BL13-XALOC, MX EXPERIMENTS AT ALBA: CURRENT STATUS AND ONGOING IMPROVEMENTS

G. Cuní, J. Benach, D. Fernandez-Carreiras, J. Juanhuix, C. Pascual-Izarra, Z. Reszela (CELLS-ALBA Synchrotron, Cerdanyola del Vallès), T. Coutinho [on leave] (ESRF, Grenoble)

Abstract

BL13-XALOC is the only Macromolecular Crystallography (MX) beamline at the 3-GeV ALBA synchrotron [1]. The control system [2] is based on Tango [3] and Sardana [4][5], which provides a powerful python-based environment for building and executing user-defined macros, a comprehensive access to the hardware, a standard Command Line Interface (CLI) based on ipython, and a generic and customizable Graphical User Interface (GUI) based on Taurus [6]. Currently, the MX experiments are performed through panels that provide control to different beamline instrumentation. Users are able to collect diffraction data and solve crystal structures, and now we start combining the feedback from the users with the development of the second stage features like data collection strategies, automated data collections, and workflows. This article describes the BL13-XALOC control system architecture, some implementation examples, and the next features to be implemented.

BL13-XALOC BEAMLINE

BL13-XALOC is the only beamline at Alba dedicated to Macromolecular Crystallography. The beamline is designed to handle not only heavily automated X-ray diffraction experiments but also less standard ones that may include a large variety of crystal sizes and unit cell parameters. The photon source is an in-vacuum undulator with full tunability in the 5-25 keV range, allowing standard multiple-wavelengths anomalous diffraction experiments at all common K/L3 absorption edges. The optics consists in a Si(111) channel-cut monochromator and a pair of mirrors in a Kirkpatrick-Baez configuration. The optical design allows two main operation modes: an unfocused mode (without focusing mirrors) having a small vertical divergence of >0.03 mrad to be used for crystals with large cell parameters; and a focused mode, where both mirrors focus the beam to 54×5.7 um2 FWHM on small crystals. The X-ray beam at sample shows an excellent energy and spatial stability. Variable focusing can be used to match the beam spot to the crystal size, or to focus at the detector. To achieve a uniform defocused beam, the slope errors were reduced down to 55 nrad RMS by using a novel method implemented at the Alba high-accuracy metrology laboratory. The end station includes a single axis diffractometer with a removable mini-kappa mount and a robot dealing with both cryogenic samples and crystallization plates. A photon counting 6-Mpixel detector offers a large active area (431×448 mm2), a fast framing rate (12 fps), a 20-bit dynamic range, and negligible dark current noise.

Figure 1: XALOC end station, main components (left to right): detector (on red table), goniometer and beam conditioning elements (on blue table), and automatic sample changer (yellow).

The beamline received the first beam in October 2011, and by February 2012 the beam was focused at sample position. On April 2012 the first dataset was already collected and could be processed for analysis. Since July 2012 users are invited to perform their experiments, and the first users' paper using BL13-XALOC data has been published on March 2013. The whole setup has the required stability, and flexibility designed to perform experiments. After each shutdown, the beam alignment recovering process does not last more than ten minutes. Since November 2012, 1559 crystals have been tested and 825 datasets of more than 30 images each have been produced.

CONTROL SYSTEM ARCHITECTURE

The BL13-XALOC control system follows a common architecture that has been designed for all beamlines in order to be as similar as possible. Basically, we use the same hardware and controller types for the same subsystems (e.g. vacuum, motion, diagnostics, etc) whenever is possible, which then leads to the possibility to share common software interfaces. Being part of the Tango community allows us to reuse any Tango device server development which may be already implemented, or contribute with new developments or modifications. As an extension to this direct access to hardware using Tango device servers, all beamlines have an installation of Sardana, which makes the different control systems very comparable. Hardware controllers may be specific but, the software access to them is common for each given type of devices. Using the Taurus framework, the user interfaces can also be very similar, and a lot of development can be shared. For the exclusive hardware that a beamline may have, like the sample changer in the case of BL13-XALOC, the proper control system component has been developed and integrated, so the higher control system layers are able to access it using

standard channels like Taurus widgets, or Sardana macros.

Sardana is a software package for Supervision, Control and Data Acquisition in scientific installations. It delivers important cost and time reductions associated with the design, development and support of the control and data acquisition systems. It enhances Tango with the capabilities for building graphical interfaces without writing code, a powerful python-based macro environment for building sequences and complex procedures, and a comprehensive access to the hardware. Just as Tango, Sardana is open source and its development model is open to collaboration, providing a free platform that fits well in small laboratories as well as in large scientific installations. Briefly, Sardana is composed by a "Device Pool", a "Macroserver", "Taurus" as a GUI framework, and a CLI interface on top of ipython.

The Device Pool is not only responsible to abstract specific hardware access but also to make sure this access is done as efficiently and as synchronized as possible. Hardware abstraction is possible by providing a set of interfaces to the outside world: Motors (discrete, continuous or pseudo), experimental channels (scalar, 1D and 2D or pseudo) and communication channels. Communication with specific hardware is achieved through the implementation of plug-ins known as "Controllers". Each interface supported by the Device Pool has a controller type. When writing a controller, one must obey a specific interface to be considered valid. Controllers can be as simple as a mapping to another Tango device server or as complex as movements based on any function of several axes.

The Macroserver is a central process that can execute Macros in a controlled environment. Macros are python classes or functions that can be sequenced. Parallel execution is also possible if there is no resource conflict. The macros' code can also be edited and debugged using the Macroserver interface.

The Taurus framework allows the creation of GUIs and CLIs that can interact with the control system. Taurus models link to any Tango device, attribute, or pool device that may be operated. Via the definition of external Taurus schemes, other type of channels can also be accessed. The connection to the Macroserver is done through a Tango device called "Door". A Macroserver can have multiple doors, and each door can only run a single macro at a time. This door has a rich interface which reports execution status and messages, online updates, and any result that the macros may return. When connected to a Macroserver, the CLI application called "spock" is fed with meta-information about the known macros and elements that are part of Sardana. This way, it can provide features like context sensitive word completion, command history, macro error handling and debugging. Graphical interfaces also receive this information enabling powerful widgets to be built. Figure 2 below depicts a Sardana architecture schema.
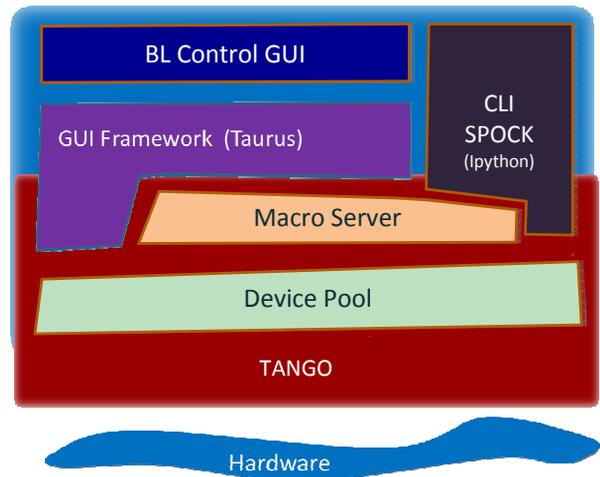

Figure 2: Sardana architecture.

CONTROL SYSTEM IMPLEMENTATION

The control system implementation in BL13-XALOC follows the architecture explained above, and has specific developments that only apply to this beamline. Some of these particularities will be highlighted in this section.

Fluorescence screen cameras, electrometers, vacuum controllers, alarms, and the archiving system are examples of completely shared development, even the graphical applications that are used are common. In the case of the pixel detector, it is controlled using LIMA [7] which is plugin-based and provide a common 2D detector interface that is also shared between beamlines. Other Tango device servers are used to access the fluorescence detector, the on-axis-view camera for sample centering, or the cryostream control.

The Device Pool contains all beamline motion axes, and measurement channels that can be used by the scientists or the users to configure or operate the beamline. Apart from the physical elements that belong to the instruments, like motors for translation or rotation, a set of pseudo-elements is also defined. These elements enrich the system giving the possibility to have complex calculations with simple interfaces. As an extension to the typical apertures based on individual blades or pitch as function of two vertical axes, in BL13-XALOC there are other pseudo-motor implementations that cover the scientists' requirements. The Pool Device controllers to change the energy, or the controllers to change the mirror focusing configuration are two examples.

Starting with the monochromator bragg angle, and calculating the current d-spacing value as a function of the mono's thermocouple values, the basic energy motor is programmed. On top of it, another energy motor is created, apart from changing the angle, it corrects the undulator gap according to a given tune and harmonic. Another layer is finally built on top. Based on the characterized small horizontal and vertical drifts of the photon beam when changing the energy, we define a motor that is able to change the energy and fix these

drifts. With this setup, the diffractometer table can be automatically realigned keeping the beam at the same position in the sample environment while changing the beamline energy. Mirror stripes could also be selected for some energy ranges.

Regarding the mirror focusing configuration, in the Fig. 3 below there is a schema showing the triple layer of motors and pseudomotors that allow to change the vertical focusing mirror configuration.
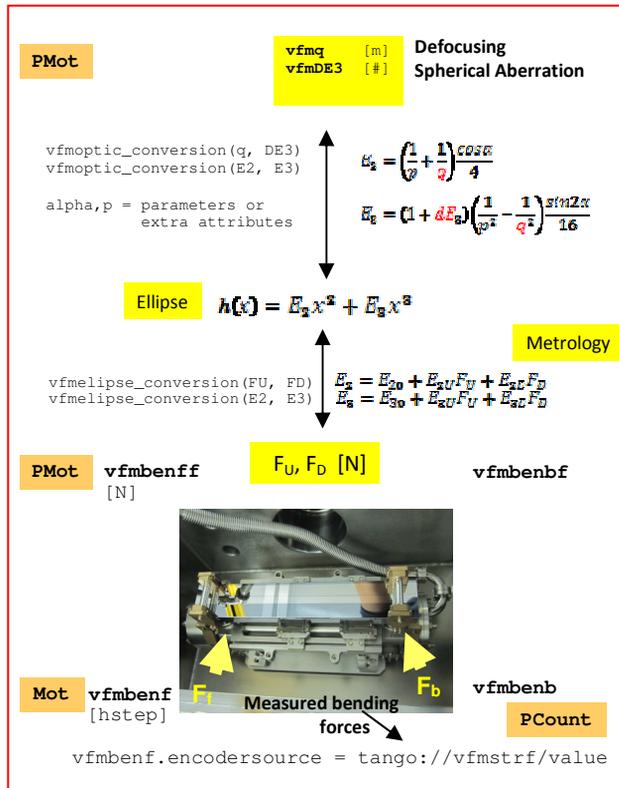

Figure 3: Example of mirror focusing pseudomotors.

At the lower level, there are the vertical focusing mirror benders installed in the front and back of the mirror (vfmbenf and vfmbenb respectively). These motors are moved in half-steps, and the position reading is done using strain gauges that provide the measured force applied. Using calibration tables, it is possible to drive those axes using force units, and on top of them, with the proper metrology and modeling, it is possible to end up with two motors vfmq and vfmDE3 that the scientists can use to change the focal point with the same tools (graphical or command line) that can be used to drive any simple motor.

The Macroserver provides a library containing standard macros used to operate a beamline, and gives a framework where all the specific procedures that the scientists determine that are useful for beamline configuration, alignment, or operation can be implemented. Some standard macros that are used in the BL13-XALOC beamline include individual motion of axis and diagnostics of beam shape, or currents in XBPMs, or a series of scans to set up the beamline for any particular experiment. The flexibility to access any

motion position or measurable value within the macro API (Python), gives the power to the beamline scientists to write any macro for any particular purpose. Simple macros as the ones to operate valves, actuators, shutters, or in/out elements have the same interface, and other macros to check environments can be programmed just as python functions. Fluorescence scans or the data collection procedures are examples of more elaborated macros.

For fluorescence scans, the implementation is based on a macro that internally executes a standard energy scan, and has extra input parameters with information like which element, transition edge and ROIs have to be used. The scan data is post processed in order to determine peak, inflection and remote energy points. This procedure uses the standard Device Pool interface of detector acquisition, and in case the fluorescence detector needs to be replaced, this change will be transparent to the macro, only the group of channels to be used to perform the scan would have to be updated.

The data collection is the core of BL13-XALOC experiments, thus, a simple but robust macro allows the preparation of the detector and trigger sources, proper speed and acceleration setup of the motion axis, and guarantees that the sample environment is ready. It is then on top of this building block that testing samples at different angles or energies, sequence of collects or snapshots are developed.

CONTROL SYSTEM INTERFACES: CLI, GUI, CUSTOM PANELS AND WIDGETS

Sardana provides a set of tools that allow the creation of simple user interfaces with little effort. In the BL13-XALOC beamline, the usage of the CLI is basically for commissioning tasks and to set up the beamline before any experiment. It is also possible to use the generic Sardana graphical application that is based on the meta-information from the Device Pool and the Macroserver. This tool automatically generates the different instrument panels with all necessary controls with an easy access. The scientists have also the possibility to define with few clicks and with drag&drop custom panels to group certain control system variables that are of their interest, or create also trends to monitor certain values which could also be extended backwards in time thanks to the integration with the archiving system.

Figures 4 and 5 below show the different interfaces for simple data collection and fluorescence energy scan.
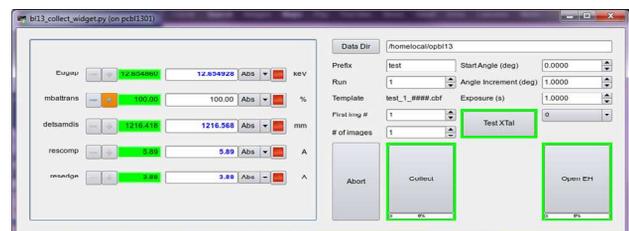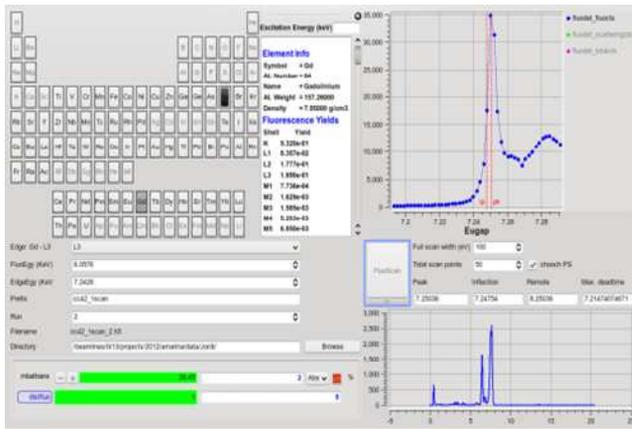

Figure 4: Simple collect interface.

Figure 5: Simple fluorescence energy scan interface.

ONGOING IMPROVEMENTS AND FUTURE DEVELOPMENTS

Even if the users are able to perform their experiments and collect data, there is a big amount of new features that were left for future releases. Some of them are already in the process of development and others are still in the long-term feature lists.

It is key for the data collection procedure to have quality data, and we are integrating the EDNA [8] framework. The Tango device server is installed and configured, and we have proven that data collected at BL13-XALOC can be processed by this server in order to recommend a set of collect strategies for a best data collection quality. A new graphical interface and macro is being designed so the possibility to have collect strategy recommendations is offered to the user.

A collect queue or multicollect feature is also being designed. The first approach is to be able to carry out different data collections with the same sample, and later on to include the automatic sample changing within the processing queue.

With the current implementation, users can perform their experiments but they have to manually interact with different interfaces. We are also starting the design of an integral application that includes all the functionality that is already available with the different widgets and panels that may help the users with a better experience. Remote operation is also being evaluated, and the first approach is to extend the whole desktop remotely.

There are more features for long-term releases which include sample tracability, mini-kappa 3D modeling for collision prevention, implementation of workflows and auto-processing programs.

CONCLUSION

Focusing on standard interfaces and share development between the different beamlines has been a good choice because it helped the parallel development of all the control system subsystems. The particular requirements of the diversity of experiments lead to flexible software interfaces and graphical tools that have been proven to be useful in a variety of situations.

Starting simple and dealing with "atomic" operations first, and later build new functionality on top had positive effects. On one hand there is a better knowledge of the low level performance of each component that had been implemented and tested, and on the other hand, the above layers can be included progressively.

Visits to other synchrotrons contributed also to have a better understanding of the experiment requirements and the solutions implemented elsewhere, having the opportunity to set priorities for day-1 features, "must" features, mid-term functionalities, and long-term interesting possibilities.

With the current implementation, both beamline scientists and users are happy with the tools that are available. This work is not finished, and we have learned that it is good to tackle features step by step, but we must continue developing and releasing new versions of this control system.

CONTRIBUTIONS

Many people have worked in this project. This control system implementation has been done mainly with the work of the Electronics and Controls sections within the Computing Division of the Alba Synchrotron Light Source. The support of the other Computing Division sections, Systems and MIS, has been also mandatory to make this project successful. We would like to specially thank the BL13-XALOC scientists Jordi Juanhuix and Jordi Benach who have been requesting the features that this system have, and also have contributed in the development of python calculations and procedures that were key during beamline commissioning phase, and are part of the beamline operation and experiment data collection.

REFERENCES

[1] J. Benach et al. "Status and Developments of BL13-XALOC, the macromolecular crystallography beamline at Alba". J. Phys.: Conf. Ser. 425 012007.

[2] D. Fernández-Carreiras et al. "The design of the Alba Control System. A Cost-Effective Distributed Hardware and Software Architecture". ICALEPCS 2011, Grenoble. FRBHMUST01.

[3] http://tango-controls.org/

[4] T. Coutinho et al. "SARDANA: The software for building SCADAS in Scientific Environments". ICALEPCS 2011, Grenoble. WEPMSO23.

[5] http://sardana-scada.org/

[6] https://www.tango-controls.org/static/taurus/

[7] http://lima.blissgarden.org/

[8] http://edna-site.org/