

Service Oriented Architectures for Science Gateways on Grid Systems

Dennis Gannon, Beth Plale, Marcus Christie, Liang Fang,
Yi Huang, Scott Jensen, Gopi Kandaswamy, Suresh Marru,
Sangmi Lee Pallickara, Satoshi Shirasuna, Yogesh Simmhan,
Aleksander Slominski, and Yiming Sun

Department of Computer Science, Indiana University,
Bloomington Indiana, USA
gannon@cs.indiana.edu

Abstract. Grid computing is about allocating distributed collections of resources including computers, storage systems, networks and instruments to form a coherent system devoted to a “virtual organization” of users who share a common interest in solving a complex problem or building an efficient agile enterprise. Service oriented architectures have emerged as the standard way to build Grids. This paper provides a brief look at the Open Grid Service Architecture, a standard being proposed by the Global Grid Forum, which provides the foundational concepts of most Grid systems. Above this Grid foundation is a layer of application-oriented services that are managed by workflow tools and “science gateway” portals that provide users transparent access to the applications that use the resources of a Grid. In this paper we will also describe these Gateway framework services and discuss how they relate to and use Grid services.

1 Introduction

A Grid is a network of compute and data resources that has been supplemented with a layer of services that provide uniform and secure access to a set of applications of interest to a distributed community of users. The most significant examples of Grid systems have come from communities engaged in distributed scientific collaborations. For example, NEESGrid [1] is a set of shared resources used by earthquake engineers. The Particle Physics Data Grid [2] is a collaboration based on sharing data and analysis tools used in the hunt for subatomic particles. There are many more examples. There is also now a very active industrial community that is defining Grid technology in terms of the requirements of data center management and application service provisioning.

In the early days, Grid systems were built with ad hoc collections of software, but the emergence of Web Services has galvanized the community around Service Oriented Architectures (SOAs). Two organizations have emerged to help organize standard for these groups. The Enterprise Grid Alliance [3], led by Oracle is defining use cases for service frameworks for the data center. The Global Grid Forum (GGF) [4] is the older and larger organization that represents both the scientific and industrial community in defining the standards for Grid technology. GGF is organized along a standards track and a community track. The focus of the standards track is the Open

Grid Services Architecture (OGSA), which is being promoted by GGF as the future SOA for Grid systems. The community track is a forum of research groups that are looking at the role of new technologies in both the scientific and vertical market domains such as telecommunications, biotechnology and media.

In this paper we will provide a very brief, high level overview of OGSA and then turn to a discussion of the service architecture that is used by virtual organizations centered around scientific applications of Grid systems. Within the Teragrid project [5] these are referred to as “Science Gateways”. The goal of a science gateway is to provide a community of users access to scientific tools and applications that execute on the back-end compute and data resources. The users should be able to use the applications as an extension of their desktop without ever knowing that there may be a massive Grid framework in the background supplying the computing power. Typically these gateways are organized around a web portal and a family of desktop tools. The portal server authenticates the user and establishes the user’s authorization to access data resources and applications. The applications often take the form of workflow templates that are instantiated and executed on the user’s behalf. As illustrated in Figure 1, the workflow engine must interact with application metadata and data services, application registries and data directories and Grid resource brokers. Notification services are used to log and monitor application progress and to create the provenance documentation needed to make computational experiments repeatable.

There are many examples of gateway systems in use today and each has a slightly different version of the service architecture that supports it. In many cases the gateway services are built directly on top of an OGSA-like SOA, and in other cases the true Grid layer is very thin or even non-existent. In this paper we will describe examples of each and we will consider one in detail that does build upon a solid Grid foundation.

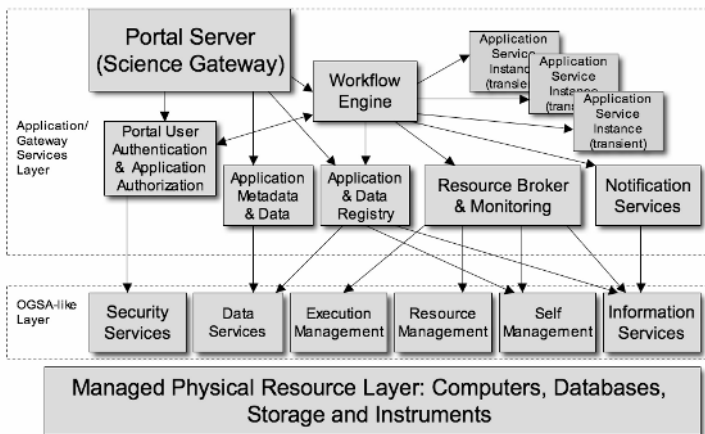


Fig. 1. Service Architecture for a Science Gateway

2 The Open Grid Service Architecture

The Open Grid Service Architecture (OGSA) is a product of the Global Grid Forum OGSA-WG led by Hiro Kishimoto. The first specification of OGSA [6] can be viewed as a profile for the organization of a standard Grid. OGSA contains six families of services which, when properly integrated, deliver a functioning Grid system. It must be noted that, at the time of this writing, there are no official implementations of OGSA, because the details of a basic service profile is still being developed. However, understanding the six core components can be a useful way to understand how Grid systems differ from other SOAs. We describe each of these service classes below.

2.1 Execution Management Services

Most Grid systems must manage the execution of computing tasks on the resources that comprise the Grid. OGSA models execution management in terms of three classes of services: Resources, Job Management and Monitoring, and Resource selection. The Resource services describe service containers and persistent state handlers. The Job Manager handles the full lifecycle of the execution of a set of jobs. It interacts with task queues on each computation resource as well as the other services involved in resource brokering and resource monitoring. The resource selection services consist of execution planning, which build schedules of jobs and resources, candidate set generation services, which produce the likely resources for running a particular job or set of jobs, and reservation services which interact with accounting and authorization systems. One interesting outcome of this work has been the Job Submission Description Language that is a schema for describing jobs. JSDL [7] is being used by a variety of Grid projects including the large Japanese Grid project NARAGI [9] and GridSAM [8] from the London e-Science Centre and the Open Middleware Infrastructure Institute [10].

2.2 Data Services

OGSA data services are intended to address the movement and management of a number of different data resources. These include the standards such as flat files, data streams and relational, object and XML databases. But they are also concerned with derivations, i.e. data that is derived by queries or transformations on other data, and Data services such as sensors. The types of activities that must be supported by the data services include remote access, staging, replication, federation, derivation and metadata generation and management. In addition, these capabilities are to be presented to the user in the form of virtualized services that hide the different implementations that are required to support different media and low-level data types. Virtualized services are a way to realize in practice the distributed systems notion of “access transparency”.

The OGSA working groups involved with defining the specific data services are still hard at work. However, there are important pieces that are currently in use. One important component is OGSA Data Access and Integration [11], which establishes the definition and development of generic Grid data services providing access to and integration of data held in relational database management systems, as well as

semi-structured data held in XML repositories. Another important contribution is the replica location service provided by the Globus toolkit GT4.

2.3 Resource Management Services

There are three categories of resource management that are of concern to OGSA. First there are the actual physical resources: computers, networks, storage systems and instruments. At the lowest level this management is done through standard protocols and frameworks like CIM and SNMP. But OGSA stipulates that there is another intermediate level where a common interface and approach is needed. This is where the Web Service Resource Framework (WSRF), a proposed standard, is most appropriate because it gives a standard way to discover and interrogate services that interact with the management interface of each resource. WSDM, Web Services Distributed Management, is an additional tool that OGSA envisions using for this activity.

The second class of resource management involves resources of a Grid such as resource reservation and monitoring. The third class is the management of the OGSA infrastructure itself. There are two type of interfaces to these management services: functional interfaces, which accomplish tasks such as creating or destroying a job, and manageability interfaces, which provide the mechanisms to manage a capability, such as monitoring a job manager. In general, these services provide resource reservation, monitoring and control of resources, virtual organization management, problem determination and fault management, metering and policy management.

2.4 Security Services

The OGSA security services are designed to make it possible to enforce the security policies of a particular Grid and its member organizations. OGSA postulates the existence of six services: a credential validation service, a trust service, an authorization service, an attribute service, an audit service and a bridge translation service. Though OGSA does not give the precise definition of these services they observe that the services must support the following capabilities:

- Authentication. The credential validation and trust service should be able to verify an identity assertion.
- Identity Mapping. The trust, attribute and bridge translation service should enable the translation of an identity that is valid in one domain within the Grid into an identity that is valid in another domain within the same Grid.

Authorization should be provided by the authorization service. The audit service tracks security-relevant events and is policy driven.

2.5 Self-management Services

An important concept in OGSA is that interactions between users and services are largely based on Service Level Agreement (SLA), which are documents that govern the way transactions are carried out. For example, when submitting a job, a user negotiates the jobs priority, a guaranteed completion time and required resources with a service level manager to arrive at a working SLA. Self-management services

automate the tasks of configuration, healing and optimization needed to keep the Grid operating correctly and meeting its SLAs. The service level management services operate by monitoring load and utilization of resources and the running state of the other services. Based on the monitoring data, the management services must do an analysis to make sure that all the SLAs can be satisfied. If not, the management services must adjust priorities or provision additional resources.

2.6 Information Services

Information services provide the mechanisms for the other Grid services to learn about dynamic events used for status monitoring and directory information that is used to discover services and logged data. The information services are typically based on a web service publish-subscribe event notification system such as WS-Eventing or WS-Notification. But dynamic directory and query services also play a critical role.

A closely related and important concept is that of naming. OGSA assumes a three level naming systems in which the top-level is a human readable name. The middle level is a persistent abstract unique identifier. The lowest level is the actual address (or addresses) of the object being named. For example, a Grid notification may state that a new resource exists identified by its abstract unique name. A user or another service can use a name resolver service or directory to obtain an address for this object. A human user can use the directory services to discover entities that correspond to a particular human readable name.

3 Actual Grid Systems

OGSA is still a work in progress, so there are no certified implementations. However there are a number of software stacks that are available that are used in different Grid deployments and many of these contain many of the features of OGSA. Several of these are available as open source systems and are used extensively in the scientific community. The Globus Toolkit GT4 [12] is the most frequently used. It contains elements of all of the OGSA core service areas except for self-management. It will be used as the core service layer for the National Science Foundation TeraGrid project [5]. It is also used in the bioinformatics Grid GeneGrid [13], and GridCast [14], a Grid to support the delivery of multi-media for the BBC. The Laser Interferometer Gravitational Wave Observatory (LIGO) [15] uses GT3, the previous version of Globus. GT3 is also used in the Network for Earthquake Engineering [1] and Cancer Biomedical Informatics Grid caGrid [16].

Another SOA for Grids, gLite [17], coming out of CERN, supports research in high-energy physics. gLite is used in the “Enabling Grids for E-Science” EGEE project [18] and the LHC Computing Grid [19]. Another large Physics project is the Open Science Grid OSG [20], which uses elements of GT4 and gLite. The Legion system, which is one of the oldest software platforms for Grid computing is being redeveloped as a web SOA by the University of Virginia and is being used in the Global Bio Grid [21]. Another early OGSA-like Grid is Discovery Net [22]. In addition to these OGSA-like SOAs used in large science Grids, there are several

commercial products that are available and in use in the enterprise computing sector. We expect that many of these will evolve into close compliance with OGSA.

4 The Application Service Layer: Scientific Gateways

There are really two types of Science Gateways. One type is a service gateway that bridges two Grids. An example of this is a proposed Gateway between TeraGrid and the Open Science Grid (OSG) that will allow OSG users access to TeraGrid resources. The second type of Gateway is a collection of tools that allow a large number of users transparent access to remotely deployed application and database services. Typically these systems use a web portal and desktop tools such as a workflow composer and visualization tools. There are many excellent examples of this type of service organization. Some projects use an extensive Grid infrastructure similar to OGSA as the underlying foundation, while others are relatively lightweight and self-contained. For example, the Taverna [23] system, which is widely used in biomedical applications, does not rely on an underlying Grid. Rather it directly orchestrates web service and other services available on the Internet. Another powerful tool to help scientists orchestrate web services and other applications is Kepler [24]. While Kepler can be used as a desktop tool to orchestrate simple services, it can also be used in a large grid-based gateway, such as the Biomedical Informatics Research Network BIRN [25]. Finally, Triana [26] is a workflow composition tool that can be used by scientists either as a desktop application, or as a component to a larger Grid Gateway, such as used with the GridLab project [27]. Another excellent Gateway that does not require extensive backend OGSA-style Grid support is ServoGrid [28], a portal for computation geophysics. On a closely related topic, the Earth Systems Grid [29] provides an excellent portal that provides access to tools for climate research. Unlike some of the others, it is based on a substantial Globus-based Grid foundation.

To provide a better idea of what a Grid based Gateway architecture looks like, we will look at one project in detail. The Linked Environments for Atmospheric Discovery (LEAD) project is an National Science Foundation sponsored effort to vastly improve our ability to predict tornadoes, hurricanes and other mesoscale weather events. The project, led by Kelvin Droegemeier at the University of Oklahoma involves seven other primary institutions¹. The use-case for LEAD can be described as follows. Vast arrays of instruments are constantly collecting data about the weather. This includes ground sensors measuring pressure, humidity, lightning strikes as well as Doppler radars and airborne detectors such as balloons and commercial aircraft, and satellites. There are also substantial data about previous weather events. In LEAD, data-mining agents will monitor the instrument data streams looking for interesting emerging severe storm conditions. When something significant is detected, the agent will broadcast notification events to a workflow engine. These events will trigger one or more waiting workflows to begin executing. The job of the workflows is to interact with resource broker services to invoke a series of data “ingest” analysis services and start up an “ensemble” of weather forecast simulations each representing a

¹ Indiana University, the University of Alabama Huntsville, the National Center for Supercomputing Applications (NCSA), UNIDATA/UCAR, Millersville University, Howard University and the University of North Carolina.

slightly different scenario. (An ensemble may be a few dozen simulations or hundreds, depending on resource availability.) As these simulations proceed other data analysis services will compare the output of each with the evolving state of the real weather. Those simulations that fail to track reality will be terminated and those that are doing well will be given more resources. As the simulations predict a developing area of trouble, the workflows will be able to direct some of the instruments, such as the newest generation of Doppler radars, to gather more detailed information in the region of concern. This improved data can be used to increase the resolution of the simulation. When the simulations begin to converge on a serious severe storm in the making, the scientific team will be alerted. At this point the scientists may choose to interact with the workflow to cause additional scenarios to be explored or to generate visualizations.

In addition to this real-time adaptive storm prediction scenario, LEAD must also support a workbench for researchers, teachers and students of all age groups. This diverse user base has vastly different requirements. The LEAD Gateway service architecture has components that support two primary activities:

- Data discovery, data and metadata management, and data storage.
- Workflow management for simulation and data analysis services.

To support these activities there are six persistent services as illustrated in Figure 2. The portal server is user's primary access point. When the user authenticates with the server (via standard https protocols), the portal server fetches the user's Proxy X.509 certificate and a set of SAML based authorization tokens. The authorization tokens determine which other services the user has access to. The portal server presents the user with a series of JSR-168 portlets that provide the interfaces to the other core services.

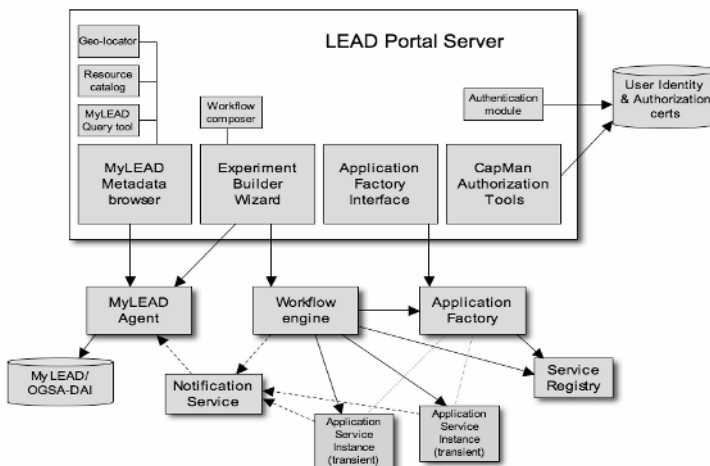


Fig. 2. LEAD Gateway Services

For most users, the primary portlet is the MyLEAD service [30] and associated tools. MyLEAD is a metadata catalog of each users experimental data and results. The tools associated with MyLEAD include an interactive query tool that allows a user to search for data based on a variety of experimental attributes, a Geo-query tool that allows a user to define data in terms of geospatial (*i.e.*, map coordinates), temporal and data attributes, and a resource catalog, that allows the user to select data from public weather services such as UNIDATA Local Data Manager (LDM) servers. The portal components of MyLEAD talk to the MyLEAD agent web service which is the front end for the MyLEAD catalog which is built on the OGSA-DAI service [11].

The “high-end” experimental users of LEAD need the ability to integrate large scale simulation and data analysis codes into experimental workflows. The way they do this is through an Application Factory service. They begin by deploying their application on some host in the back-end Grid. A description of the deployment and how it is invoked (including the types of input files it needs and output files it generates) is encoded by the scientist into a “service map document”. This document can then be uploaded into the portal application factory interface, which passes it to the application factory service. The application factory is now able to create a web service component that can launch the application as part of a workflow or directly from the portal. The WSDL for the generated application service is loaded into a service registry, which uses soft-state concept of “leases” to keep track of the state of the service. These application service instances are not assumed to be persistent because they run on remote hosts as user processes. The application factory is capable of re-starting an application service that is no longer running if it is needed by a workflow.

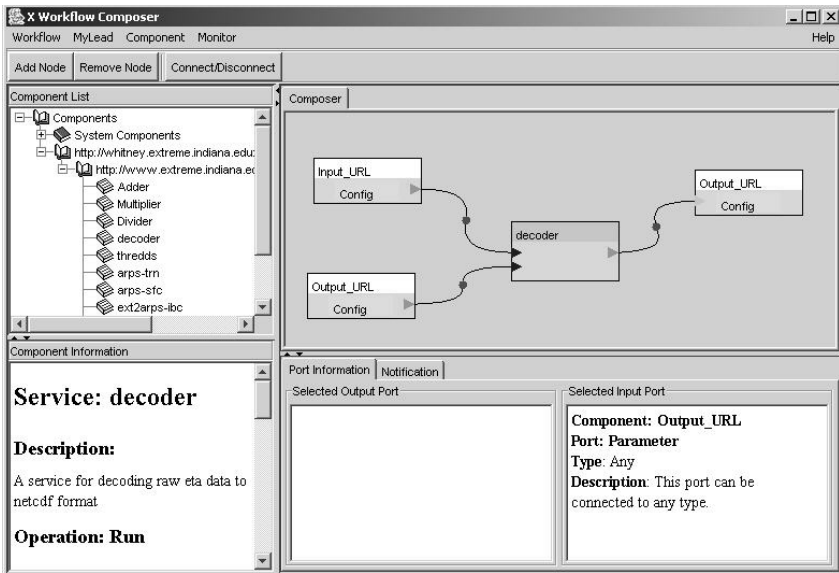


Fig. 3. The workflow composer tool

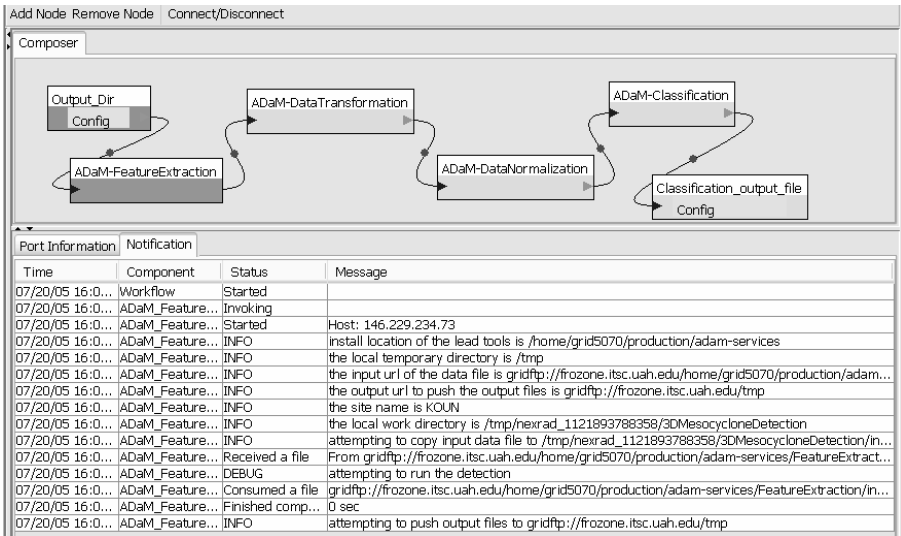


Fig. 4. The workflow composer can be used to monitor events published by a data mining workflow

The workflow service is based on a limited version of BPEL we call GPEL. It has been designed specifically to host long-running workflows that interact with application and data management services. Because BPEL is not a language that atmospheric scientists find very friendly, we have built a graphical user interface along the lines of Kepler and Triana. As shown in Figure 3, this interface allows the user to compose application components from a pallet of known application components. The composer tool compiles the graphical description into a BPEL document which is stored in the user's persistent space (i.e., MyLEAD) and in the workflow engine. Because the application services typically work by consuming and producing large files, the input and output components allow the user to specify these files by the persistent abstract unique identifier which can be used by a Grid name resolver to locate and stage a copy of the desired object.

The final service is Notification. The LEAD notification system is based on a service that implements both the WS-Eventing and an early version of the WS-Notification standards. This critical service ties together many of the components of the LEAD gateway. Every application service publishes events about its current state for each invocation of that service. Collectively, the event histories for each service invoked as part of a workflow form a record of each computational experiment. The entire event history is logged in the user's MyLEAD space as part of the record for that experiment. This gives the user the ability to examine each step of the process and also provides provenance information for every data product generated along the way. As shown in Figure 4, it is possible for the workflow composer to subscribe to the experiment event stream and monitor it as it progresses. This has proven to be an essential tool for debugging. The anticipated number of users who will actually compose and debug new workflows is expected very small. But to simplify their task, the portal contains an Experiment wizard which is able to guide a user through all the

steps required to set up an experiment, compose a workflow (or select a pre-composed workflow template and bind parameters to it), and run it. The results are automatically registered with MyLEAD.

The vast majority of LEAD Gateway users will probably only use the MyLEAD interface to browse data and apply canned transformations to them. Teachers will want to present students with simple simulation scenarios and allow them to execute them with limited capabilities. To facilitate this we are developing special tools that allow teachers to imbed a “user interface” to a workflow so that any authorized student can enact it. The user interface can be embedded into the xhtml text of an educational module.

5 Conclusions

It should now be clear that a service-oriented architecture is both an elegant and a practical way in which to provide scientific communities shared access to tools and resources. SOAs are the basis for both OGSA-style Grids and science gateways. These gateway frameworks are usually built around a web portal and a set of desktop clients that access backend application services. The OGSA group has done an excellent job of characterizing the families of services required for different classes of computing and data Grids. What remains unclear is the degree to which a gateway framework needs the full power of an OGSA-style Grid. In many communities, it is sufficient to provide a portal and tools that allow users unfettered access to public web services. These communities resist installing complex grid infrastructures because of the management overhead it requires. They want lightweight, simple solutions that can be dropped into any user’s environment with little or no effort.

In other cases, such as the LEAD example described here, a full-blown Grid infrastructure is needed to handle the task of doing state-of-the-art predictions of storms. Yet LEAD must also serve a community of educators and scientists that demand lightweight solutions. The dividing line between these two extremes involves the size of the data and the scale of computing required to satisfy the users needs. The high-end of LEAD tasks require massive data management and supercomputing support, while the educational environment requires modest data access and analysis and very limited simulation capability. In the majority of life science applications, the work is data analysis and transformation based on relatively small data sets available from public web services. Consequently, large Grid deployment is not always necessary in many bioinformatics applications.

A flexible service oriented architecture for Science Gateways is one that allows application services to be installed as simple components on any platform, yet they must be able to connect and interoperate with the larger, more secure back-end Grid services in a manner that is transparent to the user. By building the gateway service architecture as a modular component framework where one can deploy application services as needed, it is possible to create a system that works in a local environment but also exploits the power of large Grids.

References

1. Network for Earthquake Engineering <http://it.nees.org/>
2. The Particle Physics Data Grid. <http://ppdg.net>
3. The Enterprise Grid Alliance. <http://www.gridalliance.org/en/index.asp>
4. The Global Grid Forum. <http://www.ggf.org>
5. NSF Teragrid Project, <http://www.teragrid.org/>
6. Foster, I., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Kishimoto, H., Maciel, F., Savva, A., Seibenlist, F., Subramaniam, R., Treadwell, J., Von Reich, J.: The Open Grid Service Architecture, V. 1.0, www.ggf.org/ggf_docs_final.htm, GFD.30. July 2004.
7. Global Grid Forum, Job Submission Description Language. Draft specification available at <http://forge.gridforum.org/projects/jsdl-wg>
8. GridSAM – Grid Job Submission and Monitoring Web Service, <http://www.lesc.ic.ac.uk/gridsam/>
9. Matsuoka, S., Shimojo, S., Aoyagi, M., Sekiguchi, S., Usami, H., Mura, K., Japanese Computational Grid Project: NAREGI. Proc. IEEE vol. 93, no. 510, 2005.
10. Open Middleware Infrastructure Institute, <http://www.omii.ac.uk>
11. Antonioletti, M., Atkinson, M., Baxter, R., Borley, A., Chue Hong, N., Collins, B., Hardman, N., Hume, A., Knox, A., Jackson, M., Krause, A., Laws, S., Magowan, J., Paton, N., Pearson, D., Sugden, T., Watson, P., and Westhead, M.: Design and implementation of Grid database services in OGSA-DAI, *Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2-4, Feb-Apr 2005, pp. 357-376.
12. The Globus Project: GT4. <http://www.globus.org/toolkit/>.
13. Jithesh, P., Kelly, N., Donachy, P., Harmer, T., Perrott, R., McCurley, M., Townsley, M., Johnston, J., McKee, S.: GeneGrid: Grid Based Solution for Bioinformatics Application Integration and Experiment Execution. *CBMS 2005*: 523-528
14. Belfast e-Science Center, <http://www.qub.ac.uk/escience/projects/gridcast>.
15. Laser Interferometer Gravitational Wave Observatory, <http://www.ligo.caltech.edu>
16. William Sanchez, Brian Gilman, Manav Kher, Steven Lagou, Peter Covitz, caGRID White Paper, https://cabig.nci.nih.gov/guidelines_documentation/caGRIDWhitepaper.pdf
17. Light Weight Middleware for Grid Computing, <http://glite.web.cern.ch/glite/>
18. Enabling Grids for E-Science, <http://public.eu-egge.org>
19. LHC Computing Grid, <http://lcg.web.cern.ch/lcg/>
20. Open Science Grid, <http://www.opensciencegrid.org/gt4>
21. The University of Virginia, The Global Bio Grid <http://www.cs.virginia.edu/~gbg>
22. Al Sairafi, S., Emmanouil, S., Ghanem, M., Giannadakis, N., Guo, Y., Kalaitzopoulos, D., Osmond, M., Rowe, A., Syed I., and Wendel P.: The Design of Discovery Net: Towards Open Grid Services for Knowledge Discovery. *International Journal of High Performance Computing Applications*. Vol 17 Issue 3. 2003.
23. Oinn, T., Greenwood, M., Addis, M., Ferris, J., Glover, K., Goble C., Hull, D., Marvin, D., Li, P., Lord, P., Pocock, M., Senger, M., Wipat, A. and Wroe, C.: Taverna: Lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows*, to appear 2005.
24. Ludaescher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao, Zhao, J.: Scientific Workflow Management and the Kepler System. *CC:P&E, Special Issue on Scientific Workflows*, to be published 2005.
25. Grethe JS, Baru C, Gupta A, James M, Ludaescher B, Martone ME, Papadopoulos PM, Peltier ST, Rajasekar A, Santini S, Zaslavsky IN, Ellisman MH. : Biomedical informatics research network: building a national collaboratory to hasten the derivation of new understanding and treatment of disease. *Stud Health Technol Inform*. 2005;112:100-9.

26. Churches, D., Gombas, G., Harrison, A., Maassen, J., Robinson, C., Shields, M., Taylor, I., Wang, I.: Programming Scientific and Distributed Workflow with Triana Services. *CC:P&E, Special Issue on Scientific Workflows*, to appear 2005.
27. The GridLab Project. <http://www.gridlab.org/>
28. Aktas, M., Aydin, G., Donnellan, A., Fox, G., Granat, R., Lyzenga, G., McLeod, D., Pallickara, S., Parker, J., Pierce, M., Rundle, J., and Sayar, A.: Implementing Geographical Information System Grid Services to Support Computational Geophysics in a Service-Oriented Environment. *NASA Earth-Sun System Technology Conf.*, June 2005
29. Bernholdt, D., Bharathi, S., Brown, D., Chanchio, K., Chen, M., Chervenak, A., Cinquini, L., Drach, B., Foster, I., Fox, P., Garcia, J., Kesselman, C., Middleton, M., VNeFedova, V., Pouchard, L., Shoshani, A., Sim, A., Strand, G., and Williams, D.: The Earth System Grid: Supporting the Next Generation of Climate Modeling Research. *Proc. IEEE*, vol. 93, no. 485, 2005
30. Plale, B., Gannon, D., Huang, Y., Kandaswamy, G., Lee Pallickara, S., Slominski, A.: Cooperating Services for Data-Driven Computational Experimentation, *Computing in Science & Engineering, IEEE Computing in Science and Engineering*, vol 7, no. 5, pp. 24-33, 2005.