BENEFITS, DRAWBACKS AND CHALLENGES DURING A COLLABORATIVE DEVELOPMENT OF A SETTINGS MANAGEMENT SYSTEM FOR CERN AND GSI

R. Mueller, J. Fitzek, H. Huether, GSI, Darmstadt, Germany
G. Kruk, CERN, Geneva, Switzerland

Abstract

The settings management system LSA (LHC Software Architecture) [1] was originally developed for the LHC (Large Hadron Collider). For FAIR (Facility for Antiproton and Ion Research) a renovation of the GSI control system was necessary. When it was decided in 2008 to use the LSA system for settings management for FAIR, the middle management of the two institutes agreed on a collaborative development. This paper highlights the insights gained during the collaboration, from three different perspectives: organizational aspects of the collaboration, like roles that have been established, planned procedures, the preparation of a formal contract and social aspects to keep people working as a team across institutes. It also shows technical benefits and drawbacks that arise from the collaboration for both institutes as well as challenges that are encountered during development. Furthermore, it provides an insight into aspects of the collaboration which were easy to establish and which still take time.

INTRODUCTION

The idea of collaboration between CERN and GSI on the settings management system was first discussed in 2006 between the Controls groups management of both institutes. At that time the development of LSA was well advanced but completion of functionality necessary for the LHC startup in 2008 still required a significant amount work. While GSI was evaluating different possibilities for a new control system that could be used for FAIR, it was agreed that two software engineers from GSI will spend 18 months at CERN to help with the development and commissioning of LSA.

The common development had clear benefits for both institutes. For CERN it was a reinforcement of the LSA team by two skilled developers in the view of the upcoming deadline and a fresh view on the system by new external people, while for GSI it was an excellent occasion to gain a valuable experience, learn about the system and main ideas behind, and to evaluate its possible use at GSI.

Since the first impressions about the portability of LSA were very positive, after the 18 months of joint development at CERN, the two GSI developers deployed a dedicated version of LSA in their home institute for further evaluation. The first goal was to prepare a working version of an LSA-based control system for the existing SIS18 synchrotron, which would help other GSI developers and users get deeper insight in the system and be a final verification of applicability of LSA to the whole accelerator complex at GSI.

After positive feedback and successful machine development sessions on SIS18, it was decided to use LSA for FAIR and continue the collaborative development of LSA by both institutes [2–5].

BUILDING THE TEAM

The initial 18 months of common development in one place was certainly beneficial in building a solid foundation for further collaboration as it created strong bonds within the team who worked together towards shared goals. It gave everyone a good overview about the technical aspects of the control systems in both institutes as well as a good insight into the work processes of the colleagues from the other institute, their constraints and their deadlines. This certainly helped in increasing the mutual understanding and in taking certain decisions related to the collaborative development.

To keep the collaboration active and to maintain the team spirit developed during those 18 months in the longer term, it was perceived as important to maintain a regular contact. This goes beyond exchanging ideas via mail or phone calls to also having in person discussions during regular visits. For the collaboration between CERN and GSI, these take place twice a year and are used to discuss, agree and schedule major changes to be applied in the system.

DEVELOPMENT PROCESS

Modularization

Even though LSA had been organized in a modular way from the beginning, during the common development in 2007 and 2008 it has been further restructured to split the code base into generic and CERN-specific parts, allowing possible extensions by GSI.

Figure 1 shows the current package hierarchy. The common modules in the middle contain the generic settings-management framework and have no outward dependencies, so they can be compiled and released independently. The domain objects and logic related to particular accelerator models, types of equipment, infrastructure (such as timing) and operation modes are implemented in the institute-specific parts (on the left and right side in the Figure 1).

The institute-specific extensions are developed and released independently by CERN and GSI. However all changes done in common modules are subject to review and acceptance by developers from both institutes.

ISBN 978-3-95450-146-5

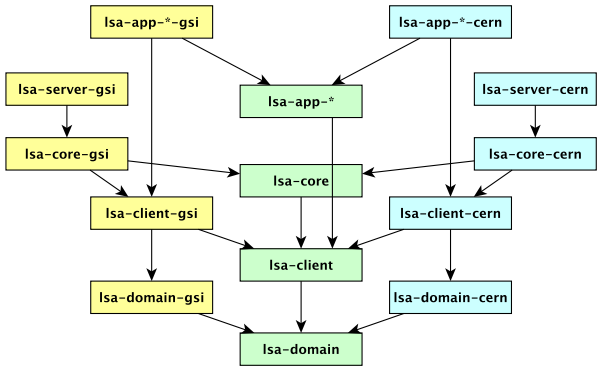

Figure 1: Layout of modules.

Changes in Common Modules

Depending on the nature of the change, the procedure to develop and review it is different. We have defined three categories of code changes that are described in Table 1.

Table 1: Categorization of Code Changes

Term	Descriptions
Major Change	Change that requires modification of the core behavior, data model or APIs with big impact on the system. Deployed only during non-operational periods (shutdown).
Significant Change	New feature or modification of existing functionality. Can typically be deployed during operational periods (technical stops).
Non-significant Change	A change with no impact on the functionality. It must always be backward compatible. Can be deployed at any time.

The *non-significant* changes are done and committed to the source repository directly by both CERN and GSI developers. Those are typically bug fixes, cosmetic changes or improvements of the documentation. Developers from the other institute usually review such changes when synchronizing with the source code repository.

The *significant* changes are discussed upfront by involved developers from both institutes. This is typically done via e-mails, phone calls or video conferences. The review of the code is usually done before committing it to the repository e.g. by sending a patch containing the change set.

The *major* changes are subjects of more in-depth analysis among all developers to find the best solution that works for everyone but also to examine the impact on the overall control system in both institutes. Such changes are typically discussed during visits which take place twice a year. They are then developed in dedicated source repository branches, and merged back to the main branch when ready for release. Deployment into the operational environment is done during longer technical stops (few days) or during shutdown periods to not influence regular operation of accelerators.

Artifacts Repository and Release Procedure

Until now, the management of all controls' software artifacts at CERN is done using Common-Build [6] (an Ant-based build system developed in-house) and a proprietary artifacts repository. Due to several dependencies on CERN's infrastructure this solution could not be used at GSI. Therefore GSI decided to use Maven, one of the major industry standard solutions.

Releases of both the common and institute-specific modules are done independently at CERN and at GSI, and are deployed into repositories at each institute respectively. All dependent artifacts developed at CERN that are necessary to build and run LSA are mirrored from CERN repository to the GSI one via a custom replication mechanism.

COLLABORATION AGREEMENT

As a formal basis for the collaboration, an agreement document is being established between the controls groups in both institutes, as an addendum to the existing overall agreement between CERN and GSI. The document defines various aspects of the collaboration, including project roles, ownership of modules, responsibilities of both parties and decision making process concerning the common development of LSA but also of any other system that may be a subject of the collaboration in the future.

The following four categories of modules have been defined in the document:

C1: Modules being part of the collaboration
C2: Generic modules used by C1 (developed by either institute)
C3: 3rd-party libraries used by C1 or C2
C4: Institute-specific modules of C1

For (C1) modules the document ensures that each party gets a free and irrevocable license on the source code. For (C2) modules, it grants usage rights. Upgrades to newer version of third party libraries (C3) need to be coordinated between the institutes. The (C4) modules are in the responsibility of each institute.

The two development teams agreed on the general practicalities of the collaboration already in 2009; however the collaboration agreement has not yet been officially finalized. Making it an addendum to the existing agreement between GSI and CERN has required an additional effort and involves other entities, like the legal departments of the two institutes.

BENEFITS, DRAWBACKS AND CHALLENGES

Based on a few years' experience there are pros and cons of the collaboration that has been seen by both parties.

Benefits

One of the biggest profits for both institutes is the joint manpower working on the system and therefore saving man-years of work on both sides.

GSI could join an already mature and relatively stable project, used in operation for several accelerators at CERN.

ISBN 978-3-95450-146-5

The foundations of the domain model were established and many conceptual problems had already been resolved. In addition several operational GUIs were implemented and could be used straight away at GSI. This meant that GSI could relatively quickly focus on the institute-specific extensions rather than spend time on developing a new framework.

For CERN, besides the additional development resources, the collaboration influenced establishing a better structure of the code and packages, which are always reviewed by GSI for portability. It also encouraged a more generic approach, making the system more flexible, also for future extensions at CERN. Since GSI was not yet in operation, in several cases new concepts or features developed in the framework could be first validated at GSI, before operational deployment at CERN.

Drawbacks

The major issue for both institutes is the overhead of introducing any significant changes in the common code. Any such modification proposed by one party must be validated by the other party. Checking usability, modeling it in a generic way, reviewing and testing against use cases in both institutes introduces additional delays as compared to if it would be done in one institute only, thereby increasing the time required from the moment a change is proposed until it is released in production.

Challenges

The principal challenges that we currently face are technical ones. Finding a generic solution in the common code that works for both institutes is often non-trivial and requires a few iterations of analysis until a satisfactory result is achieved.

Another challenging area for the collaboration is the Graphical User Interfaces. Even though functionality-wise both institutes have very similar requirements, different appearance or specific features are frequently requested by respective users. This requires designing these tools in a configurable and pluggable manner to allow customization when needed.

There are also several other technical difficulties that must be overcome to streamline the collaborative development such as easy and reliable access to generic libraries developed in both institutes, tracking changes in the database model and synchronizing them with corresponding code changes, or defining a better procedure to deal with urgent change requests that cannot wait for acceptance of the other institute. The latter one will become especially important once GSI enters into the commissioning stage and, eventually, the stable operation phase.

CONCLUSIONS

Several conclusions can be drawn concering what has been working very well and what we could have done better.

Not putting enough attention to the build and release of shared artifacts from the very beginning caused several problems and required manual work to resolve compatibility conflicts. Another area for improvement is to better organize the flow of information and make change reviews more efficient to not block one of the parties from doing necessary modifications.

The modularization of the system into generic packages and institute-specific extensions before the common development started was certainly a good decision. Also having an initial “gentlemen’s agreement” on how we plan to work together, which was later translated into a formal agreement, gave a solid basis for the future collaboration.

Although the collaboration brings many challenges and has certain drawbacks, our overall experience is very positive and we will certainly continue the common development in the future. We believe that the key aspect of a successful collaboration is the human factor. People involved must feel as a team working together, respect each other, give honest feedback and listen to feedback of others, and finally be prepared for compromises since that is what collaboration often requires in order to move forward.

REFERENCES

[1] G. Kruk et al., “LHC Software Architecture (LSA) - Evolution Toward LHC Beam Commissioning”, ICALEPCS’07, Knoxville, Tennessee, USA, WOPA03

[2] R. Mueller, J. Fitzek, D. Ondreka, “Evaluating the LHC Software Architecture for data supply and settings management within the FAIR control system”, ICALEPCS’09, Kobe, Japan, THP012.

[3] R.C. Bär et al., “Development of a New Control System for the FAIR Accelerator Complex at GSI”, ICALEPCS’09, Kobe, Japan, TUP107.

[4] J. Fitzek, R. Mueller, D. Ondreka, “Settings Management within the FAIR Control System Based on the CERN LSA Framework”, PCaPAC’10, Saskatoon, Saskatchewan, WEPL008.

[5] H. Hüther, J. Fitzek, R. Müller, D. Ondreka, “Progress and Challenges During the Development of the Settings Management System for FAIR”, PCaPAC’14, Karlsruhe, Germany, WPO005.

[6] G. Kruk, “Development Process of Accelerator Controls Software”, ICALEPCS’05, Geneva, Switzerland, FR2_5-6O.

ISBN 978-3-95450-146-5