TOOLCHAIN FOR ONLINE MODELING OF THE LHC

G.J. Müller, X. Buffat, K. Fuchsberger, M. Giovannozzi,
S. Redaelli, F. Schmidt, CERN, Geneva, Switzerland

Abstract

The control of high intensity beams in a high energy, superconducting machine with complex optics like the CERN Large Hadron Collider (LHC) is challenging not only from the design aspect but also for operation towards physics production. To support the LHC beam commissioning, efforts were devoted in the design and implementation of a software infrastructure aimed at using the computing power of the beam dynamics code MAD-X in the framework of the JAVA-based LHC control and measurement environment. Alongside interfaces to measurement data as well as to settings of the control system, the best knowledge of machine aperture and optic models is provided. In this paper, we will present the status of the toolchain and illustrate how it has been used during commissioning and operation of the LHC. Possible future implementations will be discussed.

MOTIVATION

Since the startup in 2009 the LHC has been smoothly commissioned and reached an outstanding performance. The 2011 target integrated luminosity of 1.0 fb^{-1} was already reached in June and currently increased to a total of 4.0 fb^{-1}. This was to a great extent achieved by the two-step reduction (initial 1.5m and 1.0m as of 09'2011) of β*, the extremal optical β-function in the interaction point (IP). New settings had to be generated for the LHC for each of these β* reductions. A flexible, extendable and maintainable framework was required, which allows the transfer of simulation data to the control system and verification of the generated settings which are based on this input.

The complexity of the LHC itself, combined with the required possibility to handle of a variety of complicated operational configurations, especially in the IP's, raised a demand for online modeling. There is an interest in simulations representing the current status of the machine to determine e.g. the beam position at all machine elements or the beam size according to the configuration resident in the machine. Furthermore the simulation of changes before they are driven to the machine is essential to estimate their impact concerning restrictions imposed by the aperture of the machine and machine protection.

In this paper we present the toolchain which was developed to cope with the above mentioned challenges. Applications are provided to address individual problems and a core software structure was established to provide a simulation environment used by the *Aperture Meter* [1]. The work is based on an initial implementation of a framework for the commissioning of the transfer lines [2] which was extended for the LHC commissioning [3] in 2010.

ENVIRONMENT

The main purpose of the online modeling is to establish the connection between the accelerator simulation and the real machine. The relevant systems the toolchain has to interface with, are presented in the following.

Accelerator Physics Simulation MAD-X is the de facto standard accelerator lattice design and simulation software at CERN. A large amount of machine lattice models and optics for MAD-X are available and maintained by the accelerator physics community. Alongside other features the software can calculate the optical functions and the aperture information for a given machine configuration. MAD-X input is defined via its own scripting language. The optics of an accelerator is described by a specific set of optics functions (twiss) calculated by MAD-X. The optics include the design orbit and the β-function. To calculate an optics in MAD-X, a strength value has to be defined for each lattice magnet. JMad the JAVA API to MAD-X [4] was chosen as the interface to the computing power of MAD-X. JMad provides object-oriented access to MAD-X and allows to define and maintain machine and optics definitions in `JMadModelDefinition`'s hiding the complexity of the script based interaction.

Control Infrastructure Access to the control system and online measurement data of the LHC is provided by the LHC Software Architecture (LSA) [5]. Two essential entry points are exposed to clients: The JAVA API for Parameter Control (JAPC) to retrieve measurement data and the LSA Client API for access to all LSA functions related to optics and settings. The main concepts *Parameter* (settable or measurable entity), *Context* (collection of parameter values over a period of time), *Setting* (representation of a parameter value) and *Trim* (manipulation of a number of settings) are explained in detail in [5]. Besides the online measurement data from JAPC, the CERN Accelerator Logging Service (CALS) [6] provides a JAVA API to retrieve measurement data from the logging database.

Optics (strengths and twiss outputs) have to be uploaded to the LSA database as input for the settings generation. A context (in the case of the LHC: *Beamprocess*) is defined by the sequence of optics that should be established in the machine when the context settings are driven to the hardware. In the example of the LHC squeeze all required optics are defined from the injection optics to the current 1m β* optics at the end of the squeeze beam process. The LSA settings generation is used to generate all required settings for the LHC devices. In the case of the power converter settings all parameter settings in the hierarchy are generated

from the optics strength input: PC strengths (K) down to the PC reference currents (IREF) which are driven to the hardware.

Adjustments of physics parameters like tune, chromaticity, coupling as well as orbit steering with bumps for beam crossing and separation in the IPs is done via knobs. A knob is a LSA parameter which has a unique hierarchy of dependent K parameters called knob definition. Each assigned parameter K_i is connected to the knob by an optics dependent multiplication factor f_i. Therefore a trim on the knobs setting value A is propagated to the dependent parameters like $\Delta K_i = \Delta A \cdot f_i(\text{active Optics})$.

REQUIREMENTS

The following requirements have to be fulfilled by an online modeling toolchain:

1. handling of the different machine optics configuration and data transfer to LSA
   (a) definition of optics models and tools to follow their evolution
   (b) automatic creation of optics and knobs
   (c) transfer of optics/knob data to the control system
   (d) verification of generated settings in LSA
2. provide the information of the mechanical aperture for the whole machine, including the information available from cold bore measurements and the movable device positions (e.g. collimator's and roman pots)
3. create a simulated machine, based on the current state and settings driven to the hardware
4. combine measurement data with the simulation data (e.g. orbit interpolation)
5. simulate the effect of parameter variation (virtual knobs)
6. support the extraction of all control system settings and measurements required to build a simulated model for any given time in the past

ARCHITECTURE

Package Structure Suitable for the variety of systems the online model (OM) toolchain needs to interact with, the functionality is distributed over eight 'library/service' projects and three main application projects as shown in Fig. 1.

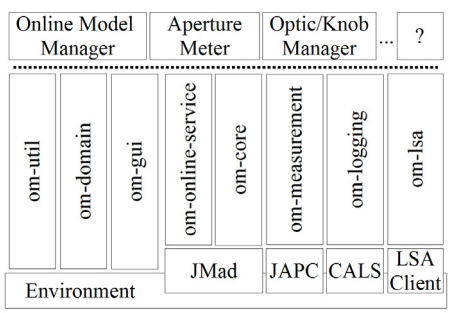

Figure 1: Online Modeling Toolchain Project Overview.

This modular design, allows to maintain and improve each part separately. The *om-util* project contains utilities for file interaction, date/time formatting, etc. The *om-domain* project contains data objects which are shared toolchain wide. Common GUI elements are gathered in *om-gui* project. The *om-online-service* project provides the online simulation from LSA settings and is explained in more detail later on. Calculations and the aperture model are implemented in the *om-core* project. The *om-measurement*, *om-logging* and *om-lsa* projects contain interfaces to interact with their respective systems, providing access to measurement or setting data. Online model applications are independent as far as possible from external packages and depend on the toolchains own packages according to the functionality required. By following this approach improvements, adjustments and bug fixes need to be performed only in one place and all application profit.

Build Project Maintaining eleven projects and ensuring that they are released in the right sequence to correctly resolve the dependencies, can be a cumbersome and time consuming task. Therefore a project was introduced into the OM framework which allows with a one click operation to release all OM packages in the correct sequence. It also provides some tools to check the currently released versions and when they have been released.

Continuous Integration and Testing To ensure that the functionality provided by the online model toolchain resumes performing in the expected manner, unperturbed by updates of software the framework depends on (JMad, LSA, ...), automatic code testing and building has been introduced. Therefore suited project/packages have been equipped with JUNIT [7] test cases covering the critical functionalities and automatic build plans have been created on the continuous integration server BAMBOO [8] provided by the CERN BE/CO group. This set-up allows to create dependency relations between the build plans of the required projects and the build plans of the OM projects. Code changes committed to the code repository trigger the building and testing of all projects related to the changed project.

FUNCTIONALITY

Optics and Knob Management

Optics Models To cope with the increasing number of optics required for simulation and operation (a total of 133 optics), it was decided to define them in `JMadModelDefinitions`. This interface is defined by JMad and is used to create a `JMadModel` which is the facade to one dedicated MAD-X process. A `JMadModelDefinition` defines the required MAD-X files to initialize the desired machine and a collection of `OpticDefinitions` which can be loaded to the `JMadModel`. An `OpticDefinition` is a named entity that

defines the required set of MAD-X input files and the sequence in which they have to be loaded in MAD-X to specify a certain machine optics. Mechanisms are provided in JMad to save the `JMadModelDefinitions` to XML files and synchronize the required input files with the optics repository, maintained by the accelerators and beam physics group. Dedicated projects have been created containing the `JMadModelDefinitions` and all required input files. These projects are in the controls software repository under version control.

When new optics are required, the corresponding `OpticDefinitions` are created and added to the proper `JMadModelDefinition`. The XML file is generated, the required input files are loaded from the optics repository into the project folder and the changes committed to the software repository. In the following build step all files are packaged into a jar file. Client application simply have a dependency to the jar which contains the required `JMadModelDefinitions`.

This set-up allows client applications to use MAD-X computation completely independently of MAD-X syntax and required input files. New optics can be used without code changes in the clients. The model definition jars represent a common source of optics definitions to ensure that simulations are performed using the same input. The versioning of the model projects allows to perform simulation on historic input data. The interaction with JMad is described more in detail in [4] and goes beyond the content of this paper.

Optic/Knob Manager Application The generation and transfer of simulation data to the control system LSA is performed using the Optic/Knob Manager. As shown in Fig. 2 it uses JMad as calculation engine and the LHC JMad model package as source for optics definitions. The available optics models are: the nominal optics model for physics production at 1m β*, the Achromatic Telescopic Squeeze (ATS) Scheme [9] optics model and the 90m β* Un-Squeeze [10] optics model for e.g. the **TOT**al cross section, **E**lastic scattering and diffraction dissociation **M**easurement (TOTEM) experiment.

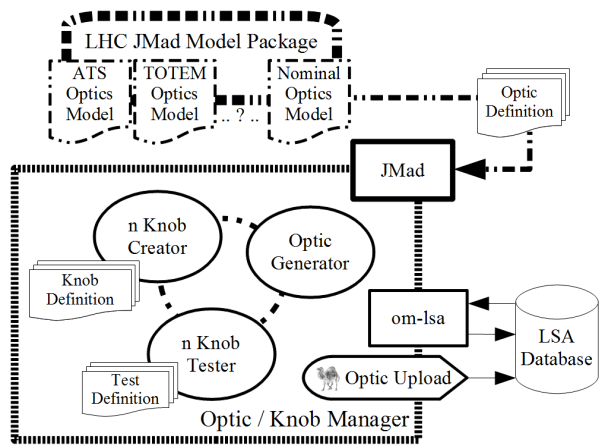

Figure 2: Optic and Knob Manager.

A simple GUI is provided in which the user can select the optics model and a set of optics to generate and upload. The `OpticGenerator` is responsible for loading the requested optics to JMad, executing possible checks for tune and chromaticity and writing the optic functions to a twiss file. A set of PERL scripts is then executed in a JAVA process to upload the generated optic data to LSA.

The same GUI also allows the selection of knobs to create for a selection of optics. The available knobs are defined by `KnobDefinitions` which have an assigned `KnobCreator` which is used to create the knob from the `JMadModel` passed. A `KnobManager` is responsible to loop over all requested optics, load the optic to a `JMadModel` and for each requested knob initialize the assigned `KnobCreator` with the model and trigger the creation process. `KnobDefinitions` are available for knobs defined by flags in the MAD-X input files (separation and crossing knobs), knobs created by MAD-X matching routines for global machine parameters like tune and chromaticity as well as for 4-corrector knobs used for local adjustment (luminosity leveling knobs). This design allows the extension of the set of available knobs by creating solely a new `KnobDefinition` and the required `KnobCreator` without touching the creation routine. `KnobTesters` are defined to ensure that newly created knobs follow definitions already present in LSA, are created correctly or produce the desired setting change when trimmed in simulation.

The import of knobs defined as a set of name value pairs is supported to define arbitrary knobs in LSA. Names can be one of the following: LSA parameter names, LSA device names or MAD-X strength names. The `om-lsa` package is finally used to transfer knob data to the control system as well as retrieve available knob definitions from the LSA database. The possibility to copy existing knob definitions to other optics is provided as well.

Setting Verification

To ensure that the settings generated in LSA are complete and correct and to apply possible corrections, the following tools have been developed in the OM toolchain based on the extraction functionality provided in the `om-lsa` project.

Beamprocess Scanner This tool can scan over the full power converter setting functions defined in a beamprocess. A snap shot of the settings is extracted at given times in the setting functions of a beam process. The settings are translated to MAD-X strength values and loaded in JMad to calculate the optic functions. The key optics parameters (tune, chromaticity, beta-beat...) can then be evaluated as function over the beam process duration. The tool has been essential for the setting optimization for the LHC squeeze [11]. It can be used to verify that the settings are valid in general (e.g. they produce reasonable tune and chromaticity evolutions) and to precalculate feed-forward correction for tune and chromaticity [12].

Online Model Manager Next to other functions, it is possible to visualize the sequence of optics in a beamprocess together with the nominal evolution of the optics key parameter values (tune, chromaticity and β*). The display acts as selection interface for the time in the beamprocess where the power converter settings should be extracted, either to display or to write them to a file in MAD-X input format. When loaded to display, the generated settings can be checked with respect to the nominal settings expected from the optics input saved in LSA. The settings saved to file can be loaded in a MAD-X script to calculate and display the optic functions. A plotting environment is provided that allows the display of multiple e.g. orbits on top of each other. The evolution of knob bump shapes can be visualized by a sequence of knob trims performed with the LSA Trim application, followed by an extraction and a MAD-X run based on the extracted settings.

Online Aperture Model

The *static* aperture model is implemented in the `om-core` package. It provides the dimension of the clearance in the vacuum layout in transverse coordinates. The data is based on the theoretical aperture with the cold bore measurement results. The model was extended with the current positions of the movable devices (collimator's and roman pots), read online from the hardware. This functionality is provided by the `om-measurement` package. The measured data from the movable devices are translated into aperture information by using the movable device information (angle, type, beam it acts on,...) that are available from the `om-lsa` package. The aperture model is an essential ingredient for the aperture meter [1].

JMad Online Service

The `JMadOnlineService` is implemented in the `om-online-service` package as the online extension to JMad. The current configuration of the machine is simulated based on the active optic and the optic dependent definitions and current settings of a set of knobs that represent the relevant orbit configuration (separation, crossing, luminosity leveling knob settings and a set of aperture scan knobs). The active optic is the one that was the basis for the generation of the settings that are currently resident in the machine hardware. A `ModelManager` allows to access, manipulate and retrieve data from the underlying `JMadModels`. Two manipulators are provided to transfer settings from LSA to the underlying `JMadModels`: One can update the models to the current state resident in the machine. The second on loads the settings that have been resident at a given point in time. This service is the basis for the aperture meter [1].

Measurement Data Treatment

Regarding measurement data the online modeling toolchain provides two more features: The interpolation of the measured orbit to all machine elements and the determination of the current state of the machine based on online data from the collimator hardware and from data distributed over the timing system. The orbit interpolation is implemented in the JMad core package and uses transfer matrices to calculate the orbit positions between two adjacent monitors. The measured orbit data are either received by a JAPC subscription to the orbit feedback service unit, or by extraction from the logging database. The identification of the online machine state is performed by merging the status and current time in the setting function retrieved from a given collimator front-end with the information about the beam mode distributed over the LHC timing system.

CONCLUSION AND OUTLOOK

The basic architecture and features of the online modeling toolchain have been presented. It was successfully used for the preparation of all LHC optics configuration for physics production as well as machine development studies and provides a good basis for future developments. The usability was shown in applications presented in this paper and as basis for the aperture meter [1]. Further developments will include the full integration of new features provided by LSA and the toolchain itself and changes towards a strict distribution of the functionality into the packages of the toolchain architecture.

ACKNOWLEDGMENTS

The authors like to thank M. Lamont, M. Strzelczyk, R. Tomas and J. Wenninger for fruitful discussion and useful advice. This work was supported by the Wolfgang-Gentner-Programme of the Bundesministerium für Bildung und Forschung (BMBF).

REFERENCES

[1] G. Müller et al., *Aperture Meter for the Large Hadron Collider*, these proceedings.
[2] F. Schmidt et al., *LHC On-Line Modeling*, Proceedings of PAC'07, 2007.
[3] G. Müller et al., *The Online model for the Large Hadron Collider*, Proceedings of IPAC'10, 2010.
[4] K. Fuchsberger et al., *Status of JMad, the* JAVA*-API for* MAD-X, Proceedings of IPAC'11, 2011.
[5] M. Lamont et al., *LHC Era Core Control Application Software*,Proceedings of ICALEPCS'05, 2005.
[6] C. Roderick et al., *The CERN Accelerator Measurement Database: On the Road to Federation*, these proceedings.
[7] JUNIT, `junit.sourceforge.net`.
[8] BAMBOO, `atlassian.com/software/bamboo`.
[9] S. Fartoukh, *An Achromatic Telescopic Squeezing (ATS) Scheme For The LHC Upgrade*, Proceedings of IPAC'11, 2011.
[10] S. Cavalier et al., *90m Optics Commissioning*,Proceedings of IPAC'11, 2011.
[11] X. Buffat et al., *Simulation of linear beam parameters to minimize the duration of the squeeze at the LHC*, Proceedings of IPAC'11, 2011.
[12] M. Pereira et al., *Feed-Forward in the LHC*, these proceedings.