# OPERATIONAL EXPERIENCE WITH THE LHC SOFTWARE INTERLOCK SYSTEM

L. Ponce, J. Wenninger, J. Wozniak, CERN, Geneva, Switzerland

## *Abstract*

The Software Interlock System (SIS) is a JAVA software project developed for the CERN accelerators complex. The core functionality of SIS is to provide a framework to program high level interlocks based on the surveillance of a large number of accelerator device parameters. The interlock results are exported to trigger beam dumps, inhibit beam transfers or abort the main magnets powering. Since its deployment in 2008, the LHC SIS has demonstrated that it is a reliable solution for complex interlocks involving multiple or distributed systems and when quick solutions for un-expected situations is needed. This paper is presenting the operational experience with software interlocking in the LHC machine, reporting on the overall performance and flexibility of the SIS, mentioning the risks when SW interlocks are used to patch missing functionality for personal safety or machine protection.

## INTRODUCTION

As the stored energy in the Large Hadron Collider (LHC) is orders of magnitude above damage level of many accelerator components, safe operation of the accelerator requires highly reliable interlocking of dangerous situations or equipment failures. The core of the LHC interlock system is the Beam Interlocks System (BIS) that is entirely implemented in hardware and designed to inhibit injection or dump the beams with extremely high safety and availability requirements. As a complement of the BIS, the Software Interlock System (SIS) provides further protection by surveying and analyzing the state of various key equipment. Its open architecture allows for fast and easy configuration of more complex logic which allows to anticipate failure rather than reacting to them. It is in particular possible to define complex interlocks that correlate the state of many different systems and that are difficult to implement as hardware interlocks. The system has been designed to be highly reliable and the software base provides the flexibility for an easy reconfiguration of the logic to respond to the changing needs of the LHC operation.

## SIS ARCHITECTURE

The SIS has already been presented in details in [1] and we will summarize shortly the key features in this paper.

The central concept of SIS are boolean expressions represented as trees. A root node of each tree is called a ”software permit”. The permit usually corresponds to a boolean signal allowing injection in the LHC or disallowing beam circulation. The leaves of the trees are channels connected directly to the equipment status values and the intermediate nodes represent some logical expressions over those individual channels. The values are usually compared to predefined thresholds or ranges that are known as ”good” references guaranteeing correct functioning of the accelerator equipment. The typical tree is calculated periodically, triggered by a repetitive, external event. The outcome of the calculation (a boolean TRUE or FALSE value) is used as an input for predefined actions (exports), typically to either cut or enable the beam.

The SIS has a layered architecture which reflects the two major tasks of the system: Data Acquisition and Data processing, Fig. 1. The first layer deals with data subscriptions providing values used later for the tree calculation. Those values are stored in an internal buffer. The Calculation layer holds the definition of the trees and it is activated upon the tree calculation event, taking already prepared values from the internal buffer. One main architectural goal was to make the analysis part as reliable as possible and thus as independent as possible of the data acquisition part.

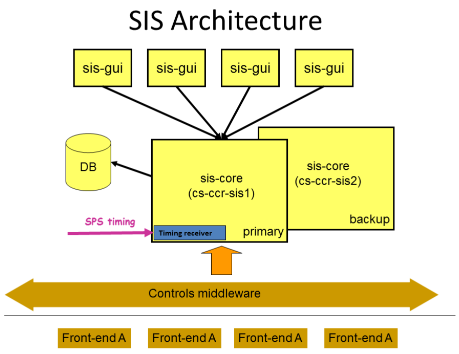
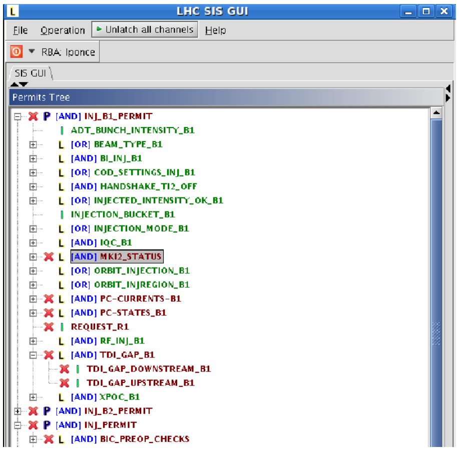


Figure 1: Schematic view of the SIS Architecture

The configuration of both layers is typically done in XML where parts of this configuration can be generated by a Velocity scripting engine [2]. To allow for easy and fast configuration, some of the logic like triggering events or nodes calculation can also be done directly in XML with the use of Groovy scripting language. Export actions and more complicated logic for data transformation are usually stored as Java classes. Extension points allow developers to provide their own implementation of components by leveraging the Spring Framework [3].

SIS is written in Java using modern JEE technologies like Spring, RMI, JMS, XML, Velocity and Groovy. As the SIS is a server side application, a Swing GUI was developed to show the system state to the operators in the control room. All permit trees are visible and dynamically updated; channel states are expressed with colours and markers, Fig. 2. Basic functionalities have been implemented for the GUI and will be extended for next run in order to improve the interaction with operators. The GUI also provides an interface for some user actions and a sophisticated analysis mechanism capable of identifying several typical fault scenarios, such as a missing data or a data with incorrect values.



```
⊟ ✖ P [AND] INJ_B1_PERMIT
        | ADT_BUNCH_INTENSITY_B1
    ⊞   L [OR] BEAM_TYPE_B1
    ⊞   L [AND] BI_INJ_B1
    ⊞   L [OR] COD_SETTINGS_INJ_B1
    ⊞   L [AND] HANDSHAKE_TI2_OFF
    ⊞   L [OR] INJECTED_INTENSITY_OK_B1
        | INJECTION_BUCKET_B1
    ⊞   L [OR] INJECTION_MODE_B1
    ⊞   L [AND] IQC_B1
    ⊞ ✖ L [AND] MKI2_STATUS
    ⊞   L [OR] ORBIT_INJECTION_B1
    ⊞   L [OR] ORBIT_INJREGION_B1
    ⊞ ✖ L [AND] PC-CURRENTS-B1
    ⊞ ✖ L [AND] PC-STATES_B1
      ✖ | REQUEST_R1
    ⊞   L [AND] RF_INJ_B1
    ⊟ ✖ L [AND] TDI_GAP_B1
          ✖ | TDI_GAP_DOWNSTREAM_B1
          ✖ | TDI_GAP_UPSTREAM_B1
    ⊞   L [AND] XPOC_B1
⊞ ✖ P [AND] INJ_B2_PERMIT
⊟ ✖ P [AND] INJ_PERMIT
    ⊞ ✖ L [AND] BIC_PREOP_CHECKS
```

Figure 2: SIS GUI: the top permit like INJ_B1_PERMIT, INJ_B2_PERMIT are marked with a P; the tree can be expanded to see the channels.

## THE LHC PERMIT STRUCTURE

At a fundamental level a Individual Software Interlock Channel (ISIC) is associated to a reading of a state, value or property of a system. The acquired parameter is analyzed (tested) and converted into a logical state TRUE or FALSE. The logical states are then grouped into a tree-like structure and combined using logical operators (AND, OR, IN RANGE,...) into intermediate nodes called Logical Software Interlock Channel (LSIC). The top of the tree corresponds to a software Permit which itself can be TRUE or FALSE. The TRUE status allow injection or circulation of the beams in the LHC, a FALSE status result into a inhibit of injection or a beam abort. The initial configuration of LHC SIS, using mainly AND and OR hard-coded logic has been extended to allow more and more complicated Interlocks written in JAVA extension pulling together multiple signals and database references. To perform its job during last run, the LHC SIS was handling 2665 device/parameter subscriptions representing some 5500 checks grouped into 7 permits:

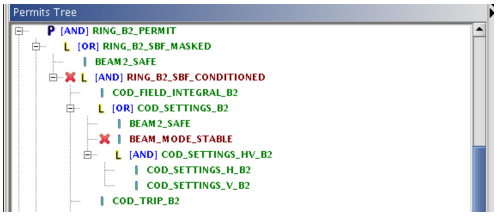
- Injection permits (beam 1, beam 2 or both beams) exported to the Beam Interlock Controller to inhibit the beam .
- Ring permits (beam 1, beam 2 or both beams) exported to the Beam Interlock Controller to inhibit dump the beam .
- Powering permit exported to the Powering Interlock Controller to abort the magnet powering

All interlocks trees are evaluated every 2 second for the LHC but can be faster if needed (in the order of 10-100 ms for the injectors SIS).

### *Interlock Masking*

Masking is a mechanism that allows operators to ignore an individual ISIC or LSIC. Masking a channel means overriding its real state and evaluating it always to TRUE. The ability to mask a given ISIC/LSIC is defined for each channel individually and the Permit signals are not allowed to be masked. The masking itself is done from the SIS GUI by operators. A role based access control framework is used to define masking rights, but the masking rights apply to all channels defined as maskable. When applied, a mask is always active, independent of beam conditions.

Therefore another way of “masking” interlocks automatically for a given period within the beam cycle or a given energy/intensity range is largely used in the SIS via the OR logic. A beam intensity, beam mode (describing the time in the beam cycle) or beam energy test is added as a ISIC with a OR logic to the channel that should be masked, see for example Figure 3. As soon as the intensity, energy or mode condition is TRUE, the interlock is de-facto masked.



```
⊟   P [AND] RING_B2_PERMIT
    ⊟   L [OR] RING_B2_SBF_MASKED
            | BEAM2_SAFE
        ⊟ ✖ L [AND] RING_B2_SBF_CONDITIONED
                | COD_FIELD_INTEGRAL_B2
            ⊟   L [OR] COD_SETTINGS_B2
                    | BEAM2_SAFE
                  ✖ | BEAM_MODE_STABLE
                ⊟   L [AND] COD_SETTINGS_HV_B2
                        | COD_SETTINGS_H_B2
                        | COD_SETTINGS_V_B2
                | COD_TRIP_B2
```

Figure 3: Example of masking using the beam intensity: the 60A power converter settings LSIC is combined with a OR logic with the BEAM_SAFE ISIC (intensity) and the BEAM_MODE_STABLE ISIC (collision period in the beam cycle)

### *Injection Permits*

The LHC injection Permits are connected to the Super Proton Synchrotron (SPS) Beam Interlock System in order to prevent injection into the LHC. In case one system is not ready or not in a nominal configuration for injection, an inhibit is sent to the the injector complex to prevent extraction or even production of the beams in the the injectors allowing a more efficient operation of the complex. As loosing one injection in the machine is not so critical for the operational efficiency, the interlocking policy can be very strict and with a large number of checks. Started mainly with checks of statuses and values in range for different equipment (Power Converters, Quench Protection, RF...) in 2010, the injection interlocks have been extended to more operational settings checks using the possibility to combine parameters published by different systems. Most of these additional interlocks were implemented to improve the machine protection level following some initially unforeseen operational conditions. One example is related to the very large bunch intensity range that has been used in the LHC during the 2012 operation: some component configurations, like the transverse feedback system, depend on the peak bunch intensity and may be damaged if the injected beam intensity is higher than the pre-configured one. No hardware interlock is available on the extracted bunch intensity but this information is available in time before injection from the injector, so a software interlock comparing the actual feedback settings and the extracted intensity in the SPS has been added as part of the injection permit.

### *Ring Permit*

The Ring permit value is exported to the LHC BIS to trigger a beam dump (either per beam or both beams) in case the evaluation result is FALSE. Taking into account the lengthy injection process and beam cycle in the LHC, dumping a circulating beam is really costly in term of efficiency. Thus the reliability of ring permit should be as high as for the BIS system. The number of ring interlocks is more limited than for injection permits. Interlocks on the orbit feedback system (to check the presence of the correct reference orbit) and a continuous monitoring of certain power converter currents have been added in the middle of the 2012 run following some critical situations. Although the beam loss monitoring system that is connected to the BIS would have eventually dumped the beam, it was decided to add SIS checks to trigger a preventive dump in those situations.

Another important ring interlock is the Orbit and Correctors Orbit Dipole (CODs) interlocking. The principle is to limit the global orbit excursions of the beams to prevent beam losses and catch un-detected orbit bumps by comparing the settings of each COD and the reading of each Beam Position Monitor (BPM) with a reference and a tolerance stored in the LHC settings database (LSA) [4]. The tolerances are defined as a balance between machine protection and availability and have been set so far quite strict at injection and during the collision period, but more relaxed during the ramp and squeeze process. SIS configuration allow to condition the reference with a beam mode or an energy with the AND/OR logic and also to read dynamically the reference from the database settings with a pre-defined periodicity.

### *Powering Permit*

In 2009 the access rules in the LHC machine during main magnet powering had to be changed to protect people in case of massive Helium release in the tunnel. An access matrix [5] describing the zones which should be patrolled and empty for each of the 8 powering sub-sectors was defined. The link between the powering system and the access system was not foreseen in the hardware interlock systems, so the SIS was chosen to implement this personal protection logic to complement an operational procedure.

For each sector that may be powered independently the SIS monitors the state of the access and of the currents in the power converters. In case the currents of some converters exceed predefined limits and the access conditions are not safe, the SIS triggers an interlock to de-power the entire sector by sending a signal to Powering Interlock System.

Initially used during the powering test campaign where access is on-going in part of the machine in parallel of the tests in another part, the interlock was implemented as a separate tree, not related to beam operation. The SIS solution proved to be very reliable and it was then decided to keep it active during beam operation even though the whole LHC machine is empty when beam is circulating. Since the signal from the LHC Access Control System (LACS) was not guaranteed during beam operation (as not needed), some unnecessary powering aborts were triggered initially. To prevent spurious false triggers the initial interlock has been combined through a OR logic with the highly reliable signal indicating the absence of persons in the entire interlocked area of the LHC. This LASS signal is connected directly to the BIS and can be monitored by SIS. No false interlocking was observed from then on.

It was initially planned to re-implement this powering software interlock using a purely hardware solution by including it in the LASS logic during the current shutdown. But thanks to the high reliability of the SIS during the first years of operation, it was finally decided to only consolidate the signal transmission from the LASS to the SIS and to continue using the SIS for the logic and for the export to the Powering Interlock System.

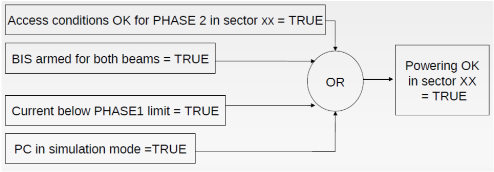

Access conditions OK for PHASE 2 in sector xx = TRUE
BIS armed for both beams = TRUE
Current below PHASE1 limit = TRUE
PC in simulation mode =TRUE
OR
Powering OK in sector XX = TRUE

Figure 4: Logic of the powering permit combining the signals coming from the access system, the BIS and the power converter current level.

## SIS PERFORMANCE AND AVAILABILITY

The LHC SIS core runs on dedicate HP server equipped with a timing card. Since the beginning of the operation in 2008 for the SPS and LHC SIS instances, only few crashes of the SPS server were observed during the 2009-2010 maintenance period with no beam operation. The problem was traced back to a concurrency problem in the timing library and was quickly fixed.

For the LHC machine, any time the beam is aborted a post mortem file is produced tracing the root cause (first trigger) of the beam dump. Extracting the data of the post mortem database for the 2012 operation period, 77 dumps are flagged to be due to LHC SIS as first input to the BIS. All events are real interlocking conditions, see Table 1, i.e. one of the ring permit changed from TRUE to FALSE status: none of the dumps are due to SIS failures, the programmed logic was always followed.

ISBN 978-3-95450-139-7

Table 1: Interlocks Channels Leading to Dump

| SIS DUMP cause | Ratio |
|---|---|
| Communication problem | 20% |
| Orbit feedback issues | 20% |
| Power converter faults | 15% |
| Beam position measurements | 10% |
| Beam Loss monitors HV | 10% |
| Others (wrong settings, masks) | 25% |

During 2012 operation, several dumps were caused by a stop of the data streams. The most affected data source was the Power Converters Function Generator Controller publication which was not received by the SIS data acquisition part for several minutes on some occasions. In such a case, the tree is evaluated to false if the data are not updated before a pre-defined time-out to avoid being blind with beams in the machine. The programmed logic was correctly followed. The problem was traced back to a problem within the middleware communication protocol used at CERN which was not protected against "slow clients".

## CONCLUSION

LHC SIS is successfully used in operation since 2008. It is a reliable solution for injection interlocks (when high reliability is less critical), complex interlocks involving multiple systems or distributed systems (like orbit) and as a fast answer to un-expected situation like feedback problems. It is all software, soft real-time, but the reliability, even if safety will never be SIL3 level, is remarkably high. During the long shut-down period some of the software interlocks will be moved to hardware interlock, but there will be more software tests coming.

## REFERENCES

[1] J. Wozniak et al., "Software Interlock System", ICALEPCS'07, Knoxville, October 2007, WPPB03, p. 403.

[2] `http://velocity.apache.org`

[3] `http://www.springframework.org`

[4] D. Jacquet et al., "LSA – the High Level Application Software of the LHC – and Its Performance During the First Three Years of Operation", ICALEPCS'13, San Francisco 2013, THPPC058.

[5] M. Gruwe et al.,"Access restrictions in LHC and SPS during LHC powering Phase II", LHC Project Document LHC-OP-OSP-0016, EDMS Number 1010617.

ISBN 978-3-95450-139-7