

STUDY ON A NEW FRAMEWORK OF THE ACCELERATOR APPLICATION SOFTWARE BASED ON ECLIPSE*

Q.Gan[#], C.H.Wang, IHEP, P.O.Box 918, 100049, Beijing, China
P.Chu, SLAC, Menlo Park, CA 94025, USA

Abstract

The development of new accelerator application software based on Eclipse is a front research task in international perspective. This paper analyses the framework of XAL accelerator software developed by SNS, and presents a new framework of accelerator application software based on Eclipse. According to the new framework, some tools of XAL are rebuilt and transplanted into Eclipse as plug-in. The plug-in software will enhance general-purpose property and can be applied in any accelerators.

INTRODUCTION

For modern accelerators, a high-level application framework is essential for control system and beam physics. The Spallation Neutron Source (SNS) develops a Java based framework, called XAL. Now, XAL has been used for SNS commissioning and operation for several years, and it has been introduced to some other accelerator laboratories.

XAL Application Framework

XAL is a Java based framework for application program development. The framework is designed to provide an accelerator physics programming interface to the accelerator. Much of the underlying connections to the EPICS control system are hidden from the user. Use of this framework allows writing general-purpose applications that can be applied to various parts of the accelerator. Also the accelerator structure is initiated from a database, so introduction of new beamline devices or signal modifications are immediately available for all XAL applications. [1]

However, XAL is mainly designed for the SNS accelerator, and there are some SNS-specific codes in the application. Moreover, the graphical user interface (GUI) of XAL is developed by AWT/Swing application programming interface (API), which isn't beautiful and smart. So it's necessary to improve the framework of XAL.

Eclipse Plug-in Framework

The Eclipse Platform, a development framework donated to the open source community by IBM, is notable because of the mature, well-designed, and extensible architecture. What is valuable about Eclipse is that it provides an open source platform for creating an extensible integrated development environment (IDE). This platform allows anyone to build tools that integrate

seamlessly with the environment and other tools.

The key to the seamless integration of tools with Eclipse is the plug-in. With the exception of a small runtime kernel, everything in Eclipse is a plug-in. This means that a plug-in you develop integrates with Eclipse in exactly the same way as other plug-ins; in this respect, all features are created equal.

In addition, Eclipse provides an advanced API to replace AWT/Swing, called SWT/JFace. SWT/JFace is a cross-platform development of the GUI development kit, which can be used to make more beautiful Java GUI.

A New Framework for XAL

An advanced framework is very important for the modern accelerator control systems and physical applications. It should provide a simple hardware interface, and many friendly accelerator tools. In this framework, the program developers neither need to face to the hardware, nor need to write many replicate functional codes. Obviously, Eclipse plug-in framework is a good choice.

Currently, Eclipse plug-in framework has been implemented in CSS (Control System Studio). CSS is a common platform for new control system applications and provides developers with management infrastructure and a centralised connection to external data sources. [2] Technically CSS is a set of essential core-plug-ins and application plug-ins selected by the user. The Eclipse features made the function and GUI of CSS more convenient and smart.

Just like CSS, Eclipse will provide many generic features for XAL like menus, preferences, help system, etc, and make the development of new applications easy. Thus, a new framework of the accelerator application software would merge both XAL and CSS plug-ins into Eclipse, its structure is shown in Figure 1.

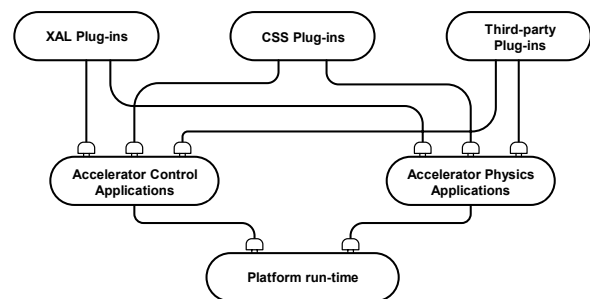


Figure 1: A new framework for XAL.

* Work supported by NSFC 10775152

[#] ganq@ihep.ac.cn

XAL PLUG-INS DEVELOPMENT

In order to reform the XAL software framework, first, split functions that provided by various tools in XAL. Secondary, classify all these functions. Then integrate similar functions to facilitate the implement of plug-in with these functions. At last, according to software requirements, select corresponding plug-in and compose the final application. Doing this will not only avoid the repeated development of codes with the same function, but also can customize functions and interfaces flexibly.

XAL Features Classification

The latest version of XAL tool set contains more than 50 tools. And all of the features are classified into two parts: the XAL core plug-ins with common features and the accelerator application plug-ins. [3]

1) Basic function plug-ins

- XAL core as a plug-in
- General purpose XML parser plug-in
- Device/PV “tree”-like browser plug-in
- Probe plug-in
- “1-D” or “2-D” scan plug-in
- Data correlation plug-in
- “Strip-tool”-like plug-in
- PV connectivity verification plug-in.
- Online model plug-in
- General purpose “knob” plug-in

2) Application plug-in used in accelerator physics

- Gaussian (and other profile shape) fit plug-in
- Wire scanner plug-in
- Emittance plug-in
- Image plug-in
- Trajectory optimization plug-in
- Beam energy measurement
- Bunch length measurement

These plug-ins are used for accelerator commissioning and operation. In addition, users can also write new features plug-in, and then combine the custom plug-ins with the above plug-ins into new accelerate applications. So that expands the function of XAL.

Data Resources

There are two primary external data resources for the new framework. The first resource is accelerator database containing the static accelerator configuration. The new framework also applies the original XAL configuration file which is generated from database. The second resource is the real time values of dynamic accelerator properties which are provided by the EPICS control system.

Interface to EPICS

Communication with the diagnostics and beamline devices is through the EPICS control system, or more specifically via the Java Channel Access (JCA) and Channel Access for Java (CAJ) interface. In the new framework, JCA and CAJ are converted into JCalib plug-

in as well. When use their related functions, only need to import the JCalib plug-in.

GUI

Eclipse can be used to develop both SWT/JFace and AWT/Swing. Compared with other GUI development kit, the user interface of SWT/JFace (including appearance and accelerator key) is consistent with the operation system. On the other hand, the SWT application is faster than other application developed by pure Java due to this feature. Therefore, the new framework will make the most of SWT/JFace to develop user interface.

However, SWT/JFace is not as flexible as AWT/Swing, and its graphical functions are relatively simple. AWT and Swing will be used occasionally. The third-party graphics package JFreeChart will also be used when necessary.

These GUI programmes will be packaged in Eclipse RCP form so that the users who are not familiar with Eclipse can call them separately. Users who are familiar with Eclipse can use the corresponding plug-in or view in Eclipse directly.

XAL PLUG-INS

It needs a lot of efforts to implement all of the XAL plug-ins. At present, there are several laboratories are committed to this work. The author is also involved in the project, and completed some plug-ins as following.

Capture Plug-in

The capture plug-in is a basic function plug-in, which is used to capture the image of application or screen. As shown in Figure 2, the capture plug-in contains two parts. The first part is a toolbar set. It can capture the full-screen, the current window or a rectangular area. The second part is an image browser used to display the captured image or a local image file, the browser also provides some general function such as zoom in, zoom out, flip, save and so on. Since the functional requirements, the first part feature uses both AWT and SWT. But the graphical interface of the second part is completely implemented by SWT.

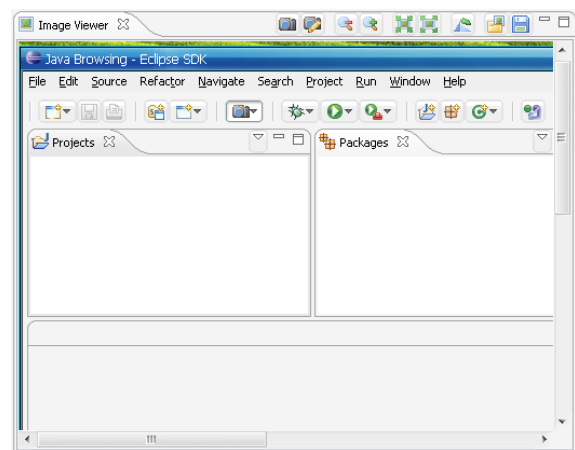


Figure 2: Capture plug-in.

PV Meter & Information Plug-in

The PV meter & information plug-in is also a basic function plug-in, which is used to get the EPICS PV information and show the PV value in a meter (see Figure 3). This plug-in calls the JCALib plug-in to build a channel access to EPICS control system, and reads the PV periodically.

As is shown in Figure 3, the PV meter & information plug-in contains two parts. The left part uses an `org.jfree.chart.plot.MeterPlot` (JFreeChart) object to display the real-time PV value. The right part is a PV information panel implemented by SWT. In addition, the plug-in provides a PV set panel in the bottom of the view.

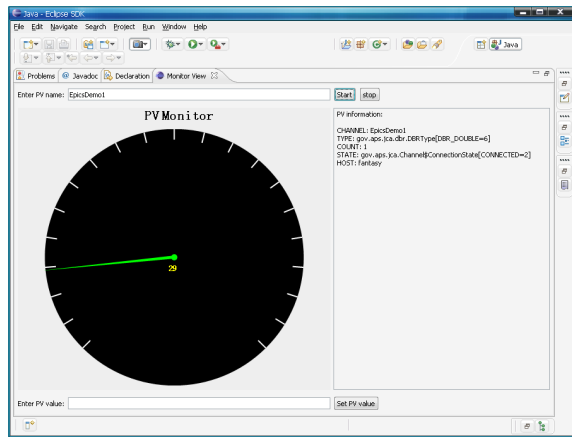


Figure 3: PV meter & information plug-in.

PV Monitor Plug-in

The PV monitor plug-in is also a basic function plug-in, which is used to monitor the EPICS PV value and draw the time curve (see Figure 4). This plug-in also calls the JCALib plug-in to get the PV value periodically.

The main part of this plug-in view uses an `org.jfree.chart.plot.XYPlot` (JFreeChart) object to achieve the function of drawing time curve. The plug-in also provides a PV set panel in the bottom of the view.

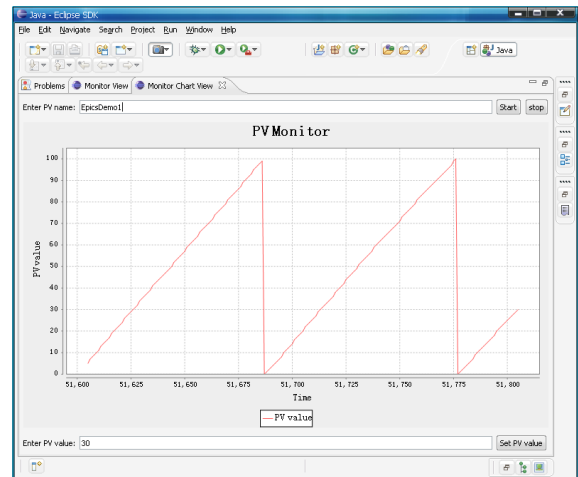


Figure 4: PV monitor plug-in.

CONCLUSION

The new framework combines the good points of XAL and Eclipse, which is a promising way for future accelerator software framework. This will be a frontier among the accelerator software community. The software development needs collaboration with SNS and LCLS. The CSNS will benefit from the collaboration for the high level applications.

REFERENCES

- [1] J.Galambos, C.M.Chu, et. al., "XAL Application Programming Framework", Nuclear Science and Engineering, 2003.
- [2] J. Hatje, "Control System Studio", ICALEPCS'07, Knoxville, Oct. 2007.
- [3] Greg White, C.M.Chu, et. al., "LCLS Online Physics Software Applications Summary", Stanford Linear Accelerator Center, 2008.
- [4] J.Galambos, C.M.Chu, et. al., "XAL APPLICATION PROGRAMMING STRUCTURE", Proceedings of 2005 Particle Accelerator Conference.
- [5] Java technology home page; <http://java.sun.com>.