THE SPIRAL2 COMMAND CONTROL SOFTWARE ORGANIZATION AND MANAGEMENT

Dominique Touchard*, Pascal Gillette, Christophe Haquin, Eric Lecorche, Evelyne Lemaitre, Patrick Lermine, Laurent Philippe# (CEA-CNRS/GANIL, Caen, France)
Jean-François Gournay, Yves Lussignol, Pierre Mattei (CEA/IRFU, Saclay, France)

Abstract

The SPIRAL2 project aims to provide a new facility able to produce and study rare ions. To ease collaboration between laboratories involved in software development, the command control team has chosen the EPICS software. At an upper layer, high level applications will be programmed in Java while the XAL framework is currently under investigation. A development skeleton, programming rules, subversion development tools are about to be fixed on to achieve the whole organization. Program developers will be able to generate generic EPICS applications which can be integrated in each IOC VME crate or LINUX box. Furthermore, Spiral2 beam control equipment will be described in a relational database and a program will be provided to automatically generate EPICS flat databases. The aim of this paper is to describe this organization and the benefits for the Spiral2 command control team.

THE SPIRAL2 PROJECT

Overview

The SPIRAL2 project aims to produce and study rare ions to fulfill the demands of the large (700 physicists) international community using GANIL [1]. It consists of a new facility to provide high intensity rare ions beams (RIB) using a Uranium carbide target fission process [2]. To achieve this goal, deuterons, protons or heavy primary beam will be produced from two kinds of sources, and then pre-accelerated by a RFQ, the injector part of the facility as shown in Figure 1. The produced beams are then accelerated by a superconducting linear accelerator, the LINAC part, and transported in Hight Energy Beam Transport lines, the HEBT part.

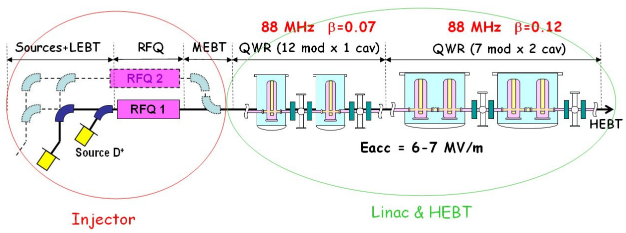

Figure 1: The Spiral2 accelerator schematic layout.

After a carbon convertor and the above-mentioned target, post acceleration of RIB produced by "the Isotope Separation On-Line Method" is assured by the existing CIME cyclotron built in 2001 for the SPIRAL project. SPIRAL2 beams, both before and after post-acceleration and depending of their kind, will be able to be used either in the present experimental area of GANIL or into a new low energy experimental hall.

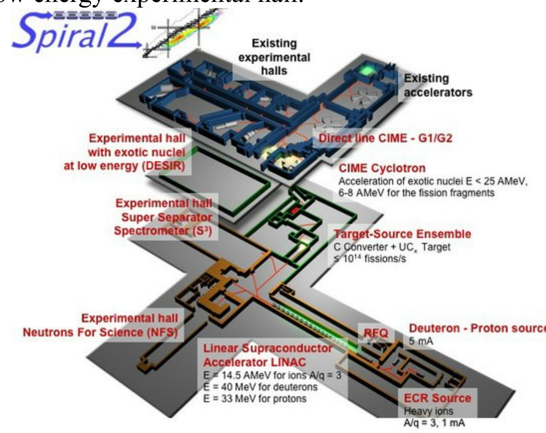

Figure 2: Spiral2 facility schematic layout.

Collaboration Organization

The European Preparatory Phase for SPIRAL2, which deals with the critical financial, legal and organizational issues related to the international character of the SPIRAL2 facility during its construction and operation phases, approached 25 partner institutions from 13 countries. As far as the SPIRAL2 command control team inside the infrastructure section is concerned by project organization, three main laboratories, IPHC (Strasbourg), IRFU (Saclay), and GANIL (Caen) are involved in the design and the deployment of the control systems set [3]. Several sites of test are planned before integrating and exploiting on the GANIL site the whole of the facility. Q/A=1/3 primary beams are being tested in the LPSC in Grenoble from the beginning of this year. The deuteron source and the first part of the Low Energy Beam Transport lines will then be tested in the IRFU SACLAY site next year.

SPIRAL2 COMMAND CONTROL DEVELOPMENT

Development Layers

To provide a homogeneous control system (for future operation), it has been decided during the detail design study of SPIRAL2 to use EPICS for the whole accelerator. EPICS is recognized as a good toolkit to build efficiently a powerful and scalable control system. EPICS has got two main development levels: Input

* touchard@ganil.fr
# Half time student from Sorbonne University, Paris, France

Output Controllers (IOC) which interface the equipment to control with a network protocol named Channel Access (CA); EPICS Extensions and associated development which constitute a first part of IOC clients.

Although a feasibility study into using XAL [4] as a global framework is underway, it is acquired that high level tuning applications which are required to make such a facility like SPIRAL2 work, are written in Java.

These JAVA applications constitute the other part of the EPICS clients.

In order to link these two software development layers, it has been necessary to establish a convention to interface equipment. A study is being done to define the rules of access to the equipment, which eventually also takes into account the XAL handlers specificities.

A Unique EPICS Distribution

So that all the labs use the same working environment, it has been decided to use Linux PCs in RHEL5, VME crate in VxWorks 6.5, EPICS 3.14.9, and JAVA 5. To avoid spending too much time testing environments, we decided to make only a major update per year, at the most.

Table 1: Version Choices

Item	Currently used	Future
Epics	3.14.9	3.14.11 to be validated
Consoles & servers	RHEL5.2	no evolution foreseen yet
IOCs	VxWorks 6.5	VxWorks 6.7 to be validated
JAVA	5	6 to be validated

Naming Conventions

In the same state of mind, as shown in Figure 3, a convention to name the equipment has been established for the whole SPIRAL2 project and is used by the whole software development team. In the same way, EPICS software macros that make each piece of software generic, respect also naming rules.

Function *Domain-Marker-Component* *(17 character max)*		Signal *(10 character maximum)*
DDDDD-MMMMM[-CCCCCCC]	:	SssssSsssss

Figure 3: Equipment naming rules.

EPICS Development Skeleton

To make each piece of EPICS software developed compatible with each other, an EPICS directory tree model specific to SPIRAL2 and named "topSP2" has been elaborated. It allows to separate the modules from the IOCs. A module is an EPICS application associated to a type of equipment or function which will be instantiated in an IOC. Besides, an IOC is allocated to a SPIRAL2 accelerator section or function.

A new EPICS template files have been designed to allow each member of the development team to create his own topSP2 structure in his working directory. These templates associated to the makeBaseApp EPICS perl command could produce either a module or an IOC skeleton that needs to be filled and wherein the piece of software has to be realized.

This model and these conventions of developments permit to deploy any topSP2 piece of software on any test site without any modification or adaptation.

Software Development Workflows

The labs involved in SPIRAL2 command control software development, have to supply work packages which have been defined by the project leader. If macroscopic cutting out shown in Figure 1 allows an easier organization, some subtleties appear which may involve the participation of several laboratories on the same piece of software. It is actually the case for the Low Level Radio Frequency module which will be finalized by each lab in charge of the development of a cavity [5] [6].

It is thus necessary to be able to integrate simultaneously on the sites of tests, the applications and patches coming from various laboratories. During improvement of the machine and the pieces of software, the module corrections and transfers between the various sites can be therefore frequent. A study of the development deployment and correction rules is thus in progress to ease the ability to react while keeping a compatible organization with the complexity of the project.

For these purposes, a versioning software control system has been set up as shown in Figure 4. Subversion software, also named SVN, which is an open source project, has especially been designed for global projects developed by a distributed team.

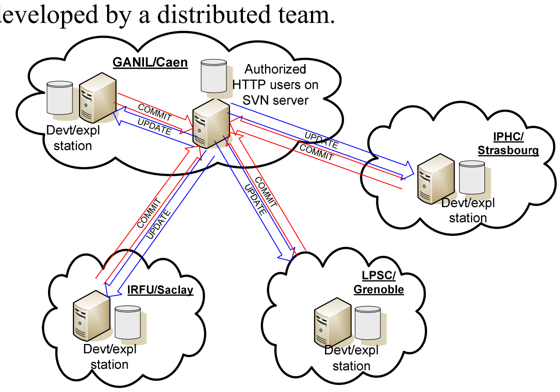

Figure 4: SVN workflow over development sites.

Each remote team can easily access to a SVN shared module or to a SVN shared IOC with an URL constructed as shown in red inside Figure 5 because of the use of a WEB server.

Likewise, the Integrated Development Environment named Eclipse and associated plug-ins are being evaluated to facilitate the implementation of development rules.

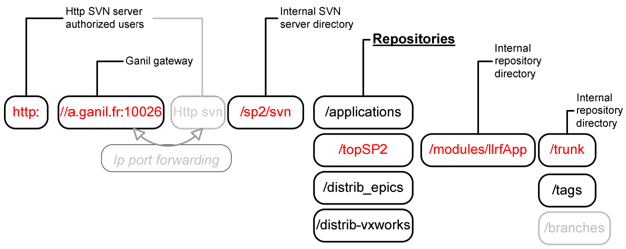

Figure 5: Simple SVN URL.

A similar solution (Eclipse and SVN) has been set for the java software management. Thanks to those tools, the XAL framework is under test to verify if it could offer keys for creating general purpose applications accessing equipment via standardized EPICS process variables.

SPIRAL2 COMMAND CONTROL EXPLOITATION

Equipment Characteristics inside a Relational Database

To describe and exploit the 4000 equipments which are envisaged for the control of the SPIRAL2 facility, a relational database and the management tools associated are developed at GANIL. The graphic interfaces written in JAVA, like the one in Figure 6 which verifies the equipment name consistency as described in "Naming convention" paragraph, allow users without any EPICS knowledge to directly handle equipment characteristics.

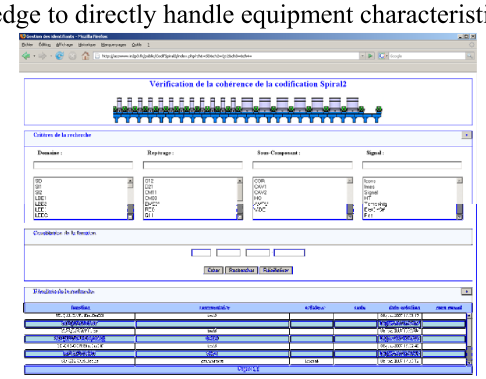

Figure 6: Name consistency application.

Automatic EPICS Database Generation for IOCs

The set of conventions and rules described above, the tools that have been deployed for development, and the equipment database, would permit to automatically generate EPICS flat files that define completely an IOC.

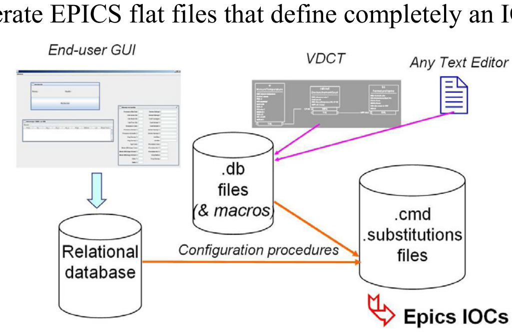

Figure 7: Epics databases generation work flow.

The Figure 7 illustrates the entire automatic process that shows with purple narrows the development part, with the oranges automatic configuration procedures and the blue end user interfaces for equipment handling.

THE LPSC TEST BENCH

As said previously, the Q/A=1/3 ECR multi beam source and the first lines coupled with the low energy beams, have been installed in the LPSC to realize the first tests of the SPIRAL2 project. The platform of command-control deployed for this purpose has been the first opportunity to test in life-size the result of the first developments made upstream. It turned out that the conventions and the organization have simplified and optimized the work in the point to make the system operational in very short time.

In the same way, first tests of the subversion solution have been made at that time and have shown that the repository model needs to be improved.

CONCLUSION AND PERSPECTIVES

Although a lot of work has still to be done, software deployment tests that have already been done show that these solutions improve the command control team efficiency. At this time, the definitive development rules are being studied and are being about to be achieved.

ACKNOWLEDGEMENTS

We would like to thank N. Menard (Ganil) who configured the SVN server with the collaboration of G. Lebertre (Ganil). He also specially configured the whole LPSC platform too, with the collaboration of P. Meyrand (LPSC) for the network configuration.

We would also like to thank F. Gougnaud (IRFU), J.-F. Denis (IRFU), C. Maazouzi (IPHC), J. Hosselet (IPHC), who are involved into the SPIRAL2 command control development and who have helped us to test some of the software tools and organization mentioned above.

REFERENCES

[1] http://pro.ganil-spiral2.eu/spiral2/what-is-spiral2
[2] T. Junquera "Status of the Construction of the Spiral2 Accelerator at Ganil" LINAC 2008, TU102.
[3] E. Lécorché et al. "First steps towards the new Spiral2 project control system" ICALEPCS 2007 TPPB07.
[4] www.ornl.gov/~t6p/Main/XAL.html
[5] M. Di Giacomo et al. "Status of the RF systems for the SPIRAL2 LINAC at the beginning of the construction phase" LINAC 2006 THP039
[6] P. Galdemard. "The development of the digital LLRF system for the SPIRAL2 accelerator" Spiral2 week 2009 spiral2ws.ganil.fr/2009/week/presentations/drivers/SP2wk09_Galdemard.pdf