ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-WEPAB319

OPEN XAL STATUS REPORT 2021

J. F. Esteban Müller*, E. Laface, Y. Levinsen, N. Milas
European Spallation Source ERIC, Lund, Sweden
T. Gorlov, A. Shishlo, A. Zhukov
ORNL, Oak Ridge, Tennessee, USA

Abstract

The Open XAL accelerator physics software platform is being developed through international collaboration among several facilities since 2010. The goal of the collaboration is to establish Open XAL as a multi-purpose software platform supporting a broad range of tool and application development in accelerator physics and high-level control (Open XAL also ships with a suite of general-purpose accelerator applications). This paper discusses progress in beam dynamics simulation, new RF models, and updated application framework along with new generic accelerator physics applications. We present the current status of the project, a roadmap for continued development, and an overview of the project status at each participating facility.

SNS STATUS

The Spallation Neutron Source (SNS) project continues to use Open XAL as the main control room accelerator physics tool. Several applications were upgraded including super conducting linac tuning and injection foil painting. We still use somewhat outdated Java 8 and Jython 2.7 for scripting. We are assessing the possibility of migrating to Python and abandoning the Java platform.

Ring Tuning Application

Accelerator model in the Ring of SNS has problems. Simple linear beam dynamics is very sensitive on quadrupole magnet parameters. The hard edge model of quadrupole produces an error of about 1% for particle transportation through the magnet compared to real 3D magnet [1]. The measurement of the integrated or effective magnetic field also has about 1% error or more. The total error from all 52 quadrupoles in the Ring has a huge impact on the linear model and makes it impossible to apply it for Ring tuning using measured magnet parameters.

We developed an Open XAL application that calculates/calibrates all integrated/effective quadrupole fields in the Ring and uses the same simple linear hard-edge model. The Ring of SNS has capability of detailed orbit measurement by BPMs during linear motion of a small bunch. This orbit and motion can be manipulated and changed inside of the Ring by corrector kicker magnets. The Ring application changes all kickers in the Ring one by one and collects all experimental position data r_i where $i = 1 \ldots N$ that is called orbit. After that it fits experimental data r_i to the model orbit in the Ring that depends on quadrupole magnet parameters $r_i(q_1, q_2, ..., q_{52})$ where q_i are "effective" quadrupole

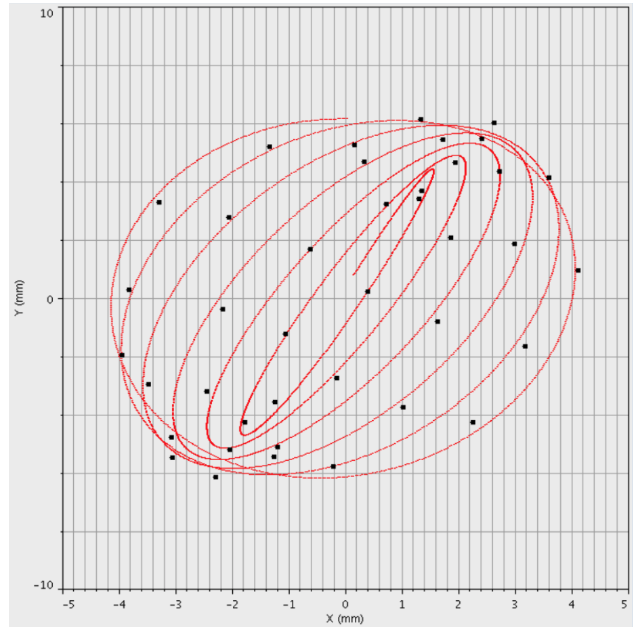

Figure 1: Example of x-y beam position of single minibunch in the Ring measured by single BPM turn by turn.

gradients. Then it calculates quadrupoles by minimizing objective function:

$$\sum_{i=1}^{N} (r_i - r_i(q_1, q_2, \ldots q_{52}))^2 \rightarrow \min \qquad (1)$$

As a result we have a working linear model in the Ring that agrees with experimental data orbit-wise as well as for some other Ring parameters like tunes. The application is supposed to perform precise painting for the Self Consistent Beam project (SCBD) [2]. It also can be applied for routine production beam tuning. Figure 1 presents an example of x-y beam position turn by turn measured by the same BPM. Both vertical and horizontal positions have equal tune and oscillate with the same frequency making a permanently tilted ellipsoid.

ESS STATUS

As reported at previous conferences [3–5], at ESS we developed an extension to the Open XAL model to match our operational requirements. We reached a point where most of our requirements are met by this model. In the past 2 years we have therefore focused on improving the Java FX application framework and developing the applications and services needed for the next commissioning phase.

* JuanF.EstebanMuller@ess.eu

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-WEPAB319

Model and Machine Description (SMF)

We developed a new element for our model to simulate cavities using 3D RF fieldmap, for cases when cylindrical symmetry cannot be assumed.

The representation of the magnets in the machine description format, which uses XML files, has been simplified. Previously all magnets shared the same attributes, although some were used only by dipoles and not quadrupoles, and others were specific to corrector magnets. Now each magnet has its own attribute bucket.

A feature to save the lattice into files has been implemented, finally allowing to modify the lattice from Open XAL applications and save the result.

We have also replaced the plugins that handle EPICS communications (*jca* and *pvaccess*) by a new one called *epics7*. *Epics7* implements both Channel Access and PV Access protocols, since only one plugin can be used at a time. The user can decide which protocol to use by appending a prefix to the PV names, either *ca://* or *pva://*.

Finally, the ChannelFactory class that wraps around the EPICS plugin now has a test mode, which will append a suffix to all PV names, both when creating a channel in a server or when used as a client. This feature is very useful for testing purposes to avoid collisions with the real PVs.

Java FX framework

All applications developed at ESS use the Java FX libraries for the graphical user interface. We developed an Open XAL extension that provides a framework to create the applications, and implements basic functionalities common to all applications. The features include a top menu bar, methods to load the default or change the accelerator, integration with the logbook, widgets, and plotting tools.

The integration with the ESS logbook, which currently is based on elog [6], has been improved in terms of performance and design, including an HTML editor. A snapshot of the dialog use to create new entries is shown in Fig 2.

The framework also includes widgets that can be used by the applications. So far, we have included 3 widgets based on a TreeView: AcceleratorTreeView, ComboSequencesTreeView, and PowerSuppliesTreeView. They display the accelerator sequences and elements, combo sequences and power supplies, respectively, and provide methods to bind actions on the TreeView to other events in the GUI.

The development of the XAOS framework [7] has been halted, but the plotting library is ready and used by all our Java FX applications.

Finally, two themes are available for Java FX applications: a default 'light' theme and a 'dark' theme. The user can switch the theme for each application by using the option in the top menu bar.

Applications

As stated above, we have upgraded our applications to make use of the Open XAL Java FX framework and the

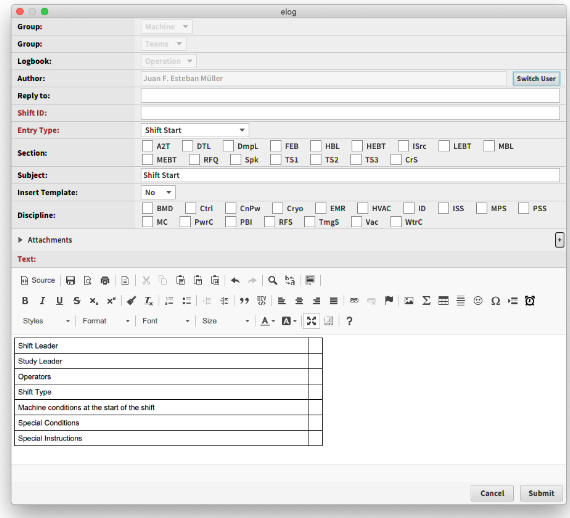

Figure 2: Example of the Java FX dialog to post a new entry in the ESS logbook.

XAOS plotting library. We have also developed several new applications.

One of the new applications is called LatticeEditor, which allows the user to browse and edit the accelerator optics files in a more user-friendly way than editing the XML files directly. This application makes use of the new saving feature to store the modified lattice to XML files. A screenshot of the LatticeEditor can be seen in Fig. 3

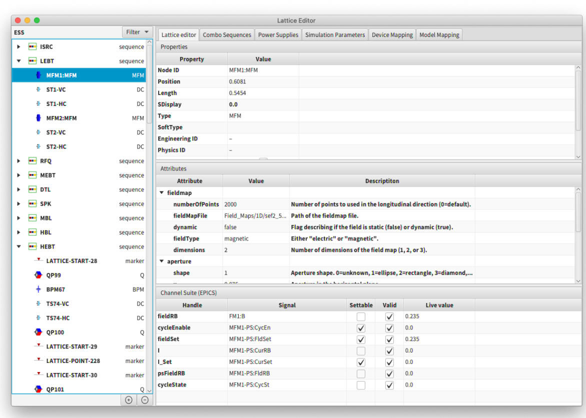

Figure 3: Screenshot of the Lattice Editor application. The TreeView on the left shows the accelerator sequences and elements, and on the right pane one can see the details of the elements. Other tabs allow users to modify other properties of the lattice.

The VirtualMachine application is a headless tool that allows simulation of the control points of the accelerator for application development. It generates an EPICS server which creates PVs for all the signals defined in the lattice files, and connect the setpoints with their readbacks. It also

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-WEPAB319

makes use of the ChannelFactory test mode to avoid collision of PV names with the real signals.

Another GUI-less application is the ModelServer. This tool runs the Open XAL model at approximately 1Hz and serves results like trajectory and beam size as a set of PVs. It runs the simulations for both: the design parameters and for 'live' parameters, obtained from the real accelerator or from a VirtualMachine instance. This tool is useful, for example, to display results of simple simulations in an OPI.

Some of the applications that require more intense simulations and comparison between real data and the machine model often start as scripts in Python. It is possible to access most of the classes available in Open XAL in Python, through an interface using JPype [8], offering a powerful, flexible and fast prototyping platform for development. Both an RF phase scan and transverse matching application are currently in this state, and will be tested and used soon. After the first tests are finished we will decide what needs to be ported to fully stand-alone Java Applications.

For the RF phase scan application, we have tested the use of REMI [9] as a potential GUI framework. REMI allows for quick development of GUI in Python, which is rendered in a web browser. While having the application accessed through a web browser provides several obvious benefits, we found that these benefits are not hugely important in a homogeneous control room environment, while the drawback of depending on Java, Python and HTML/CSS interacting in harmony reduced the initial productivity gains. We found it to be an interesting framework in general, but do not currently plan to pursue this approach to GUI development further.

Java Environment

At ESS, we have decided to use the latest Long Term Support (LTS) release for all Java applications in the control room. Currently, the latest LTS version is 11, but Java 17 will be released later this year and we will probably upgrade our applications to use that version, including Open XAL.

Regarding the Java FX framework, since we deploy the SDK files together with our Open XAL distribution and it does not affect other applications running in the control room, we decided to use the latest available version, which currently is Java FX 16.

ACKNOWLEDGEMENT

This work was partially support by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. This research was supported by the DOE Office of Science, Basic Energy Science, and Scientific User Facilities. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

REFERENCES

[1] J. G. Wang, "Particle optics of quadrupole doublet magnets in Spallation Neutron Source accumulator ring," *Physical Review Special Topics - Accelerators and Beams*, vol. 9, no. 12, Dec. 2006. doi:10.1103/physrevstab.9.122401

[2] J. A. Holmes, T. Gorlov, N. J. Evans, M. Plum, and S. Cousineau, "Injection of a self-consistent beam with linear space charge force into a ring," *Physical Review Accelerators and Beams*, vol. 21, no. 12, Dec. 2018. doi:10.1103/physrevaccelbeams.21.124403

[3] A. P. Zhukov *et al.*, "Open XAL Status Report 2019", in *Proc. 10th Int. Particle Accelerator Conf. (IPAC'19)*, Melbourne, Australia, May 2019, pp. 3341–3344. doi:10.18429/JACoW-IPAC2019-WEPTS096

[4] A. P. Zhukov *et al.*, "Open XAL Status Report 2018", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.-May 2018, pp. 3388–3391. doi:10.18429/JACoW-IPAC2018-THPAK069

[5] A. P. Zhukov *et al.*, "Open XAL Status Report 2017", in *Proc. 8th Int. Particle Accelerator Conf. (IPAC'17)*, Copenhagen, Denmark, May 2017, pp. 4676–4678. doi:10.18429/JACoW-IPAC2017-THPVA093

[6] S. Ritt, https://elog.psi.ch/elog/.

[7] C. Rosati and E. Laface, "New Approach in Developing Open XAL Applications", in *Proc. 8th Int. Particle Accelerator Conf. (IPAC'17)*, Copenhagen, Denmark, May 2017, pp. 4043–4046. doi:10.18429/JACoW-IPAC2017-THPAB137

[8] JPype Software, https://github.com/jpype-project/jpype

[9] REMI Software, https://github.com/dddomodossola/remi