g-Eclipse – A Contextualised Framework for Grid Users, Grid Resource Providers and Grid Application Developers

Harald Kornmayer[1], Mathias Stümpert[2], Harald Gjermundrød[3], and Paweł Wolniewicz[4]

[1] NEC Laboratories Europe, IT Division, Rathausallee 10, 53757 St. Augustin, Germany
harald.kornmayer@it.neclab.eu
[2] Forschungszentrum Karlsruhe, Institut für wissenschaftliches Rechnen, Postbox 3640, 76021 Karlsruhe, Germany
[3] University of Cyprus, PO Box 20537, 75 Kallipoleos Str. 1678 Nicosia, Cyprus
[4] Poznan Supercomputing and Networking Center, 61-704 Poznan, ul. Noskowskiego 10, Poland

Abstract. As the future pervasive and ubiquitous computing environment will be composed of resources from local computing, Grid, SOA and Web infrastructures, the complexity of this distributed system will increase significantly. At the same time the end user wants easy and simple access to his computing environment while he receives more responsibilities i.e. for the composition and the management of the system. In order to perform his daily work, the end user needs a general workbench toolset which supports customisation and contextualisation for the user. The g-Eclipse framework offers an eco-system to access Grid infrastructures with support for contextualised user roles. Currently, the g-Eclipse framework includes contextualised perspectives for Grid end users, Grid resource provider and Grid application developers. The abstraction layer of the g-Eclipse system, its integration in the Eclipse framework and the main use cases are presented.

1 Introduction

Wide-scale distributed infrastructures, called Grids [1], emerged in the recent years to enable the sharing of geographically distributed, heterogeneous computing, storage and network resources. These infrastructures are connected by Grid middlewares which offer the basic services to interact with the underlying Grid infrastructure, but each middleware system follows a slightly different approach, although there is a trend towards interoperable, service-oriented implementations of Grid-services.

Grid providers can build "Virtual Organisations" on top of these infrastructures, which users can belong to, in order to solve complex problems. The benefit of such a general infrastructure for scientific and commercial applications was demonstrated by many Grid projects. But many of the potential users restrain

M. Bubak et al. (Eds.): ICCS 2008, Part III, LNCS 5103, pp. 399–408, 2008.
© Springer-Verlag Berlin Heidelberg 2008

themselves from using the Grid because of the inherent complexity of using and interacting with Grid technologies. Furthermore, the Grid user is often limited to the "end user" role, but a user may also take the roles of providing resources, applications and/or services. A user of Grid resources does not cover only one role in only one context as in desktop computing, but many roles are aggregated in a single Grid user depending on the tasks he wants/has to fulfill [2]. Therefore, user-friendly and intuitive user interfaces are needed to make the look-and-feel of Grid infrastructures similar to that of existing computer desktop systems. Additionally, these frameworks must provide support for contextualisation and customisation for the Grid user to support his daily work.

The g-Eclipse project [3] currently develops such a framework for different Grid roles and contexts. As it relies on the Open Source framework Eclipse [4], the g-Eclipse project delivers extensions for the Eclipse workbench to integrate Grid resources and to support the different Grid roles. The future g-Eclipse framework aims to devise a middleware independent, integrated workbench toolset to enable contextualisation for different Grid user roles.

Results of this work are presented in this paper which is structured as follows: The different roles and contexts are described in Sec. 2. A short overview of the Eclipse framework is given in Sec. 3. In Sec. 4 the g-Eclipse architecture, the abstraction layer for the Grid resources and its integration into the Eclipse framework are presented. Sec. 5 describes a number of tasks that a user, operator or developer can perform using the g-Eclipse framework. Sec. 6 concludes.

2 Grid Roles and Contexts

Grid users act on service-oriented infrastructures in different roles and contexts. An analysis of these roles and contexts build the base for the design and development of a contextualised Grid workbench.

2.1 Grid Roles

The following main roles of Grid users have been identified within the g-Eclipse project.

Grid application users want to interact with Grid resources in the same manner as with local resources to perform their daily work. I.e. they start applications and monitor the progress of the submitted jobs. The access to their distributed data (i.e. opening, copying, moving, renaming, visualising, ...) is another important action they need to perform.

Grid operators manage the distributed Grid resources. This includes the configuration of the infrastructure and services as well as the monitoring, testing and benchmarking of Grid resources.

Grid application developers develop applications for Grid infrastructures including compiling, debugging and deployment of applications and services. They want the freedom to develop their applications with their preferred programming language.

The interaction with Grid resources should not differ from the interaction with local resources for these three Grid user roles.

2.2 Grid Contexts

Even with different roles, a Grid user can interact with Grid infrastructures in different contexts. The following contexts had big impact on the design of the g-Eclipse framework.

Virtual Organisations (VO): As Grids are seen as a tool for distributed collaborations, users organise themselves and their resources into Virtual Organisations (VO) in order to improve scalability and manageability while addressing security issues. Of course, a single user can belong to multiple VOs. A resource provider allocates his resources on a VO basis instead of a per actor basis to improve scalability. This means, that resources are supplied to VOs based on some service level agreements and not to individual actors. A VO can be seen as a virtual brace around the distributed resources and services which were connected to collaboratively solve a task. Membership in an Virtual Organisation is managed by some kind of membership service. Based on different access roles in a VO and a given membership of the users, the access to the distributed resources is managed. This collaborative aspect of Grid infrastructures is seen as very essential for the design of the g-Eclipse framework.

Projects: In his daily work, a user organise his tasks and subtasks in projects and folders. The Eclipse framework follows this approach by organising files and configurations within a *project* in the workbench. The project is a placeholder for all possible interactions which an actor initiates in a specific context. This approach would be helpful for Grid users too, as not all tasks can be organised in a common Grid workspace. I.e. an Grid user can have multiple research projects that are using resources from different VOs at the same time. It was envisioned that the information belonging to one research project should be organised into a Grid project which includes a description of the VO assigned for this project. All resources and artefacts of a Grid project (data files, job descriptions, information about submitted jobs, . . .) should be collected within the scope of one project.

3 Eclipse as Underlying Platform

To reach the goal of a reliable, contextualised framework for Grid users, the g-Eclipse team decided to reuse the Eclipse platform [4]. The Eclipse framework was designed as an open platform for a wide range of tools and the initial contribution was an integrated development environment for Java. The central point of the Eclipse architecture and framework is its plug-in architecture, a component-based software architecture that leads to a clear and modular design [5]. In the Eclipse world, every plug-in amends the functionality of other plug-ins. This is achieved by the underlying OSGi [6] framework that defines the dependencies between the different plug-ins, and how and when additional plug-ins are loaded. In addition, the Eclipse framework relies on the mechanisms of extension points and extensions. An extension point is a definition of how to enhance existing functionality. This way of building software components leads to an extensible architecture with well-defined interfaces.

The graphical front-end of Eclipse is organised with the concepts of views, editors, wizards, preference pages, etc. These components provide the basic functionality to integrate new GUI elements into the framework. These basic elements are grouped in so called perspectives. Perspectives determine the visible actions and views within the workbench, but go well beyond this by providing mechanisms for task oriented interaction with resources in the Eclipse Platform. Users can rearrange their workbench and therefore customise it to their needs and habits with the help of these components.

4 g-Eclipse Architecture

The overall architecture of the g-Eclipse framework is shown in Fig. 1. Based on the Eclipse framework, g-Eclipse provides a core Grid model [7] for the integration including abstract implementations. The user interface (UI) components of the g-Eclipse framework will reuse the components of the Eclipse framework (i.e. Views, Editors, Wizards, . . .). Middleware specific implementations can be build on top of the g-Eclipse core model and will reuse the UI components of the g-Eclipse framework. This architecture enables the g-Eclipse framework to be middleware independent and to deliver two middleware specific implementations for gLite [8] and GRIA [9] up to now.

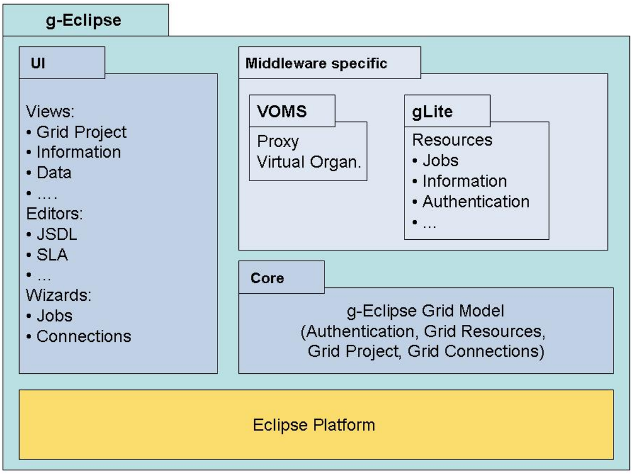


Fig. 1. The g-Eclipse Architecture

4.1 Integration with Eclipse

The integration of UI elements in the Eclipse workbench relies on the interfaces *IAdaptable* and *IResource* of the Eclipse framework. The g-Eclipse interface *IGridElement* helps to integrate the g-Eclipse Grid model in the Eclipse framework by providing the method *getResource*. All other interfaces of the g-Eclipse Grid model are based on this interface. The Grid element and its subclasses offer further methods to integrate both local entities and remote Grid elements.

4.2 The Abstraction Layer of the g-Eclipse Grid Model

The abstraction layer of the Grid model contains a multitude of Java interfaces that define the basic functionalities of the various Grid model elements. Fig. 2 shows a simplified outline of the inheritance tree of these interfaces. As this layer only contains interfaces, multiple inheritance is allowed. This is used to map the structure of local and Grid elements to the model as well as mapping the relations between the different elements. Basically, the leaves of this inheritance tree are thought to be implemented either internally by the model or externally by middleware specific implementations.

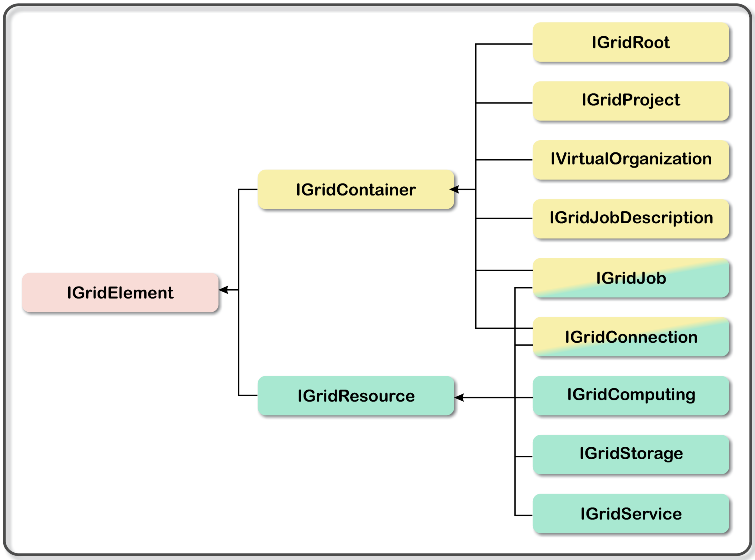


Fig. 2. The Interface of the g-Eclipse Core Grid Model

4.3 Grid Project

The Grid project shape the base for the integration of g-Eclipse in the Eclipse environment. As described in Sec. 2, a Grid project is the fundamental context for a user to interact with a Grid infrastructure. A Grid project is a direct child of the Grid workspace root and holds all information necessary to access the Grid. Within g-Eclipse, a Grid project is always connected with one Virtual Organisation. Furthermore, a Grid project follows some structure and consists of the following standard folders (see Fig. 3):

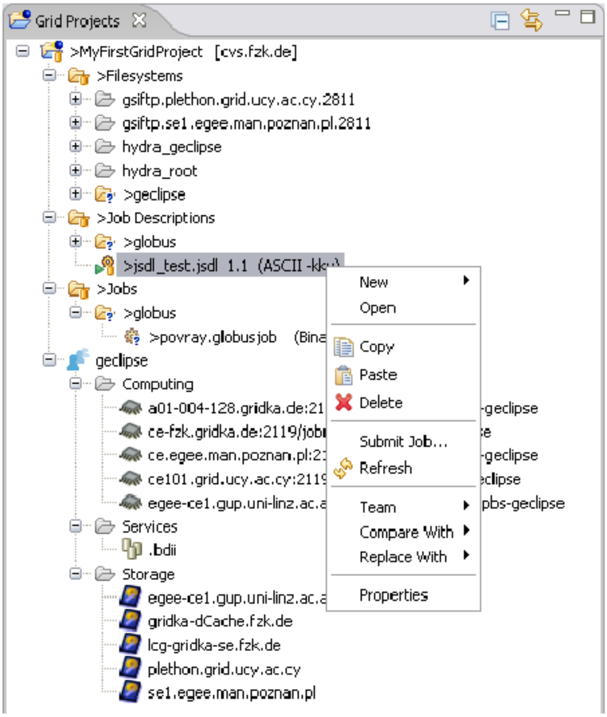


Fig. 3. The g-Eclipse Grid project view

Filesystems holds the connections to local and remote filesystems and enables the seamless access to users' data.

Job description holds the definition of Grid jobs created by the user and enables the user to start computation jobs on Grid infrastructures.

Jobs holds instances of Grid jobs, which are submitted Job descriptions. These instances will be updated frequently to get the actual status.

Virtual Organisation holds the computing and storage resources and available services of the VO of this project. This folder is purely virtual as all information about its entries will be collected from the VO declaration or from VO specific information systems.

By defining a clear correlation between a Grid project and a Virtual Organisation the g-Eclipse project nature and the corresponding Grid project view is essential for the contextualisation of the user workbench. Therefore the Grid project view will be part of the different perspectives for the different Grid user roles and their implementations as Eclipse perspectives.

5 Contextualised Grid User Perspectives

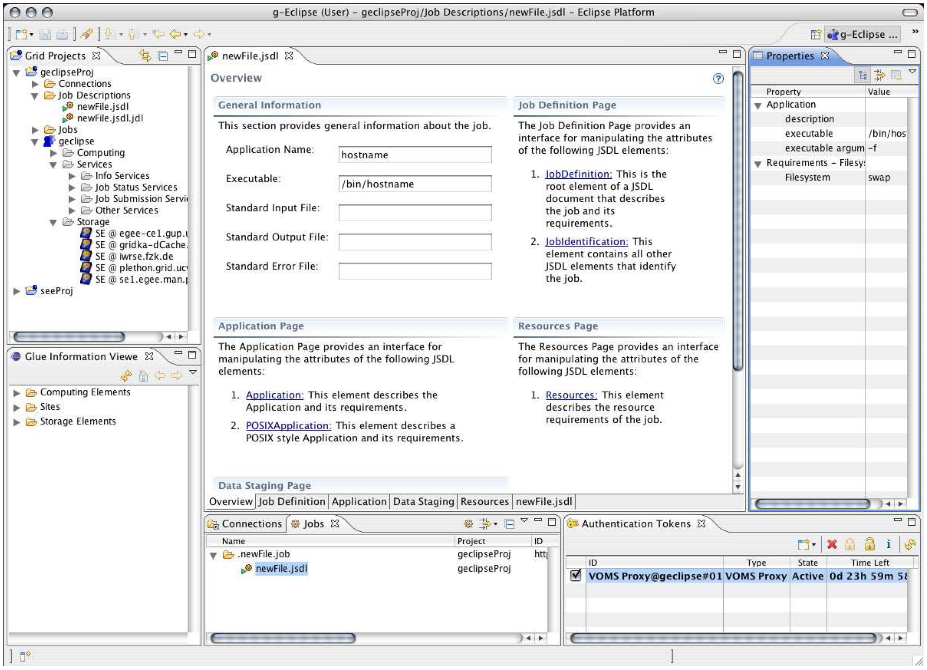
The role based contextualisation of users within the g-Eclipse framework relies on the Eclipse perspective pattern. The current g-Eclipse perspectives consist of a set of GUI elements provided by plug-ins and which combined provide an abstraction of the Grid and assist the actor in the role based Grid interaction.

5.1 Grid Application User

The goal of the user perspective is to simplify the procedure of accessing Grid resources. By using this perspective a Grid application user is able to create a Grid project including remote Grid resources, manage its data, utilise an editor to specify the job particulars and manage the execution of them on a Grid infrastructure.

The predefined user perspective for Grid application users is shown in Fig. 4. This contextualised Grid workbench perspective consists of the following components:

- The Grid project view is part of all contextualised and predefined g-Eclipse perspectives and shown on the top-left of Fig. 4. This view enables the access to Grid data, Grid job descriptions, Grid jobs and to the virtual resources of the assigned VO of the project.
- More details about the Grid resources are available from the Grid information view (see bottom-left). The shown Grid information view supports the GLUE information schema [10].
- The central part of the workbench is the editor area, where data content is shown. In Fig. 4 the JSDL multipage editor is shown as an example of integrated tools of the g-Eclipse framework. This editor supports the Grid application user in the definition of Grid jobs by supporting the JSDL schema [11].
- On the right, the properties view of the Eclipse framework shows the properties of the selected Grid element in the Grid project view.
- The Authentication Token view (bottom right) is part of the security infrastructure of the g-Eclipse framework and supports the Grid user with details about his current security tokens.
- The Jobs view, shown in the lower part of Fig. 4 shows the status of current and past Grid jobs.
- The Connection view (not shown) enables the user to create, move, copy and delete directories and files on local or remote storage devices.
- The Job Description wizard and Job Submission wizard (not shown) supports the user in the process of creating job description and submitting it to the Grid.



General Information

This section provides general information about the job.

Application Name: hostname

Executable: /bin/hostname

Standard Input File:

Standard Output File:

Standard Error File:

Job Definition Page

The Job Definition Page provides an interface for manipulating the attributes of the following JSDL elements:

1. JobDefinition: This is the root element of a JSDL document that describes the job and its requirements.
2. JobIdentification: This element contains all other JSDL elements that identify the job.

Application Page

The Application Page provides an interface for manipulating the attributes of the following JSDL elements:

1. Application: This element describes the Application and its requirements.
2. POSIXApplication: This element describes a POSIX style Application and its requirements.

Resources Page

The Resources Page provides an interface for manipulating the attributes of the following JSDL elements:

1. Resources: This element describes the resource requirements of the job.

Property	Value
Application	
description	
executable	/bin/hos
executable argum	-f
Requirements – Filesy:	
Filesystem	swap

Name	Project	ID
.newFile.job	geclipseProj	htt
newFile.jsdl	geclipseProj	

	ID	Type	State	Time Left
☑	VOMS Proxy@geclipse#01	VOMS Proxy	Active	0d 23h 59m 58

Fig. 4. The g-Eclipse Grid Application User Perspective

5.2 Grid Operator

Grid users can manage and test existing Grid infrastructures by using the g-Eclipse operator perspective. Additionally to the Grid project view and the Grid information view, the operator perspective consists of the following predefined components:

- The Batch service editor supports the operator of a Grid computing resource to manage the available batch system with the help of a graphical editor. The details of the selected batch resource are shown in the properties view.
- The Batch Job view enables the mapping from Grid Jobs to local jobs in the batch queue. With the help of this view, resources providers can organise the execution of jobs in the local batch system.
- The Terminal view enables the operator to connect himself directly to a computing or storage resource.

5.3 Grid Application Developer

Support for the development of Grid applications is provided by the g-Eclipse developer perspective. Nowadays, Grid applications are legacy applications and therefore the migration of these application must be supported by the developers perspective. Applications are developed as separate projects in the local workspace. With the help of the g-Eclipse perspective and its wizards these applications can be compiled and debugged on remote resources, deployed on selected Grid resources. For the development of more complex Grid workflows, a dedicated workflow editor is currently being integrated in the g-Eclipse framework.

5.4 Customisation

The above mentioned perspectives offer only predefined configurations for the proposed Grid user roles. As g-Eclipse is build on top of the Eclipse framework it supports the customisation of these perspectives. The layout of the perspectives is persistent between sessions. Users can customise each of the proposed perspectives to their needs and preferences. Furthermore, new Grid roles can be supported by defining a corresponding predefined g-Eclipse perspective.

6 Conclusions

The g-Eclipse project provides an extensible framework to enable various Grid actors to access Grid resources in an intuitive and easy way. g-Eclipse relies on the Eclipse framework and provides a Grid model to seamlessly integrate Grid resources into the Eclipse framework. Based on this middleware independent Grid model, the g-Eclipse framework can be extended to support different Grid middleware systems. The concepts of Virtual Organisations and Grid projects are important to contextualise the work of Grid users.

The g-Eclipse framework offers a set of tools which are composed to so called g-Eclipse perspectives to support different Grid actors. Currently, three actors – Grid application users, Grid resource operators and Grid application developers – have been identified and are supported with predefined contextualised Grid workbenches. These predefined workbenches can be customised by the Grid users to adjust the workbench to their needs and preferences. Grid actors can easily switch to different workspaces depending on the planned interaction with the Grid infrastructure.

The g-Eclipse framework has been available for potential users since end of 2007. The framework will be tested with an application from the airplane industry when support for the GRIA middleware is completed. The interest in the g-Eclipse framework is increasing and more users are expected to use the g-Eclipse framework for their applications. Reports on user experience with the g-Eclipse framework will be addressed in the future.

The g-Eclipse Grid model and its integration in the Eclipse perspectives framework offer a framework that can be contextualised and customised to the need of future Grid users. The architecture of g-Eclipse enables further mashups between Grid GUI components, SOA GUI components and GUI for Web2.0 by defining personalised and contextualised user workbenches and environments.

Acknowledgments. This work was partly supported by the EU project g-Eclipse (#FP6-2005-IST-034327). We would also like to thank the member institutions of the g-Eclipse consortium and all the project members. Furthermore, g-Eclipse is supported by the Eclipse foundation.

References

1. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. In: Concepts and Architecture, ch. 4, pp. 37–64. Elsevier, Amsterdam (2004)
2. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, December 1994, pp. 85–90. IEEE Computer Society Press, Los Alamitos (1994)
3. The g-Eclipse Project, http://www.geclipse.eu
4. The Eclipse Project, http://www.eclipse.org
5. Gamma, E., Beck, K.: Contributing to Eclipse. Principles, Patterns, and Plugins. Addison-Wesley Longman, Amsterdam (2003)
6. OSGi Alliance, http://www.osgi.org
7. g-Eclipse architecture II, Public project deliverable D1.5 of the g-Eclipse project, http://www.geclipse.eu/fileadmin/Documents/Deliverables/D1.5.pdf
8. Hemmer, F., Laure, E., Barroso Lopez, M., Di Meglio, A., Fisher, S., Guy, L., Kunst, P., Prelz, F.: Middleware for the Next Generation Grid Infrastructure. In: Proceedings of CHEP 2004, Intelaken, Switzerland (2004), http://glite.web.cern.ch/glite
9. Surridge, M., Taylor,S., De Roure, D., Zaluska, E.: Experiences with GRIA - Industrial Applications on a Web Services Grid. In: Proceedings of the First International Conference on e-Science and Grid Computing pp. 98–105 (2005), http://www.gria.org
10. Andreozzi, S., Burke, S., Field, L., Fisher, S., Kønya, B., Mambelli, M., Schopf, J.M., Viljoen, M., Wilson, A.: GLUE Schema Specification, version 1.2, Final Specification (December 3, 2005)
11. Anjomshoaa, A., Brisard, F., Drescher, M., Fellows, D., Ly, A., McGough, S., Pulsipher, D., Savva, A.: JSDL Specification, Version 1.0 (November 7, 2005)