3.5 Beamline Controls and Data Acquisition

3.5.1 Overview and General Remarks

The principal tasks of the beamline controls and data acquisition (BCDA) group are to develop, test, maintain, distribute, and support new software and hardware solutions for APS users, and to implement those solutions on SRI-CAT beamlines. The tasks of development and maintenance have become increasingly mixed in recent years, as new features are required in old software, and known problems in mature software must be fixed before new software can be layered on top.

In general, the group's development objectives have become somewhat more complex in recent years as experimental programs have matured on some stations, while new programs have started at others. New programs frequently need new software urgently enough to tolerate the bugs and shortcuts expected from rapid development, while mature programs want incremental improvements in well-tested software and no new bugs. However, the bulk of the software both programs use is the same. EPICS allows custom software in a program's local directory to override specific pieces of base software—in fact it provides two levels of this override support. This allows us to preserve some software-development efficiencies and still serve a variety of different user priorities. But the housekeeping required to maintain several different versions of base software with a variety of custom overrides for each of roughly 30 experimental programs makes development slower and more error prone.

In past years, most of our effort went into new software development, software distribution, and providing technical support to CAT developers. As the total amount of software has increased, more of our effort has gone into maintaining old software through changes in hardware and in the underlying EPICS and VxWorks software. Increasingly sophisticated applications layered on top of established software (e.g., automated alignment support based on scan software) have exposed problems that were previously ignored. In some cases, our requirements have become unique in the EPICS collaboration, and this has required us to take over development of a custom version of software that we used to rely on other collaboration members to support.

Software distribution absorbs much less effort than it did in years past, partly because maintaining our web-based system has become routine, but mostly because CAT developers now require less thorough documentation of new software, and because they are upgrading less frequently since their focus has moved from development systems to actual users. Now the distribution problem has evolved into a problem of merging software from different developers into a package distributable to all, and this too is essentially a maintenance activity.

As a software group's effort moves from development of new software to maintenance of old software, development teams normally begin to lose developers, in part because people capable of development do not want to become software "babysitters." We cannot afford to lose people. Argonne's Hot Technology program has helped to lower the financial incentive for developers to leave, and we are working to minimize other incentives. We now have an understanding within the group that a minimum of 20 percent of a developer's time should be spent on challenging projects that are interesting and that require development of new skills. Thus far, the group has lost only two developers.

The technical-support problem has changed significantly. CAT developers have become quite proficient, and their technical-support requirements have changed from mostly hand holding to mostly troubleshooting. Instead of a large number of simple tech-support problems, we are getting a small number of difficult ones. The time required is roughly the same, but different group members are involved.

What follows is a listing of software modules and objectives along with descriptions of new developments.

3.5.2 Scan-Related Software

Previously, scan software acquired a single data point at a time, and our storage and visualization tools were layered on top of this software. But the APS provides enough photons that hardware-assisted scanning is required for many applications, notably imaging applications like tomography and microscopy. The scan software was modified to acquire arrays directly from multichannel analyzers, transient digitizers, etc., and this required minimal modifications to the storage and visualization layers.

We developed a system for very rapid acquisition of image data using a waveform generator to drive a 2D piezostage and a multichannel scaler to acquire x-ray data. To implement the system, we rewrote waveform-generator software originally developed by CEBAF, and used MCS software developed by CARS-CAT. We have demonstrated a peak acquisition rate of 10^5 data points per second. No real data have been acquired with the system, as x-ray microscope users apparently are not interested in the technique, and x-ray transmission tomography can be acquired more rapidly with a CCD detector. X-ray fluourescence tomography, however, will require very fast point-by-point acquisition.

Also, users required acquisition of many more than the 15 signals our scan software supported; the latest software allows users to trigger up to four detector modules and acquire up to 85 signals at each data point. In one application, users trigger a scaler, a multichannel analyzer, and a CCD detector. In another application, 64 signals from a bank of multichannel scalers acquire data from a 16-element x-ray detector with four regions of interest per detector.

In some cases, users want the scan software to set up conditions and trigger a separate client program, such as the multichannel analyzer display program, to acquire and store the data. The previous wait-for-client support in the scan software was adequate for use by beamline staff but was not robust or simple enough for outside users.

Users wanted scan software to go straight to the peak of a finished scan, to take some of the tedium out of beamline alignment and to support peak-following scans. This capability, coupled with existing scan-parameter support, has been used to implement one-dimensional (1D) automated alignment of the zone plate and order-sorting aperture for the 2XFM x-ray microscope. The user interface for this microscope is being considered for use on other SRI-CAT x-ray microscopes.

As the scan software became more frequently driven by other software, we discovered a rare condition in the procedure used to verify the links through which the scan software drives positioners. We also discovered that the linking software was not behaving correctly when the client's request for a completion callback was denied. Both of problems have been fixed. Currently, we are testing a solution to another long-standing link-related problem: the scan software was unable to read changes in positioner limits that occurred after the link had been made.

We have developed new tools for manipulating and viewing scan data in the MDA (multidimensional archive) file format written by VME resident software; mdaTools is a collection of programs and libraries supporting this format. Currently, we have a basic listing program; a display program for 1D, 2D, and 3D data; and a program that converts a directory of MDA files into a NeXus file. We also have a modified version of the old data catcher program that displays 2D data as they are being acquired.

3.5.3 EPICS String-Sequence Record

A CAT developer asked for a new EPICS record type to facilitate the development of software for serial and general purpose interface bus (GPIB) devices. The new record extends the EPICS sequence record in the same way the string-calculation record we developed last year extended the EPICS calculation record. The record also simplifies the development of custom user interfaces and is a key element of our multistep-measurement software.

3.5.4 EPICS Optical Table Record

Previously, our optical-table software used first-order approximations to the equations relating real motor positions to virtual motor positions (e.g., tilt angles). We found an exact solution to the system of six coupled transcendental equations that describes the real-to-virtual transformation, and with help from XFD's engineering group, demonstrated an accuracy of a few microns in the position of the point about which the optical table rotates (at the National Design Engineering Show).

3.5.5 Java Interface to EPICS Channel-Access Library

EPICS modules communicate with each other using a facility named "channel access." We would like to develop EPICS client software in the Java programming language, so our programs will run on many different platforms (Unix, Windows, etc.) without modification. A first step in this development is to connect Java with channel access. Java channel access (JCA) does this. Though usable as-is, JCA is not yet implemented entirely in Java, which would require the channel-access protocol definition. When this definition becomes available, we hope to continue the development of JCA.

3.5.6 Conversion to EPICS Message Passing Facility

Previously, EPICS support for analog-to-digital converters (ADCs), digital-to-analog converters (DACs), serial and GPIB devices, and miscellaneous other devices was implemented on a second VME processor using a facility named Hideos. Hideos was too difficult to maintain and port to new VME processors and was replaced by the EPICS message passing facility (MPF). We assisted in the development of MPF by developing VxWorks board-support software. MPF requires more memory than Hideos—too much memory to run on the processors typically installed at APS beamlines. To preserve the investment in these processors, and simplify the conversion from Hideos to MPF, we produced a version of the VxWorks image that could be loaded into flash memory.

3.5.7 Merging Third-Party Software

One function of the BCDA group is to collect software applicable to APS beamlines from all sources and integrate it into a package that is easily distributed and deployed at APS beamlines. Software collected in this way includes a large amount of code from CARS-CAT: support for multichannel analyzers and multichannel scalers, fast proportional integral differential (PID) loops, and MPF versions of serial support we wrote. Also included are support for Heidenhain encoder-interpolation hardware, written by a developer at BESSY; waveform-generator support from CEBAF;

generic subroutine support from the Royal Greenwich Observatory; and slit-control software from UNI-CAT.

In one application of this software, we implemented lookup table support within a feedback loop, to maintain 10 microradian crystal alignment between a linear stage and a rotary stage. This allowed the two stages to be used together to emulate a four-meter-diameter rotation stage.

Among other developments driven directly by user requests are the following:

- Modified and tested support for Heidenhain encoder interpolators.
- Designed and tested interface board for a bank of 16-bit ADCs.
- Designed transition boards for digital and analog output devices.
- Converted Omega temperature controllers to RS232 and made cables for EPICS operation in sector 1.
- Wrote software to run and acquire data from a transient digitizer.
- Updated, fixed, and added to the BCDA web pages. Most significantly added a hardware "home page."
- Fixed problems in support for Oxford ILM202 liquid N2 level controller for MHATT-CAT.
- Converted GPIB support for Queensgate AX301 piezocontroller to serial.
- Designed and implemented remote-shutter electronics for the topography lab.
- Implemented a custom DAC channel to control the mirror bender in 3-ID.
- Designed and built a gate-valve controller.
- Designed and built a Personnel Safety System remote shutter control interface.
- Designed a standard glue-logic electronics module.
- Fabricated and tested 100 new motor-transition boards.
- Wrote software support for Omega, Lakeshore, Oxford, Wavelength Electronics temperature controllers.
- Implemented EPICS access security for the mirror lab.
- Studied wavelet signal processing techniques.
- Wrote software to perform multiple-step measurements (e.g., for dichroism experiments).
- Investigated optimization algorithms for multidimensional automated alignment.
- Modified spherical-grating mono-chromator software to use multiple gratings.
- Developed software support for the Canberra triple channel analyzer.

3.5.8 CCD Image-Grabber Software

The Portable Channel Access Server allows the core functionality of EPICS to exist on a multitude of platforms. One place where it is very desirable to take advantage of this functionality is cheap and/or specialized PC hardware. Over the last couple of years, a significant amount of work has been done with the Portable Channel Access Server to ease the merging of PC-based control and acquisition systems into the EPICS control structure. To date, PC-based CCD camera systems from four separate manufacturers have been successfully merged into our EPICS-based control system. Also, work has begun to add EPICS control of a PC-based motion control system.

In each case, we were able to use vendor-supplied drivers in their native environment in combination with in-house-developed software to provide virtually seamless EPICS-based control over these systems. This has allowed the users to expand the tools available to them without adding new and unfamiliar control systems for them to learn.

3.5.9 Insertion Device Support

Users who routinely scan the ID along with their monochromator encountered occasional difficulties in unusual circumstances, such as scans interrupted by electron-beam fill operations. To avoid these problems, we wrote a front end to the ID software that makes an ID look more like a simple motor, and that has a more robust command-completion handshake.

3.5.10 Progress toward a Standard Scripting Language

We are in the early stages of what we hope will become a collaboration with developers at ESRF and the National Synchrotron Light Source (NSLS) to implement a standard scripting language for use at synchrotron-radiation beamlines. Users of EPICS-based beamline software have long requested scripting support, and attempts have been made to use the languages tcl and Rexx for this purpose, but defects in these languages or in their availability on different platforms have so far limited progress. The language, Python, now seems to be the best candidate. EPICS developers at Fermilab and KEK (Japanese High Energy Accelerator Research Organization) have produced good connections with the EPICS channel-access layer, and we have assembled some of the required infrastructure (e.g., numeric support, plotting widgets). The project has been set aside for several months.

3.5.11 Parameter Management Software

The software that preserves beamline parameters (e.g., motor speeds and positions) through reboots of the VME crate has become reliable enough that users seldom run the manual parameter-management tools anymore. The software now defends parameters against crashes that occur while they are being saved and detects directory permissions that allow the parameter file to be written but that do not allow its length to be changed.

3.5.12 Motor Support

Motor-control software has undergone a thorough revision with four goals: to reorganize device-support software so that it is easier for third-party developers to write software for new motor controllers, to address a long list of minor problems in the motor-record definition, to support servomotors, and to allow the motor record to be used as an interface to other software, making what some users call a pseudomotor. We have also added motor support for a few new controllers and developed driver electronics for small servos.

3.5.13 System/Network Administration

- Upgraded to redundant file server and new disk array.
- Recovered from failed disks, a failed disk interface, and a failed central processing unit board.
- Upgraded network from 10 Mb hubs to 10/100 Mb switches, 1 Gb backbone.
- Tripled the number of network connections per experiment station in sectors one, two, and three.
- Installed a new network in sector four.
- Installed new tape-library system for system backups.
- Installed new operating system patches and security patches.
- Investigated technologies suitable for an SRI-CAT data store.
- Moved the file server and related equipment, as well as the computer support lab, to new locations.