

HIGH LEVEL CONTROLS FOR THE EUROPEAN XFEL

Lars Fröhlich, Bolko Beutner, Winfried Decking, Olaf Hensler, Raimund Kammering,
Torsten Limberg, Sascha Meykopff, Josef Wilgen, DESY, Hamburg, Germany

Abstract

The European X-Ray Free-Electron Laser (XFEL) will generate extremely short and intense X-ray flashes from the electron beam of a 2.1 km long superconducting linear accelerator. Due to the complexity of the facility and the sheer number of subsystems and components, special emphasis needs to be placed on the automatization of procedures, on the abstraction of machine parameters, and on the development of user-friendly high-level software for the operation of the accelerator. The paper gives an overview of the ongoing work and highlights several new tools and concepts.

INTRODUCTION

The European X-ray Free-Electron Laser (XFEL) is a research facility currently under construction in close collaboration between the European XFEL Facility GmbH¹ and DESY² in Hamburg, Germany [1–3]. It consists of a superconducting linear accelerator delivering an electron beam with particle energies up to 17.5 GeV and a total beam power up to ~600 kW, several long undulator sections enabling the generation of extremely brilliant X-ray pulses at wavelengths down to 0.05 nm, beamlines and setups for photon science experiments, and the associated infrastructure. An overview of the control system architecture is given in [4].

Compared to other accelerator facilities operated by DESY, the XFEL presents itself as a system of unprecedented complexity. It will generate ~600 μ s long *macropulses* of up to ~3000 bunches at a repetition rate of 10 Hz. Individual bunches out of these trains will then be distributed to several undulator beamlines and beam dumps using a system of fast intra-train kickers. Even the sheer number of subsystems and components—more than 1400 electromagnets for beam guidance and focusing, more than 450 beam position monitors, to pick two popular examples—makes it clear that a high degree of automatization will be needed to operate the machine efficiently. Control software should offer high level abstractions for all important machine parameters rather than forcing operators to work at the level of individual technical subsystems.

We have formed an interdisciplinary team of physicists, engineers, and computer scientists from multiple DESY groups with the goal of establishing the best possible operability for the machine. The scope of this *high level controls* (HLC) effort is a deliberately broad one. It includes the preparation of user friendly tools for various physical or technical tasks in the control room as well as the development of feedbacks and of middle layer servers. A lot of work is invested to

improve the entire software stack of libraries, toolkits, and network services. This paper gives an overview of the various projects and activities, starting at the infrastructure level and closing with a discussion of stand-alone tools. For obvious reasons, we limit ourselves to the description of a few selected examples for each category.

PROTOCOLS AND INFRASTRUCTURE

Three main control system protocols are used at the XFEL: DOOCS [5, 6], Karabo [7], and TINE [8, 9]. Much work has been invested to improve the interoperability of these protocols [10], so that the problem of network communication across protocol boundaries is mostly solved. For HLC applications, efforts have concentrated on the development and improvement of interfaces for Matlab [11] and Python.

A *configuration database* has been set up as a central network service using DESY's Oracle 12c database system. It stores a complete list of beamline components and associated information such as calibration data. In this way, inconsistencies in the configuration of distributed servers can be minimized [12].

Finally, a *virtual accelerator* has been created by duplicating most of the middle layer and data acquisition software of the XFEL and exchanging the real front-end servers with dummies generating simulated data. This virtual XFEL has proven invaluable for the testing of high level software [13].

LIBRARIES AND TOOLBOXES

A multitude of libraries has been created to support the development of middle layer servers in C++. These projects include generic class libraries for access to various accelerator components, collections of algorithms for numerical tasks and fitting routines, classes for image analysis and conversion, an easy-to-use interface for retrieving information from the central configuration database, and various utility functions.

Matlab has traditionally been the main high-level interpreted language for physics applications at DESY. A broad assortment of utility functions is therefore available and actively maintained; a report on the DataGUI library for the design of advanced graphical user interfaces is found in [14]. We are currently focusing on the implementation of Matlab classes for access to specific accelerator components.

Recently, Python has been introduced as a second high-level language. Its main application at the moment is the rapid development of graphical user interfaces using the QT framework. Toolkit and library support is still fragmentary, but expanding continuously.

In general, we are encouraging the use of modern language standards where possible. For example, big parts of our C++

¹ European X-Ray Free-Electron Laser Facility GmbH, Albert-Einstein-Ring 19, 22761 Hamburg, Germany

² Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany

library code use C++11 and all of our Python developments are based on version 3.4 of the language.

MIDDLE LAYER SERVERS

The XFEL is equipped with lots of front-end servers that are distributed all over the accelerator, undulator, experimental, and infrastructure buildings. These front-ends establish the interface between hardware components and the controls network, fulfilling functions such as data acquisition from ADCs, communication with field buses, controlling actuators and the like.

Above this level, another system of servers pre-processes data and offers advanced functionality. In contrast to the distributed front-ends, these *middle layer servers* are running on only a few powerful computers in central locations. Among other considerations, this approach allows data to be streamed and processed via a powerful shared memory based data acquisition (DAQ) system in a fully synchronous manner [15]. A major part of the HLC effort is focused on the development of such middle layer servers, with the following goals:

More physics: We are continuously integrating more physical models directly at the server level. For example, magnets can not only be controlled via their power supply current, but also through the setting of deflection angles or fields under automatic tracking of hysteresis effects. Radiofrequency (RF) stations can not only be controlled via amplitude and phase, but also by specifying a desired energy chirp on the electron bunch. Even a complete optics model with matching capabilities is provided by an optics middle layer server [16].

More automatization: Time-consuming and complex procedures should be automatized wherever possible. For example, state machine based middle layer servers handle the startup, restart, and shutdown of RF stations along with the conditioning of couplers. Another server can perform a phase scan of the RF gun and determine the optimal phase for operation.

More online measurements: The implementation of measurement procedures at the server level makes it possible to provide a bunch-synchronous measurement of beam parameters like energy, slice emittance, or bunch length.

More uniformity: Some subsystems consist of device families with quite different interfaces. Middle layer servers present a uniform interface for all devices with a specific functionality—for example, the beam profile server can measure bunch profiles using scintillating screens or wire scanners, and the charge server will integrate readings from toroids and from beam position monitors (BPMs).

Less load: Because applications retrieve their data from the middle layer, there is less network and CPU load on the front-end side where infrastructure and computing resources are costly.

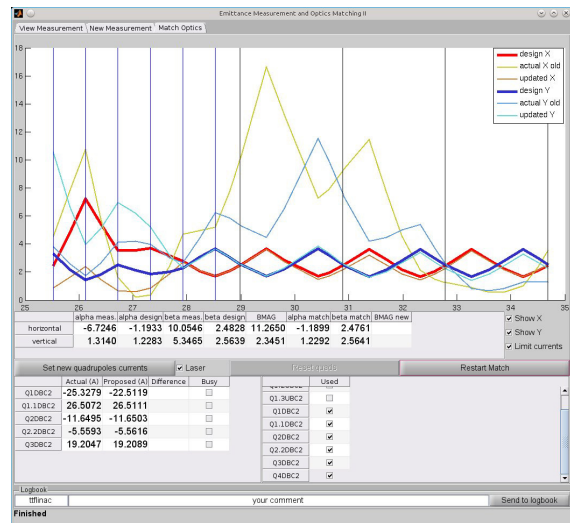


Figure 1: Screenshot of the *Emittance & Matching* tool while matching the beam optics to a periodic lattice.

More jddd: The Java DOOCS Data Display *jddd* [17, 18] is the main user interface builder used for constructing control system panels at DESY. It allows the deployment of standardized, powerful graphical interfaces with a minimum of effort. A lot of functionality that would traditionally have to be implemented as a full graphical application can therefore be integrated at the server level and use *jddd* as a user interface.

Feedbacks

Feedbacks are of enormous importance for the stability of the machine. Apart from a few very fast regulation loops acting within a single macropulse (at time scales of tens of microseconds), almost all beam-based feedbacks are implemented as middle layer software: Trajectory, bunch compression, charge, energy, and more parameters are stabilized by DAQ-based servers. Apart from the stabilization aspect, feedbacks are also important because they enable direct control of the quantity of interest—beam positions are adjusted instead of correction coil currents, beam energies instead of RF amplitudes, and so on.

APPLICATIONS

As mentioned before, a powerful platform for the rapid deployment of graphical control system interfaces, *jddd*, is available for the operation of the XFEL. *jddd* panels cover the vast majority of use cases for applications in the control room, both at the operator and at the expert level. A few cases nonetheless require the development of specialized stand-alone software—when the functionality is too complex or too seldomly used to be implemented in a server, when very specific graphical user interfaces are needed, or when the implementation in a high-level language offers other advantages such as the availability of skilled developers.

While almost all middle layer servers are written in C++ and deployed on Linux systems, applications should gener-

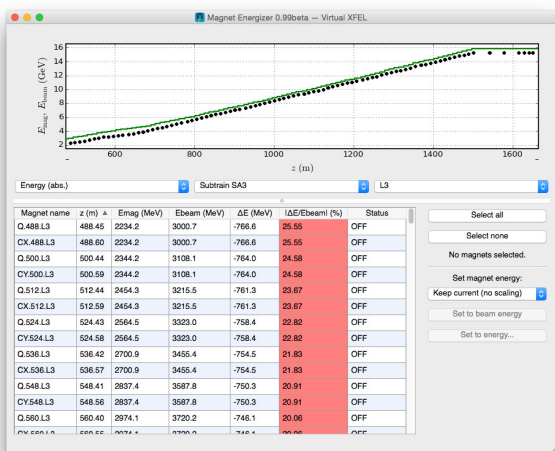


Figure 2: Screenshot of the *Magnet Energizer* tool for the scaling (setting of nominal particle energies) of magnets.

ally run on all of our three supported desktop platforms—MacOS, Windows, and Linux. Therefore, platform-independent development in high-level interpreted languages such as Matlab or Python is preferred. Java Web Start plays an important role in the platform-independent launching of such tools, and many low-level control system utilities are implemented directly in Java. Like the jddd panels, most high level applications rely heavily on the services of the middle layer server system.

The range of applications available and under development is quite broad and very similar to those used at other high gain free-electron laser facilities: To name only a few examples, an *Emittance & Matching* tool supports the measurement of the beam emittance with the 3-/4-screen method, calculates the optical functions and matches them to the design optics (Fig. 1); the *Magnet Energizer* compares the measured beam energy with the settings of magnets and adjusts them accordingly (Fig. 2); *RF Tweak* functions as a graphical frontend to a fast tracking code and can calculate longitudinal bunch profiles from RF settings and optimize bunch compression settings, taking into account radiative, wakefield, and space charge effects (Fig. 3).

CONCLUSION AND OUTLOOK

The European XFEL is a machine of considerable complexity, not only due to its sheer size and number of single components, but also because of the intricacies of its pulsed mode of operation, of its beam distribution system, and of many subsystems. To ensure the best possible operability of the entire accelerator, we are advancing high level controls developments from multiple sides:

- through the preparation of libraries, toolkits and configuration databases (at the infrastructure level)
- by providing more physical values and more automated procedures in the control system (at the server level)

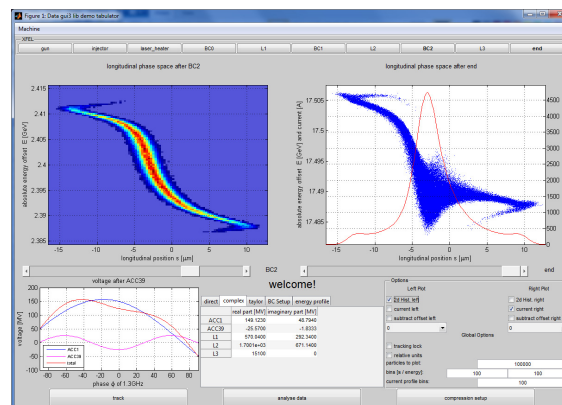


Figure 3: Screenshot of the *RF Tweak* tool for the calculation of longitudinal bunch profiles from RF settings.

- by providing powerful and easy-to-use graphical user interfaces for tasks in the control room (at the application level).

A big part of our software could be tested in the *Virtual XFEL* environment, and as many developments as possible were ported back to the FLASH [19–21] user facility, which is in many ways the predecessor of the XFEL. Experience has also been gained from the first operation of the XFEL gun with beam in February 2015. All of this makes us look forward to the injector commissioning at the end of this year with confidence.

REFERENCES

- [1] A. Aghababayan et al., “The European X-Ray Free-Electron Laser technical design report”, DESY 2006-097, DESY, Hamburg, Germany (2007).
- [2] W. Decking et al., “European XFEL post-TDR description”, XFEL.EU TN-2013-004-01, European XFEL GmbH, Hamburg, Germany (2013).
- [3] W. Decking et al., “Status of the European XFEL”, WEA04, FEL’15, Daejeon, Republic of Korea (2015).
- [4] A. Aghababayan et al., “The large scale European XFEL control system: Overview and status of the commissioning”, MOA3002, *These Proceedings*, ICALEPCS’15, Melbourne, Australia (2015).
- [5] A. Aghababayan et al., “The accelerator control system at DESY”, in: ICFA Beam Dynamics Newsletter 47 (2008), pp. 139–167.
- [6] The DOOCS website, <http://doocs.desy.de/>.
- [7] B. C. Heisen et al., “Karabo: An integrated software framework combining control, data management, and scientific computing tasks”, ICALEPCS’13, San Francisco, USA (2013), pp. 1465–1468.
- [8] P. K. Bartkiewicz and P. Duval, “TINE as an accelerator control system at DESY”, Meas. Sci. Technol. 18 (2007), pp. 2379–2386.
- [9] The TINE website, <http://tine.desy.de/>.
- [10] P. Duval et al., “Control system interoperability, an extreme case: Merging DOOCS and TINE”, PCaPAC’12, Kolkata, India (2012), pp. 115–117.

- [11] J. Wilgen and S. Meykopff, “A unified Matlab API for TINE and DOOCS control systems at DESY”, PCaPAC’ 14, Karlsruhe, Germany (2014), pp. 55–56.
- [12] L. Fröhlich et al., “Magnet server and control system database infrastructure for the European XFEL”, WEPGF006, *These Proceedings*, ICALEPCS’ 15, Melbourne, Australia (2015).
- [13] R. Kammering et al., “The virtual European XFEL accelerator”, TUD3O04, *These Proceedings*, ICALEPCS’ 15, Melbourne, Australia (2015).
- [14] S. Meykopff, “Advanced Matlab GUI development with the DataGUI library”, WEPGF142, *These Proceedings*, ICALEPCS’ 15, Melbourne, Australia (2015).
- [15] V. Rybnikov et al., “FLASH DAQ data management and access tools”, PCaPAC’ 10, Saskatoon, Canada (2010), pp. 195–197.
- [16] S. Meykopff, “An optics-suite and -server for the European XFEL”, PCaPAC’ 14, Karlsruhe, Germany (2014), pp. 52–54.
- [17] E. Sombrowski et al., “jddd: A tool for operators and experts to design control system panels”, ICALEPCS’ 13, San Francisco, USA (2013), pp. 544–546.
- [18] E. Sombrowski et al., “A HTML5 web interface for JAVA DOOCS Data Display”, WEPGF150, *These Proceedings*, ICALEPCS’ 15, Melbourne, Australia (2015).
- [19] J. Roßbach et al., “A VUV free electron laser at the TESLA test facility at DESY”, Nucl. Instr. and Meth. A 375 (1996), pp. 269–273.
- [20] V. Ayvazyan et al, “First operation of a free-electron laser generating GW power radiation at 32 nm wavelength”, Eur. Phys. J. D 37 (2006), pp. 297–303.
- [21] K. Honkavaara et al., “FLASH: First soft X-ray FEL operating two undulator beamlines simultaneously”, FEL’ 14, Basel, Switzerland (2014), pp. 635–639.