THE USE OF SOFTWARE IN SAFETY CRITICAL INTERLOCK SYSTEMS OF THE LHC

Alejandro Castañeda, Iván Romera, Frederic Bernard, Pierre Dahlen, Benjamin Todd, David Willeman, Markus Zerlauth, CERN, Geneva, Switzerland

Abstract

This paper will provide an overview of the software development and management techniques applied to interlock systems in the CERN accelerator complex. Despite the hardware based approach, software and configuration data are present in various forms and have to be treated with special care when aiming at safe, reliable and available protection systems. Several techniques and methods deployed in the LHC machine protection systems are highlighted, regarding data management and version tracking, hardware choices, commissioning procedures, testing methods and first operational experiences with the systems in CERN's accelerator complex.

THE LHC INTERLOCK SYSTEMS

The energy stored in the LHC accelerator, both in the superconducting magnets and in the circulating beams is unprecedented and an uncontrolled release could lead to serious damage of equipment. During nominal beam operation at 7 TeV, each proton beam will have a stored energy of about 340 MJ. The energy stored in the magnet system amounts to around 10 GJ. Major damage of equipment inside the cryostats will result in long repair times, as the equipment is delicate and difficult to access.

The main objective for the machine protection system is to protect the machine in case of failure. Several systems are required in order to protect the complex equipment of the LHC accelerator. The machine interlocks are part of the protection and include several systems: the powering interlock system, the warm magnet interlock system, the beam interlock system, the fast magnet current change monitors and the safe machine parameter system.

Machine protection systems should contribute to avoid causing any damage of equipment as well as to maximize operational availability by minimising time for interventions [1].

TECHNIQUES AND METHODS DEPLOYED IN THE LHC

Software is widely used in the interlock systems of the LHC, not only to execute the desired interlock functionalities over different devices but also for testing and monitoring purposes as well as to manage and to keep a close track of all its components. Thus, depending on its functionality it can be classified in four main branches: safety critical software, supervision software, software used during testing and management software. Because the goals of these groups are completely different, the way in which software is programmed, run and tested is radically different for each of them.

Before describing in more detail each of these categories it has to be mentioned that they all share a common origin which is the LHC Functional Layout Database [2]. This database is a complete description of the machine and powering layout which has been realised due to the complexity of the LHC and the need for coherent data during installation, commissioning and operation of the machine.

Any configuration source code needed by any of the applications or programs of the interlock system is created directly from this unique source which ensures coherency and correctness of process data in the different components [3]. Figure 1 visualises the data flow for the generation and deployment of configuration data.

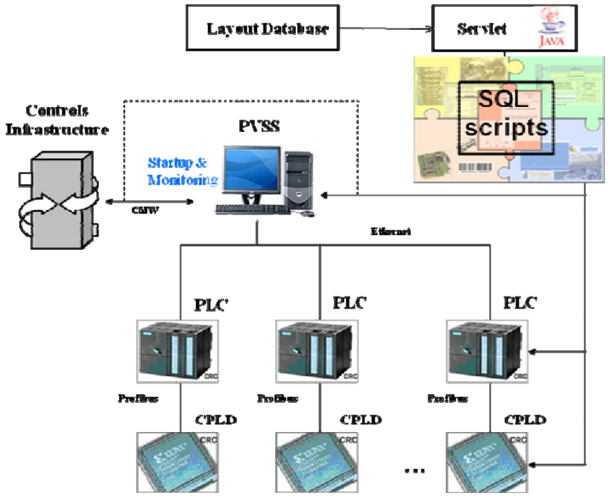


Figure 1: Generation and deployment of configuration data.

Safety Critical Software

This is software conceived to carry out the critical functionality of the interlock system at the lowest level. It runs in a fail-safe system which in the event of any kind of malfunction sets all of its outputs to the predefined fail-safe state. Time response, reliability, maintenance and cost are the most important parameters when designing this kind of software and hardware.

Deterministic time response is decisive when defining the limit between using hardware or software based solutions. With the current technology, low-level software works well in the order of milliseconds. Thus a PLC solution has been chosen for the Powering Interlock Controller (PIC) and the Warm magnet Interlock Controller (WIC).

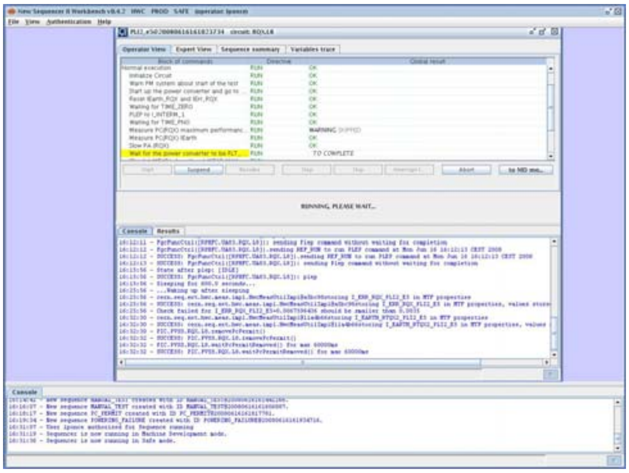
For security it is important to keep things as simple as possible thus for the PIC and the WIC the PLC runs in a continuous loop, it scans its inputs, executes the user program and writes its outputs. Some of the tasks inside the user program have a higher priority than others. Tasks directly related to the interlock functionality or the history logging have highest priority. Other tasks such as communication with the supervision have a lower priority. To achieve this, the PLC calls the functions from two separate routines: there is a defined loop time routine which carries out the critical processes allowing for real time operation because its execution is deterministic within a defined time frame. Finally there is a free running block of code that executes as fast as it can, carrying out the lowest priority tasks [4].

Critical and non critical parts of the program are kept isolated from each other in different source codes, which diminishes risks and facilitates maintenance tasks in the future, e.g.: if some aspect of the non critical part needs to be upgraded it can be done without perturbing the critical part of the program (the system must still retested, this separation of code greatly reduces regression testing time).

Testing Software

This class comprises the code and applications written to test and validate the systems. Any interlock system has to be 100% tested and validated, meaning that any possible combination of events that might occur has to be checked to be 100% sure that the system is going to react as expected before being declared operational.

During the individual system test phase of each equipment, these programs run in dedicated test benches that simulate all possible signals connected to the interlock system. Once the validated system has been installed in place and connected to the various client systems specific tools are employed to test all systems together, the so-called "LHC Sequencer", see Figure 2.

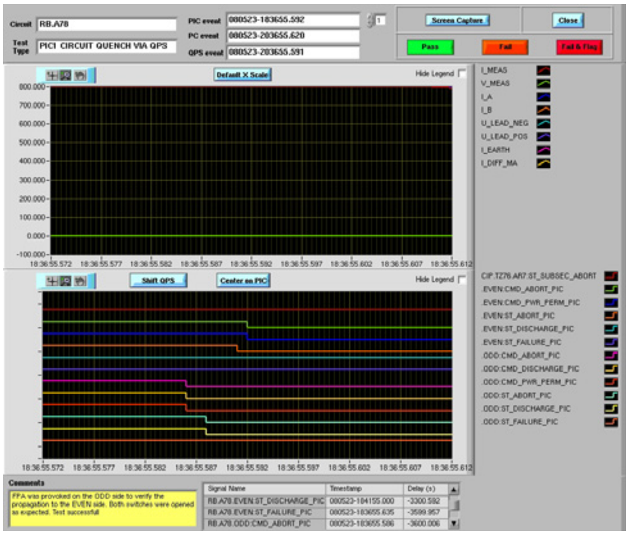


Block of commands	Directive	Global result
Normal execution	RUN	OK
Initialize Circuit	RUN	OK
Warn PM system about start of the test	RUN	OK
Start up the power converter and go to ...	RUN	OK
Reset IEarth_RQX and IErr_RQX	RUN	OK
Waiting for TIME_ZERO	RUN	OK
PLEP to I_INTERM_1	RUN	OK
Waiting for TIME_PNO	RUN	OK
Measure PC(RQX) maximum performanc...	RUN	WARNING SKIPPED
Measure PC(RQX) IEarth	RUN	OK
Slow PA (RQX)	RUN	OK
Wait for the power converter to be FLT...	RUN	TO COMPLETE

RUNNING, PLEASE WAIT...

Figure 2: LHC Sequencer application

The LHC Sequencer allows for simultaneous equipment access and execution of the established test procedures for each of the systems. The results of the tests are then analyzed by the different equipment experts by using the "Post Mortem" tools, as shown in Figure 3.



Circuit RB.A78
Test Type PIC1 CIRCUIT QUENCH VIA QPS
PIC event 080523-183655.592
PC event 080523-203655.620
QPS event 080523-203655.591

Comments
FPA was provoked on the ODD side to verify the propagation to the EVEN side. Both switches were opened as expected. Test successfull

Signal Name	Timestamp	Delay (s)
RB.A78.EVEN:ST_DISCHARGE_PIC	080523-184155.000	-3300.592
RB.A78.EVEN:ST_FAILURE_PIC	080523-183655.635	-3599.957
RB.A78.ODD:CMD_ABORT_PIC	080523-183655.586	-3600.006

Figure 3: Post Mortem application

The Post Mortem system gathers all transient data generated in the equipment systems by each test launched from the LHC Sequencer and presents it in a predefined way to facilitate the analysis task of the expert. It is also responsible for collecting the validation signatures of each expert thus finally passing or failing a test. Both tools communicate between each other through a dedicated Front End Software Architecture (FESA) class and the LHC Software Architecture database (LSA), defining the necessary tests for each circuit, its status and its history.

Supervision Software

This is the software deployed to constantly monitor the status of the interlock system. It is important to highlight that these applications are not needed for any of the critical protection functionality to be executed by the interlock systems. Even if the supervision completely stops, still all systems and processes on the critical lowest layer will still keep assuring the equipment protection.

The SCADA system used for the interlocks is PVSS, its server, based on the UNICOS Framework, synchronizes the requests coming from the LHC Sequencer or from the supervision.

The SCADA system plays an important role during the commissioning phase since it allows certain interlock parameters to be masked without modification of PLC source code. Furthermore it provides additional security mechanisms by verifying the coherence of all configuration files in every interlock system

Data Management Software

These are tools used to manage the interlock project allowing tasks such as: issue tracking, asset management, repository database, project status, test record, task assignment, book logging, modification history, etc.

These software applications do not perform tasks directly related to the interlock function itself, they are in any case crucial to keep things in order in the complexity of the interlock system. The interlock system has a large number of parts that have to be interconnected, tested, revised, repaired and upgraded. Thus a tool to keep a close follow-up of all of this is an absolute necessity.

For this purpose, a tailor made application has been designed, the "PIC Manager". It provides all the capabilities mentioned before in a user friendly web interface, allowing anyone involved in the project to access directly from any computer. This means: flexibility, accessibility, quick look ups, quick reporting and up-to-date data as shown in Figure 4.

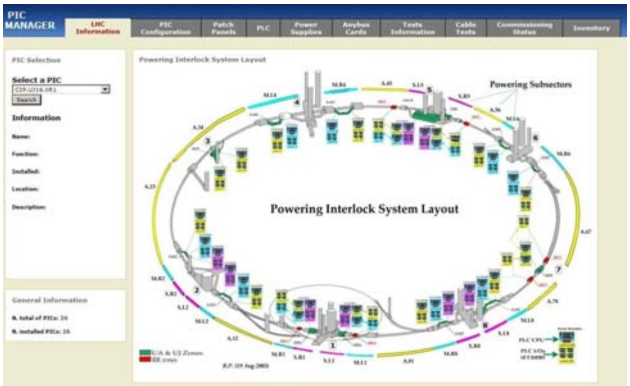


Figure 4: PIC Manager application

In addition, a strict versioning any program is carried out with CVS (Concurrent versions system) which is a standard tool widely used at CERN. CVS allows all the capabilities needed such as version tracking, file comparison, modification history and change traceability.

FIRST OPERATIONAL EXPERIENCES IN THE LHC

Commissioning Phase

The commissioning of the machine in the presence of such large interlock systems requires powerful tools for testing and diagnostic purposes to trace interlock related issues.

A set of high-level software applications has been developed for automating and validating the interlocks following the commissioning procedures. The LHC Sequencer is the tool in charge of orchestrating a variety of tests in an automatic way.

Experience during the different hardware commissioning phases of the LHC has proven that, in many cases it is necessary to mask a certain number of interlocks in order to carry on testing when not all final systems are yet installed or ready for testing. For the time being tracking of these changes have been followed up in a manual way, e.g.: meetings, e-mail exchange, parameters displayed at supervision level, etc. Something which can be coped with only when there are not many changes, but which will quickly become unacceptable for parallel commissioning. For this reason it is necessary to develop tools that systematically keep track of masks in an efficient and reliable way including information such as: interlock masked/unmasked, date, reason for the change, systems affected and what can/cannot be done with the current status of the interlock. This could be managed by an application able to compare the current status of each interlock in the machine with its reference value broadcasting a warning in case of mismatch.

FUTURE REQUIREMENTS

Tracking of Interlock Conditions

To maintain the high level of security, none of the interlock conditions is to be changed during operation without repeating the commissioning procedures for this system.

Before starting the machine after a shut-down every interlock system should be (at least partially) re-commissioned by using the LHC Sequencer.

In order to manage changes of the interlock conditions during operational period, the reference settings should be set. In case of an interlock change, a new nominal setting will be established and compared with the reference setting. In the future, the entire interlock tree should be managed on this way.

CONCLUSIONS

Experience has proven that the use of software in safety critical systems of the LHC has to be handled with great care as it is of vital importance in any aspect: safety, supervision, testing and management. Special attention has to be paid in case of any modification of the interlock conditions, by tracking and re-commissioning of the systems affected by the changes. Its correct functionality has to be 100% guaranteed.

Still some efforts have to be made to improve the tracking facilities and management of the interlock parameters modifications by using automated tools.

REFERENCES

[1] F.Bordry et al., "Machine Protection for the LHC: Architecture of the Beam and Powering Interlock Systems", Project Report 521, December 2001.

[2] http://layout.web.cern.ch

[3] J. Mariethoz, F.Bernard, R.Harrison, P.Le Roux, M.Peryt, M.Zerlauth, "The importance of Layout and Configuration data for Flexibility during Commissioning and Operation of the LHC Machine Protection Systems", EPAC'06, Edinburgh, Scotland.

[4] R. Harrison, "Powering Interlock Controller Software Functionality", AB/CO/MI functional specification, Geneva, CERN 2007.