

## **Integrated Development Environment for Creating and Executing Customized Beamline Alignment and Data Collection Scripts**

E. Kondrashkina<sup>1</sup>, S. Kozyrenko<sup>1</sup>, S. Devarapalli<sup>2</sup>, S. Stepanov<sup>2</sup>

<sup>1</sup>*BioCAT at the APS, Illinois Institute of Technology, Argonne, IL, 60439 USA*

<sup>2</sup>*GMCA-CAT, Biosciences Division, Argonne National Lab, Argonne, IL, 60439 USA*

(October 18, 2004)

High-throughput operation of beamlines at modern synchrotrons may be hindered by frequent realignment of beamline optics, especially when the beamlines are multi-purpose. An example is the BioCAT 18ID beamline at the APS providing beamtime for research in X-ray spectroscopy, diffraction and scattering areas. At the BioCAT, reduction in the beamline realignment time was previously achieved by performing fast on-the-fly scans which were by orders of magnitude faster than the commonly used step scans. Use of this custom-built fast-scan software was eased by friendly GUI. However, the hard-coded graphical user interfaces have become an obstacle on the way of utilizing the software for construction of scan job sequences that required by the tasks of automating beamline alignment and data collection. A solution was found in creating an integrated development environment that provides a possibility to graphically create beamline control scripts using the fast-scan procedures combined with logic operators and data analysis functions. The software is written in Java and uses EPICS Easy Channel Access for Java (EZJCA) implemented on top of JCA package. Scripts are saved in XML format and executed by Apache Ant. The first functionality test results are discussed.

### I. INTRODUCTION

High-throughput operation of beamlines at modern synchrotrons may be hindered by frequent realignment of beamline optics, especially when the beamlines are multi-purpose. BioCAT 18ID beamline at the Advanced Photon Source (APS) is built as a high-performance, high-throughput facility for studies of biological samples with multiple techniques such as X-ray absorption spectroscopy, small- and wide- angle diffraction and scattering [1]. This diversity imposes strong requirements on the beamline control software. Among those are the needs to (a) - minimize the time used for beamline realignment and beam refocusing in order to make possible rapid turnaround when switching between the techniques, and (b) - to facilitate the beamline control by supplying a friendly graphical user interfaces (GUI) to software letting novice users and non-physicists learn and operate beamline effectively.

To satisfy these requirements the BioCAT has implemented a Fast Scans software package [2] that was built on top of EPICS (Experimental Physics and Industrial Control System) [3] widely used at the APS. The fast on-the-fly scans synchronize continuous motion of servo- and stepper- motors as well as their assemblies with simultaneous recording of x-ray intensity. Only several-seconds long, those scans are used for beamline alignment and data acquisition speeding up the operation by orders of magnitude in comparison with the performance of the standard step scans. A friendly GUI makes the software easy to operate. The Fast Scans package has been recently installed at another beamline at the APS – GMCA-CAT (23ID) where it demonstrated good performance in beamline commissioning.

However, on the way to fully exploit the capabilities of the beamlines hardware and to meet user needs for custom, non-routine scans as well as to automate beamline alignment procedures the hard coded programs and graphical interfaces become an obstacle. An Integrated Development Environment (IDE) is necessary for providing users a possibility to graphically and interactively construct custom types of scan sequences with minimal effort in learning the software. With this purpose the Integrated Beamline Environment for Scanning and Setup (IBLESS) has been designed

and its core functionalities were implemented at the BioCAT.

## II. DESIGN OVERVIEW

The IBLESS project fully utilizes the Fast Scans software package. The scan routines comprise a library of high-level beamline commands that are safe to execute within a user script and are supplied with graphical interfaces for collecting input parameters. The library is enriched in the new project with the logical commands such as *if/then* and *for* and with a set of analysis functions such as simple *math*, *find maximum*, *find FWHM* (full-width on the half maximum), *find Y* etc. A possibility is provided for users to add their custom functions and executables to the command library in order to further use them in script composition.

The main window of the IBLESS project is represented in Figure 1. Users can create custom scripts graphically by dragging and dropping the commands from the command lists (see at the middle of the window in Fig.1) to the editing panel (left panel of the window in Fig.1). The script presentation is easy to understand (no \$ signs, variable declarations, complicated syntax, etc.).

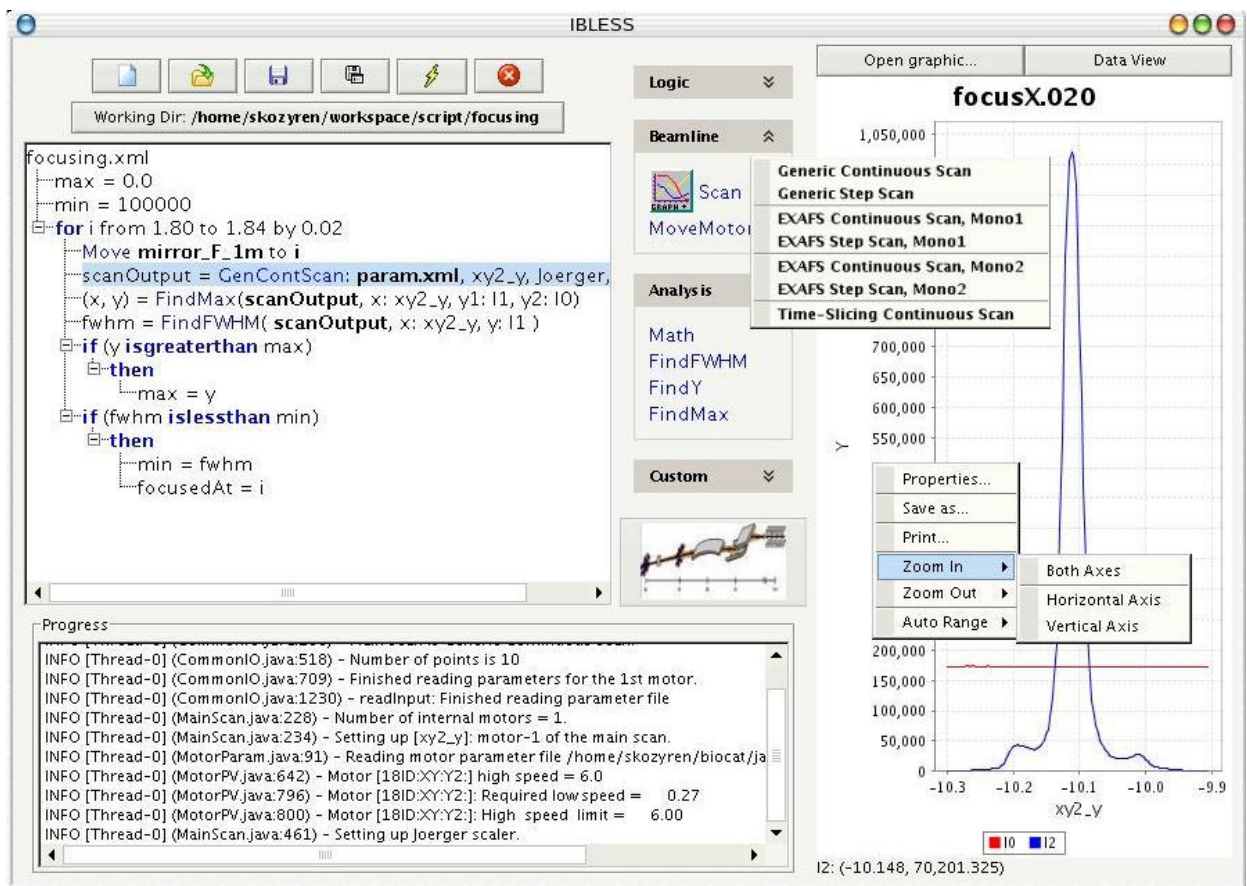


Fig.1 The main window look and feel of the IBLESS software.

The execution progress is shown by highlighted line in the script and log messages appeared in the output panel (bottom panel of the window in Fig.1). Exceptional situations are handled by the

software and script can be aborted automatically or by user with bringing hardware to safe conditions.

An interactive graph panel (right panel in Fig.1) displays the data resulted from the script execution or it can be used by user for looking through the collected data or to do a simple data analysis.

## II. IMPLEMENTATION OF THE PROJECT

The IBLESS software is written entirely in Java. Among numerous reasons for this choice, there are such advantages provided by Java as portability, potential for remote beamline controls through Java applet running in a Web browser, better performance than provided by popular scripting languages like Perl or Python [4] which is very important for real-time beamline control. Furthermore, Java served as programming language for graphical interfaces in the Fast Scans project. For the IBLESS project the scan programs written in C have been translated into Java.

There exists an EPICS channel access interface for Java called Java Channel Access (JCA) [5]. We have implemented an Easy Java Channel Access (EZJCA) library on top of the JCA that mimics the Easy Channel Access (EZCA) library available for C [6], so that migrating from C to Java was very straightforward. We have done performance tests that did not show any time delays significant to the project after this transition.

A number of other open source software packages available for Java are used in the implementation of the IBLESS project.

Scripts that are created by user in the editing panel of the IBLESS GUI are stored in XML format. This decouples script text presentation to the users from its actual functionality. XML parsing is performed by dom4j -- the open source software library for working with XML, also written in Java [7].

Execution of the scripts is carried out by Apache Ant [8], the open source Java build system. It provides very useful extension mechanism that permits using it for a wide variety of tasks.

All beamline commands, logic operators and analysis functions are the Apache Ant extensions. New functionality can be implemented with little effort by inheriting from Ant *Task* class and overwriting *execute()* method. Moreover, no knowledge of how IBLESS works or any IBLESS modifications are required to add new commands. Execution of custom user programs is supported by Ant *exec* task.

Logging of the execution process utilizes the log4j package [9]. The log is sent to the progress panel of the main IBLESS window.

The graph panel for displaying data resulted from scripts execution is implemented by using JFreeChart library [10]. It provides an interactive graphics panel with the support for multiple Y-axis, auto scaling, zooming and many others.

## III. DISCUSSIONS AND CONCLUSIONS

IBLESS provides to user a much easier way to compose a script than any standard scripting language such as PERL or Python, because the script is written graphically and the command parameters are collected through graphical interfaces, because the number of commands is small and solely related to beamline controls and the most of commands (like scan commands) are much

higher level than in PERL or other standard scripting language.

We are aware about other existing beamline control software packages, for example SPEC (which is a commercial product) that are capable of interfacing to the EPICS layers and capable of scripting. None of these packages are currently capable of delivering the fast scan functionality taking into account the specifics of the BioCAT beamline hardware and as so they would be unable to deliver the required level of performance.

Testing scripts created within the IBLESS project have demonstrated the proof of concept. Further improvements of user input validation and fail-safe mechanisms of scripting are required. Script for full automation of beamline alignment is under development. Our estimates show that the script will contain from about 50 to 100 operations. The automation is expected to result in the significant increase of experiments throughput at the BioCAT beamline. The GM/CA CAT is planning to deploy IBLESS for alignment of their three beamlines for high throughput macromolecular crystallography.

- [1] R. Fischetti, S. Stepanov, G. Rosenbaum, R. Barrea, E. Black, D. Gore, R. Heurich, E. Kondrashkina, S. Wang, K. Zhang, T. Irving and G. Bunker. The BioCAT Undulator Beamline 18ID: A Facility for Biological Non-Crystalline Diffraction and X-ray Absorption Spectroscopy at the Advanced Photon Source. *J. Synchrotron Rad.* v. 11, p.399-405, (2004).
- [2] E. Kondrashkina, S. Stepanov, R.F. Fischetti, and T. Irving. Fast on-fly scans at the BioCAT beamline. In: 11<sup>th</sup> Users Meeting for the Advanced Photon Source Program & Abstracts, Argonne, IL, USA, October 9-11 (2001), p.107.
- [3] <http://www.aps.anl.gov/epics/>
- [4] <http://www.flat222.org/mac/bench>
- [5] <http://www.aps.anl.gov/xfd/SoftDist/swBCDA/jca>
- [6] <http://www.aps.anl.gov/epics/extensions/ezca>
- [7] <http://www.dom4j.org/>
- [8] <http://ant.apache.org/>
- [9] <http://logging.apache.org/log4j/docs/>
- [10] <http://www.jfree.org/jfreechart/>