USER INTERFACES FOR THE SPIRAL2 MACHINE PROTECTION

L. Philippe, P.Gillette, G.Normand, GANIL, Caen, France

Abstract

Spiral2 accelerator is designed to accelerate protons, deuterons, heavy ions with a power from hundreds of Watts to 200kW. Therefore, it is important to monitor and anticipate beam losses to maintain equipment integrities by triggering beam cuts when beam losses or equipment malfunctions are detected; the MPS (Machine Protection System) is in charge of this function. The MPS has also to monitor and limit activations but this part is not addressed here. Linked to the MPS, five human machine interfaces will be provided. The first, "MPS" lets operators and accelerator engineers monitor MPS states, alarms and tune some beam losses thresholds. The second "beam power rise" defines successive steps to reach the desired beam power. Then, "interlock" is a synoptic to control beam stops state and defaults; the "beam losses" one displays beam losses, currents and efficiencies along the accelerator. Finally, "beam structure" lets users interact with the timing system by controlling the temporal structure to obtain a specific duty cycle according to the beam power constraints. In this paper, we introduce these human machine interfaces, their interactions and the method used for software development.

THE SPIRAL2 FACILITY

The SPIRAL2 facility (Fig. 1) is based on a high power, superconducting linac driver, which will deliver a high intensity, 40 MeV deuteron beams as well as a variety of heavy-ion beams with mass-to-charge ratio of 3 and energy upon to 14.5 MeV/nucleon. A possibility of construction of a second injector for heavy-ions with a mass-to-charge ration of 6 is incorporated in the design. The main RIB production scheme of SPIRAL2 is based on the fast-neutron induced fission of uranium target. Using a carbon converter, a 5 mA deuteron beam and a high-density (up to 11g/cm^3) 2.3 kg uranium carbide target, the fission is expected to reach a rate up to 5*10^{13}/s. A direct irradiation of the UC_2 target with beams of protons or ^{3,4}He could also be used. The extracted 1+radioactives ions will be injected in the 1+/n+ charge breeder (ECR ion source) and post-accelerated by the existing CIME cyclotron.

The SPIRAL2 project is divided in two phases (Fig 1):

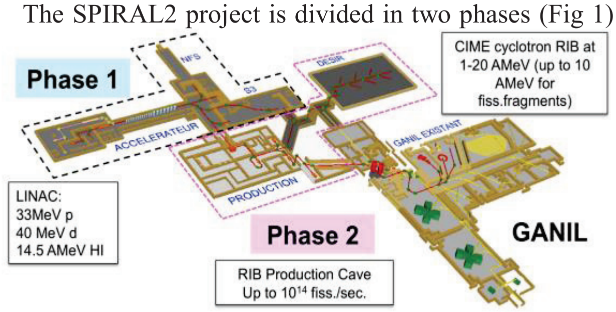

Figure 1: Accelerator layout.

MACHINE PROTECTION SYSTEM

Overview

The machine protection system is in charge of:

- Limiting heath increase resulting from the beam power inside the machine devices,
- Controlling the operating range of the accelerator,
- Minimizing radiations: This function isn't addressed in this paper because it will be implemented in a separate subsystem due to security constraints. For more information see reference [1].

System Architecture

Figure 2 below shows the machine protection system architecture:

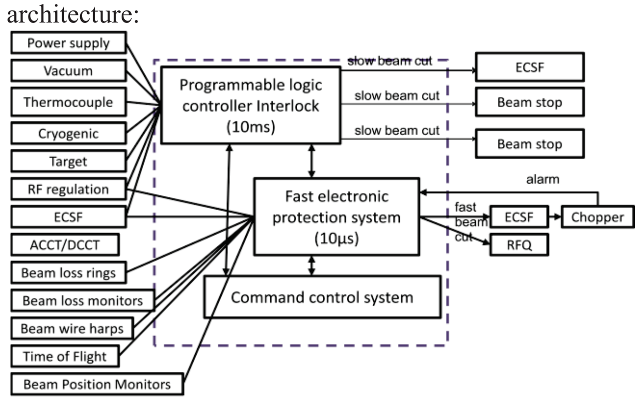

Figure 2: MPS architecture.

Many types of devices act upstream the system in order detect disjunction, vacuum losses and beam losses.

The MPS system reacts on the beam by decreasing the beam duty cycle (beam chopper and RFQ pulsations handled by the ECSF system) and by inserting an ad hoc beam stop.

MPS system is composed of three subsystems:

- PLC interlock triggers slow beam cut and manage accelerator mode.
- Fast electronic protection system triggers fast beam cut and warn PLC interlock that keeps safe the accelerator.
- Command control system provides human machine interface (HMI), operating alarms, archiving, threshold management.

As far as the command control is concerned, five HMI are designed to assure monitoring and control of MPS system. These softwares are presented in the next chapters.

MACHINE PROTECTION SYSTEM HMI

"MPS" HMI lets operators and accelerator engineers monitor MPS states, alarms and tune some beam losses thresholds.

ISBN 978-3-95450-139-7

Threshold Management

For each diagnostic used in the MPS context a threshold management subsystem will be implemented either inside the device's electronic or in a separate electronic card. When the threshold is exceeded an alarm is issued to the fast electronic protection system (Figure 2). The latter should decide to cut off beam.

- The functions about threshold management are listed under mentioned Compute threshold offline
- Apply threshold
- Modify some threshold inline
- Verify threshold

Figure 3 below shows an example of threshold display:

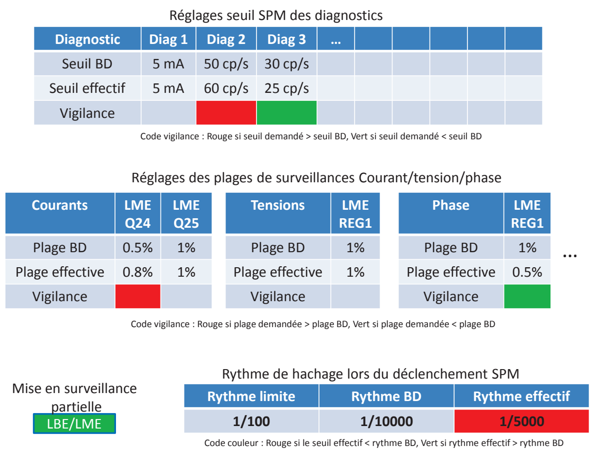

Figure 3: Threshold management part of MPS HMI.

MPS State

The visualization of the MPS state is composed of :

- A synoptic that displays the main components (beam stops, diagnostics) related to the MPS context (top of figure 4).
- A table that summarizes MPS alarms.

Operators can also from this interface change the duty cycle or put devices under supervision.

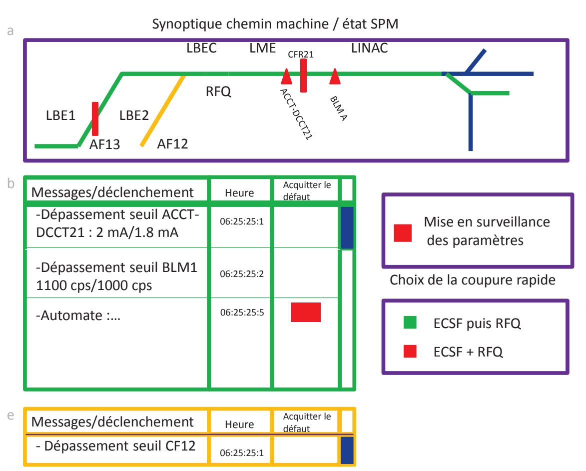

Figure 4: SPM State part of MPS HMI.

BEAM POWER RAISE

"Beam power raise" guides operators for raising progressively the beam power. Successive steps to reach the desired beam power are given by a 3D matrix. Checking of the machine status and conditions with the steps defined by this matrix is performed within the Interlock PLC. The three axes are the following:

- **Machine Path:** A path from the beam source to a beam stop. Currently 60 machine paths are defined. Each machine path if of type corresponding to the beam's progress level (source, injector, experiment rooms).
- **Beam Power mode:** Average beam power at the final beam stop. 300W, 1kW, 2kW, 6kW, 10kW, 50kW, 200kW.
- **Beam Type:** Deuterium, proton, H2+, He3 1+ He3 2+, He4 2+, Heavy Ion, No beam.

Depending on the beam type and desired destination, series of machine path are defined from the source in use. For each machine path, the beam power must be increased by step.

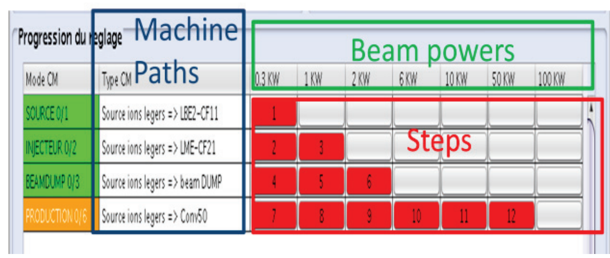

Figure 5: 3D Matrix representation in beam raise HMI.

Beam power raise is in charge of

- According to the beam to be produced, constructing the matrix and presenting it to operator (Figure 5).
- Asking for a step change in respect of the conditions defined in the Interlock PLC.
- Showing conditions to reach for each step and potential stumbling block. This information is provided by the PLC interlock system.
- Verifying parameters applied to the accelerator in coherence with the current step within the 3D matrix. Therefore, to provide this coherency, a specific communication between the Interlock PLC and the program has to be set.
- Backing-up accelerator parameters automatically at each line change (ie: change of machine path).

INTERLOCK

The "Interlock" HMI is a synoptic displaying the state of all beam stops and act on them. Four states are defined for a beam stop :

- IN
- OUT
- Indeterminate
- Radiological protection lock : inhibit the command by the control system

ISBN 978-3-95450-139-7

The figure bellow shows an example based on the GANIL control command's HMI interlock. For Spiral2, CSS Boy is envisioned for this application and is still to be developed.

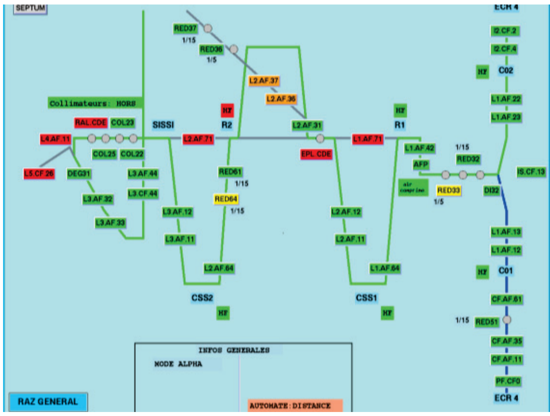

Figure 6: Interlock HMI for GANIL accelerator.

BEAM LOSSES

"Beam losses" HMI concerns three parts:

- **A synoptic** that displays beam losses (Figure 7).
- **A visualization of beam alignment and beam efficiency** (Figure 8).
- **A stream archiving and retrieval data module:** it will be assumed by the global CSS archiver and databrowser mechanisms with potential in-house report or view.

The display for beam losses will represent them by gauges with a color in function of delta between current value and threshold. Furthermore, "Beam losses" HMI should access threshold management.

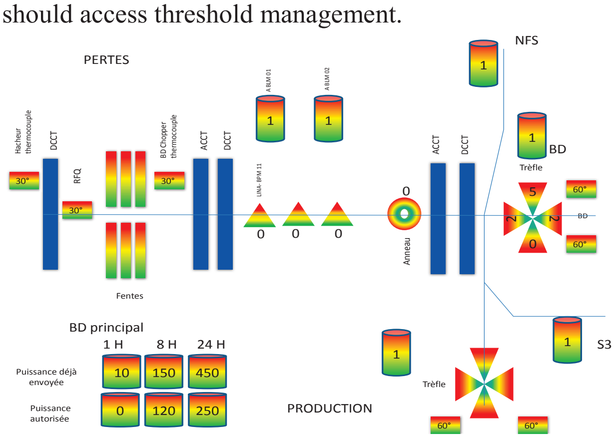

Figure 7: Beam losses view.

The synoptic should be frozen in case of MPS beam cut in order to graphically locate beam losses at the time of the MPS trig.

Another view within the application is shown on the following. It displays beam efficiency with gauges and alignment at different positions of the accelerator. Specific JAVA/XAL software allows operators to optimize the alignment.

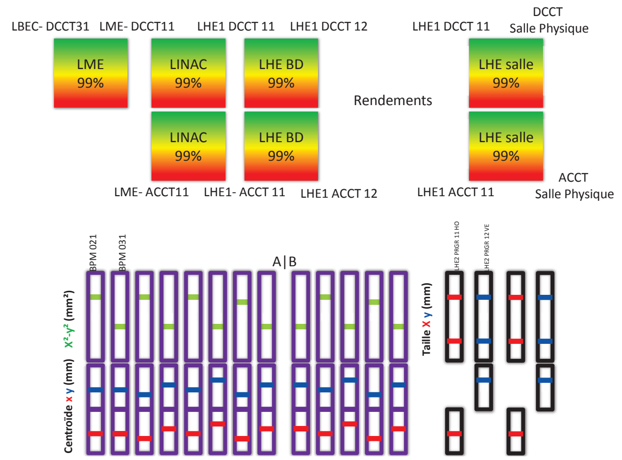

Figure 8 : Display of machine efficiency and beam alignment and size.

BEAM STRUCTURE

This application gathers all interactions with devices having an effect on the beam time structure: sources, choppers, RFQ.

Specifications include:

- Visualizing the duty cycle pulsation as well as the individual pulsations of each of the devices composing the final signal.
- Managing lists of duty cycle configurations: the configuration to get a specific duty cycle defines a set of values of each device acting on the final duty cycle (a given duty cycle can be obtained by different configurations based on the pulsations of the source, chopper and RFQ).
- The duty cycle offline preparation: For each beam raise step of the 3D matrix, the objective is to determine the duty cycle configuration to reach with which elements.
- Backing-up duty cycle applied automatically after each beam power raise.
- Backing-up duty cycle applied when a backup of the accelerator parameters is applied.
- Actions on the duty cycle are validated by the PLC interlock.
- Specifying the duty cycle to apply in case of MPS's beam cut.

METHOD AND TOOLS

Method

A functional specification was produced for each application.

From this specification we defined:

- Interactions between the application with other systems as well as with users. UML use case diagram are used.
- A HMI layout, designed from the model realized with end users

ISBN 978-3-95450-139-7

UML is a modeling language in the field of software engineering. ***Use case diagram*** is one of UML diagrams and represents users' interactions with the system. To create a UML use case diagrams we should first identify:

- Actors involved either human or other system.
- Main functions.

An important point at this step of conception is to identify clearly the border of the system to be provided. This border of the system and interactions with external systems allow defining interfaces between systems. Depending on the context, it could consider either the entire system (PLC, fast electronic and command control) or only the command control subsystem. So, we can then determine clearly which function is implemented in which subsystem. Once use cases are defined and validated by end users in order to make sure there is no function missing, we should prioritize the use cases.

The next step is to determine behavior of system, how actors interact with the system considered as black box? UML ***Sequence diagram*** is the best way to do this. UML sequence diagrams show interactions between objects along a time sequence.

In the mean time, we should define in addition to the model how the users navigate inside the application and how the users use the application during operation process. UML ***Activity diagram*** permit to apprehend this part of the system. UML activity diagram is a workflow representation with activities.

At this step, we have formalized requirements and should go from requirements abstraction level to system abstraction level.

To do this, we should define from use cases and mockup which classes are necessary. A class is a computer representation of a real object important for the system to implement. Then, we defined interactions between classes thanks to UML ***Class diagram***. At this step, UML class diagrams don't include any choice of technologies, frameworks, etc….

From class diagrams and black box sequence diagram, we can then define how the system implements functionalities, considering now the system like a white box. UML sequence diagram is also used at this point.

Finally, we add in class diagram choice of technologies, frameworks, etc… to implement the software.

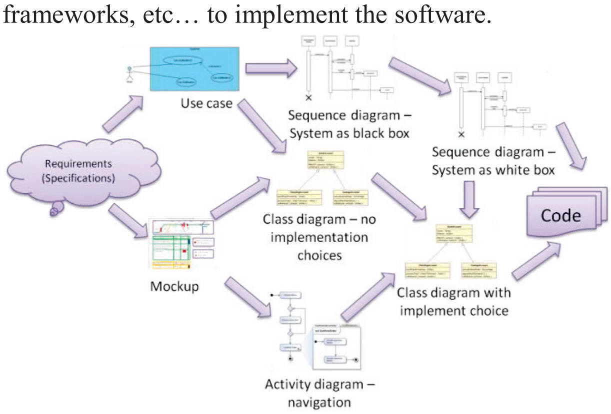

Figure 9: Explanatory diagram of the method.

At this step, it only remains to write the code!

This method is based on the method detailed in reference [2].

Tools

In order to improve efficiency, we use a case tools Visual paradigm [3]. Visual paradigm support UML 2.4 and ERD for database architecture. These case tools allow defining links between diagrams that help us to navigate between the steps of the methods. It also permits to generate JAVA code, database diagrams from UML class diagrams, database generation. Its usage is very helpful and let us concentrates to important aspects.

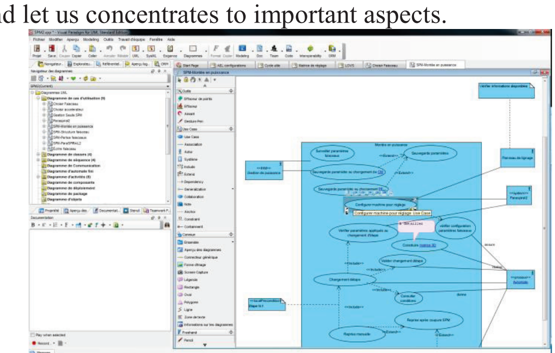

Figure 10: Visual paradigm Use case diagram.

CONCLUSION & NEXT STEPS

We are in the step of consolidating requirements. An important issue is to determine which subsystem should implement which function or part of function and to catch links and sequences between subsystems.

There is still a lot of work to do before implementing the system and many people are involved in this thematic so we must coordinate works besides the software code production itself.

The Spiral 2 control system progress is detailed in reference [4].

ACKNOWLEDGEMENTS

Other contributors are: D. Touchard (Epics environment and tools) and C. Haquin (real time) from the Ganil control group, C. Berthe from the Ganil facilities management group, C. Jamet and B .Ducoudret from Ganil electronic group.

REFERENCES

[1] C. Berthe et al, "Machine Protection System for the SPIRAL2 facility", MOPPC042, ICALEPCS 2013, to be published.

[2] Pascal Roques, "UML2 Modéliser une application web», EYROLLES (2008), ISBN 978-2-212-12389-

[3] Visual Paradigm for UML is a product of Visual Paradigm International Company, http://www.visual-paradigm.com/product/vpuml/

[4] C. Haquin et al, "The Spiral2 control system progress towards the commissioning PHASE", MOCOAAB03, ICALEPCS 2013, to be published.

ISBN 978-3-95450-139-7