ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-MOPWI024

ACCELERATOR ONLINE SIMULATION PLATFORM*

P. Chu#, Y. Zhang, MSU, East Lansing, MI 48824, USA

Abstract

A platform for accelerator online beam simulation has been established for various accelerators. This modelling platform supports multiple simulation codes for different sections of the complex machine which cannot be properly modelled with a single online simulation tool. Model data for the platform is stored in a relational database which is designed to accommodate most simulation data. The stored data is accessible with physics intuitive data API (Application Programming Interface). Presently, the platform is supporting Open XAL, MAD-X and IMPACT simulation codes. In addition to the model data storage and access, tools such as data comparison and simple graphing capability are also included in the platform.

INTRODUCTION

Modern accelerators with high precision requirements often rely on good online model for their machine tuning. However, each machine is likely having its own unique physics simulation issues which requires different modelling tools to deal with. For example, some lower energy machines are sensitive to space charge effects which need a good space charge code, while higher energy machines can be calculated without space charge. This model dependent machine simulation usually requires customized software code for each simulation tool. On the other hand, a properly designed platform with minimal overhead can provide a universal solution for various machines. Such a platform has been developed and prototyped.

One relatively easy way to achieve a platform for various modelling tools is shown in Fig. 1 as centre piece of the physics application architecture. The key components in this approach for the online simulation platform are a data container which can accommodate model data for different modelling tools, a set of common data access APIs for the data container, and data adapters between the data container and various modelling codes. For this model platform implementation, the model data container is a relational database, as shown in the middle of Fig. 1, which will be described in the next section. Besides the database itself, it is necessary to provide a set of data access API instead of tedious SQL (Sequential Query Language) statements for querying the saved model data. For each modelling tool, the input data should be generated programmatically so the model run can be automated, while the model output data should be parsed and saved into the database via convenient API. These tools are the model data adapters which will be explained in details later.

Additionally, both offline and online model data can utilize the same platform. Open XAL [1] as a physics application framework provides the interface to the control systems. Even fitted empirical machine model can be included in this platform.

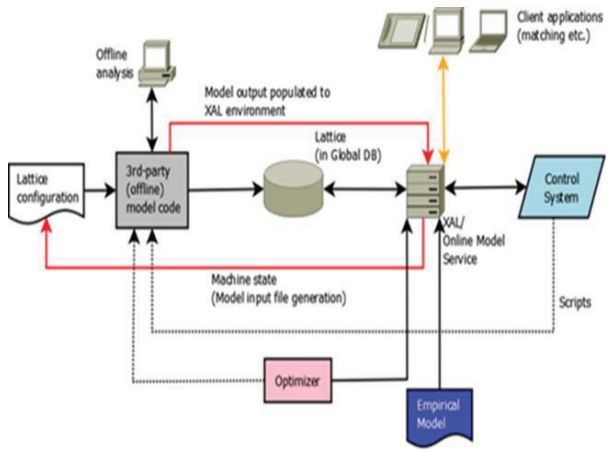

Figure 1: Online Model Platform schematic diagram.

MODEL DATABASE

The model database used for the data container is part of an accelerator database effort from the collaboration among FRIB, NSLS-II (National Synchrotron Light Source II) and ESS (European Spallation Source). The blue print for this database was from IRMIS-3 [2]. Originally there were two separate sub-schemas, one for lattice data including device settings and the other one for model run data. Because there is significant overlap between these two data areas, it is efficient to merge them into one schema to minimize data duplications. There are 20 tables in this database schema with a potential hook point to another part of the accelerator database, namely the Configuration Database which is for accelerator hardware data. The key feature for this database is to use "property name and value" pair to store most model data, i.e. the model parameter names are also part of data as opposed to the traditional way of defining the parameter names as database column labels. The detail of the Lattice and Model database tables can be found in the Lattice/Model Schema and API Release Note from the Open EPICS Model download area [3].

DATA ACCESS API

Besides the database itself, there is a set of data access APIs also developed as part of the Lattice and Model package. The software package is Java based which takes full advantage of the latest Object to Relational Mapping (ORM) technology, a mapping mechanism between

*Work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661
#chu@nscl.msu.edu

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-MOPWI024

objective approach of the Java programming language and the relational database structure, in JPA (Java Persistence API) included in standard Java releases since Java version 7.

The data access API discussed here is modelling tool independent and is solely for the purpose of getting model data in and out of the database. There are two groups of software packages: one is site-independent named with org.openepics… and the other one is for FRIB only whose package names starting with edu.msu.frib.xal… Because the classes in edu.msu.frib.xal… are all calling XAL API, the package names contain "xal". XAL/Open XAL provides a rich set of APIs to facilitate many general-purpose functions for physics data. A set of entity classes (org.openepics.model.entity package) which performs the ORM functions were programmatically generated using NetBeans Integrated Development Environment (IDE), which is based on JPA, i.e. each database table has a corresponding Java class for its description. A set of convenient query APIs are included in the org.openepics.model.api package and org.openepics.model.extraEntity package; these convenient APIs are a collection of commonly used queries. A few supporting packages such as edu.msu.frib.xal.exl2DB are for modelling elements, beam parameters and lattice-element relationship. Since many of the features provided by the data access API rely on Open XAL, it is necessary to have an API to generate Open XAL optics configuration file programmatically. The generator for Open XAL initialization files is located in edu.msu.frib.xal.db2xal package and this program can generate 3 files: *.impl for hardware type definition, model.params for model initial condition parameters, and *.xdxf for optics details. With these APIs, the model data can enter the database and client applications can extract saved model data out of database easily.

MODEL DATA ADAPTERS

In order to interact with various modelling tools, it is needed to generate each modelling tool's input file(s) programmatically and extract information from its simulation results. Fortunately, Open XAL has a well-defined accelerator lattice and optics data structure with fully support API. Based on the Open XAL API, there are utility tools for generate a few modelling tools' input files. Previously, MAD-8 and Trace-3D were fully supported with DYNAC [4] partially supported. We have added MAD-X [5] and IMPACT-Z support to the input file generation collection in Open XAL. Note that the MAD-X lattice generator uses similar approach as Open XAL for drift space calculation, i.e. only specify modelling elements' locations and lengths while letting the modelling tools to calculate the drafts in between at runtime, as opposed to the MAD-8 convention of explicitly defining the drafts. For the MAD-X model run parameters, presently Polymorphic Tracking Code (PTC) run commands are selected as it supports acceleration in a linac.

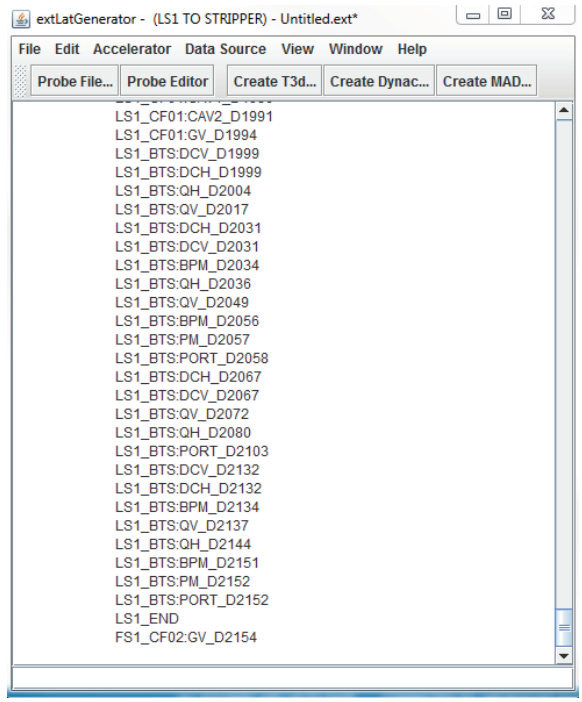

Figure 2: Open XAL Lattice Generator.

There is an Open XAL GUI application (shown in Fig. 2) for these lattice generators but it is much more useful to access the lattice file generation via API which can be called within user's own routines. The newly added code support has not been added to the Open XAL GUI application but only API available.

The APIs for parsing model output files and upload the data to RDB is included in the Lattice and Model Database package because they are independent from Open XAL; the APIs are currently in the edu.msu.frib.xal.model2DB package which may be renamed to org.openepics.model2DB in the future because they are site independent. Presently, supported model data parsers and database uploaders include Open XAL, MAD-X, and IMPACT-Z.

APPLICATIONS

With the model platform ready, some applications can take advantage of it. Because the platform can accommodate data for several modelling tools, one direct application is to compare among these models. Also, it is necessary to validate the Open XAL Online Model against other well-established codes. Several simulation codes have been chosen for Open XAL to compare to: IMPACT-Z, MAD-X, and COSY Infinity. Previously, Trace-3D was also selected for comparing to the original XAL Online Model. It is usually quite tedious to write parsers, prepare input files, and convert units among various simulation codes.

The actual benchmark work for Open XAL is still ongoing but a proof-of-principle comparison has been done. In Fig. 3, it shows the difference between Open XAL and MAD-X β function in transverse direction x for

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-MOPWI024

a small section of the FRIB Folding Segment 1 right before the charge stripper; the section contains quadrupoles and a bend dipole. The entire process was done with JYTHON scripts. First, a JYTHON script generate MAD-X input file using Open XAL API described above; then run MAD-X against the generated input file and save the simulated data into the Lattice/Model DB with the data access API; in parallel, run Open XAL and also save its result to the same Lattice/Model DB via the data access API; and finally extract both models' data and plot the difference. One can easily extend the JYTHON scripts for further detail benchmark comparison work. The data plot software

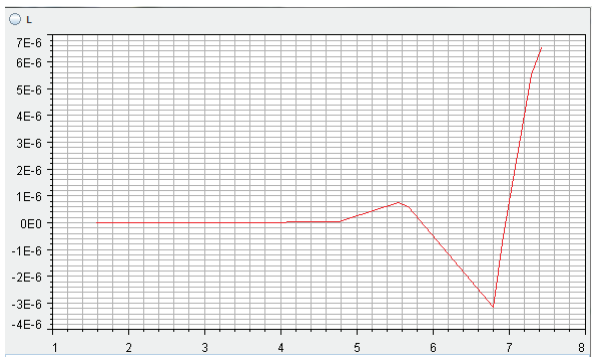

Figure 3: Benchmark Example: Difference between Open XAL and MAD-X.

shown in Fig. 3 is provided by Open XAL.

The overall performance of this benchmark system is very satisfactory for its general simulation code coverage and easy to use features. Additionally, one can, in principle, convert any code's units to whatever chosen ones and save both data in the original and the converted units in the database. The database data extraction API will then take care of retrieving the data in properly converted units and compare among models accordingly.

Other example application for the platform is a swappable modelling tool virtual accelerator or machine simulator. A prototype for such a virtual accelerator has been implemented in a JYTHON script for Open XAL and MAD-X. In principle, most model-based applications can take advantage of the platform.

CONCLUSION

We reported the model platform design work in a previous IPAC [6] and the platform is now prototyped and fully functioning. The original propose also included model run-control such as multi-particle simulation run scheduling which requires much more R&D efforts for services. The present platform architecture is much simpler and can cover most of the physics application usages. The proof-of-principle work has been completed. The remaining work will be adding features and more modelling tools support to the platform which should be relatively straightforward.

ACKNOWLEDGMENT

The authors would like to thank the DISCS Database Collaboration colleagues for their help on the model database development. The authors would also like to thank Drs. J. Qiang and Q. Zhao for their help on IMPACT-Z discussion, and Dr. L. Yang for his kind advice on MAD-X usage.

REFERENCES

[1] Open XAL project web site:http://xaldev.sourceforge.net
[2] http://irmis.sourceforge.net
[3] http://sourceforge.net/projects/openepicsmodel
[4] http://dynac.web.cern.ch/dynac/dynac.html
[5] http://mad.web.cern.ch/mad
[6] P. Chu et al., "Online Physics Model Platform," TUPPC048, IPAC'12, New Orleans, USA (2012).