Settings Management within the FAIR control system based on the CERN LSA framework

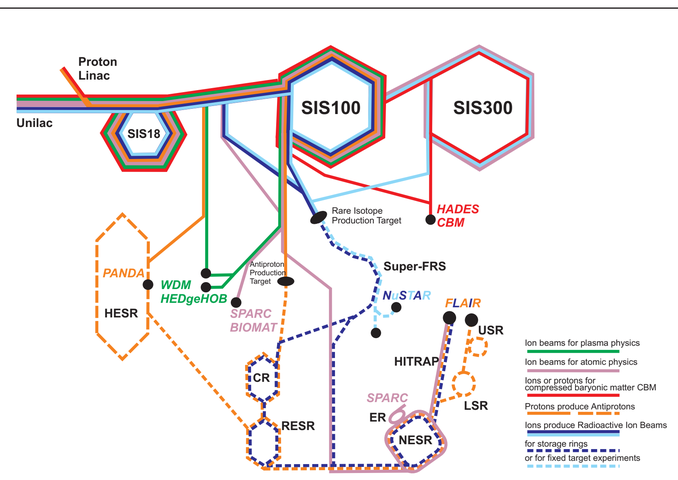
J. Fitzek, R. Mueller, D. Ondreka *GSI, Darmstadt, Germany*

Abstract

A control system for operating the future FAIR (Facility for Antiproton and Ion Research) accelerator complex is being developed at GSI. One of its core components is the settings management system.
At CERN, settings management and data supply for large parts of the CERN accelerator complex is done using the LSA (LHC Software Architecture) framework. Several concepts of the LSA framework already fit the FAIR requirements: Generic structures for keeping accelerator data; modular design; separation between data model, business logic and applications; standardized interfaces for implementing the physical machine model. An LSA test installation was set up at GSI and first tests were performed controlling the GSI synchrotron SIS18 already applying the new system to the existing facility.
These successes notwithstanding, there are issues resulting from conceptual differences between CERN and FAIR operations. CERN and GSI have established a collaboration to make LSA fit for both institutes, thereby developing LSA into a generic framework for accelerator settings management. While focussing on the enhancements that are necessary for FAIR this paper presents also key concepts of the LSA system.

FAIR

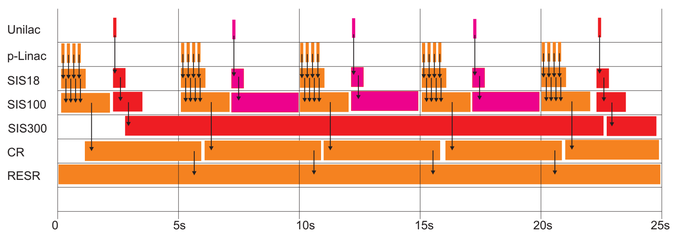
- GSI/FAIR complex will consist of 12 accelerators (realized in several subsequent stages)
- Pulse-to-Pulse switching between particle types
- Increased complexity: many more experiments at a time (time-sliced), up to five beams in parallel
- Will be build while the existing machine is running



GSI/FAIR beamlines, P. Schuett, GSI 2010

FAIR Control System

- Currently in design phase
- Controls around 12000 devices
- Integrated control of the whole accelerator complex
- Support for all operation aspects of the GSI/FAIR machines
- Commissioning of the system already with parts of the existing machine
- Existing facility shall be kept running continuously
- New control system is planned to start behind the linear accelerator UNILAC, leaving the control of the injector untouched, but tightly integrated
- Within the FAIR control system: need for a new settings management system for modeling accelerators and calculating/modifying settings in an integrated way
- Collaboration on the CERN settings management system LSA started in 2007 and is ongoing



FAIR beam pattern example, P. Schuett, GSI 2009

LSA settings management framework

Covers all important aspects of settings management

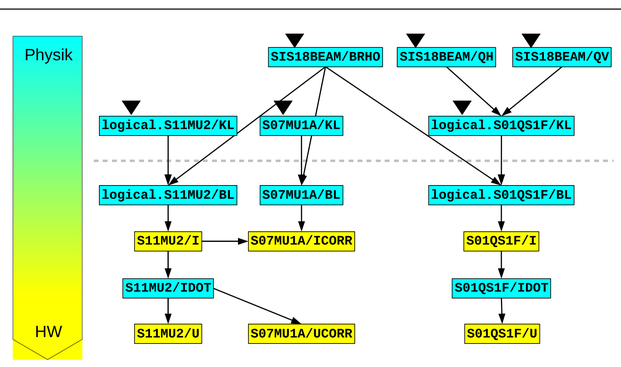
- Optics (twiss, machine layout)
- Settings generation and management
- Translation from physics to hardware parameters
- Operational exploitation, hardware exploitation, equipment control, beam based measurements

Modeling accelerators within LSA

- Parameter hierarchy: from physics to hardware parameters
- Rules for propagating setting changes along the hierarchy
- Physics model kept in a separate package, enables physicists to easily implement the machine model themselves
- Import of optics and device information
- System calculates good initial settings, allows consistent corrections, correction and theory values stored separately

Architectural principles

- Object oriented, platform independent (written in Java)
- Designed in a very generic way, not CERN accelerator specific
- Separation of concerns: model, business logic, applications
- Business logic is visible through façade classes
- Device communication is realized using an abstraction layer called JAPC (Java API for Parameter Control)



SIS18: parameter hierarchy example

LSA test system at GSI

✔ Test server setup in 2008/2009

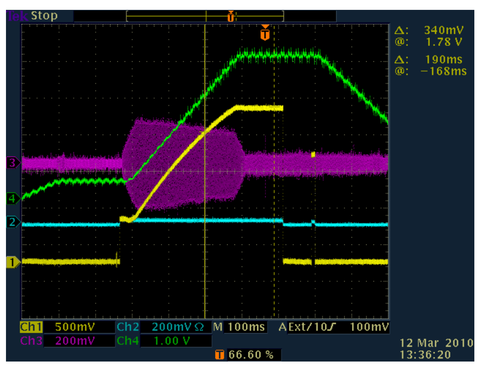
1. Oracle database instance, few tables prefilled
2. LSA test server - Java process on a Linux machine
3. GSI dummy implementations for the few unfulfilled references (like role based access)
4. GSI implementation for device access (JAPC plugin)
5. Deploy and test of generic LSA applications

✔ Modeling SIS18 with LSA in 2009/2010

- The FAIR data supply project team started with modeling the existing synchrotron SIS18 in LSA
- Importing accelerator layout, device information, optics
- Implementing the physical model of SIS18: parameter hierarchy, propagation rules
- Defining first test cycles based on today's operating

✔ Test in march 2010

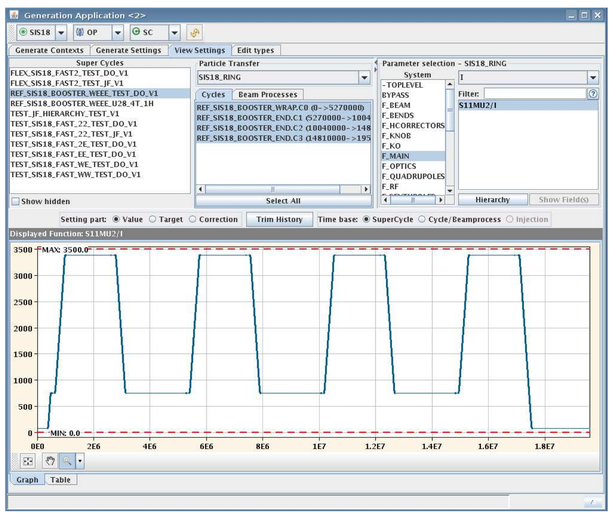
- Successful first test with beam as proof of principle
- Test with one/two cavities, one/two shot extraction



SIS18 LSA test with beam: single shot extraction

LSA at GSI - Outlook

- Focus on modeling the ESR (Experimental Storage Ring) in LSA to create a proof of concept also for storage rings
- Implement further operation modes of the SIS18 synchrotron for testing
- Implement specific applications on top of LSA supporting the future FAIR operations



Generic LSA application showing a SIS18 cycle

Extending the LSA core

Remove restrictions found due to fixed cycle time periods and therefore rather static operations at CERN

- Remove fixed times in cycle scheduling
- Automatically adapt the cycle length while changing settings (e.g. energy trims)

Add possibility to model full accelerator chains instead of only single accelerators

- Model accelerator chains
- Define inter-accelerator dependencies, e.g. link extraction and injection energy of two subsequent accelerators
- Propagate changes along accelerator chains
- Connect cycles of different accelerators belonging to the same beam production chain

Enhance build tool possibilities

- Desirable to hold laboratory specific sources in the labs version management system
- Add the possibility to perform a self defined, independent release cycle at GSI
- Support for multiple (local and remote) binary repositories
- Perhaps migrate to a well established (open source) tool?

Summary

- The LSA prototype installation and SIS18 modeling showed how quick other accelerators can be modeled thanks to the generic concepts in LSA.
- Successful first tests with beam proved that LSA fits very well for settings management and data supply within the FAIR control system.
- LSA is extensible and supports institute specific implementation (lab specific packages) as well as temporary development (configuration for exchanging implementation).
- Restrictions in LSA have been found and will be tackled within the well-established and ongoing collaboration.
- The collaborative goal is to make LSA a flexible framework for settings management for complex accelerator operations.

presented at PCaPAC 2010