ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA062

FIRST PRODUCTION USE OF THE NEW SETTINGS MANAGEMENT SYSTEM FOR FAIR

J. Fitzek, H. Hüther, R. Müller, A. Schaller, GSI, Darmstadt, Germany

Abstract

With the successful commissioning of CRYRING, the first accelerator being operated using the new control system for FAIR (Facility for Antiproton and Ion Research), also the new settings management system is now used in a production environment for the first time.

Development efforts are ongoing to realize requirements necessary to support accelerator operations at FAIR. At CRYRING, new concepts for scheduling parallel beams are being evaluated. After these successful tests and the first production use, the focus now is to include major parts of the existing facility (synchrotron SIS18, storage ring ESR and transfer lines) into the system in the context of the Controls Retrofit project. First dry runs are planned for Q4 this year.

The settings management system is based on the LSA framework [1], that was introduced at CERN in 2001 and is being developed and enhanced together in a collaboration with GSI. Notwithstanding all successes of LSA at both institutes, a review study was set up with the goal to make the LSA framework fit for the future. Outcomes of this study and impacts on the settings management system for FAIR are being presented.

USING THE NEW SETTINGS MANAGEMENT SYSTEM

The CRYRING heavy-ion storage ring is a Swedish in-kind contribution to the FAIR project. The machine includes its own linac with a MINIS type ion source, an RFQ linear accelerator and an electron cooler. It was set up at GSI (see Fig. 1) and successfully commissioned in 2015 with several beamtimes until now, including an injection test from the existing ESR storage ring performed in 2016 and recent successes with longer beam storage times [2].

While the other GSI accelerators are in shut-down for the FAIR upgrade, CRYRING represents a unique opportunity, since it is the first machine fully operated with the new control system for FAIR. With the equipment controllers, device class implementations with FESA [3], new middleware and service layer, the settings management system based on the LSA framework and new Java-based applications, new developments on all layers of the control system [4] have been successfully brought into operation.

Making LSA Ready for Production Use

For the FAIR settings management system, this first use in a production environment posed several challenges. So far, the machine developments done with LSA have been performed using a test database where basically everyone was allowed to enter data manually at any time. Since LSA is strongly data-driven, the data integrity is very important.

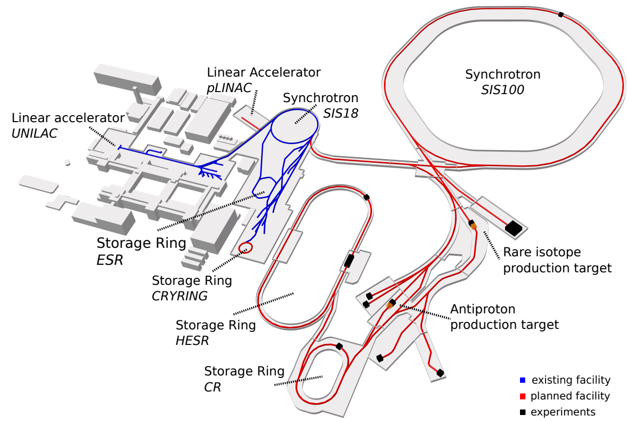

Figure 1: FAIR facility, FAIR.

Therefore, on the new production database, no manual user access to the data is allowed. Only scripts can do modifications, which either pull data from other databases (e.g. accelerator layout, devices, calibration curves, etc.) or from data files that are kept under version control together with their import scripts (e.g. optics, twiss information). Efforts are ongoing to provide a Java-based importer toolset for all necessary data. Also, the production database itself was not set up as a structural copy of the existing test database. Instead, an agreed-upon database state was taken from CERN and it was transformed into a set of defined scripts, making it possible to set up a new database at any time, using these structural scripts, together with the data content scripts mentioned above.

Besides these measures on the database, also the release process had to be formalized. So far, tests with LSA have been mostly performed from the developer's workspace. For production use, releases have to be performed inline with control system releases. The whole control system is developed with a major release every half year. In order to formalize the release of LSA at GSI and make it possible to introduce bugfixes on the last release state whenever needed, release branches were introduced on all levels, LSA as well as applications (roughly 50 artifacts). Now, developers are able to continue development for the next release, even introducing incompatible changes, without compromising the current production state.

Realization of New FAIR Concepts

The major development that is ongoing in the LSA framework is to realize the parallel beam scheduling concepts designed for operation of the FAIR facility. FAIR operation poses unique challenges with up to five parallel beams and pulse-to-pulse switching between particles. The idea is to move away from an accelerator-oriented to a more

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA062

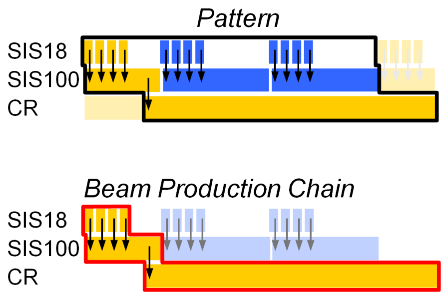

Figure 2: Patterns and Beam Production Chains as new concepts for scheduling beams, D.Ondreka, GSI.

beam-oriented view. Consequently, also matching software concepts are developed to support this change in paradigm.

The new concepts: Beam Production Chains to represent single beams in the machine and patterns grouping those chains into scheduling blocks that are executed in the machine, see Fig. 2, have been realized in the LSA framework [5]. The existing concept of Beam Processes, which describe single steps in the machine like injection, ramp, extraction as the central objects for storing settings has been kept. Beam Production Chains are composed of those Beam Processes. New at GSI, and also new in LSA, is the idea that those contexts can span accelerator boundaries. Beam Production Chains and patterns have first been successfully used at CRYRING, with the limitation that, so far, only one accelerator was involved.

To be able to introduce Beam Production Chains into the LSA framework, the limitation that Beam Processes could only be scheduled with fixed times has been lifted. Instead, a new relative scheduling mechanism has been implemented. Now, one can schedule that the Beam Process for the ramp comes “after” the injection (begin-end) and that a Beam Process for the transfer line is scheduled parallel to the extraction in the ring (begin-begin with same length). Advanced concepts for scheduling Beam Processes like that one Beam Process in the ring spans multiple Beam Processes in another machine (begin-begin with Beam Process A, end-end referencing Beam Process B) has still to be realized to support the booster mode with four extractions from the SIS18 into the SIS100, see Fig. 3.

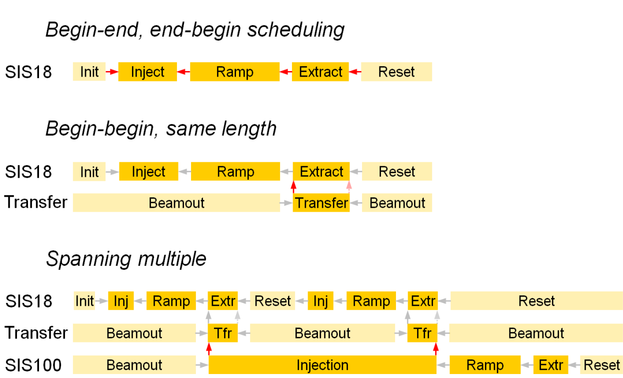

Figure 3: Relative scheduling of Beam Processes in LSA, example with booster mode and two injections into the SIS100.

Calculation of the Timing Schedule in LSA

With CRYRING, also the new White Rabbit-based timing system is used for the first time. To program the timing system, a schedule is generated in LSA using the standard mechanisms: Alongside other physics rules that are used to calculate settings for high level physics down to hardware parameters, there is also a rule that is called for the timing system. Input parameters are all other parameters that influence the schedule like bumper fall time, extraction time, ramp length depending on ramping speed etc. Since settings in the LSA framework exist per Beam Process, the timing system settings are also generated per Beam Process. That fits to the principle that Beam Processes are the smallest atomic unit that gets executed later on in the machine. However, the interface of the timing system expects one “execution schedule” for the whole facility. Therefore, settings are merged together and converted into the specific timing system format during data supply of the timing system. Besides that, the timing system is just treated in LSA like yet another “device” and therefore all mechanisms of LSA can be used seamlessly.

At the moment, data supply of the timing system is done using a rather low level interface. Developments are ongoing to abstract and logically separate different components of the Beam Scheduling System (BSS). In the future, changing the timing schedule will much more flexible and comfortable.

RETROFIT PROJECT

Within the scope of the Controls Retrofit project, the existing accelerators SIS18 and ESR, including the transport lines between them, will be retrofitted to be controlled fully with the new control system beginning with the beamtime in 2018. Major efforts have been going into that project since it was started in 2016. Regarding the settings management system, the following steps have been taken.

Importing Master Data

Together with the machine coordinators and machine modelers, accelerator data has been created in LSA. The accelerator layout, especially for the transfer lines, has been translated into the LSA structures. Particle Transfers that describe sections with same optics and same timing (typically a ring, or a part in the transfer lines between two junction points) and their subdivision into Accelerator Zones (with similar beam properties) has been completed.

A first import of device data into LSA with a newly provided script has just been performed. The existing SIS18 physics model has been imported successfully. After that, models for the transport lines and the ESR will be added and will be tested in the development environment before importing a final version into production.

As discussed above, data should be imported into the production database in a structured way. Although data and a physics model already exists for machines mentioned, it was imported manually and, as it was used as prototype, the data was not yet imported using the officially supported

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA062

mechanisms. Since now many data flows have been set up properly (like calibration curves coming from the magnet group into the central GSI/FAIR Component Database, then into the Controls Database, then into LSA), the production database will be filled solely using those procedures.

Integration of Existing Equipment Controller Software

When communicating with devices to send new set values, software on the device level written with the FESA framework seamlessly integrates with the settings management system. Devices can be imported into LSA together with their interface definition. This is the case for CRYRING, where only new FESA device classes are used. However for the Controls Retrofit project, the majority of existing device software has been realized with an existing equipment controller framework at GSI called Device Access. In principle, communication with those devices is possible through a special middleware plugin that was written specifically for that purpose. Successful machine development shifts with LSA in 2010 already made use of this plugin [6].

In those tests however, only a small subset of device properties were used. In 2018, all the devices in the machines taking part in the Controls Retrofit project will be exclusively accessed from the new control system, making it necessary to fully support their interface. Certain features, like parameterized calls to devices and complex data structures, are not supported by the new settings management system. Instead, LSA supports a defined set of fixed data types including arrays and 2d-arrays allowing for generic implementations (basically no "custom" data types). Therefore, properties of the existing device class implementations have been reviewed and modified to fit to the new guidelines.

Existing devices are also connected to the old timing system. This setup will stay until their full replacement with FESA device class implementations and a connection to the new White Rabbit-based timing system. As new FESA devices are typically Beam Process-based while the Device Access devices are still cycle-based, a mapping has been defined. A bridge that provides translation functionality between the new and the old timing will ensure, that events necessary for existing devices are also sent to the old MIL-based timing network. In the LSA schedule, also those events for the existing devices are foreseen.

Quality Measures

In the recent past, first steps were taken in LSA to allow for testing the LSA framework in both institutes independently of the test database content. Those "self-sufficient" test cases represent today a good portion of the JUnit test cases in LSA with a growing number. Still, significant effort remains to convert the rest of the testcases and achieve a yet higher test coverage.

In addition to those generic LSA tests, specific integration tests with devices have been setup at CRYRING. For that purpose, each FESA device class also has a "mock" representation that allows for automated interface tests. However for the Controls Retrofit project, this does not exist in the same manner for the existing Device Access framework. Ideas have been discussed and tried out, but in prospect of the full replacement with FESA device classes in the coming years, no more effort is put into it.

For integration tests, the idea is to set up a full integration test environment for all components of the control system. First components are already deployed, but much more effort needs to be put in to profit from this test environment.

As tests can never fully replace the real machine, especially when it comes to real-world accelerators with no laboratory test copy of it, tests on the accelerator without beam are a necessity. As not only the settings management system is new, but also the rest of the stack of the control system, early dry runs have been scheduled starting this October.

ORGANIZATION OF THE COLLABORATION

Synchronizing Features

To better support collaboration and allow exchanging features more easily, a common synchronization ("sync") branch was introduced in the source code repository. Only features that are ready and tested within one institute are then also committed to the common sync branch, see Fig. 4. Before, features could only be transferred from CERN to GSI or vice versa by using the CERN SVN trunk. Doing so, it was almost impossible to get one feature without retrieving unfinished changes. To mitigate that problem, source code merging was mostly only done directly after a major release at CERN when the trunk was rather stable and did not contain too many open developments.

With the new setup, each institute is able to pull features from the common code base when technical stops or accelerator shutdowns allow introducing changes. The sync branch only contains finished features and therefore always represents a stable state of the code. Organizationally, new features to be introduced to sync are agreed upon beforehand and can then only be introduced after completing a review process on both sides.

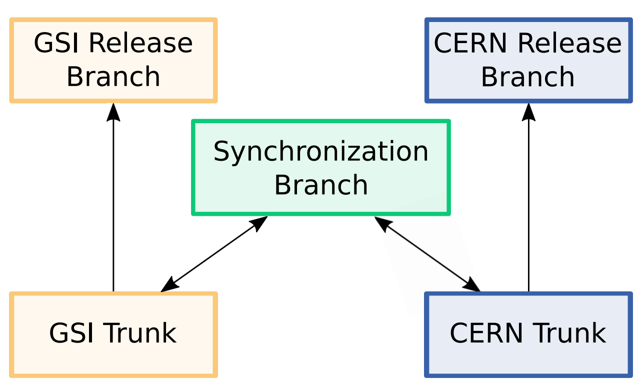

Figure 4: Repository setup including the new synchronization branch for feature exchange.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA062

The next step will be to enable testing and nightly builds on this common code base, which requires setting up a matching sync database.

Technically, this setup only works because the GSI branches of the LSA core products, as well as the sync branch itself, reside in the CERN SVN repository. Therefore, it is possible to use the merge feature of SVN to exchange features. In the future, when both institutes switch to using Git, it might be possible to merge across different Git repositories and keep the GSI specific code at GSI.

Versioning of Database Changes

As with the code changes, it was also complicated to identify the database changes, that correspond to the code changes. Furthermore, there was no definition of which database changes belonged to a specific version of LSA. Changes in the Java codebase and in the database scripts were only related by keeping both references in the same bug/feature description of the issue tracking system at CERN.

At GSI, Liquibase was introduced as tool for database versioning. Besides versioning scripts along with Java releases, the introduced mechanism also supports different database structures and content for the development, integration and production databases. Further details are presented in [7].

At CERN, database versioning tools are also evaluated at the moment, Liquibase and Flyway are strong candidates. Once a versioning tool has been established also at CERN, the exchange of features via the sync branch could become much easier as it can be accompanied also by the corresponding database change.

LSA REVIEW

LSA Review Study at CERN

The LSA framework has been developed over 15 years now. Bigger refactorings took place in the past, but typically targeting only a specific functionality in LSA like refactoring the package that sends data to the devices. However, the overall logical building blocks have not been questioned so far. The goal of the review study initialized at CERN was to really question LSA at its core, i.e. its structure and its functionality. The questions were not only what could be improved or what is missing in the framework, but also what does not fit the scope of the system. Developers from GSI contributed to this study with their 10 years of LSA experience.

One of the major findings was that the LSA code base has become quite large and gets harder and harder to maintain. One proposal of the study is to inverse the logical structure and have a very slim settings management core and add further functionality in additional layers. A list of other proposals deals with specific functionalities where the question always was if it should actually still be part of the LSA core or be moved elsewhere (e.g. institute-specific code that still remains in the common core). Smaller refactoring proposals were also part of the outcome. Examples are: improving the parameter relations management and moving it from a database implementation up to a more maintainable Java version, improving the settings archiving mechanism and improving testing of the core.

Due to the tight time and resource constraints at CERN and GSI, it is at the moment not possible to aim for a big LSA refactoring that tackles all of the proposed changes from the study. The intention is to take the proposals one by one and realize them during the planned shutdowns. Even though the idea is to keep the outside interface to clients stable where possible, especially restructuring the code base might also affect at least package names. At CERN, the next time window for introducing backward incompatible changes will be the long shutdown LS2, scheduled for the end of 2018. This fits also to the constraints at GSI, which require LSA being left as stable as possible for the beamtime mid of 2018 and only afterwards introducing substantial changes.

LSA Performance Review at GSI

Frequent changes of top-level parameters (like energy) are performed at GSI. Such trims affect a great part of the parameter hierarchy as they lead to recalculation of many settings, which takes some time. For CRYRING, which is rather small, this is not yet a problem. But in order to be able to commissioning the much larger upcoming FAIR accelerators, measures had to be taken.

As result of this optimization of LSA, a huge speedup could be achieved that now allows the user to change top-level parameters without much noticeable delay. Details can be found in [8]. The next step will be to merge these changes into the common code base, so that they will be also available at CERN.

OUTLOOK

Besides these great achievements and first production use of LSA, efforts are still ongoing to realize the Pattern concept. Next steps are to finalize the features needed for the Controls Retrofit project in 2018 like being able to execute several parallel Patterns if they target disjunct parts of the facility (e.g. a parallel SIS18 and CRYRING beamtime) and also to implement a more advanced relative scheduling.

Storage ring functionality will be the big upcoming topic after the beamtime in 2018. Features like stop points in the schedule and several alternative following Beam Processes with the execution path defined by the operator at runtime are central requirements.

In the collaboration, the idea of exchanging features through the synchronization branch still has to evolve. As the sync branch will become much more important, also a nightly build setup with a corresponding sync database are the next logical steps.

First successes with feedback applications based on LSA [9] have demonstrated the potential of the new settings management system. Open interfaces allow for easy integration with other systems, e.g. simulation tools. At CRYRING,

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA062

a steering mechanism based on genetic algorithms is being put into place [10]. Other feedback applications will follow. Since some are not written in Java, the question of how to best integrate them still has to be answered.

REFERENCES

[1] G. Kruk *et al.*, "LHC Software Architecture (LSA) - Evolution toward LHC Beam Commissioning", in *Proc. ICALEPCS'07*, Knoxville, Tennessee, USA, paper WOPA03.

[2] F. Herfurth *et al.*, "The commissioning of CRYRING@ESR", presented at COOL17, Bonn, Germany, paper THM13.

[3] A. Schwinn *et al.*, "FESA3: The new Front-end Software Framework at CERN and the FAIR Facility", in *Proc. PCaPAC'10*, Saskatoon, Canada, paper WECOAA03

[4] R. Bär *et al.*, "News from the FAIR control system under development", in *Proc. PCaPAC'14*, Karlsruhe, Germany, paper WPO004.

[5] H. Hüther *et al.*, "Realization of a Concept for Scheduling Parallel Beams in the Settings Management System for FAIR", in *Proc. ICALEPCS'15*, Melbourne, Australia, MOPGF147.

[6] J. Fitzek, R. Müller, and D. Ondreka, "Settings Management within the FAIR Control System based on the CERN LSA Framework", in *Proc. PCaPAC'10*, Saskatoon, Saskatchewan, Canada, paper WEPL008.

[7] S. Müller *et al.*, "Conception and Realization of the Synchronization of Databases Between Two Research Facilities", presented at ICALEPCS'17, Barcelona, Spain, Oct 2017, paper TUPHA041.

[8] A. Schaller *et al.*, "Optimized Calculation of Timing for Parallel Beam Operation at the FAIR Accelerator Complex", presented at ICALEPCS'17, Barcelona, Spain, Oct 2017, paper TUPHA019.

[9] R. Steinhagen *et al.*, "Beam-Based Feedbacks for FAIR – Prototyping at the SIS18", in *Proc. IPAC'17*, Copenhagen, Denmark, paper TUPIK046.

[10] S. Appel *et al.*, "Automatized optimization of beam lines using evolutionary algorithms", presented at ICALEPCS'17, Barcelona, Spain, Oct 2017, paper THPHA196.