LSA- THE HIGH LEVEL APPLICATION SOFTWARE OF THE LHC AND ITS PERFORMANCE DURING THE FIRST 3 YEARS OF OPERATION

Delphine Jacquet, Roman Gorbonosov, Grzegorz Kruk, Pablo Pera Mira

CERN, Geneva, Switzerland

Abstract

The LHC Software Architecture (LSA) [1] project was started in 2001 with the aim of developing the high level core software for the control of the LHC accelerator. It has now been deployed widely across the CERN accelerator complex and has been largely successful in meeting its initial aims.

The main functionality and architecture of the system is recalled and its use in the commissioning and exploitation of the LHC is elucidated.

INTRODUCTION

Even though the project's main target was to provide the control system for the LHC, it started with the idea that much of the accelerator control functionality is common to all particle accelerators and therefore the LSA system and associated tools could be re-used and extended for other machines. Since many operational aspects of the LHC were still under discussion at the beginning of 2000s, the initial design was based on the SPS requirements and on operational experiences from the previous control system used for the LEP accelerator. The initial milestones of the system included LHC transfer line tests in 2003 and 2004, first operational deployment in the LEIR accelerator in 2005 and progressive replacement of the existing SPS software in 2006.

By 2007, the foundations for the LHC control were well established and the system was used for the LHC beam commissioning. There were still a number of features under development but most of them were completed and deployed by the time of the LHC start-up in 2009.

LSA PRINCIPLES

LSA is a software suite, covering the most important aspects of the accelerator control. It is composed of a data model, a set of software modules based on that model, providing accelerator control services, and a set of generic applications using these services.

The data model includes:

- Definition of accelerators and transfer lines layout and all the elements
- Information about all controlled devices and their interfaces (properties)
- Beam optics and twiss parameters such as phase advance, beta function or dispersion
- Device parameters and associated settings

The software modules provide all necessary functionality for accelerator control:

- Access to the configuration data (Oracle database[2])
- Generation of initial settings based on optics
- Coherent settings changes (aka trim) with history and rollback
- Abstraction of the equipment access layer
- Hardware control and measurements

The core functionality of LSA is the management of device's settings. One of the main concepts is the parameter which represents a settable or measurable property of a device like voltage or current. Parameters can be organized in hierarchies which describe relationships between them. Typically roots are the high-level physics parameters and leaves are the hardware parameters. Any change of physics parameter value is propagated to the hardware level and sent to the equipment, see Fig 1.

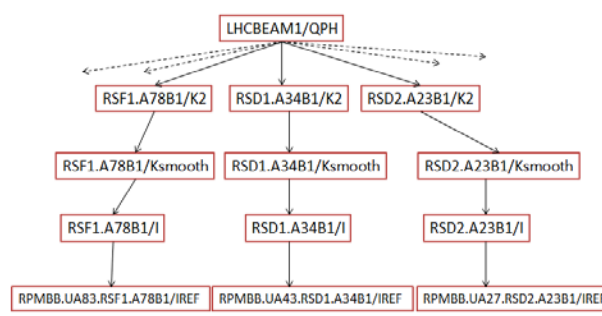

Figure 1: Parameter hierarchy for beam 1 chromaticity.

LSA is based on 3-tier architecture. The logic is implemented at the server side that exposes a simple interface used by LSA generic applications or many other client applications. All equipment and database accesses are made through the server, see Fig. 2.

ISBN 978-3-95450-139-7

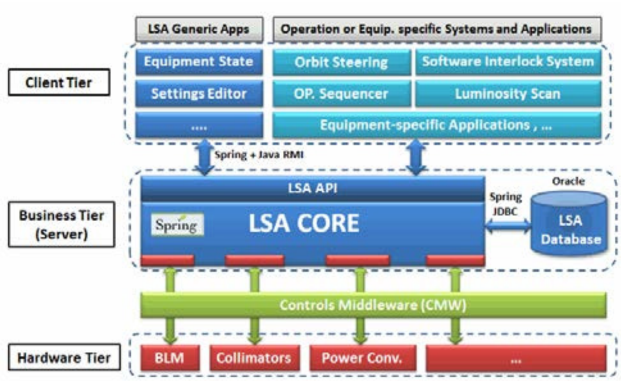

Figure 2: LSA architecture.

THE LHC CYCLE

The LHC cycle is composed of several phases necessary to bring beam from injection to luminosity production. Phases of arbitrary length (injection for example) alternate with phases where pre-defined functions are loaded into the hardware and executed (e.g. ramp).

- Pre-cycle: all superconducting magnets have to execute a predefined current function that resets the magnetic history. The pre-cycle ensures the reproducibility of their field at injection. The cycling properties depend on the magnet type.
- Injection: several injections are necessary to fill the LHC (e.g. twelve injections per beam for physics production). Beam parameters like chromaticity, orbit and tune are corrected during this process.
- Ramp: ramp functions are loaded into the power converters, the RF cavity control, the tune and orbit feedback systems and the collimators. All functions are triggered synchronously by a timing event. Beams are accelerated from 450 GeV to 4 TeV.
- Flat Top: LHC stays at high energy, necessary beam measurements and corrections can be made.
- Squeeze: beams are squeezed to increase the collision efficiency. Squeeze functions are loaded into a sub-set of power converters (mainly quadrupoles and sextupoles) and collimators whose position has to follow the beam size.
- Collision and luminosity production: Beams are put into collisions. Once the beam parameters are corrected, “Stable beam” is declared and experiments start to collect data.

Some of the LHC equipment, such as the power converters, the RF cavity controllers or the collimators, have been designed to support two types of settings:

- Scalar: during phases of undefined length like injection or flat top, scalar values are loaded to hardware. These values are set synchronously when beam parameter corrections are applied.
- Function: during phases like ramp or squeeze, function settings are loaded to the hardware. Functions are triggered by timing events, no modification can be made while a function is executed by the equipment, although real time tune and orbit feedback are operational.

The LHC cycle is driven by services provided by LSA: loading of function or scalar settings to the hardware, control of the settings change or triggering of timing event. LSA also ensures the settings continuity along the cycle. Whenever a scalar value is modified for a given parameter at phases like injection and flat top, LSA incorporates the new setting into the function of the following phase, so that the underlying equipment can smoothly start its execution.

LSA manages also devices whose parameters have to be control asynchronously with respect to the operational phases, or parameters constant along the cycle (e.g. collimator energy thresholds, instrumentation configuration settings). All primitive value types are supported including scalar, array or string.

LHC MODELS IN LSA

Optics Model

The LHC optics are uploaded from MADX[3] into the LSA database. The optic definition includes the normalized strengths, twiss parameters and pre-calculated coefficients for beam parameter changes (tune, chromaticity, coupling). Optics parameters are measured during the run and compared to the theoretical model. The calculated corrections are fed-forward to LSA. The applied corrections result in a measured optic very closed to the model.

Magnetic Model

FIeld DEscription for the Lhc (FIDEL) [4] provides models for determining superconducting magnet field behaviour

- Static effects like saturation and residual magnetization depend on the magnet current and are reproducible from cycle to cycle.
- Dynamic effects, decay and snapback, depend on the powering history and are not completely reproducible from one cycle to another.

The FIDEL model coefficients are uploaded into the LSA database, and the correction of static effects is integrated with LSA settings generation:

- Calibration curve for each magnet: Field/current function.
- Calculation of the settings of high order magnets used to compensate for magnetic field errors.

ISBN 978-3-95450-139-7

- Generation of the pre-cycle functions that depends on the magnet types.

LSA also provides corrections of the decay and snapback: the decay is predicted from the powering history and the dynamic model. Corrections are calculated and applied during the injection process. Just before ramp the snapback is predicted according to the time spent at injection and the ramp settings are corrected, see Fig. 3.

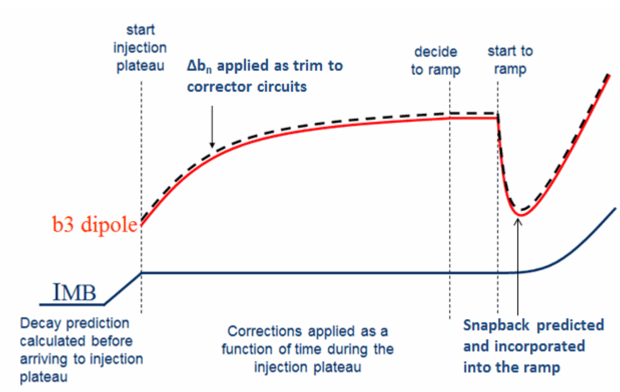

Figure 3: Compensation for decay and snapback on the injection plateau.

LSA INTERFACE

Thanks to 3- tiers architecture that exposes a clear API, LSA provides information on all LHC aspects and functionalities that can be easily used by any software.

The LHC sequencer [5] is an operational application that executes predefined sequences of tasks. Tasks are created to access all LSA services with predefined arguments (e.g. set of a specific parameter to a defined value, load of settings into a set of equipment, trigger of a given timing event, etc.). To ensure the complete reproducibility and homogeneity of the operational process, the sequencer is used for all phases of the LHC cycle and executes all the necessary actions to go through the cycle.

LSA is also used by various specialist application like orbit feedback that needs the optic information, specialized python scripts for optics correction, the luminosity scan application that sends trims to change the beam position at the interaction points, GUIs of instruments use LSA to managed their configuration settings (e.g. beam loss monitor, BCT…).

PERFORMANCE

Completeness and Flexibility

LSA modular architecture facilitates the implementation of new requirements and solutions for specific equipment constraints. LSA supports many types of parameters. The parameters are organised either as standalone, or in complex hierarchies. Settings can be changed at any level of the hierarchy in a coherent way. This flexibility allows for most of the LHC equipment to be integrated into LSA, like power converters, collimators, instrumentation etc…

Settings management includes functionalities like settings copy or comparison from one cycle to another, complete history of settings modification with the reload possibility.

LSA covers every aspect of the accelerator control, from the theoretical values given by the optics to the real settings that need to be applied after automatic or manual corrections. This ensures the coherence of settings from high level parameters to hardware, and beam parameters can be controlled very precisely at every phase of the cycle.

Reproducibility and Stability

Thanks to the very good magnetic model used for static and dynamic corrections of the field errors, plus the systematic pre-cycling of all the magnets, the LHC magnetic field is remarkably reproducible. As a consequence, optics, tune, chromaticity and orbit are stable from one cycle to another.

In addition, the corrections of the feedback systems for orbit and tune [6] are regularly fed forward to LSA settings to reduce the systematic errors [7].

Reliability

As the central control system, LSA system has to be available 24/24 hours 7/7 days. Despite being used by many applications and for every phase of the LHC cycle, almost no machine downtime can be assigned to LSA. Quality insurance is enforced by a strict release policy and intensive testing of the software before deploying new versions.

Universality

LSA is based on principles that are common to all accelerators: devices/parameters/optics/cycles. It is flexible enough to be adapted to cycling machine or colliders and can include specific requirements of various accelerator operation constraints. After its deployment in SPS and then in LHC, it is now replacing the legacy settings management system in the PS/Booster complex.

CONCLUSION

Developing a control system for a complex accelerator as the LHC was a big challenge. From basic and simple concepts, LSA provides functionalities that answer the LHC operational constraints. A good control system is the key point of operation efficiency and LSA had a crucial role in the success of the LHC. After 3 years of operation,

ISBN 978-3-95450-139-7

the software is stable and new requests are seldom. This may not be the case with new operational conditions after LS1, but thanks to the qualities demonstrated by LSA until now, no major issues are expected at the software level.

REFERENCES

[1] G. Kruk, S. Deghaye, M. Lamont, M. Misiowiec, W. Sliwinski "LHC Software Achitecture[LSA] – Evolution toward LHC beam commissioning", ICALEPS'07, Knoxville, USA.
[2] C.Roderick, R.Billen "The LSA database to drive the accelerator settings", CERN-ATS-2009-100, ICALEPS'09, Kobe, Japan.
[3] H.Grote, F.Smidt "MAD-X An upgrade from MAD8" PAC'03, Portland, USA.
[4] Nicholas Sammut "Magnet Modelling", LTC extended meeting March 2008, CERN.
[5] V.Baggiolini, R.Alemany-Fernandez, R.Gorbonosov, D.Khasbulatov, M.Lamont "A sequencer for the LHC Era" " CERN-ATS-2009-114, ICALEPS'09, Kobe, Japan.
[6] R.Steinhagen "Real time control at the LHC", PAC'11, New York, USA.
[7] M. Pereira, X. Buffat, K.Fuchsberger, M. Lamont, G.J. Müller, S. Redaelli, R.J. Steinhagen, J. Wenninger Feed forward in the LHC", ICALEPS'11, Grenoble, France.

ISBN 978-3-95450-139-7