Graphical User Interface Design For High Throughput Crystallography

James Fait[1], John Chrzas[1], and Bill Lavender[2]
[1]SER-CAT, Building 436B, 9700 S. Cass Ave. Argonne, IL 60439 USA, and [2]Illinios Institute of Technology, BCPS, 3101 S. Dearborn St, Chicago, IL 60616 USA.

Productivity at a beamline depends as much or more on the control system the users interact with as it does inherent brilliance of the X-ray source or speed of the detector. Ease of defining the experiment, stability of the beam, quick and repeatable alignment of the sample, robust optimization of intensity on sample, ability to routinely handle 15x15x30μm crystals, routine movement from 7 KeV to 15 KeV, a user notes section, control of sample robotics, robust strategy for data collection, and archiving the data automatically all make for a highly productive trip to the beamline. The control system GUI used on both beamlines at Sector 22 at the APS, SER-CAT, has been redesigned with these ideals in mind, and both the philosophy and results of this work will be presented.

Beamline Design Philosophy

First, requirements were placed on the beamline hardware and control systems at a high level to provide specific capabilities for usability. The hardware and controls hardware design specifically addressed the following:

- **Provide for fixed exit including corrections for beam shift due to dispersion and sagital focus radius**

  The monochromator is designed with both Bragg normal and Bragg parallel translations with sub-micron resolution. These are then implemented with 5th order polynomial fits to maintain fixed offset to sub-micron levels over the entire energy range. This depends on the repeatability and accuracy of the mechanical systems.

- **Provide repeatable mechanical systems that allow for simple optimization routines**

  Each of the mechanical systems is designed to allow for specific measurable fiducials for setup, that are then measurable by automated scans for those attributes of the system which cannot be fixed to enough precision to maintain the beam on sample. At this time, we have scans for first to second crystal tune, and slit scans for the final beam position with feed-forward for goniostat sample position. These are necessary as the second crystal does have a non-repeatable shift in tune with change in focus, and the input beam from APS does move with time.

- **Provide reliable diagnostics for users to determine when staff intervention is necessary**

  At SER-CAT, we have provided a series of specific diagnostics for known issues that occur with the beamline systems, and publish specific diagnostics that users can utilize if they suspect that there is a problem with the system. The diagnostics include fine (1 µm tip) alignment needles, phosphor balls, and standard test crystals with collection protocols. The results of the most recent staff standard data collection are posted at all times as an indication of beamline readiness.

- **Crystal visualization systems that allow for 1 to 2 micron resolution**

  Crystal visualization is required for accurate centering, either automatically, or manually. Commercially available long-distance microscopes are aperture limited at about 1.5 µm. Further development along this line will require custom optical systems to be designed.

- **Automation of crystal centering, either loop centering or point and click centering**

  We have algorithms for point and click centering implemented at this time, though the micro x-y sample stage used on our ID line fails very frequently, and has been taken out of service. We are in the process of implementing new micro x-y stages for sample centering. We have an ongoing collaboration with ALS for loop centering automation, and are working on extending these algorithms to crystal centering.

Control System Philosophy

- **Modular control system based on commercial components from many vendors**

  One challenge is the variety of commercial vendors that provide hardware for the SER-CAT beamlines. In spite of design decisions to reduce the number of hardware vendors as much as practical, it was still necessary to deal with specific challenges presented by hardware products that were chosen for specific tasks on the beamline.

MarCCD

MarUSA provides the primary beamline data collection detectors for diffraction. We have 3 detectors, a Mar300, a Mar225, and a Mar165, all of which use the same software, which is closed source. The MarCCD program has exposed two control interfaces. The first interface, which is currently in use, MarCCD acts as a master for a very limited set of beamline commands, and drives an external program for the beamline control interface, with a several second/frame overhead penalty. The second interface, MarCCD is slaved to a control server that allows MX to drive the CCD directly. This allows us to remove the additional overhead that is seen when MarCCD is the master.

Delta Tau PMAC-2

The Delta Tau PMAC-2 is an industrial grade motion control platform that is capable of controlling up to 32 motors per controller, in addition to digital and analog I/O, real time capture capabilities, sophisticated programming capabilities, forward and inverse kinematics, and sub-millisecond hardware synchronization capabilities. These capabilities are used extensively for data collection and detector positioning, and are also used for all motors.

SIS

SIS provides both multi-channel counter/timer and PCI/VME interfaces. The PCI/VME interface allows for very high thoughput, which still needs further development. The SIS 3801 multi-channel counter/timer is used to capture slew scans, where multiple inputs are captured along with encoder step up / step down to give absolute position for each scan position. During a slew scan, each position is obtained 'on the fly', which allow for very fast scans.

XIA

XIA Saturn multi-channel analyzer used for fluorescence scans, with the possibility of collecting either MCA spectrum or individual SCA ROI through SIS 3801 slew scans. Setup is done via MX driver or Windows software, communicating with the module via an EPP interface. This requires the system to run on a Windows system, local to the module.

- **Client-Server model, over TCP-IP**

  A client-server model of operations was chosen, with TCP-IP chosen for the transport. This allows for a distributed system, with servers maintaining the state information and clients connecting and exiting as needed. This flexibility allow for special use GUI and text interfaces as well as the main GUI that users interact with.

- **Multiple operating systems, multiple servers for a single client**

  With the flexible hardware and large number of devices that are present on the SER-CAT beamlines, multiple servers are utilized for load leveling and device compatibility. Some of the devices use 3rd party shared libraries that are only available on specific operating systems, which requires the server run on that OS.

- **Interoperability with EPICS**
  As EPICS is used for undulator control and ring control interaction, the control system was required to interoperate with EPICS. The MX system readily handles the EPICS interactions without any special configuration.
- **Device drivers must be readily implemented for new hardware**
  Beamlines evolve continually, and new hardware usually needs new device drivers. MX allows for new device drivers to be added with relative ease without investing major amounts of time to learn the entire system. Addition of a new device driver by a programmer familiar with the data structures and driver format is possible with a week of programming and testing. An expert can often do this within a day.
- **Vendor hardware and software must be able to be integrated, regardless of cooperation of vendor or availability of reasonable interfaces**
  Much of the hardware comes with specialized software that provides the only methodology of communication with the hardware. Some vendors are very cooperative, but don't have a clue as to the real requirements of beamlines or control systems. These vendors often require multiple visits by the vendor to the site, or by a developer to the vendor, or email exchanges to sort out the ambiguities or missing "features" in the API. Other vendors, however, are very reluctant to cooperate, and have required extensive coercion to get even rudimentary interfaces implemented for the hardware they have provided. Depending on the hardware, tools such as serial port protocol probes have been used to reverse engineer the communication protocols where documentation has been unavailable or incomplete.
- **Beamline Users should not be aware of these issues.**
  While all of these issues are key to the developer and beamline management and staff, the users should not be aware of any of these issues. Most of the users are uninterested in the details of the control system, but simply want the results.

Core Control System

The core control system for the SER-CAT beamlines is based on a distributed set of Linux based PC computers, with PMAC-2 motion controllers for the majority of motion control, and SIS 3100-1100 PCI-VME interfaces for VME based instrumentation. In addition, there are specialty computers attached to the XIA MCA, and the ALS robot.

- **MX used for server and core libraries**

MX is used on each of these computers to provide server capabilities on each of these computers, with the mxserver and mxupdate processes running on each computer to serve the resources to the client processes.

- **MX python library used for User interface**
  MX provides language bindings for tcl/tk and python. The GUI uses the python binding for communication to the lower level interfaces, and extensive use of event driven callbacks and threading techniques for performance are used.
- **MX provides device abstraction, EPICS interface, Client-Server and network protocols**
  The MX system also provides the entire abstraction for the network communication model, including abstraction of EPICS devices, client-server separation, network protocols, and access control.
- **MX drivers for Slew scans, PMAC coordinate system motions, XIA MCA system, SIS VME interface, and other devices have been developed for this purpose**
  During the construction and commissioning period from January 2000 to June 2003, many new drivers were developed as new hardware was selected for implementation at SER-CAT. Specific hardware choices for motion control, VME device control, multi-channel analyzer and trajectory control were made, and device drivers were required for each of these choices. Some of these drivers were not actually completed until 2004, and further development of device drivers is still anticipated, as beamlines are not static, but evolving with advances in technology.

GUI Design Philosophy

- **Group information and controls by function. Controls will be duplicated if needed.**
  Information and input must be available where you need it. The wxPython event interface allows for multiple widgets to receive the same information in an efficient manner, allowing for minimal overhead for display of the same information on multiple panels.
- **Log information in a portable, easily displayed format**
  The one consistent problem noticed by staff and users over several years is that after a short while, the user has forgotten the conditions that the experiment has been conducted under. To prevent this, log information from all aspects of the experiment are stored in a common log with project information, and this information is cloned when a new project is started, i.e. when a new directory for data is selected. The Logbook panel also has provisions for storing images and user notes in the log, as well.

The log is stored as HTML, with standard headers and footers, so that typical browsers are capable of displaying the log at any time.

- **Attempt to make GUI intuitive in design such that an minimally trained user will be able to use the beamline unassisted by staff**

  The GUI has been designed in consultation with a number of users of varying experience, to assure that the layout and conventions used are intuitive to users that are using the facility. Some of these users are first time synchrotron users, and find the GUI very easy to adapt to. Many of the more experienced users also enjoy using the GUI, as they find the placement of controls logical and sensible compared with some GUIs used elsewhere.

- **Tabbed format chosen, in consensus with APS BioMac crystallography TWIG for GUI standards.**

  At the APS, the BioMac Crystallography TWIG (Technical Working Interest Group) met to set a common look and feel for graphical user interfaces at APS. This working group set a working standard for graphical user interface, which included a tabbed notebook interface with specific naming conventions for 4 tabs, and naming conventions and presence requirements for specific motions, such as energy, wavelength, filters/attenuators, edge selection, etc. Also included are streaming video images on the hutch and sample panels.

- **Core format conforms to the APS standard**

  The APS standard only recommends preferences for Hutch, Sample, MAD, and Collect, allowing for additional panels at implementers choice. SERGUI adds Logbook and Analysis panels to handle these particular functions, with project management implemented both on the Logbook panel and on the Collect panel, and Strategy is implemented as part of Analysis. Both of these are add on components to the basis APS standard.

- **Emphasis is placed on usability of GUI, with testing by staff and users at multiple stages**

  Throughout the development cycle, the GUI was tested first by the staff, and then by a select group of users on the beamline, with the results of the testing returned to the programmers as both written and oral feedback. This resulted in many iterations of changes to the GUI, with a large number of changes as the code evolved into the current form. As much as possible, the code was written in a generic form, so that components can be reused in other projects, as appropriate.

GUI Components

- **Mxplot – a plotting package written in tcl/tk for 2D graphs**

This is a BLT-based replacement for plotgnu/Gnuplot. It adds x-y cursors, the ability to zoom, and a button for printing the plot. Currently, it only works on Linux and Unix platforms. This is available at http://mx.iit.edu/

- **X-GEN – used for display and analysis of frame image data, and for data collection strategy**

  Written by Dr. Andy Howard, at Illinois Institute of Technology. This package is freely available at http://xgen.iit.edu.

- **wxPython – the core graphics components for the GUI are derived from this library**

  wxPython is a **GUI toolkit** for the Python programming language. It allows Python programmers to create programs with a robust, highly functional graphical user interface, simply and easily. It is implemented as a Python extension module (native code) that wraps the popular wxWidgets cross platform GUI library, which is written in C++.

  wxPython is a **cross-platform** toolkit. This means that the same program will run on multiple platforms without modification. Currently supported platforms are 32-bit Microsoft Windows, most Unix or unix-like systems, and Macintosh OS X.

  wxPython is available at http://wxpython.org/.

wxGlade – A GUI builder for wxWidgets

wxGlade is a GUI designer written in Python with the popular GUI toolkit wxPython, that helps you create wxWidgets/wxPython user interfaces. This GUI designer is used to define the geometry and widget properties for the GUI, with the methods added to widgets afterward. WxGlade is available at http://wxglade.sourceforge.net/.