GridSphere's Grid Portlets

Michael Russell [1], Jason Novotny [2], Oliver Wehrens [3]

[1] *Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, Am Mühlenberg 1, 14476 Golm, Germany*
e-mail: {russell/wehrens}@aei.mpg.de

[2] *University of California,* San Diego, CA, USA
e-mail: jnovotny@ncmir.ucsd.edu

[3] *Center for Computation & Technology, Louisiana State University, Baton Rouge, LA 70803*
e-mail: wehrens@cct.lsu.edu

Abstract: Grid portals are the most popular means for delivering user interfaces to Grids. Grid portals build upon the familiar Web portal model to offer virtual communities of users a single point of access to computational resources. The GridSphere Portal Framework is a Portlet JSR compliant portlet container that offers a set of base classes and tools for developing Web applications. The Grid Portlets web application, developed by the GridSphere Project, builds upon the core features in the GridSphere portal framework to provide developers with a framework for developing Grid portals.

Key words: Grid portals, portlet frameworks, JSR 168 – compliant portlets

1. INTRODUCTION

The Web has proven to be an effective means for delivering information, business services, and commercial applications to virtual communities of users. Web portals [1], like Yahoo [2] or MSN [3], offer users a single point of access to a wide variety of services and technologies. Grid [4] portals build upon this model to provide a single point of access to computational resources: clusters, data servers, applications, scientific instruments and computing services. The Web is by no means a perfect solution for delivering applications (slow response times, lack of interactivity and/or difficulty in supporting DHTML [5] across browser platforms). What makes the Web attractive is the relative low cost for delivering new functionality. Nearly every desktop computer has a Web browser installed and most Web browsers now offer support for presenting media and dynamic content through plugins. It's because of the wild success of the Web and the ubiquitous Web browser that Grid portals have become the most popular means for delivering user interfaces to Grids.

2. THE GRIDSPHERE PROJECT

The GridSphere Project [6] was created by the GridLab Project [7] to leverage the most relevant standards, best-practices and technologies to offer a framework for developing Grid portals. One of the most exciting standards to gain adoption by the general community is the Portlet Java Specification Request (JSR 168) [8]. The Portlet JSR defines an application programming interface (API) and model for packaging and presenting Web content as *portlets*. Portlets are Java classes that have a clearly defined interface and life cycle. Portlets are hosted by a *portlet container* and can be presented in a Web page in any manner supported by the portlet container. The Portlet JSR makes it possible to distribute and share Web applications more easily, creating a potential marketplace for Web applications and a means for collaborating on Web portal development on a much larger-scale.

The GridSphere Portal Framework [9], developed by the GridSphere Project, is a Portlet JSR compliant portlet container that offers a set of base classes and tools for developing portlet applications. GridSphere has gained wide usage in the Grid community. The UK E-Science Program [10], D-Grid [11], K*Grid [12] and many other projects around the world have adopted GridSphere as their Grid portal development platform. GridSphere even has a growing user base in the general Web community. This is in part because of GridSphere's ease of installation, its support for customization, its documentation and many other features. But this is also due to the fact that GridSphere does not itself contain any support for using Grid technologies. It contains only the core functionality necessary to develop a Web portal. Instead, the GridSphere Project developed support for Grid development in a web application called Grid Portlets.

3. GRID PORTLETS WEB APPLICATION

The Grid Portlets web application, released for the first time in June 2005, builds upon the core features in the GridSphere portal framework to provide developers with a framework for developing Grid-enabled portlets.
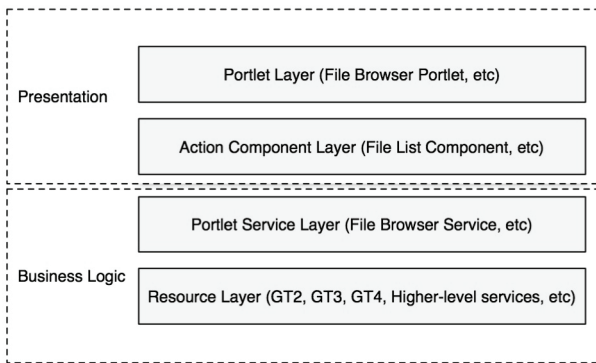
Fig. 1. Grid Portlets has a layered architecture

Grid Portlets has a layered architecture (see Fig. 1). At the top-most presentation level, Grid Portlets provides a collection of simple, easy-to-use, well integrated portlets that can be used on their own or in conjunction with third-party web applications to provide a complete Grid portal solution. These portlets are built upon reusable Java Server Pages (JSP) [13] based user interface components, called *action components*. The *Job Submission Portlet*, for example, utilizes the *File List Component* to enable users to "browse" for executable files, directories in which to run the executables, where to save job output and so on. The same component is also employed by the *File Browser Portlet* to display two "file browsers" in one window for searching for files and transferring files from one location to another. At the next layer down Grid Portlets offers developers a collection of *portlet services*, Java components, for performing tasks on the Grid. These portlet services define an API for interacting with Grid resources. Finally, the resource layer binds the portlet service layer to particular Grid infrastructures and technologies.

4. GRID PORTLET SERVICES

The portlet services contained in Grid Portlets offer a high-level API and model of the Grid, enabling developers to reuse the functionality offered in Grid Portlets to develop custom Grid portal web applications. The Grid Portlet Services API is a collection of Java interfaces and base classes that:

– Can be supported with Globus Toolkit 2 (GT2), GT3, GT4 and other service-oriented technologies [14].
– Provides support for persisting information about resources and tasks performed by users on the Grid.
– Is extensible, builds upon simple concepts to provide more complex services, resources and tasks.

The Grid Portlet Services API is supported by base implementations that make it easy to develop support for particular Grid middleware technologies. Because GT2 is the most widely deployed and supported version of the Globus Toolkit, Grid Portlets is distributed with implementations for utilizing GT2 resources, such as the Globus

Gatekeeper [17] and GridFtp [18]. However, support for Open Grid Services Architecture (GT3) [15] and Web Service Resoucre Framework (GT4) [16] based resources are offered in the GT3 Portlets and GT4 Portlets web applications respectively. These web applications provide GT3 and GT4 implementations of the Grid Portlet Services API and are distributed separately.

4.1. Resources

Central to the Grid Portlet Services API is the notion of a *resource*. Grid Portlets defines a resource as "anything that can be utilized". Resources have a *distinguished name* within the context of Grid Portlets, contain a set of *resource attributes*, and can contain other *child resources*. This very simple concept is quite useful and easy to extend (see Fig. 2). Grid Portlets defines several key resource concepts that extend from this base definition:

– *Hardware resource* – A hardware resource represents a particular host on the Grid. Hardware resources contain service resources, software resources and hardware accounts, all of which are described below.
– *Service resource* – A service resource is a resource that represents a "service" that is accessible over a network. All service resources have at least one "port" associated with them, through which they can be invoked by clients, and a "protocol" for communicating with the service.
– *Software resource* – A software resource is a resource that represents a "software". At minimum, a software resource has a "path" on a given hardware resource.
– *Resource account* – A resource account represents an "account" on a resource. For example, a *hardware account* represents an account on a particular hardware resource.
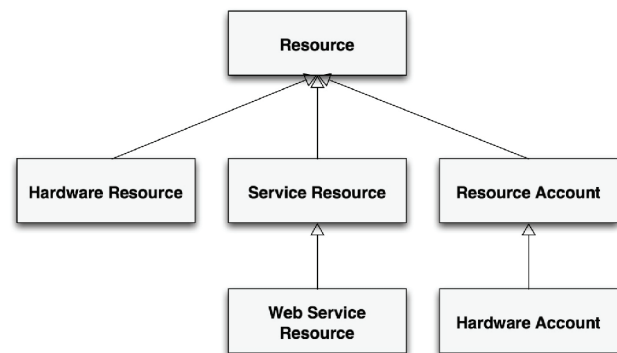

Fig. 2. Grid Portlets resource model

4.1.1. Tasks

Grid Portlets defines a task as a resource for performing a particular operation (see Fig. 3). Tasks are created by submitting a *task specification* to a *task submission service*. A task submission service will first verify the task specification and then submit the task specification to the appropri-

ate *task resource*, returning the resulting task to the user of that service. Tasks have *task statuses*, indicating whether they are *active*, *completed*, *canceled* and so on.

Several portlet services defined in Grid Portlet Services are, in fact, task submission services that provide higher level APIs pertaining to a given problem domain. For example, the *Job Submission Service*, described below, is a task submission service for submitting *job specifications* to remote *job resources*. Job specifications are task specifications that provide methods for defining a job to perform, where a job is a task for executing a particular application with a given set of arguments, input files and so on.

4.2. Key Portlet Services

Grid Portlets defines several portlet services interfaces, the most important of which are described here.

4.2.1. Resource Registry Service

The *Resource Registry Service* is responsible for making resources available to portlets and other portlet ser-
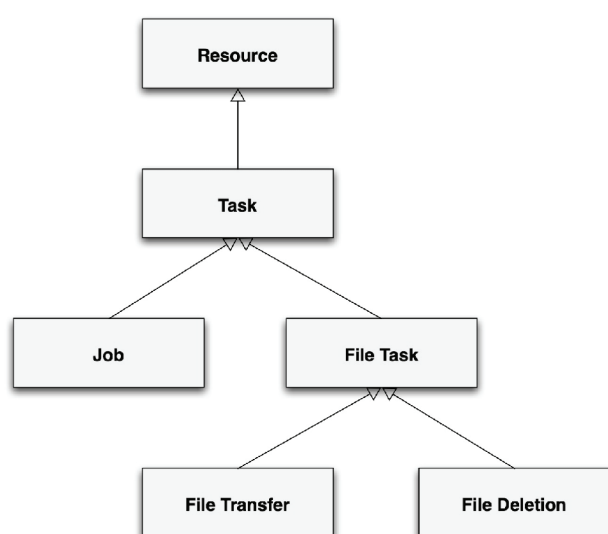

Fig. 3. Grid Portlets task model

vices, where resource descriptions are maintained in the Grid Portlets database. Resource descriptions are registered in the database through one of two means, with a file called Resources.xml, located in the WEB-INF directory of the Grid Portlets web application context, and with *resource provider* services, described below. At startup time, the Resource Registry Service will delete any hardware resource entries currently stored in the Grid Portlets database, save the hardware resource entries described in Resources.xml to the database, then activate the resource provider services that have been registered with Grid Portlets.

4.2.2. Resource provider services

Grid Portlets supports the ability to discover and monitor resources with resource provider services, where resource provider services are responsible for querying (or registering Grid Portlets for notification) the various *information resources* that have been registered with Grid Portlets. Several resource providers can be active at any one time. Grid Portlets is distributed with an *MDS2 Resource Provider Service* that will periodically query *Grid Resource Information Service* (GRIS) [19] resources registered with Grid Portlets for information about the hardware resources on which they are hosted.

4.2.3. Credential Manager Service

The *Credential Manager Service* is responsible for managing *credential contexts* and making active GSS [20] *credentials* that are registered for those contexts available to portlets and other portlet services. A credential context associates a *distinguished name (DN)* with a portal user, records the last time an active credential with that DN was registered with the Credential Manager Service and other useful information about credentials with that DN. A DN can be registered with only one user, while a given user can be associated with multiple DNs. Essentially, Grid Portlets supports the use of multiple credentials, allowing users to gain access to resources from multiple *virtual organizations*.

4.2.4. Credential Retrieval Service

Grid Portlets supports the ability to retrieve GSS credentials from remote *credential repository resources*, such as MyProxy [21], with the Credential Retrieval Service. The *Credential Retrieval Service* is responsible for managing *credential retrieval contexts*. A credential retrieval context associates information for retrieving a credential with a particular DN to a portal user. A portal user can be associated with multiple credential retrieval contexts, thereby allowing a user to retrieve multiple GSS credentials.

4.2.5. File Browser Service

The *File Browser Service* provides methods for creating *file browsers* to remote *file resources*. A file browser represents a *stateful connection* to a file resource and provides methods for listing files, changing directories, creating directories, transferring files between locations and other basic file operations. The File Browser Service is employed by the *File Browser Portlet* and other portlet services for gaining access to remote files.

4.2.6. Logical File Browser Service

Grid Portlets offers a *Logical File Browser Service* for creating *logical file browsers* to *logical file resources*. Logical File Browser Service naturally extends the File

Browser Service and is employed by the *File Browser Portlet*, described below, to provide the same look and feel for browsing logical files as it does for browsing physical files. Logical file browers extend the file browser concept to include additional support replicating files and so on.

4.2.7. Job Submission Service

The *Job Submission Service*, as described above, is a service for submitting jobs to remote *job resources*. Job resources are service resources that provide access to one or more *job schedulers*, where job schedulers define one or more *job queues* to which jobs can be submitted. The Job Submission Service can support job submission to multiple types of job resources at runtime. The GridSphere Project and contributors to the GridSphere code base have developed support for job submission to Globus Gatekeeper (GT2), Master Managed Job Factory Service (MMJFS) [22], and the Grid Resource Management Service [23]. Users of the job submission service can opt to allow the job submission service to determine which type of job resource to use for a given job specification or choose on their own.

Additionally, the Job Submission Service provides methods for listing and deleting records of all jobs previously submitted by users. All job information, in fact all tasks in general, are persisted in the Grid Portlets database. This means Grid Portlets can be shutdown and restarted without losing the ability to monitor jobs.

5. GRID PORTLETS

The portlets contained in the Grid Portlets web application leverage the Grid portlet services described above to provide a generic, yet powerful set of user interfaces for using resources on the Grid.

5.1. Resource Registry Portlet

Grid Portlets is essentially configured through the *Resource Registry Portlet* (see Fig. 4). Here, portal administrators can specify the set of resources their Grid portal makes available to its users by editing the Resources.xml file online.

5.2. Resource Browser Portlet

The *Resource Browser Portlet* (Fig. 5) enables users to view information about the hardware resources that have been registered with Grid Portlets, including the services, job queues and accounts that are available on those resources.

5.2.1. Resource Profiles

The Resource Browser Portlet supports a plug and play model, allowing developers to easily customize the information that is displayed about each resource. Grid Portlets defines a *resource profile* as a set of user interfaces for viewing resources of a particular resource type. Multiple resource profiles can be registered per resource type with the *Resource Profile Registry*. If only one resource profile
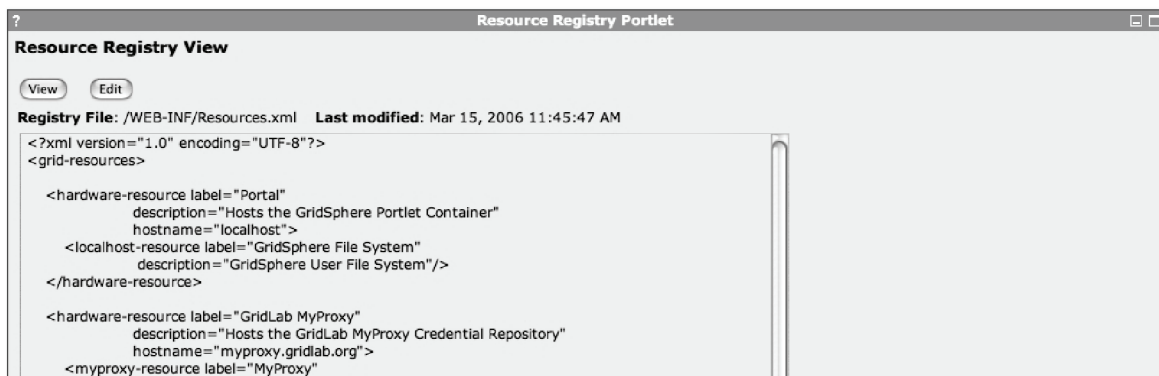

Fig. 4. Edting the resource registry with the Resource Registry Portlet
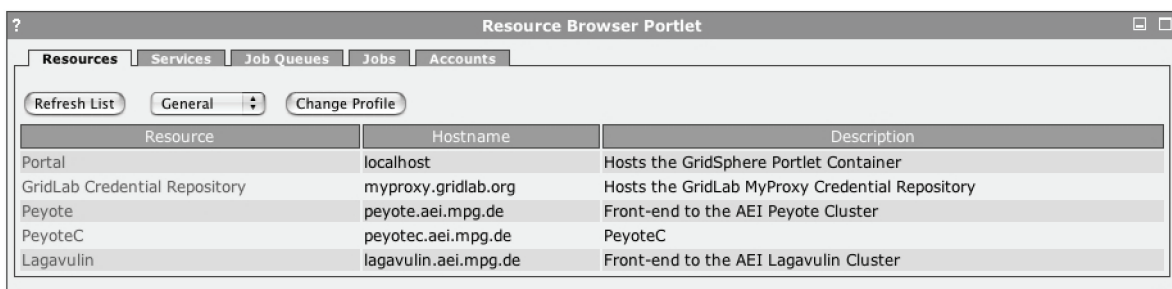

Fig. 5. Viewing resources with the Resource Browser Portlet

is provided for a given resource type, the Resource Browser Portlet will simply display that resource profile. If there is more than one resource profile per resource type, then the Resource Browser Portlet will present a list box to allow users to select the appropriate resource profile with which to view a given resource. The order in which this list is displayed is configurable. By default, Grid Portlets offers a *Generic Resource Profile* per resource type, showing high--level information about each resource. Portal administrators can easily override this profile, or add new profiles, depending on the resource profiles that are available in the web applications that have been deployed to the web application server.

In general, resource profiles are associated with particular resource provider services. Resource profiles can be registered to display only when certain information resources are available to the Grid portal. For example, the Grid Portlets web application is distributed with a MDS2 resource profile for viewing hardware resources. The ability to select this profile to view a hardware resource depends upon whether a GRIS resource has been registered for that hard-

ware resource. Selecting the MDS2 profile for the given hardware resource will then load a user interface that displays information that has been collected from that GRIS resource. Figure 6 depicts the GridScreen profile, developed for use at AEI.

5.3. Credential Manager Portlet

The *Credential Manager Portlet* (Fig. 7) enables users to delegate credentials to the portal from credential repositories, such as MyProxy, and monitor the status of those credentials. When a user defines a new credential retrieval context, they can specify whether to enable that credential for *single sign-on*. When single sign-on is enabled, the next time the user logs into the portal their credential will be automatically delegated to the portal if they enter their credential retrieval password.

5.4. File Browser Portlet

Users can browse for files on remote file systems with the *File Browser Portlet* (Fig. 8). The File Browser Portlet
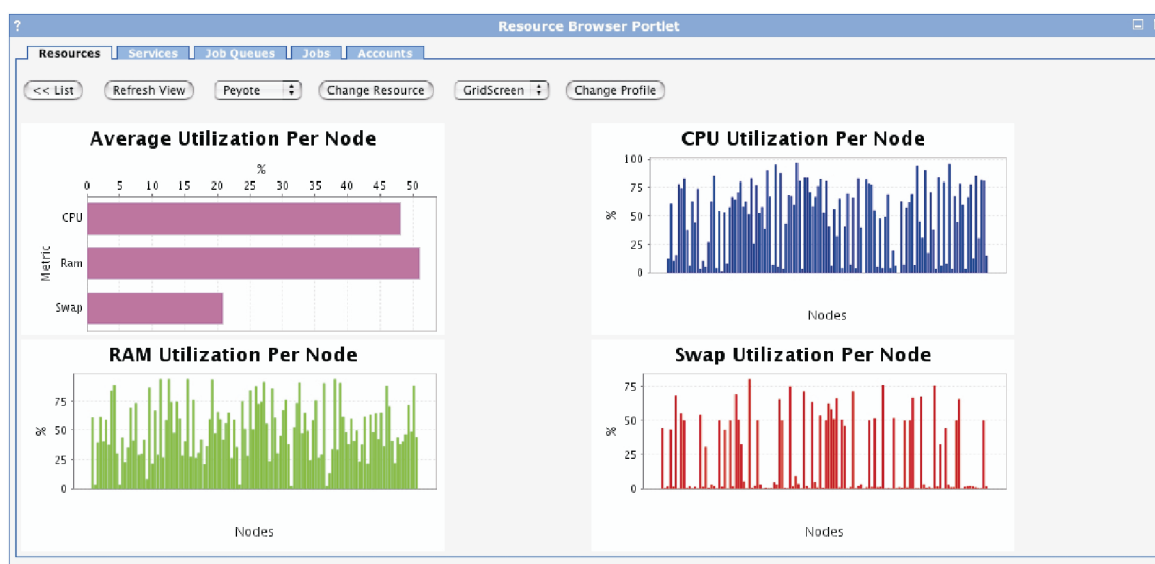

Fig. 6. Viewing a cluster with the GridScreen Profile, developed at the Albert-Einstein-Institut

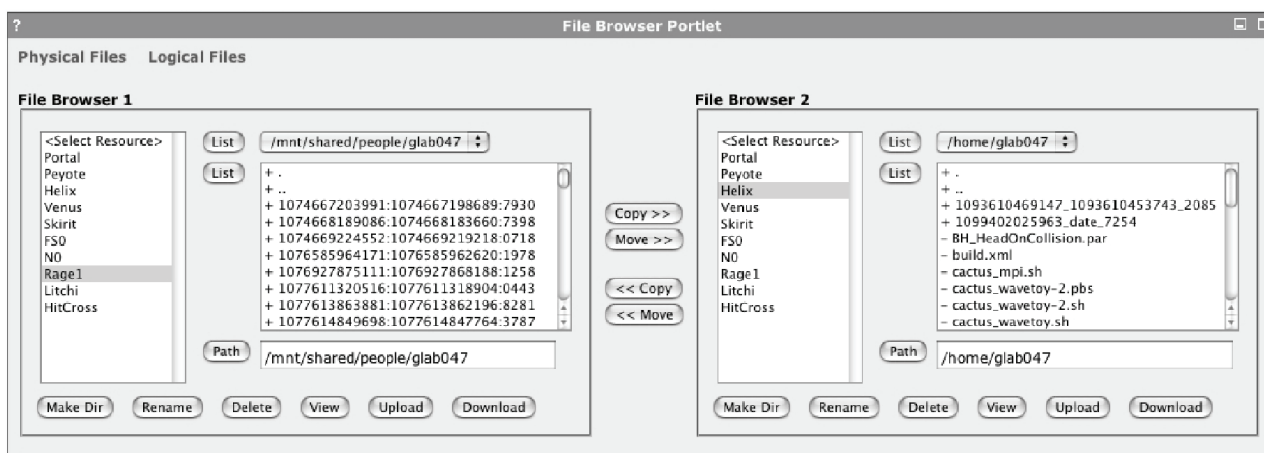
Fig. 7. Defining a new credential retrieval context

Fig. 8. Selecting files to transfer with the File Browser Porlet

displays two file browser user interfaces side-by-side, enabling users to transfer files and directories between locations. The file transfers are performed asynchronously, so that control returns immediately to the user after starting a file transfer. Users can then monitor the progress of the transfer with the *File Activity Portlet*, described below. The File Browser Portlet also supports the ability to create new directories, delete files and directories, edit text files online, upload files to remote file systems and download files to a user's local host, so long as the files aren't too large.

Because the File Browser Portlet uses the File Browser Service, it can support multiple types of file resources. Grid Portlets is distributed with two types of file resource implementations, a *secure directory file resource* and a *Grid FTP file resource*. The secure directory file resource allows users to securely interact with files stored on the web application server, while the Grid FTP resource allows users to interact with remote files over the Grid FTP.

5.4.1. Browsing logical file systems

If logical file resources have been registered with Grid Portlets, then the File Browser Portlet will automatically make those logical file resources available to users for browsing. The semantics for browsing logical files are essentially the same, the only difference is that the logical file interface defines additional methods for associating physical files with logical file entries. On the other hand, file transfers are not supported between a logical file resource and a physical file resource, because this does not

make sense technically speaking. Logical file systems operate at a layer above physical file systems.

5.5. File Activity Portlet

The *File Activity Portlet* (Fig. 9) allows user to monitor the progress of file tasks, such as *file copy tasks*, *file move tasks*, and *file upload tasks*.

5.6. Job Submission Portlet

The *Job Submission Portlet* enables users to submit jobs to remote job schedulers and monitor the status of jobs. Because the Job Submission Service employs the Job Submission Service, it supports the ability to submit jobs to multiple types of job resources. If more than one job resource is registered with Grid Portlets, then Grid Portlets will make each job resource type available as a "job submission service" selection when a user clicks on the New Job button.

5.6.1. Job Profiles

The *Job Submission Portlet* (Fig. 10) supports a plug and play model, allowing developers to easily add user interfaces tailored to particular applications. Grid Portlets defines a *job profile* as a set of user interfaces for specifying and viewing jobs. Multiple job profiles can be registered with the *Job Profile Registry*. If only one job profile is registered, the Job Submission Portlet will simply display that job profile when a user clicks the New Job button. If there is more than one job profile available, then the Job

Fig. 9. Monitoring the status of file tasks with the File Activity Portlet

Fig. 10. Selecting a job submission service with the Job Submission Portlet

Submission Portlet will present a list box to allow users to select the appropriate job profile with which to edit jobs. The order in which this list is displayed is configurable. By default, Grid Portlets offers a *Generic Job Profile* (Fig. 11) that presents a generalized interface for specifying a job. Portal administrators can easily override this profile, or add new profiles, depending on the job profiles that are available in the web applications that have been deployed to the web application server.

Fig. 11. Editing a job with the Generic Job Profile

6. ADDITIONAL FEATURES

Because Grid Portlets builds upon GridSphere's core features, it supports the ability to localize the portlets to display in a particular language as well as to display user interfaces in different themes. However, Grid Portlets goes beyond these key features to offer a more powerful way for developing user interfaces. All of the user interfaces contained in Grid Portlets can be reused in third-party web applications that employ the action component model.
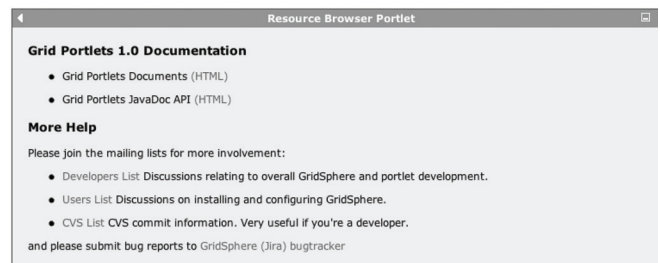

Fig. 12. Accessing help from the Resource Browser Portlet

6.1. Online Help

Grid Portlets makes the documentation written for Grid Portlets accessible in the HELP mode of each portlet. In fact, any portlet that employs the action component model will have access to this documentation (Fig. 12) in its HELP mode.

7. CONCLUSION

The GridSphere Project is proud to offer the Grid Portlets Web Application to the general community. We have many plans in store for Grid Portlets, including future co-releases of GT3 Portlets and GT4 Portlets projects which offer GT3 and GT4 implementations of the Grid Portlets Services API respectively.

Acknowledgments

References

[1] *Web Portals. A Word Definition from Webopedia*, http://www.webopedia.com/TERM/W/Web

[2] Yahoo! Personalized content and search options. Chatrooms, free e-mail, clubs, and pager, http://www.yahoo.com/

[3] MSN. Features personalization, channels of content sites like Carpoint, and integration with Hotmail e-mail. http://www.msn.com/

[4] I. Foster, *What is the Grid? A Three Point Checklist*, GRID-Today, July 20, 2002.

[5] *Dynamic Hyper-Text Markup Language HTML. A Word Definition from Webopedia*, http://www.webopedia.com/TERM/D/dynamic

[6] GridSphere Project, http://www.gridsphere.org/

[7] GridLab: A Grid Application Toolkit and Testbed, http://www.gridlab.org/

[8] Java Community Process: JSR 168 Portlet Specification. Project Website. Dec 1, 2004, http://www.jcp.org/jsr/detail/168.jsp

[9] J. Novotny, M. Russell and O. Wehrens, *GridSphere: An Advanced Portal Framework,* EUROMICRO, 2004, http://www.gridsphere.org/gridsphere/gridsphere?cid=presentations

[10] UK National E-Science Centre, http://www.nesc.ac.uk/

[11] D-Grid Program, http://www.d-grid.de/

[12] Korean National Grid Program, http://gridcenter.or.kr/

[13] Java Community Process: JSR 152 Java Server Pages 2.0 Specification, http://www.jcp.org/en/jsr/detail?id=152

[14] Globus Toolkits, http://www.globus.org/toolkit/

[15] I. Foster, C. Kesselman, J. Nick and S. Tuecke, *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration,* Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[16] I. Foster, J. Frey, S. Gram, S. Teucke, C. Czajkowski *et al.*, *Modeling Stateful Resources With Web Services,* http://www-128.ibm.com/developerworks/library/ws-resource/ws-modelingresources.html, March 5, 2004.

[17] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith and S. Tuecke, *A Resource Management Architecture for Metacomputing Systems*, Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, pp. 62-82, 1998.

[18] W. Allcock, Editor, *GridFTP: Protocol Extensions to FTP for the Grid, Global Grid Forum Draft Standard* (April 2003), http://www-isd.fnal.gov/gridftpwg/draft/GridFTPRev2.pdf.

[19] K. Czajkowski, S. Fitzgerald, I. Foster and C. Kesselman, *Grid Information Services for Distributed Resource Sharing,* Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[20] RFC 2853 – Generic Security Service API Version 2: Java Bindings, http://www.faqs.org/rfcs/rfc2853.html

[21] J. Novotny, S. Tuecke and V. Welch, *An Online Credential Repository for the Grid: MyProxy,* Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[22] V. Silva, *The Master Managed Job Factory Service (MMJFS): A custom GRAM client for the Globus Toolkit 3.x,* http://www-106.ibm.com/developerworks/grid/library/gr-factory/

[23] K. Kurowski, B. Ludwiczak, J. Nabrzyski and A. Oleksiak, J. Pukacki, *Improving Grid Level Throughput Using Job Migration and Rescheduling Techniques in GRMS. Scientific Programming*, IOS Press. Amsterdam The Netherlands **12(4)** 263-273 (2004).
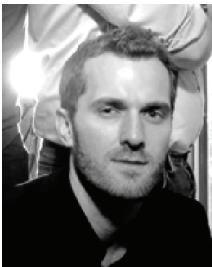
MICHAEL RUSSELL began working under the direction of Ian Foster at the University of Chicago in May 2000 where he lead development of the Astrophysics Simulation Collaboratory Portal and helped to disseminate information about the Globus Project. In May 2002 he moved to Berlin, Germany to manage the Grid Portals Work Package of the GridLab Project at the Max Planck Institute for Gravitational Physics (Albert Einstein Institute). While working on the GridLab Project, he and his colleagues created the GridSphere Portal Framework. Now, as group leader of the GridSphere development team at the Poznan Supercomputing and Networking Center (PSNC), he is helping to deliver Grid portal solutions for the Open Middleware Infrastructure Institute for Europe, HPC Europa and several other projects in Europe.

JASON NOVOTNY is the chief architect of the GridSphere portal http://www.gridsphere.org and is involved with overseeing the development of the production bioinformatics research portal at the University of California, San Diego. Jason has been involved in distributed computing research and Grid computing since the early beginnings in 1998 and authored the Grid Portal Development Kit (GPDK) and co-authore the MyProxy online credential repository software. His interests include distributed computing frameworks and middleware, distributed system security, high performance networking, and high level tools and components for developing distributed computing applications using J2EE. Jason Novotny has been involved with developing and deploying distributed applications and Grid middleware for the DOE Science Grid, the NCSA/Alliance Virtual Machine Room (VMR) and the NASA information Power Grid (IPG) and has been an active participant in Grid Forum since its inception.

OLIVER WEHRENS is one of the founders and a core team member of the GridSphere Portal Framework. Currently he is head of the portal development in Germany's D-Grid project and coordinates the efforts there as well as working at CCT in the portals environment group. He is involved with Grid and Supercomputing since 1997.