

A RAPID APPLICATION DEVELOPMENT FRAMEWORK USED FOR LHC HARDWARE COMMISSIONING TOOLS AND OTHER ACCELERATOR RELATED APPLICATIONS

C. Charrondière, O. Andreassen, D. Kudryavtsev, M. Nybo, A. Raimondo, H. Reymond,
A. Rijllart, M. Zerlauth, CERN, Geneva, Switzerland
V. Shaipov, R. Sorokoletov, JINR, Dubna, Russia

Abstract

A set of tools were developed for the LHC Hardware Commissioning team to analyse and validate the electrical circuits, the powering systems and their associated protection equipment. The choice was made to develop these tools using the Rapid Application Development Framework based on LabVIEW™, because it was the most suited to fulfil the requirements of flexibility, adaptability, quality, integration into the LHC accelerator control software and light maintenance. This paper describes the Rapid Application Development (RADE) framework, the motivation behind its development, the applications which use it focussing on the interfaces to C++ and Java. It also reports on the experience during the LHC Hardware Commissioning, together with two other examples of the use of RADE for very different applications: CLIC Two-Beam Test Stand viewer and the LHC Multi-Alignment Control System.

INTRODUCTION

The two counter rotating proton beams of the LHC [1] are guided by 1232 superconducting dipole magnets operating at 1.9K and more than 8000 other superconducting magnets. A current of 11850A flows in the dipoles, to create the 8.33T magnetic field required to bend the 7TeV beam around the 27km ring of the LHC. Before starting operation, all the electrical circuits were tested and validated during the Hardware Commissioning [2].

A number of tools were developed to validate the technical systems involved in the powering of the superconducting magnets (interlocks, power converters, quench detection, etc.); these tools gather the data, perform calculations and check or assist the operator to assess the correctness of the results. These tools are based on LabVIEW™ and were implemented using the RADE Framework. Because of its integration with the accelerator control system, RADE could also be used to develop other applications like the Two-Beam Test Stand (TBTS) [4] viewer for the CLIC test facility, and the Multiple-Alignment Control System (MACS) for the alignment of the LHC collimators [5].

RADE ARCHITECTURE

The Rapid Application Development Environment is targeting test and development applications. It was designed to design and maintain, in a short time, rapidly evolving applications, like those needed during studies typical of CERN machine development runs. It can also be used for expert diagnostics tools and test facilities.

Software Technology Evolution

RADE makes use of several communication layers as shown in Figure 1, to adapt and interface to the systems and the protocols used in the CERN accelerator control environment.

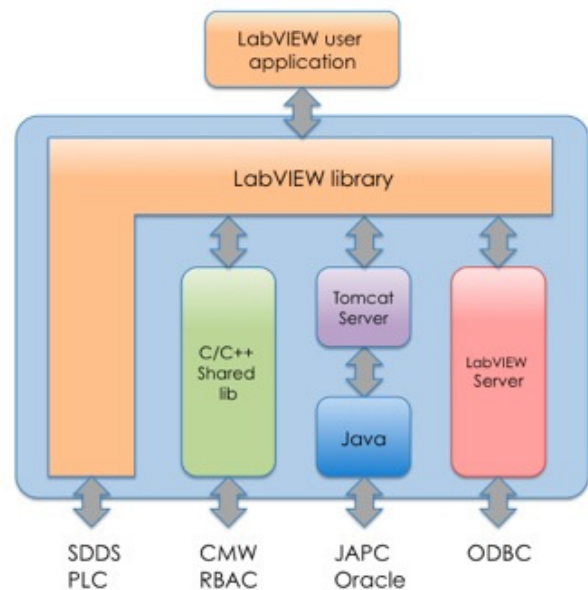


Figure 1: Communication layers in RADE.

The development effort was concentrated in the integration to the CERN accelerator control environment.

JAPC Interface

JAPC [6] is a communication layer to control accelerator devices using Java. Client programs can access JAPC parameters with *set* and *get* data, or can wait to be notified of value changes using the subscription mechanism. JAPC is a unified API for all the parameter types present in the control system. The diversity of devices is handled below the JAPC interface where each type of device has its own implementation.

For our purpose we have implemented a LabVIEW™ to JAPC interface via a Tomcat server (Figure 1). The same mechanism is used to access ORACLE databases.

CMW Wrapper

The Common MiddleWare (CMW) [7] is based on the Remote Device Access (RDA) client package to access accelerator Front End Computers (FESA FEC). All user interactions with RDA are associated with dynamically

allocated data buffers. A “set” request requires the use of a component that allocates a data buffer. One can then sequentially insert data, then the data in the buffer is sent and the buffer is released.

We implemented the LabVIEW™ to CMW interface in the form of a wrapper for the library.

It consists of a component palette for LabVIEW™ and some DLLs. The RDA C++ interface for Windows was used to create the wrapper; however, all Windows-specific code was encapsulated so that porting it to Linux became straightforward. The communication layers are shown in Figure 2.

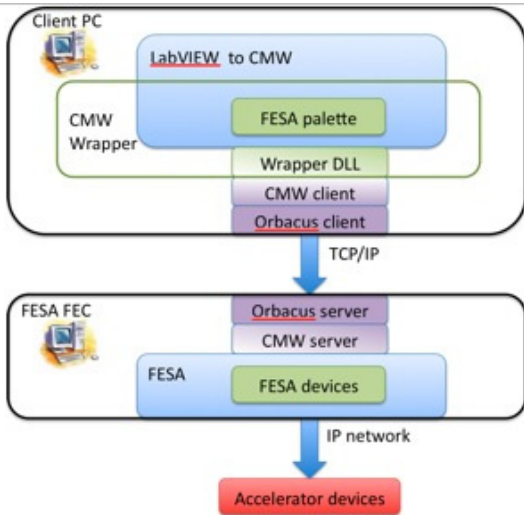


Figure 2: Communication layers LabVIEW to CMW.

SDDS Library

SDDS [8] stands for Self Described Data Set. This file format is commonly used at CERN and can contain any data type. The header of the file describes its content.

The SDDS library in RADE was implemented in standard LabVIEW™. It has functions to read the header and to direct read access parts of the data. For the digital signals only the state changes are output together with their timestamps. The library can read both ASCII and binary format.

PLC Library

The communication to the PLC is written in standard LabVIEW™ using the “Fetch-Write” protocol from Siemens™ via TCP-IP. The “Fetch-Write” has to be declared in the PLC to authorise an external device to access its data blocks. For more flexibility we are developing a wrapper for LIBNODEAVE [9], an open source library to access Siemens™ PLCs.

RADE IN APPLICATIONS

RADE in Hardware Commissioning

During the Hardware Commissioning tests are carried-out by specialist teams to qualify the electrical circuits

for operation. All the 1600 circuits are tested according to a predefined sequence of test steps (interlocks, current ramps, firing of heaters, opening of switches, etc.) and the behaviour of the systems is verified to be as expected. In this last phase, a set of applications are used to check or assist the operator to assess the results of the test procedure.

The Sequencer, written in Java, drives the equipment through the test steps. A set of C++ programs gather and store test data in SDDS files and in an ORACLE™ database. The Post-Mortem Analysis tool, developed on RADE, performs the data analysis. The collected data is presented in the Event Analyser (Figure 3); it carries-out the analysis particular to each circuit type, by presenting the data extracted from the SDDS files and from the ORACLE™ database to the operator.

RADE allowed to rapidly create expert GUIs and analyse data using graphical programming. All RADE components were used during the implementation of the Post Mortem analysis (Figure 4).

In the 2008 campaign, 11735 test steps were performed involving Post Mortem; this peaked in August, 3234 test steps were executed.



Figure 3: Post Mortem Event Analyser .

RADE in Two-Beam Test Stand

The Two-Beam Test Stand (TBTS) in the CLIC experimental hall (CLEX) aims at testing the two-beam acceleration scheme for CLIC, the CERN Linear Collider. Beam stability, beam loss issues and RF structures are studied in this set-up.

All the signals are acquired through FESA devices, which are easily accessible using the CMW wrapper or the JAPC interface. Data from the Beam Position Monitors (BPM) and RF signals is collected and presented in many different views. The RADE framework was used to implement the acquisition as a server task. Each graphical user interface is a client taking only the data it requires to display in the appropriate format, thus avoiding data duplication.

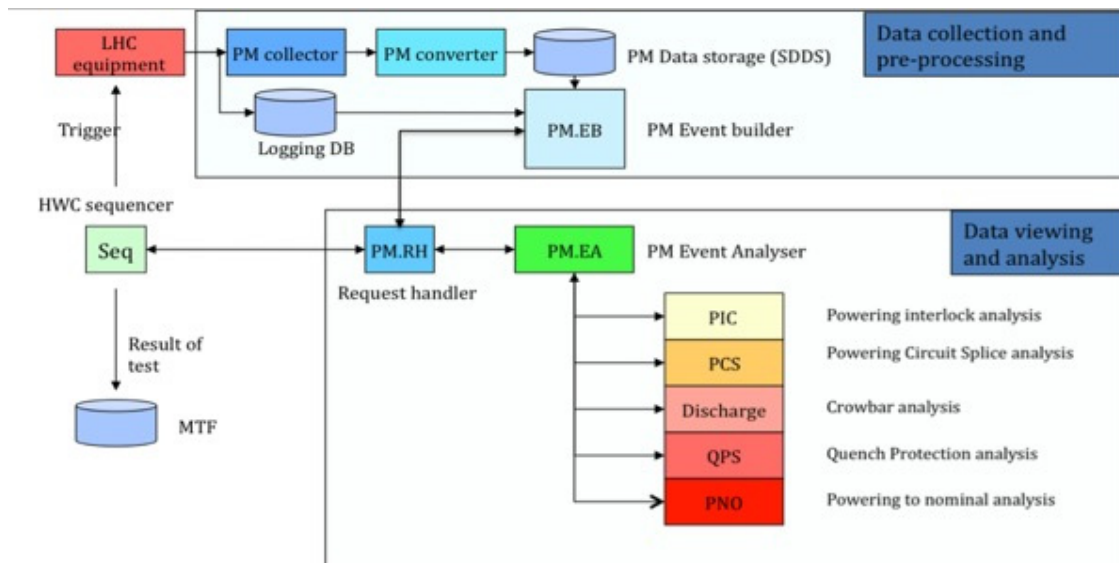


Figure 4: Post Mortem system overview for hardware commissioning.

RADE in the Multi-Alignment Control System

MACS is a remotely controlled survey system to verify the alignment of the LHC collimators. The system is based on stretched wire offset measurements combined with digital photogrammetry; it will determine the transversal position of the survey system with respect to the surrounding reference magnets. Once this is known the reversal of this process is used to determine the position of the collimators with respect to the survey system. The whole system has been installed on a train, controlled by a Siemens™ PLC, attached to a monorail and can be remote controlled by operators outside the LHC tunnel (Figure 5).

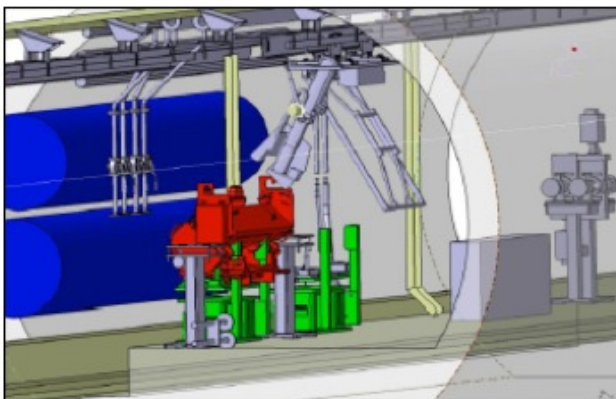


Figure 5: Collimator survey wagon.

The MACS application was developed using RADE to control the PLC and communicate with measurement instruments such as: AICON™ camera, Wyler™ Zerotronc inclination sensors, Leica™ Disto, Aeroel™ XLS35 laser micrometer. The approach has saved development time and provided a high quality GUI.

CONCLUSION

The described applications show that RADE is efficient and flexible with a good integration into the CERN accelerator control infrastructure. RADE is a basis to be extended to other interfaces and protocols, like DIP (Data Interchange Protocol[10]), accelerator timing and alarms.

REFERENCES

- [1] L. Evans, "The Large Hadron Collider – Present Status and Prospects", *IEEE Trans. Appl. Supercond.*, Vol. 10 No. 1 (2000), 44-48.
- [2] R. Saban, "LHC Hardware Commissioning Summary", EPAC08, Genoa Italy.
- [3] A. Raimondo, "Rapid Application Development Environment Based on LabVIEW" - <https://edms.cern.ch/document/904425/1>.
- [4] V. Ziemman, "The Two-Beam Test-Stand in CTF3", EPAC06, Edinburgh UK.
- [5] O. Aberle, "Collimator integration and installation example of one object to be installed in the LHC", EPAC08, Genoa, Italy.
- [6] V. Baggiolini, "JAPC-the Java API for parameter control", ICALEPCS2005, Geneva, Switzerland.
- [7] K. Kostro, "The control middleware (CMW) at CERN status and usage", ICALEPCS2003, Gyeongju, Korea.
- [8] R. Soliday, "New features in the SDDS tool kit", PAC2003, Portland, Oregon, US.
- [9] LIBNODAVE, "Exchange data with Siemens PLCs", <http://libnodave.sourceforge.net>.
- [10] DIP description, <http://ts-dep-lea-int.web.cern.ch/ts-dep-lea-int/dataexchange-DIP.htm>.