# EXPERIMENT BASED USER SOFTWARE

D.K. Chevrier, Canadian Light Source Inc., Saskatoon, Canada
M. Boots, Department of Physics and Engineering Physics,
University of Saskatchewan, Saskatoon, Canada

*Abstract*

The Spherical Grating Monochromator (SGM) and Resonant Elastic-Inelastic Xray Scattering (REIXS) beamlines are located at the Canadian Light Source (CLS). A novel approach to software design has been undertaken to simplify user interactions with these beamlines. While the SGM and REIXS beamlines are structurally different, the techniques available are quite similar. The software is developed to provide seamless acquisition of data, strong data management tools, and easy transition between beamlines for end users. The end result is software focussed on experiments rather than software focussed on beamlines.

## INTRODUCTION

One reality of modern science is that 90% of "conducting an experiment" involves sitting at a computer and interacting with software. Traditionally, the CLS has found the resources to develop *beamline software* for each new beamline. In principle, this is a good thing. However, as the facility grows and matures there is a sense that the software used at the beamlines needs to evolve as well. As the vision of the CLS – "[t]o be a global leader and a recognized centre of excellence in synchrotron science and its applications" [1] – makes clear, the purpose of the facility is to support science. As such, evolving our software from *beamline software* to *experiment software* seems like a way to better support science. It is important to note that having *beamline software* is a natural part of the software progression. When a beamline is under development and commissioning, the essential first requirement for software is to provide direct and detailed control over all the separate components that make up the beamline. The importance of this existing software should not be questioned: there would be no way do any science, nor to evolve user software to the next level, had this critical work not been done.

With this background in mind, there are clear ways to address long-standing user issues and improve the experience and efficiency of conducting research at the CLS. The evolution from beamline-centered software to experiment-centered software is accompanied by an evolution from engineering software to designing a *user experience*. That is, there is a shift from the relatively straight-forward task of stating that "software requires the ability to do *functions* A, B, and C using *widgets* X, Y, and Z" to a more holistic need for software that "makes it *quick* and *intuitive* for users to do *tasks* A and B". Because of this change from concrete to descriptive requirements, there are competing types of requirements to keep in mind. In principle the requirements of functionality, appearance, and connectivity will compete with each other as each component is designed and developed. Thus, every component within the software needs to work properly, look appeasing to the user, and be able to connect with other related tasks the user wishes to do.

In addition to a discussion about the concepts and ideas of making an experiment centered software package for users, some time must be devoted to exploring how this can be best achieved from a programming standpoint. While important, the examination of the programming principles will take a backseat to the fundamental vision.

From the inception of this project, we sought to cast as wide a net as possible to determine what users needed out of experiment based software. A summer student was given the task of shadowing users on a number of different beamlines looking for features that were exceptional, tasks that could be simplified, and common irritations that users experienced. Additionally, a workshop was conducted at the CLS Annual Users' Meeting to act as a focus group for new software concepts. A number of outstanding ideas were generated and have been incorporated into the current design.

## USER CONCEPTS

Would it not be wonderful if users could sit down and just start doing experiments when they first get to the beamline? Could it be made so software would help users with their experiments – giving them guidance when needed and remaining unobtrusive when not? Would it be so bad if users only needed one software tool from the beginning of their experiment until the end? The vision of experiment based user software is to offer all of these opportunities to users, regardless of their experience level or background, in a way to allows them to concentrate on the science they know. At the same time, the user experience needs to be as pleasant and efficient as possible. The question we must pose is whether it is possible to achieve this and, if it is, how best can that be done? Presuming it is possible, the software evolves from controlling individual acquisitions and beamline actions to managing the acquisition, the data, the beamline, and the experimental process as a whole.

### *Acquisition Management*

Currently, many users experience a steep learning curve when they arrive at the CLS, the steepest part of which is becoming familiar with the unique controls of the beamline they are working on. A common experience might be that of an expert user doing simple x-ray absorption spectroscopy (XAS) at the SGM beamline. Although this user likely knows as much as, if not more than, the beamline staff about the scientific technique itself they are still forced to learn how to conduct XAS on the SGM beamline – which controls to set, how to setup a scan, which detectors to look at, and so forth. Any time a user changes to a new beamline, the learning curve is repeated. On the other hand, if the technique was explicitly supported in software, two general consequences are expected. First, the lag time stemming from that learning curve could be eliminated as the details of how to coordinate an XAS scan could be programmed for each beamline with no need for the user to get involved if they do not wish to. Second, there would be a seamless transition between the two beamlines; and, not only would the user be able to start immediately, but they would already be familiar with the software. Because of these advantages, technique-based acquisition was one of the first features implemented.

Just as the concept of focusing on the technique rather than the beamline puts the science and the experiment first, the controls for a scan can also be put in scientific terms. Many beamlines will have a chart or a graph posted allowing users to look up the correct set of beamline parameters to achieve, for example, a desired flux and resolution. After choosing the curve they wish to emulate, it is up to the user to move the beamline components to the correct positions. However, since the user was principally concerned with balancing flux and resolution in the first place, could the software not have allowed the user to set these values directly? Furthermore, with appropriate feedback, the user can vary the flux and resolution settings to see what beamline configuration best suits their needs, also giving them a means to learn about the beamline details if they desire. Because of the importance of placing science first, this particular feature has already been implemented.

There are a great number of other concepts that would place science at the forefront as well. Routine users at the CLS are familiar with using a table or similar method to manually define the range of an XAS scan. However, they are primarily interested in scanning their samples for particular elemental edges. While the notion of entirely removing manual entry of a scan range would likely disrupt users, the idea of featuring an interactive periodic table is another way to allow the users to focus science. Since many users need to consult a handbook for the edge energy of the element they are interested in, our goal is to remove the middle-man and allow users to do this direclty in software. Furthermore, such features make it easier for scientists outside of physics and chemistry to use the CLS. This feature, while both important and achievable, is still currently under development.

A key priority for the project as a whole, but with particular focus on acquisition, has been to make the common tasks a user does easy and intuitive. If a beamline has a particular technique that is used more often than the others, or a task that has to be repeated for every technique, then these features have to be designed solidly with great attention to detail and usability.

Finally, while the prior concepts have put emphasis on single acquisitions, it is important to note that all users do many scans while they are at a beamline. Sometimes these scans are done individually with users making decisions between each acquisition; in other circumstances users wish to arrange to do one scan many times, or even to do several different scans in sequence. The concept of a workflow manager is provided to allow users to automate tasks – whether scanning samples, moving between samples, or changing the beamline configuration in some other arbitrary manner. This feature has also been implemented in the initial version.

### *Data Management*

The users of the CLS are accustomed to a process of collecting data; visualizing it in a cursory manner with a limited set of analysis tools; and, finally, transferring all of the raw data to their own computers and re-starting the analysis procedure from scratch. Most beamlines at the CLS offer no tools to assist them in either organizing or logging their experiments before, during, or after acquisition. Common experience shows that almost all groups will record most of the same data by hand into either a logbook or a word processing document. Since this is the case, there is an obvious advantage to having this information automatically collected for the user and stored with their data. Furthermore, since there are no existing tools to help organize data, the addition of a database for scans has been implemented to make it easy for users to sort their data how ever they see fit.

To make the database easy and intuitive to use, many features have been implemented already. Chronological sorting by experiment date and “run” – visit to the facility – is automatic, but users can also create their own *experiments* and sort their scans how ever they wish. This supports users in long-term research conducted across multiple visits to the facility, or across multiple beamlines. A single scan can belong to multiple experiments, if the user wishes, and scans from any run can be placed in any experiment.

A number of familiar user interface paradigms have been adapted to the database so that users can easily identify, select, and organize their large data sets. List views and detailed thumbnail views will soon be implemented to provide additional context and information -- such as beamline configuration -- for each scan. A “logbook” view is also under development: providing a convenient supplement or alternative to paper-based logbooks, and reducing mistakes that users can easily make when recording their experimental process. Finally, drag-and-drop features give users the opportunity to move scans to experiments as well as open scans in the visualization window. Simple features, like selecting multiple scans and collapsible sections, make it even easier to view and move large sets of data.

Along with a database for managing the data users have collected at the beamline, there is also the capability to import data – whether from the CLS or else where. A general structure exists for implementing a new import plugin and, while some coding is required, existing importers for older SGM data and data from Beamline 8 at the Advanced Light Source (ALS) will act as templates. Finally, the importers are optimized to handle large data sets so common users with normal amounts of synchrotron data – the normal amount being “a lot” – can easily see all their historical synchrotron data at once. Of course, if data can be imported in a given format, the software should be capable of exporting as well. One of the short-term goals is to create a framework to do this efficiently. Not only will it allow users to export the data they have collected at the CLS, it will also allow them to combine data from many facilities and export it all in a common format of their choosing.

Finally, since a user's data should be accessible whether or not they are at the beamline, the data management segment of the software has been designed to be easily separated from the rest of the acquisition tools so users can take it home with them.

### *Beamline Management*

While tools to manage acquisition and user data go a long way to putting science first in the software, there is still the issue that a number of tasks that need to be done regularly on the beamline do not fit into either category. If the aforementioned management systems work as designed, then one of the few remaining barriers to allowing the users to focus almost exclusively on science will be these beamline-specific tasks.

A perfect example of such a task is dealing with samples. Between transferring samples into or out of the chamber, labeling them on the sample plate, and aligning them in the beamline, managing samples can be a substantial undertaking for new, and even experienced, users. Because of these factors, sample management has been given a prominent spot in the initial software design. A central location has been designed to view the sample plate in the beamline; move to and label samples of interest; and, recall sample positions or reload old sample plates. The added benefit of specifying the location of and labeling samples is that the software can automatically associate scans with samples. This association propagates to the database, allowing the user to easily browse by the sample names they have chosen.

In addition to managing samples for acquisition, there is still the matter of transferring samples into and out of the chamber. Like many other beamlines at the CLS, the SGM beamline has a number of manual steps that must be performed to do a sample transfer. Normally, users follow a transfer manual but often have trouble flipping between segments. We are currently testing a software guided manual that allows users to select the transfer task they need to do and gives as much feedback as the beamline has to offer. Furthermore, additions in the near future will add optional pictures, or possibly brief videos, to give additional help as required.

Another beamline task that users often have difficulty with is troubleshooting – particularly determining if the beam's signal strength is appropriate. Normally, users need to ask the beamline scientist which controls to monitor as well as what the feedback value should be. Rather than having users memorize the expected current for different configurations and since beamline characterization has already been done for the flux and resolution settings, another design slated for immediate completion is visual feedback for the signal strength. This simple explanation conjures the image of a cellphone's signal bar, which is exactly how we intend to implement the visual interface.

## PROGRAMMER CONCEPTS

Developing this software has presented many challenges: once complexity and interconnectivity reach a certain level programming, undoubtedly, becomes more difficult. However, there has been little doubt that these obstacles could not be overcome – with enough time and code, almost anything seems possible on a modern computer. That being said, we have placed a strong focus on trying to the make the software as easy to code and expand as possible. Some of the design features are discussed in the final sections.

### *Code Design*

As the intention has always been to make the software work across beamlines, the design stresses the use of decoupling and inheritance. The base concepts have been to decouple associated ideas – scan configuration from scan control for instance – and to make "dumb general classes" which are inherited by "smarter specialized classes". Where possible, generalization has taken a backseat to such decoupling and inheritance based on the observation that generalized code tends to do everything in a mediocre fashion while specialized code tends to do one job very well. Our hope is that having the specialized implementation classes completed for a set of beamlines will act as a roadmap for programmers who wish to extend capabilities to their own beamlines.

### *Code Management*

As of September 24th, the project has grown to over 450 files and over 66000 lines of code. Thankfully, the Git version control system has been used to manage the source since the project began. In addition to working well for the initial development period, Git will allow the project to be opened up to a larger community for development – we expect this to happen before the end of 2010. In addition to code management, Doxygen has been used as the documentation suite. Git and Doxygen have been integrated so that the online documentation manual is automatically updated whenever code changes are committed to the version control system.

## CONCLUSION

While there remains substantial work to be done, the experiment based user software project has come a long way in a short period of time. By focussing on putting science first and refining the user experience, we hope to deliver software that users enjoy using both at the beamline and when organizing their data at home. With beta testing underway at the SGM beamline, the time is ripe to open the project up to a larger community of contributors, including other CLS staff, CLS users, and collaborators from the larger synchrotron community.

## REFERENCES

[1] Canadian Light Source Inc., Retrieved September 2010; http://www.lightsource.ca.