modify the list of PVSS-based components installed in a project.
  - The tool resolves dependencies across specific versions of the PVSS-based components

Two project management modes are possible in order not to perturb the operation of some critical services of the detector control system:

- **Local:**
  - Enables the interactive and local installation of PVSS-based components.
  - The System Configuration Database is updated accordingly to reflect the changes.
- **Centralized:**
  - The desired configuration of a set of PVSS projects is defined in the central FW System Configuration Database.
  - This information is accessed by the DB-agents of the FW Component Installation Tool running on the remote nodes that process the installation requests.
  - The DB-agents handle *silent* and *interactive* requests.

The consistency between the contents of the running projects and of the System Configuration DB is checked at regular intervals and alarms are raised as required

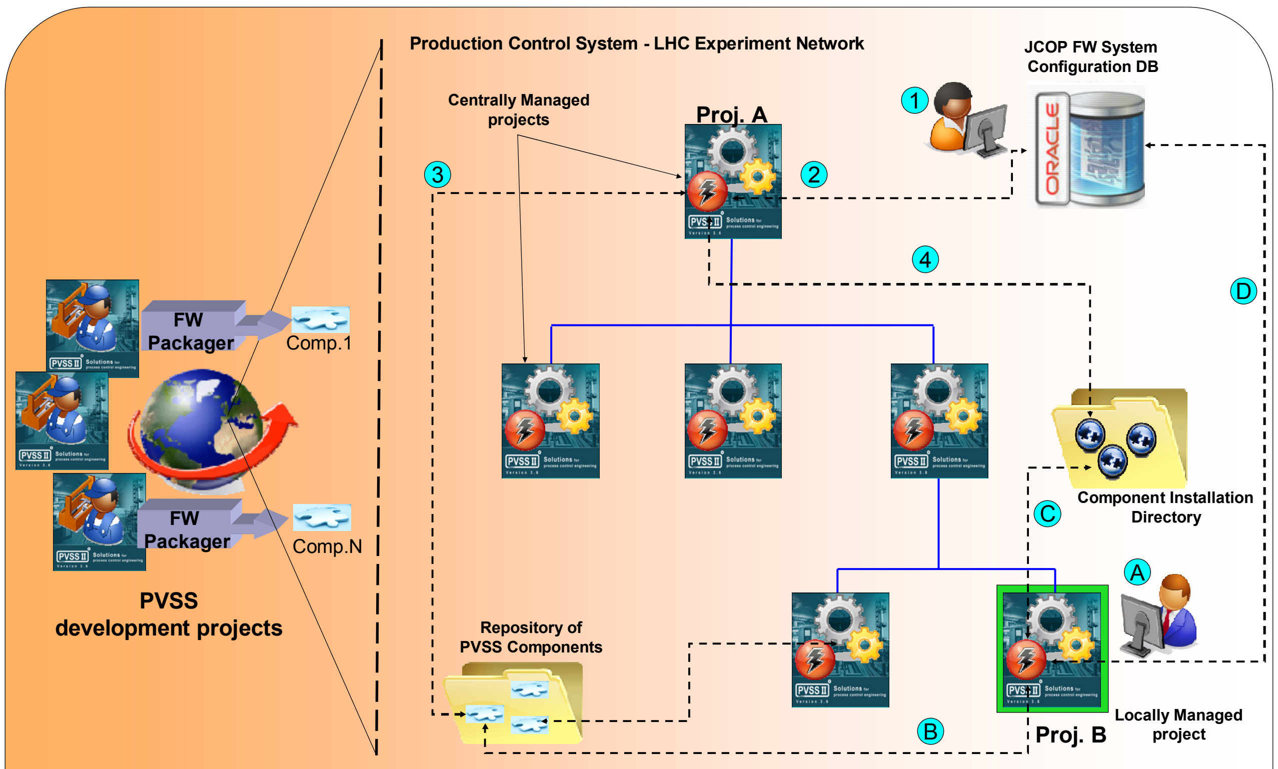

Figure 2: PVSS software management cycle. Control applications developed all around the world are packaged using the FW packager and copied to the experiment network. For centrally managed projects, the desired configuration is defined in the System Configuration DB using the FW DB Editor (1). The DB-agents in the remote projects access this information (2) and perform the installation of the components from a central repository (3) into a shared folder(4). For locally managed projects, an expert defines interactively the components to be installed in a single project (A) and the DB-agent carries out the installation requests (B, C) and updates the information in the FW System Configuration DB (D).

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.

References

[1] http://www.pvss.com
[2] D. R. Myers et al, “The LHC Experiments’ Joint COntrols Project, JCOP”, ICALEPCS 1999, Trieste, Italy.
[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] M. Gonzalez, K. Joshi, F. Varela, “The System Overview Tool of the Joint Controls Project (JCOP) Framework”, ICALEPCS 2007, Knoxville, USA.