

The GRIDCC project

The GRIDCC collaboration

Contacts for this paper are David Colling and Andrew Stephen McGough
E-mail contact for this paper: d.colling@imperial.ac.uk and asm@doc.ic.ac.uk

Abstract

The GRIDCC project is integrating into the Grid remote interaction with instruments, along with distributed control and real time interaction.

The GRIDCC middleware is being designed with use cases from a very diverse set of applications and so the GRIDCC architecture provides access to the instruments in as generic a way as possible. The middleware will be validated on a representative subset of these applications.

GRIDCC is also developing an adaptable user interface and a mechanism for performing complex workflows in order to increase both the usability and the usefulness of the system.

Wherever possible the GRIDCC middleware builds on top of other middleware stacks allowing the effort to be concentrated on the more novel elements of the project.

The GRIDCC project is a collaboration between 10 organizations in 4 different countries and is funded by the European Union.

1. Introduction

The GRIDCC project is a 3 year project funded by the European Union which started in September 2004. There are 10 project partners from Greece, Italy, Israel and the United Kingdom. The first complete release of the software will be during the second year of the project.

Recent developments in Grid technologies have concentrated on providing batch access to distributed computational and storage resources. GRIDCC is extending this to include access to and control of distributed instruments. In this paper we define an instrument to be any piece of equipment controlled through a computer interface such as telescopes particle accelerators or power stations. Instruments work in real-time and their successful operation often requires rapid interaction with conventional computing/storage resources and/or other instruments.

The control of instruments is often an interactive process. The real-time and interactive nature of instrument control provides a critical requirement for the definition of acceptable Quality of Service (QoS) constraints for interactions between the different Grid components.

GRIDCC is developing these definitions, where appropriate, building on work that is being carried out in the various standards organisations. In order to meet these defined QoS constraints it is often necessary to provide mechanisms for reserving the instruments, and subsequently for the compute and storage elements they will require.

GRIDCC will run a representative set of applications in order to validate the software produced by the project. These applications will control and monitor different types of instrument, acquire data from these instruments and then analyse this data on existing Grid test-beds. This will demonstrate integration of this project with existing Grid projects. The project will deploy applications with a significant social impact (control of a network of power grids generators, meteorology, geo-hazards monitoring, analysis of neuro-physiological data), engineering impact (distributed telecommunications measurements) and scientific impact (control and monitor of a high energy physics experiment, far remote operation of an accelerator facility).

To perform an experiment an application scientist will inevitably need to perform more than one task. They may wish to perform an experiment using a piece of equipment, analyse the results of this experiment and then potentially perform further experiments using the instrument in light of this analysis. This is but one simple example of what an application scientist may wish to perform. Thus there is a need to provide support for *workflows*, defined as the execution and management of sets of dependent and potentially heterogeneous tasks. The term "task" is used to indicate a general piece of work while a "job" refers specifically to a computational task. This must be powerful enough to deal with the myriad of possible

workflows that the application scientist may wish to describe, but also presented in a manner to the scientist which is intuitive and simple for them to work with. It should not be assumed that the application scientist is a competent software engineer nor should they be forced to try to be such.

Application scientists should be presented with a similar interface to the Grid and the instruments to that which they are already familiar. For this the GRIDCC project is developing a Virtual Control Room (VCR) environment which will present to the scientist a tailored interface both to the Grid and the instruments.

One of the barriers to placing instruments onto the Grid in the past has been the potential that inexperienced users may (unintentionally) damage the instrument by miss-use. There is thus a strong requirement to protect the instrument through appropriate checks and measures. These include local problem solvers which are capable of diagnosing situations which would cause damage to a local instrument. This could be monitoring the temperature of motors on a telescope and closing it down in the case of overheating. Likewise there is a need for global problem solvers. These are required to diagnose problems which cannot be diagnosed or remedied by a single instrument. This could be monitoring the number of power stations that are online and preventing more from coming online when this would cause too much current to pass through part of the power grid.

The goal is to build a widely distributed system that is able to remotely control and monitor complex instruments that range from a set of sensors used by geophysical stations monitoring the state of the earth to a network of small power generators supplying the European power grid. These new applications introduce requirements for real-time and highly interactive operation of Grid resources. To achieve this goal we have three main objectives:

1. To develop generic Grid middleware, based on existing building blocks (Grid Services) which will allow the remote control and monitoring of instruments such as distributed systems.
2. To incorporate the new middleware into a few significant applications that will allow the software to be validated both in terms of functionality and quality of service aspects: European Power Grid, Meteorology, Analysis of Neurophysiological data, RemoteOperation of an Accelerator Facility, High Energy Physics Experiment.

3. To disseminate widely the new software technology, the results of the evaluation on the test-beds and to encourage a wide range of enterprises to evaluate and adopt our Grid-oriented approach to real-time control and monitoring of remote instrumentation.

2. The architecture

In this section we present the overall architecture for the GRIDCC project. The architecture is composed of 8 service groups, see Figure 1, each of which may be decomposed into a number of individual services. These services are outlined below:

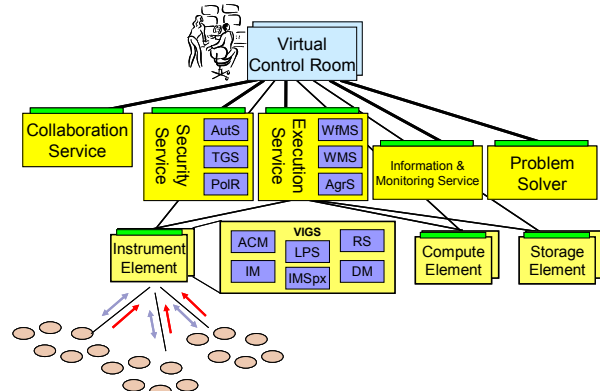


Figure 1 The GRIDCC Architecture

- **Virtual Control Room (VCR):** The VCR is the application scientists interface into the GRIDCC system. It should be tailored to the scientist's needs and present information in a manner which is understandable to the scientist. It should allow interactive real-time control of instruments along with the ability to work collaboratively with other scientists. Workflows may be constructed through the VCR, though they may be presented in a tailored manner for the application scientist.
- **Information Service (IS):** The IS provides a repository for information collected within the architecture. It may be interrogated to discover information.
- **Global Problem Solver (GPS):** The GPS uses information from the IS in order to diagnose potential situations which may lead to damage of the instrument or other elements. It is also responsible for taking whatever actions are appropriate to remedy these situations.
- **Security Services (SS):** These services are pervasive throughout the GRIDCC architecture and provide the mechanisms for performing Authentication and Authorisation of users.
- **Execution Service (ES):** The ES is responsible for taking a workflow describing what the

application scientist wishes to be performed and ensuring that it is executed within any QoS requirements which may exist.

- Instrument Element (IE): The IE is an abstraction of the instrument (or group of instruments) into a standard interface which can be used within the rest of the GRIDCC architecture.
- Compute Element (CE): The CE is an abstraction of a resource (or group of resources) which can perform computation into a standard interface which can be used within the rest of the GRIDCC architecture.
- Storage Element (SE): The SE is an abstraction of a resource (or group of resources) which can store data into a standard interface which can be used within the rest of the GRIDCC architecture.

Much effort has already happened within the e-Science community to develop Grid middlewares. These middlewares have in general been tailored towards non interactive computational job execution, such as Condor^[1], Globus toolkit (versions 1 through 4)^[2], Grid Engine^[3] and LSF^[4]. Rather than ignore the substantial effort made by these projects we seek here to extend them with the added functionality. Wherever possible we use existing code and community standards in developing GRIDCC.

The GRIDCC architecture implementation is thus being developed within gLite^[5] the Grid middleware developed within the European community from the EGEE^[6] project. Other potential candidates that were identified included the Globus toolkit version 4 and the OMII service infrastructure^[7]. Although all of these architectures are at approximately the same level of maturity the decision to go with gLite was based around the accessibility to the developers as there is an overlap of members of the GRIDCC and gLite development team.

The Grid community has, over the last few years, been moving rapidly between different underlying middleware architectures. These have included such technologies as JINI^[8], JXTA^[9], OGSII^[10] and many proprietary protocols. Stability is now being approached within the Grid community with most developers moving towards Web Service^[11] oriented middlewares. The increased use of firewalls between sites on the Internet has caused significant problems when using many of the original middlewares (such as JINI). As JINI was developed for use in a local environment without firewalls it uses a large range of ports when communicating between instances. This has caused major problems when deploying between administrative domains. This problem has been

resolved by the development of JINI2. However, it was felt that Web Services provided a better solution.

There are many workflow description languages available for use within the Grid. These include BPEL^[12], WS-Choreography^[13] and YAWL^[14]. Again with the intention of leveraging good quality implementations the choice was made to use BPEL as this appears to be the more favored option with a number of commercial and open source engines becoming available.

3. The subcomponents

In this section we describe the different subcomponents of the system. Interested readers should refer to the GRIDCC architecture document^[15] for further details.

3.1 The Virtual Control Room

The Virtual Control Room is the GridCC user interface. It allows users to build complex workflows which are then submitted to the Execution Service, it can connect to resources directly and be used to control instruments in real-time. The VCR also extends human interaction with Grid resources through its Multipurpose Collaborative Environment (MCE). The VCR is an instance of the MCE for a specific GRIDCC application. The VCR should be intuitively easy to use for the application scientist and present the required functionality in a manner which the scientist is familiar.

Web-based portals for the delivery of scientific information, experimental and computational services using a Grid infrastructure have already proven effective in meeting these requirements. We thus use Grid portal technology to implement the VCR. The MCE is being built from components defined with public interfaces, semantics and standard behaviors. The MCE provides a set of collaborative services such as chat, audio/video conferencing, instant messaging as appropriate to aid control of an application.

The simple schema presented in Figure 2 describes the internal architecture of the VCR.

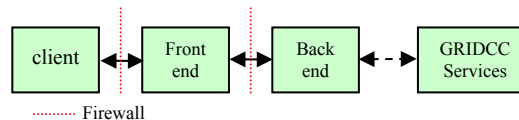


Figure 2 The VCR Architecture

The Client, which provides the Human Computer Interaction (HCI) component, is being implemented via a Web browser running on a client computer or PDA and will present portlets to the user. The user has to log on to the portal, a step often referred to as

authentication. This will activate the user's current "context". Users can configure the portal to their own personal "view" and this will be stored when they log off. The user can select the portlets they require for a particular job and to customise these.

The portlet JSR-168 specification^[16] handles the presentation end of information enabling re-use of portlets in different containers. In order for containers to present their contents as services the Web Services for Remote Portlets standard (WSRP)^[17] allows portals to retrieve content from other portals via their portlet containers and other data sources.

The front end provides access to the GRIDCC architecture through a Web Service to which the Client Web Browser will communicate. The Front end, often behind a firewall, will provide the initial security for scientists connecting with GRIDCC. In many Grid environments, though, instruments and other resources are held behind further firewalls. In this case the Back end can be used to proxy requests through the firewall. If no firewalls are required then the Back end may be ignored.

3.2 The Compute and Storage Elements

As stated above we are using the gLite architecture as the biases for the GRIDCC project. As such we make use of and extend the existing Compute Element (CE) and Storage Element (SE). However, the CE and SE provided currently through gLite do not support reservations or the making of agreements. Effort is currently underway within gLite to add functionality for reservations (and agreements) to the SE which will allow a specified amount of space to be reserved for a given amount of time. We are also extending the CE in order to provide similar functionality.

As the CE is implemented in gLite through one of many underlying Distributed Resource Management (DRM) systems, most of which currently don't support reservations we are developing techniques to provide "fake" reservation functionality for these CE's.

3.3 The Instrument Element

The Instrument Element (IE) is a concept unique to GRIDCC. The term IE describes a set of services that provide the needed interface and implementation that enables the remote control and monitoring of physical instruments. The term "instrument" is used in this context to define a piece of equipment that needs to be initialized, configured, operated (start, stop, standby, resume, application specific commands), monitored or reset.

The IE is viewed by users as a set of Web Services. Web Services are an excellent choice when there is a

need to provide a common language to the cross domain collaboration and to hide the internal implementation details of accessing specific instruments, which is the case in the IE, as described in Figure 3.

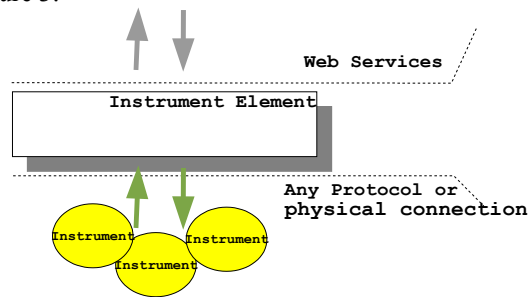


Figure 3 The role of the Instrument Element

The interface of the IE consists of three basic components:

- Virtual Instrument Grid Service;
- Resource Service;
- Information and Monitoring Service.

The above are generally public services that can be accessed by anyone, provided that they have the appropriate credentials which are checked by the *Access Control Manager*.

The IE is comprised of more components not directly accessible by external entities but indirectly through the edge services described above. They implement the operational functions and comprise the core engine of the IE. These components are the following:

- Instrument Manager(s) (IM);
- Local Problem Solver.

In Figure 4 we can see where these components are placed and how they interact together in a single installation. Specific details for the functionality of each of the components are described below.

The IM are the parts of the instrument element that perform the actual communication with the instruments. They act as protocol adapters that implement the instrument specific protocols for accessing its functions and reading its status. Since the instruments are heterogeneous in nature there is a need to support many IM's in the same container, one instance for each logical set of instruments.

Each IM is composed of four subcomponents:

- The *command gateway* accepts control requests from the Virtual Instrument Grid Service and forwards them to the instrument;
- The *command gateway* accepts control requests from the Virtual Instrument Grid Service and forwards them to the instrument;
- The *monitor manager* is responsible for collecting monitoring data (status, errors, etc.)

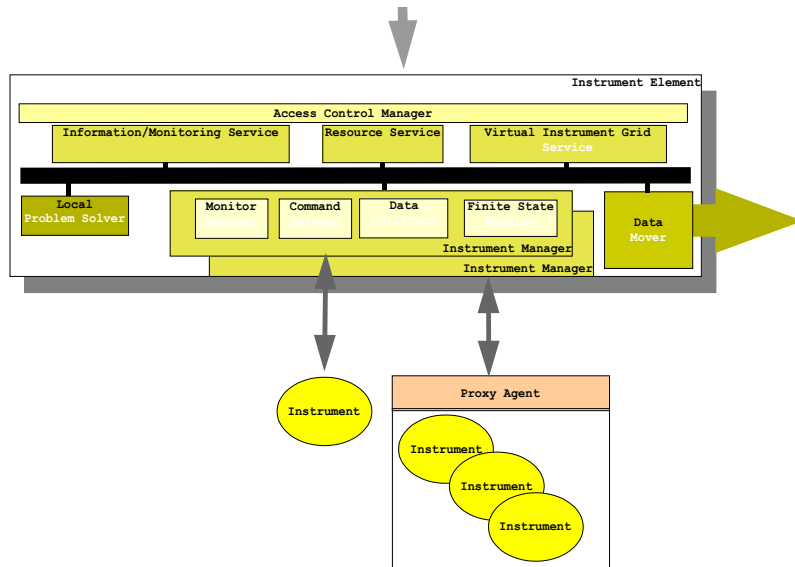


Figure 4 The Instrument Element Architecture

and provide it to the IS and/or the Local Problem Solver in a common format;

- The *data collector* component gathers the flow of data (if any) from the instruments. It then passes this data to the *Data Mover* which provides the interface with any external storage or processing elements;
- The *finite state machine* reflects the collective state of instruments controlled by the IM. It receives the status from the individual instruments and updates its state accordingly.

To transfer instruments from one IE to another, all that is required is to remove the IM instance from the original IE and instantiate a new one on the new IE. The other services (edge and core) react to this action accordingly.

In general, each IM is collaborating locally with the interfacing services to translate external requests for control or monitoring to requests that the instruments understand and can react to

The Local Problem Solver component has been introduced to add both problem solving and prediction capabilities inside the IE. It is positioned in the core of the IE and can collect alarms, errors, warnings, messages from all the IM's (more specifically from the monitor managers as described above), and to propose wherever possible appropriate recovery actions or raise alarms. Artificial Intelligence techniques like rule-based experts system can be used for this task.

The local problem solver interacts with the information and monitor service in order to make available its status and alarms, warnings or other

information to be processed by a larger scale system, like the Global Problem Solver to external entities.

The Access Control Manager component is responsible for checking user credentials and deciding whether a request should be processed by the IE (this process is referred as authentication and authorization). Once it is decided to proceed with the request it forwards it to one of the edge services.

The Virtual Instrument Grid Service (VIGS) is the interface of the IE that is responsible for parsing control requests that target one or multiple instruments and forwarding them to the corresponding IM, with the collaboration of the Resource Service. This service is very critical since it has real time requirements and should react according to the control requests that arrive from external entities.

The Resource Service (RS) provides a catalog of all the instruments managed by the IE. With this service one can obtain specific information about the instruments themselves, learn about topology settings and about configuration parameters for each instrument. This information can then be used to prepare requests for setting or controlling the instruments in a specific manner and send to the instrument through the VIGS.

Moreover, the RS deals with the logical partitioning of a set of instruments by a user with the objective of operating only a subgroup of them. This use case arises when an IE is handling a set of instruments that must be controlled by a single user or group to avoid conflicts with other requests. The instruments are then locked by the RS and accept requests only from that user / group. In turn when the

need for the partition no longer required, the instruments are unlocked and can accept requests from all users. More sophisticated allocation and advanced reservation schemes can be added using the Execution Service.

The Information and Monitoring Service uses a publish / subscribe system to disseminate monitoring data to the interested partners. More specifically each instrument, through the monitor manager publishes monitoring information to the Information and Monitoring Service arranged in topics. The local problem solver uses the same approach to publicise each state and corresponding data. The external entities, in turn, subscribe themselves as listeners to the topics and can receive synchronously or asynchronously the flow of monitoring data from the IE. The topics can be organized in subjects of general nature, specific to one or a set of instruments, in severity levels (info, warn, error) etc.

The Information and Monitoring Service (IMS) can act as a standalone module, where listeners have to subscribe to it locally, or be part of a larger scale distributed system where the publishers and subscribers can register themselves in their local IMS and interact with all the IMSes in the Grid transparently, as seen in Figure 5. The choice is dependent on the requirements of the application under consideration

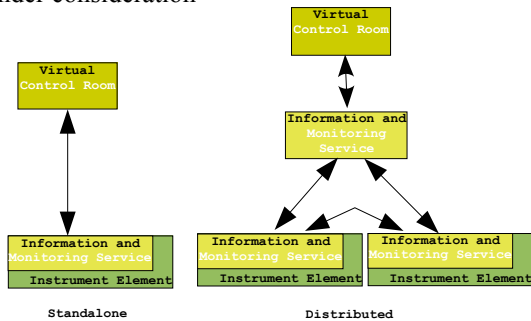


Figure 5 The Information and Monitoring service

3.4 Workflows and Execution Services

The Execution Service may be regarded as the heart of the GRIDCC architecture. It is responsible for taking the workflow from the VCR and ensuring that it is executed within the QoS constraints defined.

The term Execution Services refers to a group of services responsible for the execution and management of a workflow (Figure 6). The ES is also responsible for the reservation of resources, such as computing, storage and network bandwidth. The ES may also be responsible for the reservation of instruments, however, it may be the case that the scientist has made a reservation for an instrument out-with the GRIDCC architecture. The co-allocation

of multiple resource instances is a specific workflow example, where the individual units consist of atomic allocation operations. The ES individual components are:

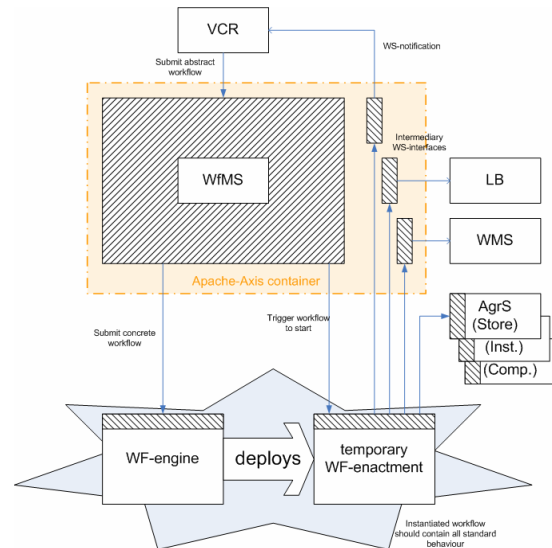


Figure 6 The ES architecture

- The Workflow Management System (WfMS)
The WfMS comprises of the Planner and the Observer. The planner decomposes the workflow into a number of independent activities that may be submitted to the Workflow Management System (WMS)^[18]. It may determine the resource on which some of the tasks operate and if appropriate make reservations on these resources. It modifies the workflow document to match this and submits it to a workflow engine. The workflow may now be executed. The Observer monitors the running application and determines if the workflow needs to be modified. This may be due to the original workflow not performing within the stated QoS requirements or more appropriate resources becoming available within the Grid. If deemed appropriate the Observer may trigger the Planner to modify the workflow.
- The Workload Management System (WMS)
The WMS provides a mechanism for submitting tasks to all of the CE's. The WMS may be instructed to use a given resource, if a reservation has been made on that resource. Alternatively the WMS may be able to select the most appropriate resource based on criteria defined by the user and the software developer.
- The Agreement Service (AgrS)
The AgrS^[19] is responsible for making reservations with computation, storage and instruments this is achieved through the process of reservations.

3.5 The information systems

The purpose of the GridCC Information Service (not to be confused with the Information and Monitoring Service of the Instrument Element) is to store and provide information about components, such as services, resources and instrument elements, of a Grid system. This information can be static, slowly changing or dynamic and can serve a variety of different purposes.

Static and quasi-static information supports the discovery of resources and services performed by the Execution Services. The principle aim is to aid the discovery of static, or at least slowly changing resources, such as the properties, configuration and availability of the IE and other Grid resources needed by a particular application. In practice many systems currently exist which provide this or similar functionality (e.g. products supporting the UDDI protocol^[20], RGMA from EDG/EGEE^[21], the Globus MDS^[22] etc.).

3.6 The problem solvers

Automated problem solving in a Grid environment is a feature of the GRIDCC project. Two levels of problem solver can exist within GRIDCC: the first local to a given IE, the Local Problem Solver (LPS). The second global to the entire system, the Global Problem Solver (GPS). The LPS solve problems related to the function of a given instrument and are embedded within that IE, whereas the GPS solve problems related to the whole system.

3.7 Security

The applications for which GRIDCC is designed mean that security –Authentication, Authorization and Accounting— are especially important in this project. The damage that could be caused by a security incident resulting in misuse of the components of the National Power Grid or the multi-billion Swiss Franc CMS detector is vast. Yet the security system is constrained to both to be light, so that the GRIDCC QoS requirements can be met, and to inter-operate with security systems of other Grid systems. In order to satisfy these two, almost mutually contradictory, requirements GRIDCC has decided to use a split security system. When interacting with the components of other Grid projects the GSI security will be used and the user identified by their X.509 proxy certificate, however when interacting with the IE the user will be identified by a Kerberos ticket. The level of security

provided by both systems is equivalent. Though Kerberos has a lower overhead.

4. The applications

A variety of applications have been considered within the GRIDCC project and their use cases have had a strong bearing on the design and implementation of the software. These range from meteorology, geo-hazards, neurophysiology, and power grids to accelerator control and high energy physics experiment control. While it is hoped that many of these applications will use the GRIDCC software the project itself has chosen 3 applications to take through to completion. These are:

4.1 Control and monitor of a High-Energy Physics (HEP) experiment

This application involves three basic elements: a very high number of instruments, a large number of concurrent “users” (sometimes in the form of automatic monitoring tasks) and a high-rate of incoming data. These characteristics result in stress-tests for the monitoring of data streams, the network(s) involved, the QoS capabilities (prioritizing the data stream against normal traffic), as well as the capability to respond to different types of error conditions arising from normal operation. This latter element is very important, for the analysis of what should result in a system stop and what should not, has to take place under the stringent real-time requirements of the experiment’s data-taking process.

4.2 Far Remote Operation of an Accelerator

This application also involves large numbers of instruments (mainly sensors). It is different from HEP Control and Monitor in (a) the rate of incoming data: it is far smaller than the data in HEP but of potentially far higher importance which introduces new requirements on the reaction times to alarms (b) the level of the human-machine interface, which again is higher than that in HEP.

4.3 Power Grids

In electrical utility networks (or power grids), the introduction of very large numbers of ‘embedded’ power generators often using renewable energy sources, creates a severe challenge for utility companies. Existing computer software technology

for monitoring and control is not scalable and cannot provide a solution for the many thousands of generators that are anticipated. GRIDCC technology would allow the generators to participate in a Virtual Organization, and consequently to be monitored and scheduled in a cost-effective manner.

A specific testbed application will be built and demonstrated within the GRIDCC project by means of computer simulation and emulation. Existing software at Brunel University will allow the real-time simulation of a representative power network and the associated generators. New software will be created to interface the generator simulations to the GRIDCC environment. Distributed generator scheduling algorithms will be modified to utilize GRIDCC technology. The test bed will demonstrate the performance of the emulated system under various conditions, ranging from light power system loading (where energy economics is most important) to power system emergency conditions (where overloaded power circuits necessitate co-ordinated generator control to avoid power black-outs).

5. Conclusions

The GRIDCC project is integrating instrumentation into traditional computational/storage Grids. The project started in September 2004 and the first full release of the GRIDCC software will be during the second year of the project. The software produced will be validated on at least three of representative applications.

6. References

- [1] The Condor Project, <http://www.cs.wisc.edu/condor/>
- [2] The Globus Alliance, <http://www.globus.org/>
- [3] The gridengine project, <http://gridengine.sunsource.net/>
- [4] Platform LSF, <http://www.platform.com/products/LSF>
- [5] EGEE Middleware Architecture, EU Deliverable DJRA1.4, Jul 2005 (<https://edms.cern.ch/document/594698/>)
- [6] Enabling Grids for E-science, <http://public.eu-egee.org/>
- [7] The Open Middleware Infrastructure Institute, <http://www.omii.ac.uk>
- [8] S. Liang, The Java Native Interface, Programmer's Guide and Specification, Addison-Wesley, 1999
- [9] Project JXTA, <http://www.jxta.org>
- [10] Open Grid Services Infrastructure (OGSI) v1.0, <http://forge.gridforum.org/projects/ggf-editor/document/draft-ogsi-service1/en/1>
- [11] Web Service, W3C, <http://www.w3c.org/TR/ws-arch/>
- [12] BPEL4WS. <http://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>
- [13] WS Choreography Model Overview, <http://www.w3.org/TR/2004/WD-ws-chor-model-20040324/>
- [14] W.M.P. van der Aalst, L. Aldred, M. Dumas, and A.H.M. ter Hofstede. Design and Implementation of the YAWL system. Proceedings of The 16th International Conference on Advanced Information Systems Engineering (CAiSE 04), Riga, Latvia, June 2004. Springer Verlag.
- [15] The GRIDCC Architecture Document, <http://www.gridcc.org/getfile.php?id=1382>
- [16] JSR 168: Portlet Specification, <http://www.jcp.org/En/jsr/detail?id=168>
- [17] OASIS Web Services for Remote Portlets (WSRP) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
- [18] P. Andreetto, S. Andreatto, G. Avellino, S. Beco, S. Borgia, V. Ciaschini, F. Giacomini, A. Giannelle, A. Guarise, A. Krenek, D. Kuril, A. Maraschini, M. Marchi, L. Matyska, M. Mezzadri, S. Monforte, M. Mordacchini, M. Mulac, F. Pacini, M. Pappalardo, G. Patania, J. Pospisil, F. Prelz, D. Rebatto, E. Ronchieri, M. Ruda, Z. Salvat, J. Sitera, J. Skrabal, M. Sgaravatto, A. Terracina, M. Vocu, and L. Zangrando. Practical Approaches to Grid Workload and Resource Management in the EGEE Project. In Proceedings of the International Conference on Computing in High Energy Physics (CHEP2004), Interlaken, Switzerland. 27 September-1 October 2004.
- [19] T.Ferrari, E.Ronchieri, gLite Advance Reservation Architecture <http://edms.cern.ch/document/508055>
- [20] OASIS UDDI, <http://www.uddi.org/>
- [21] Rob Byrom, Brian Coghlan, Andy Cooke, Roney Cordenonsi, Linda Cornwall, Martin Craig, Abdeslem Djaoui, Steve Fisher, Alasdair Gray, Steve Hicks, Stuart Kenny, Jason Leake, Oliver Lyttleton, James Magowan, Robin Middleton, Werner Nutt, David O'Callaghan, Norbert Podhorszki, Paul Taylor, John Walk, Antony Wilson, Production Services for Information and Monitoring in the Grid, in proceedings of the UK All Hands meeting 2004.
- [22] Globus: Monitoring and Discovery System, <http://www.globus.org/mds/>