

# SUPERVISION SOFTWARE FOR THE INTEGRATION OF THE BEAM INTERLOCK SYSTEM WITH THE CERN ACCELERATOR COMPLEX

M. Audrain, D. Anderson, M. Dragu, K. Fuchsberger, J.C. Garnier, A.A. Gorzawski, M. Koza, K. Krol, A. Moscatelli, B. Puccio, K. Stamos, M. Zerlauth, CERN, Geneva, Switzerland

## Abstract

The Accelerator complex at the European Organisation for Nuclear Research (CERN) is composed of many systems which are required to function in a valid state to ensure safe beam operation. One key component of machine protection, the Beam Interlock System (BIS), was designed to interface critical systems around the accelerator chain, provide fast and reliable transmission of beam dump requests and trigger beam extraction in case of malfunctioning of equipment systems or beam losses. Numerous upgrades of accelerator and controls components during the Long Shutdown 1 (LS1) are followed by subsequent software updates that need to be thoroughly validated before the restart of beam operation in 2015. In parallel, the ongoing deployments of the BIS hardware in the PS booster (PSB) and the future LINAC4 give rise to new requirements for the related controls and monitoring software due to their fast cycle times. This paper describes the current status and ongoing work as well as the long-term vision for the integration of the Beam Interlock System software into the operational environment.

## INTRODUCTION

Operating a complex machine like the LHC requires handling failure modes in a safe and dependable way. To satisfy the need for machine protection and prevent the resulting damage an energetic beam can cause to the accelerators, the Beam Interlock System was defined as an interface between equipment systems and systems responsible to protect the machine during beam operation. To operate and monitor the BIS, a set of supervision and controls software has been provided since the deployment of this protection system in the SPS, the transfer lines towards LHC and the LHC itself.

Further deployment of the BIS since 2012 in the LHC injectors for the new LINAC4 and the PSB accelerator re-defined the need for a more generic monitoring solution of the BIS due to the fast machine cycles and multiple beam destinations. Moreover the LS1 (2013-2014) provides a unique opportunity to unify and consolidate the current set of supervision software for the BIS towards a unique and modular application for all accelerators.

This paper presents first the architecture of the BIS supervision software; secondly, the generic solution provided to monitor cyclic machines; finally, the generic BIS supervision software and the set of BIS operational checks.

## ARCHITECTURE

The hardware of the Beam Interlock System holds four different board types to ensure the protection of the accelerators. Those boards are interfaced to the CERN technical

network through the Front-End computer (FEC). More details on the complete architecture and principles of the Beam Interlock System are described in [1].

To allow machine operators and BIS experts to monitor and control the BIS boards remotely, one can distinguish four software layers as seen on Fig. 1.

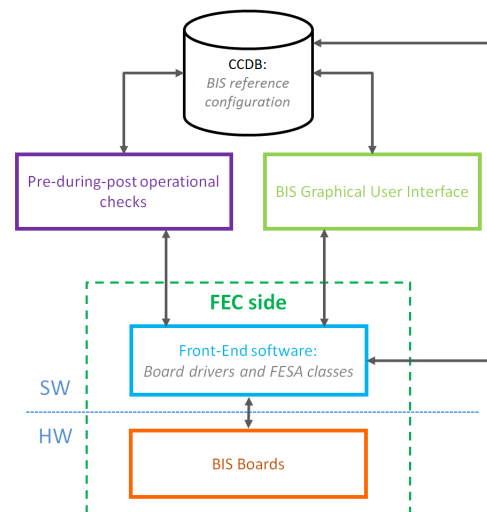


Figure 1: The architecture of the BIS software.

Those layers can be organized from bottom to top as described hereafter:

- *Front-End software*: concerns the low level software to monitor and operate the BIS boards from outside the FEC. It is composed of the board drivers and the Front-End Software Architecture [2] (FESA) classes. The BIS software developer team judged that Java is more maintainable and reusable. Therefore, the team chose to keep this front-end software layer as simple and generic as possible to export the specificities of all board types and data processing to the Java layers.
- *Graphical user interfaces (GUI)*: defines the main user-oriented entry point to monitor and operate the BIS. It is used both by machine operators and BIS hardware experts. As detailed in the two following sections, emphasis is made on creating a central and generic BIS application, adapted to all accelerator specificities.
- *Pre-Current-Post operational checks*: provides additional functionalities to assert that the BIS is functioning correctly for operation. Further details are provided in section "BIS in operation".
- *Controls Configuration Database*: provides reference configuration used by the BIS operational checks and Java applications.

A design for the deployment of the BIS in cyclic machines was required as well as unifying this design in a generic application. The next two sections will present the details of implementation satisfying those two needs.

## CYCLIC MACHINES SUPPORT

The BIS deployment in 2012 in the LINAC4 and PS booster (PSB) required a new mechanism to collect and display the BIS interlocking information.

The retrieval of BIS information in the LHC is done as an asynchronous 1Hz subscription to the FESA classes from the supervision software which is frequent enough since a usual LHC beam fill can last several hours. However, in the case of the LINAC4, the beam cycle is 1.2 seconds long with the beam pulse extracted from the source only lasting 400  $\mu$ s [3].

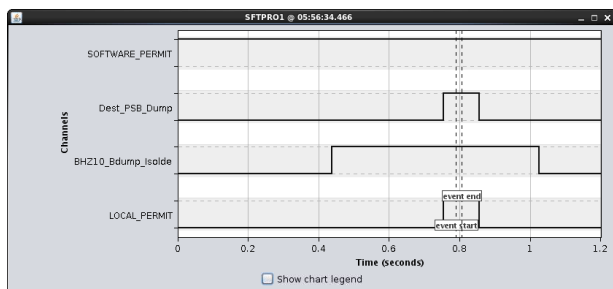


Figure 2: Display of the input and output permits for the PSB extraction device for one cycle. The time is relative to the cycle start time. The start and end beam extraction window events are represented by two vertical markers.

As the BIS hardware is fully asynchronous and cycle independent, monitoring software must map the hardware data to the cycle data to determine if the beam is allowed or not for a given cycle. The chosen solution to retrieve the cycle information is to write the cycle related timing events in the Beam Interlock Controllers history buffer (internal board events rolling buffer). At every cycle, the cycle start time is written in order to split the history buffer records into cycles in the FESA class. Moreover, the beam permit evaluation time window is also recorded in the history buffer as a single event or two events representing a time range. The representation of those events can be seen in Fig. 2.

The BIS cycle view retrieves the history buffers to calculate and display the summary of all permits of the BIS during this events window. The calculation is that if any logical permit value falls at least once to `false` during the retrieved events window, then the summarized value will be `false`, otherwise it will be `true`. This logic is consistent with the hardware beam permit calculation. The BIS cycle view is generic enough to display every input and output permit, per cycle, for one to several Beam Interlock Controllers, and possesses filtering functionalities. More details on the view can be seen in Fig. 3.

Time	User cycle	SOFTWARE_PERMIT	Dest_PSB_Dump	BHZ10_Bdump_Is...	LOCAL_PERMIT
05:53:48.866	ZERO	OK	OK	OK	OK
05:53:47.666	NORMGPS	OK	OK	OK	OK
05:53:46.466	SFTPR01	OK	OK	OK	OK
05:53:45.266	ZERO	OK	OK	OK	OK
05:53:44.066	SFTPR01	OK	OK	OK	OK
05:53:42.866	ZERO	OK	OK	OK	OK

Figure 3: The BIS cycle view for the PSB extraction device. It displays the latest cycles from top to bottom.

## GENERIC GRAPHICAL USER INTERFACE

In order to reduce the maintenance for the multiple operational BIS Java applications, a central and unified application is designed. This application possesses two generic needs. First, the BIS boards data must be organized and displayed depending on the board type and the user rights. Secondly, users must be allowed to send commands to multiple BIS boards depending on their rights.

The emphasis is to define a generic graphical user interface. However hardware specificities are taken into account to be able to answer every need in terms of functionalities. Hereafter are the three specificities the application offers:

- **Accelerator context:** Users can select on which accelerator context the BIS should be monitored. Switching from a context to another provides different overviews representing all the BIS boards present for this accelerator as shown on Fig. 4.
- **Board types:** Depending on the selected BIS board in the application, different monitoring views and commands are provided to the users.
- **Role based access :** Depending on the user rights, different monitoring views and commands are available. For example, a BIS expert will be able to see the specialist views monitoring specific registers and will be able to send additional expert commands. The application will provide read only functionalities if the user is not logged in.

## BIS IN OPERATION

At the same time as the BIS application is consolidated, all the BIS operational checks software dedicated to commissioning and operation have to be maintained, reviewed and extended for new accelerators. Hereafter are detailed the four different aspects that needs to be taken care of to be able to restart the BIS in an operational state after LS1:

- **Pre-operational checks:** executed by the LHC sequencer [4] to determine whether or not LHC operation can start. For the restart of the BIS after LS1, the pre-operational checks will be extended to include additional consistency checks against BIS reference configuration from the CCDB. Concerning, the BIS pre-operational checks for cyclic machines like the SPS or LINAC4, specifications are being made to integrate

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2014). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

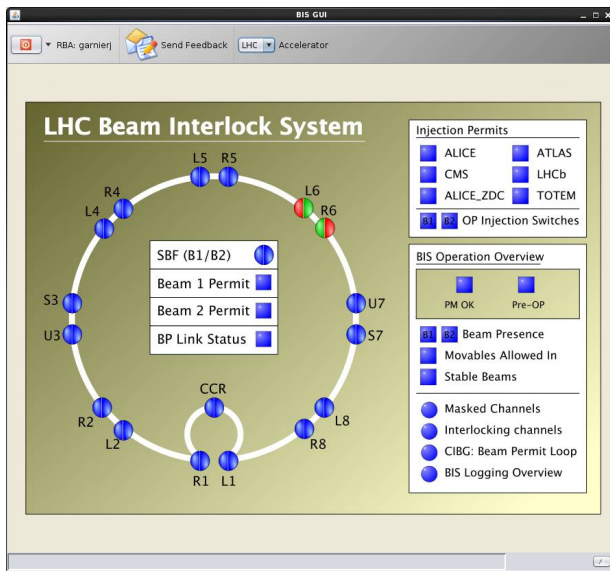


Figure 4: The unified BIS GUI overview for the LHC accelerator context. Most of the devices are not started (displayed in blue) due to the Long Shutdown 1.

cycle dependent DiaMon [5] checks to ensure the BIS is in a correct state. Future commissioning of the BIS in the LHC will also be integrated in the Acctestng [6].

- *During-operation checks*: carried out by the DiaMon framework for real-time asynchronous checking of the BIS in operation. For the BIS restart after LS1, the DiaMon checks are being specified to monitor the LHC and SPS BIS as well as latest deployments in LINAC4 and PSB.
- *Post-operational checks*: are LHC specific and rely on the Post-Mortem (PM) system [7] which collects systems data and analyses it to identify the cause of every beam dump, possibly inhibiting LHC beam injection. These checks are being commissioned for post-LS1 restart.
- *BIS Data Logging*: In order to be able to analyse the entire past activities of the BIS, the history buffers are stored in the Logging database [8]. Future history buffer events will be decoded before being stored in the Logging database, to provide more readable data.

More details on the BIS operational checks are described in [9].

## CONCLUSION

Early 2015 will see the restart of the LHC after the consolidation that has been made during LS1. This maintenance will allow the LHC experiments to capture the collisions produced by 2 beams circulating at a post-LS1 energy of

6.5TeV to allow for an efficient start-up. With the first LHC sectors being ready for commissioning, the first deployments of the BIS software layers has already started. The main tasks remaining to ensure proper monitoring of the BIS after LS1 are the following:

- Deployment of the board drivers and FESA classes to ensure a full scale monitoring of the BIS for all accelerators.
- Finish the ongoing consolidation and unification of the BIS Graphical User Interface.
- Recommissioning of all BIS software layers to be ready for post-LS1 operation.

Once the BIS is in operation, further tasks are planned to define a board registers decoding server for the BIS and the Safe Machine Parameters (SMP) allowing any demands for any critical parameters or interlock information to be provided to any Java applications running at CERN. The Long Shutdown 2 will see the end of the current hardware solution for the BIS and the definition of a new version of this system (BIS2). Moving towards the complete unification and genericness of the BIS software layers will ensure maintainability for the adaptation of the software to this new BIS2 hardware.

## REFERENCES

- [1] B. Puccio et al., "The CERN Beam Interlock System : principle and operational experience", proc. of IPAC 2010, Kyoto, Japan.
- [2] M. Arruat et al., "Front-End Software Architecture", proc. of ICALEPCS 2007, Knoxville, Tennessee, USA.
- [3] <http://linac4-project.web.cern.ch/linac4-project>
- [4] R. Alemany-Fernandez et al., "The LHC sequencer", proc. of ICALEPCS 2011, Grenoble, France.
- [5] W. Buczak et al., "DIAMON2 - Improved monitoring of CERN's accelerator controls infrastructure", proc. of ICALEPCS 2013, San Francisco, CA, USA.
- [6] D. Anderson et al., "Testing Quality and metrics for the LHC magnet powering system throughout past and future commissioning", proc. of IPAC 2014, Dresden, Germany.
- [7] R. Gorbonosov et al., "Plugin-based analysis framework for LHC Post-Mortem analysis", proc. of ICALEPCS 2013, San Francisco, CA, USA.
- [8] C. Roderick et al., "The LHC Logging Service: Handling Terabytes of On-line Data", proc. of ICALEPCS'09, Kobe, Japan.
- [9] I. Romera et al., "Pre-Operation, During-operation and Post-Operational verification of protection systems", proc. of ICALEPCS 2011, Grenoble, France.