

LEVERAGING THE ECLIPSE ECOSYSTEM FOR THE SCIENTIFIC COMMUNITY

T. Lam¹, A. Götz²

¹ANSTO, Sydney, Australia, ²ESRF, Grenoble, France

ABSTRACT

Eclipse, a successful open source project from the mainstream IT industry, has recently started making impact on the scientific community. This revolutionary software technology does not only change the software practices for the developers, but also improves the way how end-users interact with machines for automated scientific experiments and operations. With the support of nearly 100 companies included IBM, HP and Intel, Eclipse grows rapidly as an ecosystem. As a multi-language IDE and tooling platform, Eclipse encourages and supports agile and test driven development. Eclipse also provides extensible rich client platform (RCP) for building multi platform solution for scientific application.

An Eclipse RCP based multi control system (TANGO, EPICS and SICS) graphical user interface framework, the GumTree Platform (ANSTO, Australia), will be discussed as a case study to illustrate how developers leverage the Eclipse ecosystem to maximise the user experience with beamline instruments. Two more Eclipse based applications, Maestro (NASA/JPL, US) and the Parallel Tools Platform Project (LANL, US) will also be discussed.

INTRODUCTION

Software development for scientific projects, such as instrument control and data analysis, is going to two extremes: large scale multi-million dollar project and small rapid application development (RAD). By statistics the failing rate for software projects is relatively higher than any other engineering projects [1]. In commercial software industry, any big mistake can even take away the company's entire future. Although the nature of failed projects involves many factors, appropriate software tools and mature technologies can minimise the risk of the projects. Base on the need of satisfying the divergence of software development, the scientific computing community is gradually adopting a number of software engineering practices and methodologies from the mainstream software industry [2]. In order to increase project efficiency and reducing project risks, software techniques like agile method, extreme programming (XP), model driven architecture (MDA), test driven approach, design pattern, code refactoring, and aspect programming have been applied to scientific computing projects.

Eclipse [3], an award winning universal tooling platform, is currently the only open source project which has the best integrated support on those methodologies mentioned above. Eclipse is also an expanding community; it is expected to have great influence on many vertical industries, including the scientific computing community.

THE ECLIPSE PHENOMENA

In a simple sentence description, Eclipse is an open source integrated development environment for Java. Eclipse was originally aimed to provide a united platform for different IDE products from IBM. The Eclipse project, which began at the end of 1998, has an ambition to "eclipse" the leader of the IDE market: Microsoft Visual Studio. Within few years, Eclipse has evolved from a Java IDE (version 1.0) to a universal tooling platform (version 2.0), and finally evolves to an application framework for building rich client application (version 3.0). Commercial software development tools, such as IBM Rational Tool, Websphere Studio, and Borland JBuilder have been developed based on Eclipse. Eclipse is currently managed by the Eclipse Foundation, a not-for-profit consortium with over 100 members, including, HP, IBM, Nokia, Intel BEA and Borland. The consortium has received an initial fund of 40 million US dollars by IBM. Both the open source community and giant companies gain momentum to Eclipse. According to Mike Milinkovich, the Executive Director of the Eclipse Foundation, the biggest challenge for Eclipse is to cope with its rapid growth from its

community.

PLUG-IN ARCHITECTURE

Eclipse itself is an open platform based on an extensible plug-in architecture. This allows developers to contribute new features to further enhance Eclipse. Each plug-in in Eclipse is considered as a collection of contributions which adds values to the Eclipse workbench. The Eclipse platform further defines a set of extensible features (known as extension point) such as menu, wizard and editor. Developers can make use of the platform API and easily extend those extension points.

ECLIPSE ECOSYSTEM

The Eclipse consortium is currently hosting 8 top level projects and over 30 sub level open source projects. There is also countless number of commercial and open source Eclipse related products, plug-ins, and distributions (much like Linux distribution with variety of third party plug-ins preloaded) available from the Internet. This virtual development ecosystem takes care of software development, application life cycle, data management, business operations, and even scientific operations.

Here is a list of top level Eclipse Projects:

- **The Eclipse Project** supports the Eclipse platform infrastructure Java IDE and plug-in development support.
- **The Eclipse Tools Project** contains several IDE tools for visual and model driven development.
- **The Eclipse Technology Project** holds many Eclipse based software, such as aspect programming, application life cycle management, eLearning application, communicate framework and parallel computing.
- **The Eclipse Web Tools Platform Project** concentrates on J2EE web development support in Eclipse.
- **The Eclipse Test and Performance Tools Platform Project (TPTP)** provides a common platform for third party vendors to build next generation software testing application.
- **Business Intelligence and Reporting Tools Project (BIRT)** is used to design and generate report from variety of data source.
- **Data Tools Platform Project (DTP)** attempts to provide tools and a platform for building data centric applications.
- **Device Software Development Platform (DSDP)** is the latest Eclipse project that provides framework for embedded device programming.

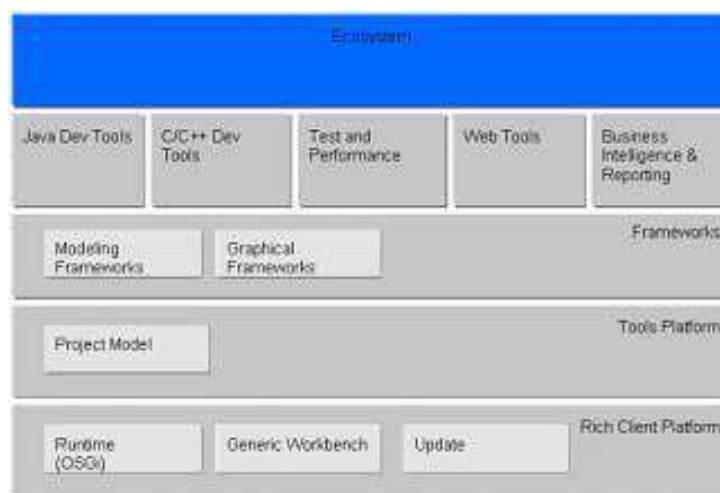


Figure 1: Eclipse Ecosystem Architecture.

ECLIPSE AS A TOOLING PLATFORM

Eclipse is sometimes referred as the 21st century version of emacs. As a multi-platform Java application, Eclipse runs on Windows, Linux MacOS, and some 64-bit operating systems. Incremental build, unit testing, CVS team support and ANT automated build system are built-in to support agile and test driven approach for Java development. Model driven and cross language (C, Python, Ruby, etc) are supported by installing third party commercial or open source plug-in into the Eclipse IDE.

LEVERAGING ECLIPSE ECOSYSTEM

For small project with very limited resource and budget, proven technologies are also the key for rapid development. More often, open source projects are consider minimising the initial project cost and base code maintenance. Since Eclipse 3.0, the platform introduces a new technology called Rich Client Platform (RCP). Eclipse RCP is an application framework which allows developers to build multi-platform application that uses natively rendered graphical widget (SWT). RCP includes many advance IDE features for application developer to reuse: integrated help system, wizard, plug-in architecture support, and online update feature. Application developers can concentrate more the domain specific logic for their applications.

RCP technology is currently being used for the following areas:

- Online stock trading system
- Knowledge context management system
- Primary school management
- Workspace collaboration environment
- Bittorrent file sharing
- Geographical mapping data display

SCIENTIFIC APPLICATION EXAMPLES

Many Eclipse managed project and third party plug-ins can be integrate with RCP application. The following Eclipse technologies are considered for scientific application:

- Report tooling (BIRT) can be used to generate experiment report and publish to the web server.
- Graphical editor framework (GEF) is ideal for users to construct graphical model on experiment planning and simulation.

- Java Swing and Windows OLE / ActiveX are embeddable in RCP, and hence legacy programs can be support in any RCP application.

There are few examples on RCP for the scientific community.

GumTree Project (<http://gumtree.sourceforge.net/>)

GumTree is a highly integrated workbench for performing scientific experiment [ref]. The GumTree Project, leads by Dr. Nick Hauser from the Australian Nuclear Science and Technology Organisation (ANSTO), is the first Eclipse RCP for scientific application. Based on the modular Eclipse plugin architecture, GumTree is capable for adapting different control system, graphics rendering, and data format. GumTree attempts to combine instrument control, simulation and online data analysis for scientists to evaluate the quality of the data acquired during an experiment. GumTree will be deployed on at least six neutron scattering instrument for ANSTO's new OPAL reactor. Future release of GumTree will support GRID for complex data analysis.

The GumTree project is currently seeking for collaborators from the control system and data analysis community to join and make use of GumTree for various scientific applications.

Plug-in Type	Plug-ins
Control System	EPICS, TANGO, SICS
Data Format	NeXus, HDF, Excel CSV, XML
Visualisation Engine	OpenGL, ISAW, Ptplot, VTK
Miscellaneous	Microsoft Office and Media Player support

Table 1: Available Plug-ins for GumTree

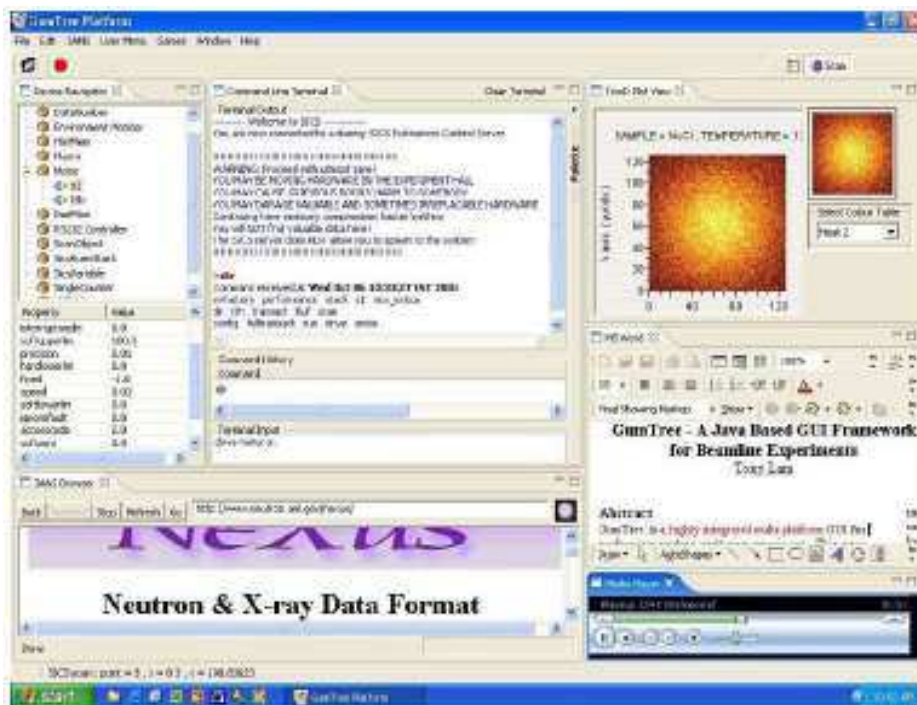


Figure 2: GumTree workbench for instrument control.

Maestro

Maestro is another Eclipse RCP example developed by NASA's Jet Propulsion Laboratory. Maestro serves two purposes: Mars rover operation and mission-critical spacecraft planning tool. The development group leads by Jeff Norris is expected to deploy Mastro on the 2007 Phoenix mission and the 2009 Mars Science Laboratory mission [4].

Parallel Tool Platform (<http://www.eclipse.org/ptp/>)

Parallel Tool Platform (PTP), proposed by the Los Alamos Laboratory, is an official Eclipse project that provides the following services:

- A runtime system that provides the user with real time information on the status of their parallel machines and jobs running on these machines.
- A parallel debugger that allows the user to launch a parallel job under the control of a debugger, then perform typical debug operations on the parallel processes.
- FORTRAN language support within Eclipse.

EPICS Office (<http://epics-office.desy.de/>)

EPICS Office is the next generation of EPICS Applications based on Eclipse RCP and using a new API which is currently under development by Cosylab contracted by Deutsches Elektronen Synchrotron (DESY).

CONCLUSION

Eclipse is a very powerful platform for developers handling daily programming task. With Eclipse RCP technology and other Eclipse technologies, the ecosystem provides a rich application framework for building scientific applications. Eclipse's open source model is currently changing the business model of software development tool market [5], and also striking the scientific community as more and more Eclipse based applications will be available to scientific users. Beside the scientific area, the Eclipse consortium is aiming to target more vertical industries, and university software engineering courses. If the growth is expected to continue, Eclipse may have great impact like Java to the society in near future.

REFERENCES

- [1] R. Charette, "Why Software Fails", IEEE Spectrum, September 2005.
- [2] P. Hathaway, "An Integrated and Agile Approach to Delivering Extensible Instrument Control System Software at ANSTO", NOBUGS 2004, Villigen, Switzerland, October 2004.
- [3] Eclipse Homepage, <http://www.eclipse.org/>
- [4] S. Schram, "Nasa Explores Eclipse Rich Client Platform", http://weblogs.java.net/blog/scottschram/archive/2005/03/nasa_explores_e.html
- [5] T. O'Reilly, "Open Source Business Model Design Patterns", EclipseCON 2005, Burlingame, CA, March 2005