MX at the Advanced Photon Source

William M. Lavender
Biological, Chemical, and Physical Sciences Department
Illinois Institute of Technology, Chicago, Illinois 60616

Introduction

MX [1] is a new data acquisition and control toolkit developed in collaboration between the Industrial Macromolecular Crystallography Association CAT (IMCA-CAT) and the Materials Research CAT (MR-CAT). MX is written in ANSI C and runs on a variety of operating system platforms including Linux, Solaris, SGI Irix, HP/UX, SunOS4, Microsoft Windows 95/98/NT, and MSDOS. At present, most graphical user interfaces are implemented using Tcl/Tk and [incr Tcl]. When used standalone, MX uses BSD sockets or Winsock to implement client/server functionality. In addition, MX can interoperate with other control systems such as EPICS [2]. Source code and more information about MX can be found at the MX web site *http://www.imca.aps.anl.gov/mx/*.

MX Design

The design of MX was motivated by the observation that most tasks performed by beamline control around the world tend to share a lot of features in common, despite the fact that the actual computers and beamline hardware may vary widely. Traditionally, many beamlines have developed control systems that are heavily tied to the particular mix of hardware and software in use, resulting in software that was not easily transportable to other beamlines.

The MX solution to this problem is to create an abstract device driver interface between the beamline application software and the underlying hardware. By doing this, complicated code like data acquisition scans can be implemented in generic code usable by all controller types, while controller specific code, such as how to get a motor to move, is isolated in an MX device driver for the controller.

A major advantage of this approach is that it minimizes the amount of new source code required to implement a driver for a new type of controller and makes it easier for programmers to write that code. A second advantage is the fact that this makes it simple to switch between controller types at a moment's notice. We have had a number of occasions where a controller has broken down and the only available controllers were of a different type. MX makes it possible to seamlessly switch between controller types without programs or users noticing.

The use of the abstract device driver interface also makes it easy to develop pseudomotors, that look to application programs just like a real motor controller. This allows users and programs to do things like scanning the monochromator energy or XAFS electron wavenumber just as easily as they scan the monochromator theta angle.

Having provided an uniform device driver interface to the low level hardware, it is also desirable to provide a uniform high level application programming interface (API). Since different programmers may choose different programming languages, it was considered important to implement MX in a language that was easily invokable from other languages. For this reason, MX is implemented in ANSI C, since routines written in C can be invoked from almost any programming language currently in existence.

Tcl/Tk was chosen as the first language to implement MX GUIs in since it is both cross platform (Unix, Win32, and Mac) and it is relatively straightforward to teach to novice programmers. The core functionality of the MX Tcl/Tk interface is written using the [incr Tcl] object package, but programmers who simply want to use the already existing object types should be able to use the interface without knowing [incr Tcl].

MX Installations

The IMCA-CAT and MR-CAT beamlines have fairly similar designs upstream of the experimental hutches, so the beamline control system designs share a lot of similarities. In each case, the monochromator is controlled exclusively using servo motors. The theta stages are controlled by McLennan servo controllers, while the motors for positioning the second crystals are controlled by Newport MM4000s on some beamlines and by Compumotor 6K controllers on other beamlines.

Downstream of the monochromators, the beamlines have a typical selection of mirrors, slits, tables, and so forth. These devices are controlled by a mix of several different kinds of controllers. Many of the stepper motors are controlled by OMS VME-58s via the EPICS motor record, while many of the rest are controlled by Compumotor 6K or 6000 controllers. Other controllers used include Newport MM4000 or MM4005 controllers and Huber slit controllers from X-ray Instrumentation Associates. All the beamlines use Joerger VSC-16 counter/timer modules controlled via the EPICS scaler record.

MX is beginning to be implemented at several other beamlines at the APS, ALS, and NSLS. To support this, drivers for a variety of other types of controllers such as PMAC motor controllers from Delta Tau have been written.

MX is also designed to be usable on inexpensive systems as well. For example, several X-ray generator based systems at Illinois Institute of Technology are currently using MX with IMS Panther motor controllers and Am9513 counter/timers.

MX Application Programs

Several application programs for MX have already been written. For crystallography users, the most important application written so far is the *Imcagui* program. *Imcagui* is designed to be the primary interface for crystallographic users of the IMCA-CAT beamlines and only includes beamline controls that crystallographers are likely to be interested in. The top level window includes controls for the monochromator and undulator energies, slit control, and X-ray attenuator control.

A second window in *Imcagui* allows crystallographers to perform automated fluorescence and absorption scans in order to help select the energies for performing Multi-wavelength Anomalous Diffraction (MAD) experiments. Also provided is an associated window that allows the users to select the element and absorption edge from a pictorial representation of the periodic table of elements. *Imcagui* then makes it easy to send the results of these scans to be processed by the external programs Benny and Chooch [3], which use Kramers-Kronig relationships to compute f' and f'' in the neighborhood of the absorption edge. This is then used to select appropriate energies for MAD measurements.

Another important program is the *Optimize* beamline optimization program. This provides beamline users with a series of several buttons, each of which runs a preprogrammed optimization scan. Typically, there will be buttons for pitch and roll scans of the second crystal in the monochromator, vertical and horizontal slit scans, and vertical and horizontal table scan. The output of this scan is sent to a fitting program that attempts to fit a Gaussian to the recorded data. The users are then provided with the center of the fitted peak and asked if they want to move the scanned motor to the center of the peak. The program is designed to allow beamline users to perform simple optimizations of the X-ray intensity without beamline staff intervention. It is not intended to be used for setup of the beamline from scratch.

A third graphical user interface is the *mxgui* program. *Mxgui* is designed to provide a generic interface to the MX control system. It provides motor control, scaler control, and the ability to setup and execute data acquisition scans. While *mxgui* is not intended to be oriented towards any particular type of experiment, it does provide an interface to most of the internal functionality of MX. Thus, many experiments such as XAFS experiments can be performed using the facilities of *mxgui* alone.

The final program to be mentioned here is a character terminal based program called *motor*. *Motor* is a command line oriented program that can be run in an xterm or a DOS command window. *Motor* is capable of performing most of the motor, scaler, and scan control functionality available in *mxgui* and the other programs. It also has a number of capabilities that are unique to it such as the ability to run a script of motor and scan commands without the presence of a user. In addition, the fact that motor is command line based allows it to be readily used over low bandwidth connections such as modems.

Future Plans

Currently, most of the planned core functionality of MX has already been implemented. The support for macromolecular crystallography experiments is largely complete, while XAFS support is almost as complete. Thus, future developments are likely to focus on experiment types that are not yet heavily supported by the existing application programs such as powder diffraction or small-angle scattering. Work will also continue on adding drivers for an ever expanding set of hardware controllers and foreign control systems. Finally, new network server interfaces will be developed to export the functionality of the MX toolkit to the clients of other control system protocols such as EPICS or CORBA.

Acknowledgements

The author would like to thank J. Chrzas, J. Fait, A. Howard, and L. Keefe, of the IMCA-CAT staff, N. Leyarovska and C. Segre of the MR-CAT staff, H. Tostmann of the University of Florida, and J. Kropf and K. Kemner of Argonne National Laboratory for their help and advice during the development of the MX toolkit. Use of the Advanced Photon Source was supported by the U. S. Department of Energy, Basic Energy Sciences, Office of Science, under Contract No. W-31-109-Eng-38. Part of this work was supported by the companies of the Industrial Macromolecular Crystallography Association through a contract with Illinois Institute of Technology (IIT), executed through the IIT's Center for Synchrotron Radiation Research and Instrumentation. The remainder of this work was supported by the member institutions of MR-CAT and the US Department of Energy, Basic Energy Sciences, under contract DE-FG02-94ER45525.

References

[1] W. M. Lavender, "MX: A Beamline Control System Toolkit," Proceedings of the 11th U. S. National Synchrotron Radiation Instrumentation Conference, Stanford, CA (1999). To be published.

[2] T. M. Mooney, B. K. Cha, K. A. Goetze, D. R. Reid, and J. R. Winans, "Beamline control and data acquisition software," *Rev. Sci. Instrum.* **67**, 1–5 (1996).

[3] http://lagrange.mrc-lmb.cam.ac.uk/doc/gwyndaf/gwyndaf.html