Journal of Neutron Research, Vol. 14, No. 2, June 2006, 167–175

Taylor & Francis
Taylor & Francis Group

GumTree—a Java based GUI framework for beamline experiments

T. LAM, A. GÖTZ, F. FRANCESCHINI and N. HAUSER*

Bragg Institute, ANSTO, Lucas Heights, Sydney, NSW, Australia

GumTree is a highly integrated multi-platform graphical user interface (GUI) for performing neutron and X-ray scattering experiments. This open source project is built upon the foundation of the Eclipse Rich Client Platform, using the Java programming language. The GumTree Project at ANSTO consists of a general application framework (GumTree Platform) and a set of specific components for the OPAL reactor Neutron Beam Instruments Project (NBIP).

The GumTree Platform provides the essential GUI components for common beamline operations via an application programming interface (API) and Eclipse's extension mechanism. GumTree is being adapted to several instrument control server systems, providing an easy-to-use front-end for users and simple-to-extend model for software developers.

Keywords: Java; Eclipse; Graphical User Interface; Control system; Data Analysis; OPAL Reactor; GRID; eResearch

1. Introduction

The human—computer interaction for instrument automation has been improved dramatically in the past decade, due to remarkable advances in software technology. The graphical user interface (GUI), in particular, has made a quantum leap in providing a simpler way to access instruments for scientific experiments. As the gateway to instruments, the GUI maximises the functionality that can be exposed to users, without users having to learn any device specific commands. Software developers generally create GUI applications by using variety of widget libraries, but those applications are generally too *ad hoc* for a particular instrument or difficult to port to other desktop platforms.

As part of the IT infrastructure for the Neutron Beam Instrument Project (NBIP) [1], the computing team at ANSTO is currently developing a novel application framework that will lead to the development of a next generation scientific workbench for performing multiple beamline experiments. This new framework is aimed to minimise the development effort required to write a new beamline instrument GUI client from scratch. Common look-and-feel across instruments should also reduce the learning curve for users who may operate more than one instrument at the OPAL reactor and other facilities.

*Corresponding author. Email: nha@ansto.gov.au

Journal of Neutron Research
ISSN 1023-8166 print/ISSN 1477-2655 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/10238160600766336

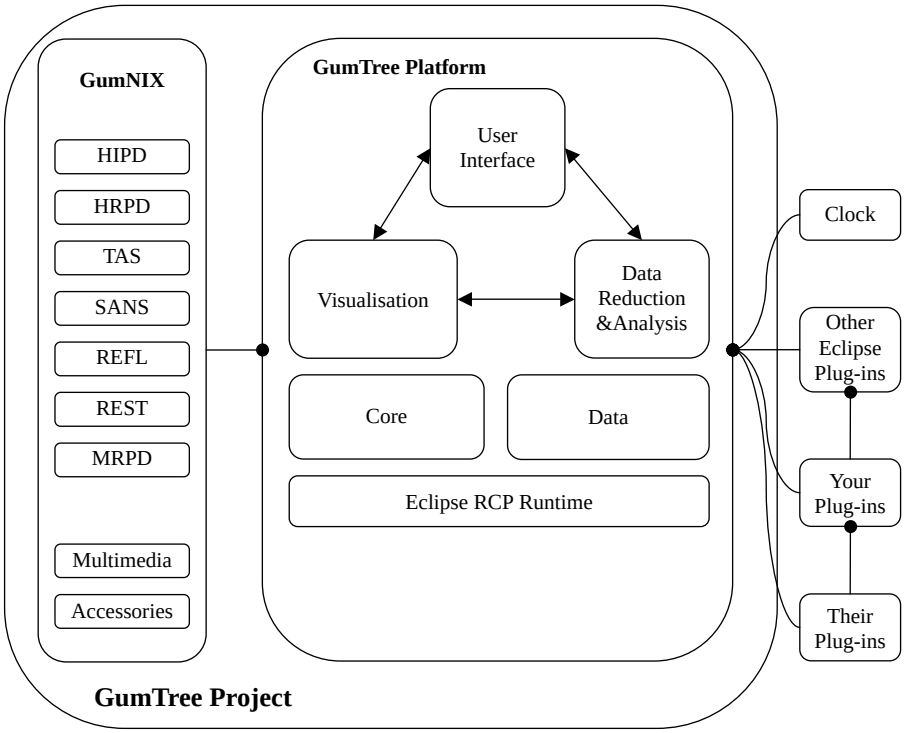

Figure 1. GumTree project overview.

2. GumTree concept

This GUI project has a codename "GumTree[1]" [2] (Graphical User interface for Multiple and Time Resolved ExpEriments). GumTree serves three purposes:

1. Client for instrument automation control and status/data acquisition.
2. Application for visualising live or offline data.
3. Workbench for data reduction and analysis.

GumTree is a highly integrated scientific workbench, which takes care of all aspects of scientific experimentation. Users could stay within a single application from experiment planing to publishing. GumTree is control system neutral. Support is planned for SICS[2] and TANGO[3], and Experimental Physics and Industrial Control System (EPICS[4]) in the near future. Gumtree is also visualisation package neutral (ISAW[5] [5] is used as the primary 1D, 2D and 3D data display package figure 1), and data analysis tool neutral (data can be exported to external analysis programs or be processed within GumTree). Future versions of GumTree are planned to have GRID computing support for data analysis with high performance computing and large dataset handling requirements.

[1] GumTree is that ubiquitous Australian tree where Aussie animals live.
[2] SINQ Instrument Control Software from PSI [3].
[3] CORBA based control system from ESRF [4].
[4] EPICS instrument control system from Argonne [5].
[5] ISAW data visualisation package from IPNS [6].

GumTree targets a wide range of user groups, from expert instrument scientists to novice university students. It is designed to be easy to use, user friendly and intuitive to new users, but without limiting the flexibility of an experience user.

To ensure the satisfaction of the software for local and overseas users, GumTree will be available to major OS platforms with internationalisation support. To address this issue, two possibilities were considered: a lightweight web application and heavy weight desktop application. Lightweight applications like web interfaces run inside a web browser (thin clients), have minimal or no installation and are multi-platform, but they may suffer slow response, and lack in functionality [7]. Although heavy weight applications (fat clients) provide richer functionalities, traditionally they are platform dependent and require users to install, upgrade and configure. Our team has decided to develop GumTree as a desktop application in order to maximise the user experience. The solution to some of the problems associated with a heavy weight client have been solved by the technology selection described in next section.

3. Base technologies

Two main software technologies are used in the GumTree Project: Java and the Eclipse Rich Client Platform (RCP) [8].

3.1 Java

Java is a high level object oriented language, which has many advantages for programmers:

- It offers a rich set of application programming interfaces (APIs) for common programming tasks (networking, I/O, data structure, multimedia, etc); and
- Same piece of code can be executed on most desktop operating systems.

The main disadvantage of using Java for building a desktop application is the performance of the JFC/Swing GUI library[6]. Also, many subjective reviews have criticised Java Swing components for a lack of sex appeal.

3.2 Eclipse RCP

Standard Widget Toolkit (SWT) is a Java based API for calling the platform's native graphical widgets. SWT makes the look of Java applications more native without paying a performance penalty, whereas swing emulation consumes huge amounts of system resources. IBM and OTI originally developed SWT for the Eclipse Java integrated development environment[7] (IDE) project [10]. Both Eclipse and SWT support major platforms such as Windows, Linux, and Mac OSX. Eclipse is currently being used as the IDE application framework for IBM Websphere Studio, Rational XDE (eXtended Development Environment) products and Palm OS Development Suite, just to name a few.

Since the release of Eclipse 3.0, developers have the option of removing IDE specific features and building their own applications on top of Eclipse. The core layer of Eclipse framework is now called the Rich Client Platform (RCP). Eclipse is a plug-in based

[6] JFC/Swing. Java foundation class libraries supplied with Java 2 Standard Edition to support building GUI and graphics functionality [9].

[7] An IDE is a software application for developing software applications.

application, where adding new features is as simple as to "copy and paste" a new plug-in into Eclipse's directory. Eclipse RCP has offered many advantages to application developers:

- *Targeted platform*—support for all major desktop platforms.
- *User configuration*—preferences are stored in eXtensible Markup Language (XML[8]) format.
- *Help system*—a Tomcat[9] web server is embedded into Eclipse to provide help support in any web browser.
- *Updating content*—seeks for updates and new features automatically across the internet.
- *Efficiency*—plug-ins do not consume memory unless they are activated by the platform.
- *Resources*—books and training are readily available from commercial and academic sources.
- *Cost*—this is an open source project under the Common Public Licence (CPL).

In particular, Eclipse RCP solves the multi-platform and upgrade problems common to heavy-weight client.

4. Design principle

Obviously some GUI features will be common across the neutron scattering instruments at OPAL, and some features may require extra configuration. It is important to develop an extensible framework so that GumTree can be adapted for different instruments. A reusable application framework can also benefit other scientific GUI programmers to reduce their software development time.

4.1 Framework design principle

There is a tendency towards the unification of common programming tasks into a single framework in the modern software community. For example, J2EE and NET frameworks are the enterprise platforms for web development, and XNA platform [11] from Microsoft is trying to unite all aspects of game programming (graphics, audio, AI, network, etc.) into a single development platform. Similarly, the GumTree project is attempting to unify GUI programming tasks for beamline experiment into an application framework.

A good framework design should consist of the following elements [12]:

- *Modular*—GumTree is divided into a set of Eclipse plug-ins to encapsulate various components in the application. Framework modularity helps improve software quality by localizing the impact of design and implementation changes.
- *Reusable*—GumTree comes with a set of Java APIs for developers to reuse the components that have been developed in the framework. This avoids duplication of effort.
- *Extensible*—The functionality in GumTree can be extended via the Eclipse extension point mechanism. Unforeseen features or any functionality not implemented from the design phase can be integrated into GumTree easily.

Developing a framework is a challenging task. Additional issues must be addressed:

- *Learning curve*—A steep learning curve cannot be avoided in working with a powerful platform. However, the GumTree framework closely follows the Eclipse framework

[8] http://xml.org/.
[9] http://tomcat.apache.org/.

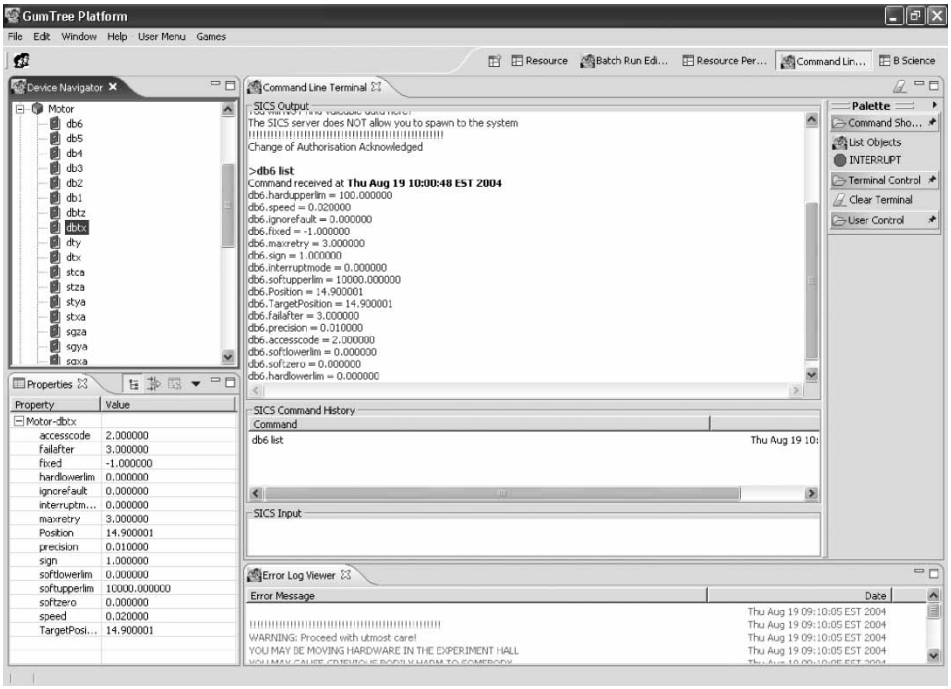

Figure 2. Command line interface for control system in GumTree.

design. Both frameworks apply the object oriented design pattern to increase code comprehension and reuse. So if a developer can master one of those platforms, she can become an expert in both.

- *Integration*—Eclipse provides an interface for integrating external libraries like Microsoft's COM technologies[10], AWT/Swing GUI, and any function that can be called by the Java Native Interface (JNI). This creates the possibility for developers to embed Windows applications (Word, Excel, media player, etc.) and Java applications into the GumTree workbench.
- *Maintainability*—The core of GumTree, Eclipse RCP, is managed by a consortium with 50 + members from widely recognised commercial companies: IBM, HP, Ericsson, Intel, Borland, Red Hat, Fujitsu, etc. Eclipse RCP evolved constantly with regular new version releases every 6 weeks, and stream integration builds everyday.

4.2 Extension point mechanism

Extensibility and integration are two very important aspects of GumTree, and they are handled by Eclipse's extension point mechanism. Eclipse and GumTree define a set of extension points for a developer to contribute new features (aka extensions) into the workbench or existing GUI components. Extensions are normally registered in the plug-in manifest (XML file), and they are activated by the platform upon request. For example, a scientist may want to add a new function to the scientific calculator in GumTree. She has to write some code and register it as an extension to extend the calculator extension point.

[10] Component object model http://www.microsoft.com/com/default.mspx.

Table 1. Selected Eclipse UI components and their corresponding application in GumTree.

Eclipse UI components	*Application in GumTree*
Views	View is an internal window inside the workbench. Each view in GumTree will provide unique functionality for performing specific tasks during the experiment.
Editors	Legacy IDE editor component from Eclipse can be used as a control system macro script editor.
Perspectives	Perspective groups a set of UI components for a particular task. Perspectives like command line interface, scan, and batch run editor have been included in the GumTree platform.
Cheatsheets	Cheatsheet is a checklist to help users complete and automate common tasks. Ultimately instrument scientists will supply their own cheatsheets for troubleshooting and guiding users through instrument setup.
Wizards	Wizard is a step-by-step guide to acquire user input. GumTree has a set of wizards to setup a batch run, new ancillaries and other operations.
Activities	Activity extension point is an effective method to filter out unwanted UI components to users. Instrument-specified plug-ins will be assigned with unique activities so unwanted views become invisible once GumTree starts in a particular instrument mode.
Intro screen	Intro screen provides application overviews, feature lists and tutorials to new users. It serves as the starting point to GumTree.

No code in GumTree needs to be modified. Generally developers can customise GumTree to adapt to a specified instrument via the extension point mechanism.

4.3 Control system support

GumTree has an abstract control system layer for adapting different instrument control servers. A dummy control server is included for training and offline simulation for instrument users. Theoretically GumTree supports a single control system at runtime, but the proxy object (wrapper to the control system) can communicate with another proxy object that connects to another control system. Such a top-level proxy is also known as the super proxy (figure 2).

4.4 Data format

An internal data object is required for passing information between different components, such as the control system proxy and visualisation package. The internal data structure in GumTree is based on the SDO[11] specification. Users may export acquired data to external programs, or use GumTree as a data browser by importing external data.

4.5 UI

Eclipse RCP defines views, perspective, activities, actions, wizards, menu and help as user interface (UI) components (table 1).

GumTree will deliver a variety of UI components for helping users to setup and perform beamline experiments. Common UI components include batch run editor, instrument scan view, experiment status view, new ancillary wizard, tutorial and extensive help topics.

[11] Service data object http://dev2dev.bea.com/pub/a/2005/11/sdo.html.

Additional GUI components can be plugged into the workbench via extension points. Some generic UI components are reusable and can be subclassed via the platform API.

5. GumTree architecture

The GumTree Project has been divided two subprojects: GumTree Platform project and GumNIX project. The GumTree Platform is a general-purpose scientific workbench with an application framework (API + extension points) for extending its functionality. An instance of the GumTree workbench supports a single instrument (i.e. instrument mode), or zero instrument (scientific workbench mode). An instrument can be as simple as single device, as long as it has a socket for GumTree to establish connection for communication. Instrument information and specified UI components are stored in a group of instrument plug-ins.

5.1 GumTree platform

The GumTree platform is a collection of plug-ins figure 3. The plug-ins are grouped into components according to their functionality. Each component is separated into two conceptual layers, the core (non-user interface logic) and UI (user interface related code). Eight components have been identified in the GumTree Platform:

- *Platform core*—platform runtime and resource management
- *Control system facility*—command line terminal
- *Data format support*—internal data object format, NeXus data import and export, NeXus browser
- *Accessibility support*—speech synthesis and other UI accessibility support

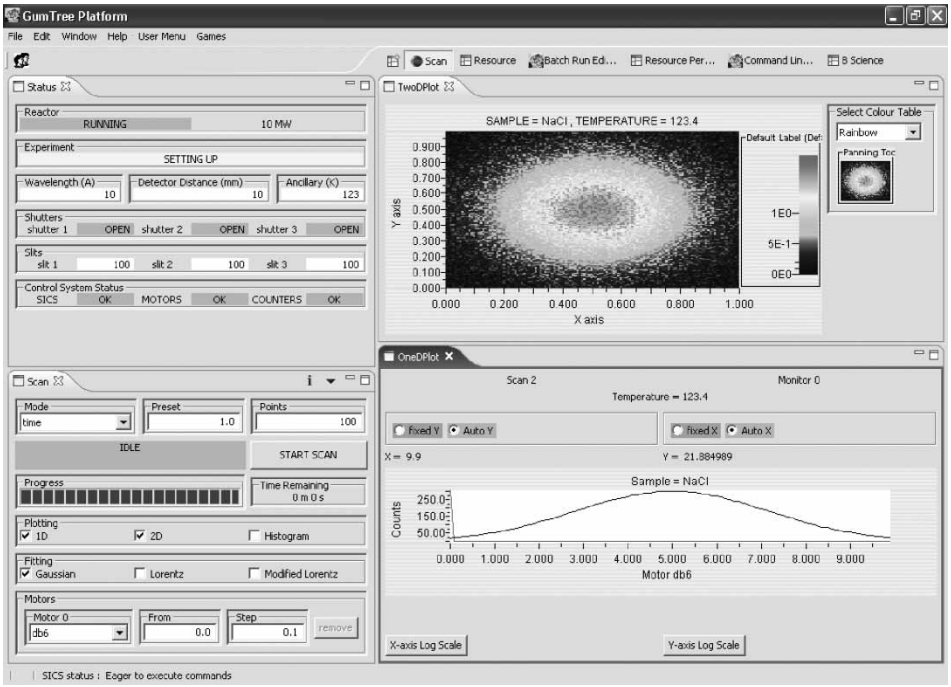

Figure 3. Scan and 1D/2D live data display in GumTree.

- *Standard workbench UI*—user interface framework
- *Data visualization support*—1D, 2D and 3D visualization
- *Data reduction and analysis support*—third party data reduction and analysis wrapper API

5.2 GumNIX

GumNIX[12] is a collection of plug-ins for all NBIP instruments figure 3. An instrument plug-in may contain the essential information for GumTree to drive the instrument. It also extends and contributes instrument specific UI components to the workbench. Instrument scientists can configure GumTree via plug-ins to filter unwanted UIs. GumTree can be started without a control system connection. In this case, GumTree serves as a workbench for data visualisation and analysis, without controlling an instrument.

5.3 Third party plug-ins

Third party plug-ins could be included in GumTree to extend its functionality. For example, a multimedia plug-in provides music for entertainment and video for interactive training purpose.

6. Development cycle

The GumTree development team deploys the extreme programming (XP) methodology to maintain software quality. The Eclipse IDE is the primary development tool for this project, and it has XP support such as unit testing built-in. Additional in-house Eclipse development tools could be developed to simplify GumTree development. Hence, non-experienced programmers can contribute new UI features to GumTree with minimal programming effort.

Meetings with NBIP instrument scientists and user groups are held on a regular basis for gathering user requirements. Milestone releases have also been scheduled every quarter. Users and instrument scientists to evaluate and feedback user stories to GumTree developers.

7. User scenarios

It is hoped GumTree will be used in neutron scattering and synchrotron institutes.

7.1 ANSTO-OPAL (Australia)

As discussed previously, the main driving force of GumTree Project is the need of a GUI for the neutron scattering instruments at OPAL. Targeted instruments are high intensity powder diffractometer (Wombat), high resolution powder diffractometer (Echidna), triple axis spectrometer (Taipan), residual stress diffractometer (Kowari), neutron reflectometer (Platypus) and small angle neutron scattering (Quokka).

[12] GumNIX stands for GumTree Neutron beam Instrument eXtensions.

8. Conclusion

The GumTree Platform helps developers to reduce time to build a GUI for beamline instruments. Developers can take advantage of the framework to improve the functionality of the GUI and maximise the potential of the instrument to users. It is hoped that GumTree will be used on various neutron scattering and synchrotron instruments and that the GumTree Platform will provide a benefit to the scientific computing community.

References

[1] NBIP Homepage, http://www.ansto.gov.au/ansto/bragg/2005/nsrrr.html
[2] GumTree Project Homepage, https://sourceforge.net/projects/gumtree
[3] SICS Homepage, http://lns00.psi.ch
[4] TANGO Homepage, http://www.esrf.fr/computing/cs/tango/tango.html
[5] EPICS Homepage, http://www.aps.anl.gov/epics/
[6] ISAW Homepage, http://www.pns.anl.gov/computing/isaw
[7] T. Williams, Eclipse-based applications: Java on the desktop revisited, http://www.eclipsecon.org/EclipseCon_2004_TechnicalTrackPresentations/17_Williams_May_Dovich.pdf
[8] Eclipse Rich Client Platform, http://dev.eclipse.org/viewcvs/index.cgi/~checkout ~ /platform-ui-home/rcp/index.html
[9] JFC/Swing Homepage, http://java.sun.com/products/jfc/index.jsp
[10] Eclipse Project Homepage, http://www.eclipse.org
[11] Microsoft XNA Homepage, http://www.microsoft.com/xna
[12] M. Fayad and D. C. Schmidt, Object-"Oriented Application Frameworks", Commun ACM **40**(10) October (1997).