

GeneGrid: A Practical Workflow Implementation for a Grid Based Virtual Bioinformatics Laboratory

David R. Simpson
Noel Kelly, P.V. Jithesh
Paul Donachy, Terrence J Harmer
Ron H Perrott

Belfast e-Science Centre
www.qub.ac.uk/escience

Jim Johnston
Paul Kerr
Mark McCurley

Fusion Antibodies Ltd
www.fusionantibodies.com

Shane McKee

Amtec Medical Limited
www.amtec-medical.com

Abstract

Bioinformatics needs solutions to capture and process genomic data, to assist diagnostic and therapeutic strategies. Industrial UK e-Science project, GeneGrid, harnesses GT3 Grid technology, to create a framework enabling the generation and analysis of genetic information from distributed sources. This paper presents the **GeneGrid Workflow and Process Management (GWPM)** component, the operational driver for the system, how it exploits a grid based architecture to meet the real life requirements, and some initial user results and experiences.

1 Introduction

The GeneGrid project is a collaborative industrial UK e-Science project [1]. Its aim is to exploit Grid technology, existing micro array & sequencing technology and the large volume of data generated through screening services to develop specialist datasets relevant to a particular disease or condition being studied. The advantage is that study of all the genes that are related to a disease or condition can be done *in silico*. This is achieved by the establishment of a grid based framework that integrates the generation and analysis of specific genetic information from various distributed international sources, public domain data sets and other unique data generated by e-Science projects.

1.1 Background

Genome expression monitoring and mapping is having great impact on clinical diagnosis and therapy, and bringing new power to clinical medicine. As the field progresses new probes for cancer, infectious diseases and inherited diseases are being identified, deeper understanding of how genetic damage occurs and how genes alter response to drug therapies. The Human Genome Projects achievement in mapping the human

genome is one of the largest and most publicised scientific endeavours in this area. The DNA sequencing data produced still contains much untapped data that needs to be converted into meaningful information. At present within bioinformatics there is a need to develop better solutions to capture, analyse, manage, mine and disseminate these vast amounts of genomic data, to assist with the development of diagnostic and therapeutic strategies.

1.2 Commercial Partners

The key stakeholders in this project are Fusion Antibodies Ltd and Amtec Medical Ltd, who have a common goal of advancing bioinformatics genomic research. Fusions core objective is to use GeneGrid to accelerate its target identification for protein and antibody discovery, while Amtec will focus on the development of potential products from the molecular mining of the inherent wealth locked within the data GeneGrid will produce. At present the individual companies do not have any dedicated in-house bioinformatics specialisation or HPC capability. They generate large amounts of data but relating this to the global environment is problematic. The low speed of data transfer between parties, lack of high performance computing power and lack of encompassing security mechanisms across the

disparate administrative boundaries and organizations is a blockage to rapid advancement of this important area of Science and research.

2 GeneGrid Overview

GeneGrid aims to provide a practical, easy to use and secure system, which harnesses and shares the power of distributed HPC resources, enabling more comprehensive and efficient interrogation of the global data pool available to biotechnologists. This addresses the immediate needs of both partner companies, who can now utilise bioinformatics expertise and HPC access to remove obstacles to their genomic and protein research. Additionally the project aims to implement an underpinning scaleable and extendable architectural base, so that the addition of extra functionality, resources, or user capacity can be readily achieved.

2.1 System Architecture

The Grid based architecture presented here is based on the Open Grid Services Architecture (OGSA) model [2] derived from the Open Grid Services Infrastructure specification [3] defined by the GGF [4] OGSi Working Group. OGSA represents an evolution towards a Grid architecture based on Web services concepts and technologies. It describes and defines a service-oriented architecture (SOA) composed of a set of interfaces and their corresponding behaviors to facilitate distributed resource sharing and access in heterogeneous dynamic environments. Figure 1 presents a block diagram of the overall GeneGrid architecture and introduces the main components of the system.

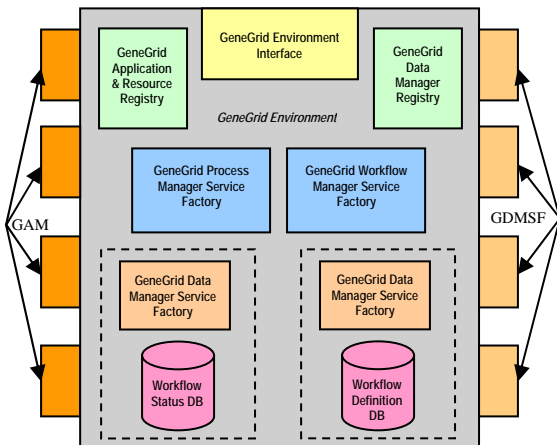


Figure 1 – Component Block Diagram

The project has been split into three core components, Data Management (GDM), Workflow & Process Management (GWPM), and Application Management (GAM), with the addition of a GeneGrid Environment Interface (GENI) facilitating user access. As is indicated by the diagram one of the central features of the project has been the development of a Grid based workflow infrastructure to enable the definition and automatic management of processes, and procedures, involved in information transfer between human and machine, or, from machine to machine

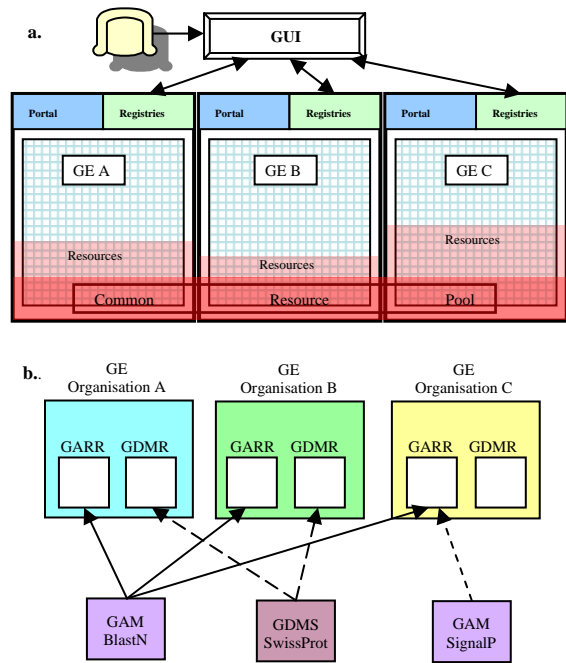


Figure 2 – Environments and Shared Resources

The ‘GeneGrid Environment’ (GE) is the collective name used for identifying the core elements of the system to a large extent this is what constitutes the ‘virtual laboratory’. Data (GDM) and application (GAM) resources are ‘integrated’ into the ‘lab’ as required and available. Any user accessing the GE can, if allowed, access all services registered with the GARR and GDMR. These services facilitate the submission, tracking and execution of workflows. An organisation would be encouraged to establish and maintain its own GE, consequently managing access and submission permission for their services. Additionally it enables the controlled sharing of applications and data set resources with other organisations on a per-resource basis. Figure 2

presents two diagrams which illustrate the resource sharing of the 'environment' concept.

2.2 GeneGrid Components

It has already been stated that GeneGrid is composed of three core components, Application Management (GAM), Workflow & Process Management (GWPM), and Data Management (GDM) and before looking in more detail at the GWPM component it's important to understand a little more about the other core components and two initially smaller components whose importance will increase as the development matures, namely Resource Management and the GeneGrid Environment Interface (GENI).

The GeneGrid Data Manager (GDM) is responsible for the access, integration and storage of data within GeneGrid; it consists of 3 types of co-operating services with elements existing both inside and outside the GE. The GDM utilises and extends OGSA-DAI [5] infrastructure to interface to a number of heterogeneous databases, integrating them into GeneGrid. These databases fall broadly into 2 categories: Biological, and GeneGrid Specific. The GDM Registry (GDMR) and the GDM Service Factory (GDMSF) are persistent services tied to the creation and destruction of their host container; on the other hand the GDM Service (GDMS) is a transient service. The always present GDMR provides a directory of all GDMSFs. A GDMSF spawns a GDMS when a client requests access to a specific database, the created GDMS manages the interaction and is terminated by the client, or when its lifetime expires. The GDM is covered in detail in [6].

The integration of bioinformatics applications into GeneGrid is a requirement addressed by the GeneGrid Application Manager (GAM). Applications may be local or remote and conceptually inside or outside the GE. These are viewed as GeneGrid resources even though each does not necessarily constitute being an individual physical resource. The applications that are initially available in GeneGrid are BLAST [7] (several variants), Transmembrane prediction and Signal peptide prediction. The GAM provides an abstracted interface to these whilst also affording the system a degree of immunity to change. The impact on the rest of the system of upgrading to enhanced and/or new applications is minimised by the GAM interface. The GAM component also has ownership of the GeneGrid Application and Resource Registry

(GARR). Similar to the GDMR, the GARR is a persistent service which is always present in any GE, and whose location is provided to all services on their start up. It advertises the location and capabilities of all GAM services and other services (e.g. GWPM services) not handled by the GDMR. GAM is covered in further detail in [7].

Resource Management is the key to the smooth operation and expandability of a GE. Initially GeneGrid will share this functionality between the GAM and GWPM core components. However as the development progresses and a GE expands in terms of both capability and complexity it is envisaged that it may become a core component in its own right. Performance information is gathered on all resources within a GE, this is achieved in one of two ways; (a.) GAM services report back information to the GARR on their application and host environment, or (b.) a GeneGrid performance 'agent' which is deployed on all host platforms integrated into a GE, is 'switched on' to periodically report information back to the GARR. In first releases of GeneGrid a simple GeneGrid Node Monitoring Service (GNMS) will co-ordinate this data collection and expose a basic 'traffic light' mechanism on resource usage. Essentially a GARR interface service which indicates, on a per resource basis whether a resource in the GE is *usable - usable:active - not usable*. This will be the source of information for basic resource mapping and scheduling functionality implemented by the GeneGrid Process Manager (GPM).

Users access GeneGrid via the GeneGrid Environment Interface (GENI). This is a web based portal built on top of a GridSphere [8] framework and executable on any suitably enabled web browser. The GENI performs a number of roles; it provides an easy to use interface to GeneGrid for end users; it facilitates the submission and querying of workflows; it enables the accessing of results; and provides the capability to perform administration and 'housekeeping' task on the GE. Transcending the roles listed above the GENI implements the first stage of the GeneGrid Security Policy based on the Globus Grid Security Infrastructure [9] X.509 certificates to authenticate all users before granting access. Subsequently user credentials are forwarded on as required to other GeneGrid services where further stages of access restriction and control are implemented. As a result of having the GENI end users should have no

knowledge of the underlying services and architecture of GeneGrid, another helpful feature from a security perspective.

3 GeneGrid Workflow & Process Management (GWPM)

Workflow is defined by the WfMC [10] sponsored e-workflow site [11] as: *The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant (human or machine) to another for action, according to a set of procedural rules..* Specifications for the design and implementation of workflows in grid and web service technology are still emerging, efforts such as the generation of WS-BPEL [12] from WSFL [13] and XLANG [14] have helped with standards development but work is ongoing. Project constraints dictated that a simple, light-weight, but extensible and scalable workflow architecture was required for GeneGrid. With no definitive direction regarding standards options involving collaboration with other projects were explored. However these presented us with drawbacks in terms of resources and the overhead of having to port unwanted

functionality. Additionally many of these projects were still in development with incompatible timescales and with no obligations to us or our stakeholders. Consequently a practical match was not readily available so the GeneGrid Workflow and Process Manager schema was conceived. The creation of the GWPM architecture has not intentionally followed the path of any other workflow development however it has been influenced by work done by NMI Grid Workflow Project [15], the myGrid project [16], and the GridCast project [17]. It uses the Java language to implement OGIS [3] based Globus Toolkit [18] (GT3) services which operate on data captured in XML files generated using a predefined XML workflow schema.

3.1 GeneGrid Workflow Context

The GWPM sub-system is expressed in the form of modular components which expose functionality through their grid services. These have a clearly defined outward appearance (interface) and behaviour, with control and data flow effected by the interaction between these services. The GT3 implementation of open grid

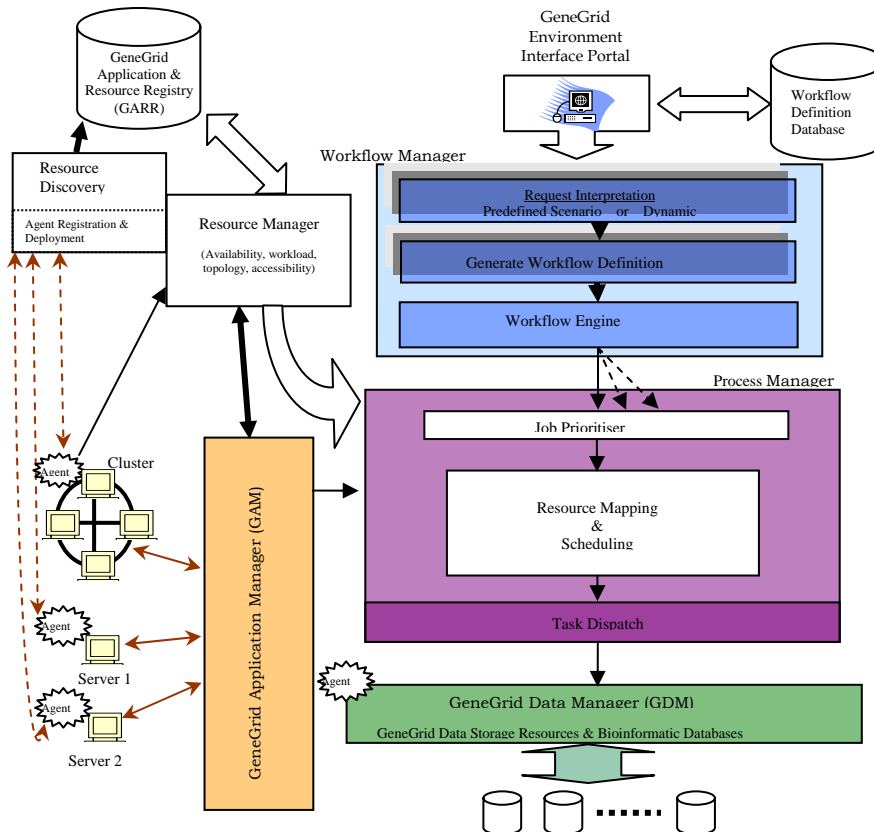


Figure 3 – GeneGrid GWPM Context

services infra-structure (OGSI) enables the deployment of these services on different computing resources taking advantage of opportunities for distributed and parallel processing. Figure 3 shows an abstracted schematic of the GeneGrid system it provides a context for the illustration of the GWPM sub-components and their interfaces to other parts of GeneGrid. (For clarity some control and data flows have been omitted.) A GeneGrid ‘workflow’ is comprised of tasks which may be autonomous or have dependency linkages to other tasks. Tasks are grouped and entered by the user via the GENI, based on an XML schema to define them in an XML ‘Workflow’ file. This is the operational driver for the GeneGrid system, and contains all the necessary information and commands to enable the workflow to be executed by the system.

3.2 Workflow Schema

The workflow schema is an XML schema [19] file that defines the structure, content and semantics of the workflows. Within GeneGrid we refer to these as *Workflow Definitions* and there are 2 types: a) *Master Workflow Definition* (MWD) – only one of these is ever active at any time in a GE. It provides a complete description of all the tasks and instructions its GE can execute; b) *User Specific Workflow Definition* (SWD) – these are formed from a sub-set of tasks and instructions in the MWD and allow users to save their most frequently used workflows. These definitions are stored in the Workflow Definition Database and accessed by the GENI. Where & how these are accessed is handled by the GDM [6]. The following is an example of the top level structure of an XML workflow document for a simple use case scenario

GeneGrid XML Workflow Document Example

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow
  workflowId="20040316151613422" submitDate="2004-03-15"
  submitTime="16:30:55:000" userName="userName1" >

  <wfDescription>
    A sample XML workflow file.
  </wfDescription>

  <aTask taskName="blast1" taskId="1" dependsOn="nothing"
  taskStatus="created" >
    <blastN>
      <!-- ***** blast specific commands -->
    </blastN>
  </aTask>
  <aTask taskName="StoreRes1" taskId="2" dependsOn="blast1"
  taskStatus="created" >
    <StoreRes>
```

```

      <!-- ***** store results specific commands -->
    </StoreRes>
  </aTask>
  <aTask taskName="FormatRes1" taskId="3" dependsOn="blast1"
  taskStatus="created" >
    <FormatRes>
      <!-- ***** format results specific commands -->
    </FormatRes>
  </aTask>
  <aTask taskName="Translation1" taskId="4"
  dependsOn="FormatRes1" taskStatus="created" >
    < Translation >
      <!-- ***** Translation specific commands -->
    </ Translation >
  </aTask>
  <aTask taskName="Transmembrane" taskId="5" dependsOn="
  Translation1" taskStatus="created" >
    < Transmembrane >
      <!-- ***** transmembrane specific commands -->
    </ Transmembrane >
  </aTask>
  <aTask taskName="StoreRes1" taskId="6" dependsOn="
  Transmembrane " taskStatus="created" >
    <StoreRes>
      <!-- ***** store results specific commands -->
    </StoreRes>
  </aTask>
  <aTask taskName="Signal" taskId="7" dependsOn=" Translation1"
  taskStatus="created" >
    < Signal >
      <!-- ***** signal prediction specific commands -->
    </ Signal >
  </aTask>
  <aTask taskName="StoreRes1" taskId="8" dependsOn=" Signal "
  taskStatus="created" >
    <StoreRes>
      <!-- ***** store results specific commands -->
    </StoreRes>
  </aTask>
</workflow>
```

Figure 4 is a diagrammatic representation of the workflow described above.

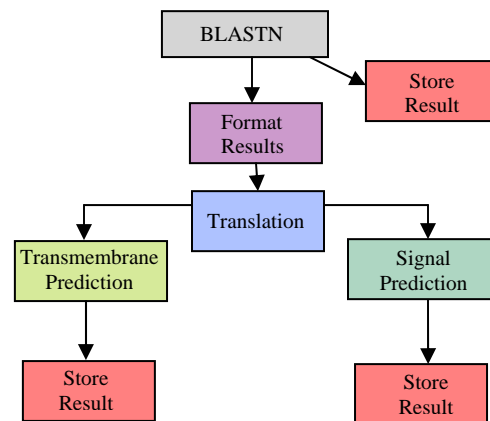


Figure 4 – Simple ‘Use Case’ Workflow

3.3 GeneGrid Workflow Manager (GWM)

The workflow manager is the entry service of the GWPM architecture. It hides the underlying processing from the portal (GENI) and is responsible for the following functionality:

- Input of pre-defined or interactively generated instruction sets (workflows).
- Ensure the validation and/or secure execution of any given workflow.
- Capacity to handle the execution of more than one workflow at any time up to a TBD maximum value.
- Partition a workflow into its constituent instructions ('tasks') and pass these on for execution by other parts of the system.
- Monitor and track the progress of any executing workflow.
- Enable access to both intermediate results (from tasks) and final completed workflow results.

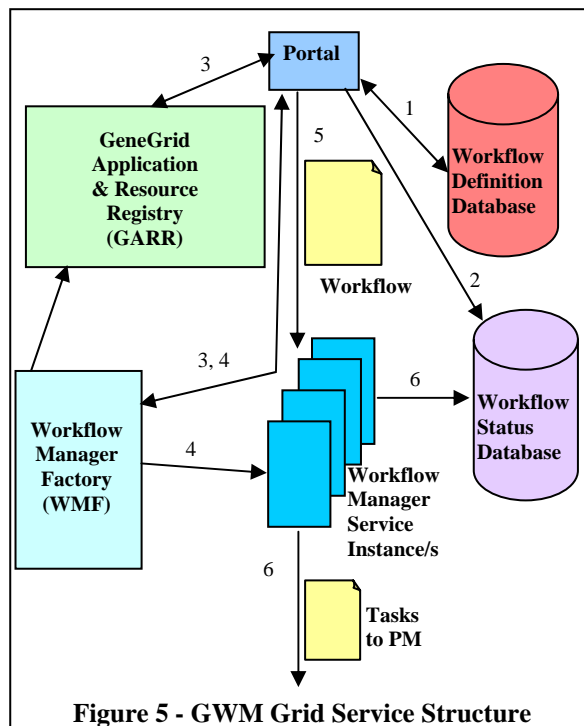


Figure 5 - GWM Grid Service Structure

Figure 5 and the associated bullet points describe this functionality.

1. A user accesses GeneGrid using the 'Portal', which allows workflow templates or stored workflows to be selected from the Workflow Definition Database.
2. The user submits the new or selected workflow for execution. The Portal creates a record of the workflow in the Workflow Status Database.
3. The Portal gets the location of the Workflow Manager Factory from the GARR (the WMF having already registered its existence) and

contacts the WMF to request the execution of a workflow.

4. The WMF will then return the location of a Workflow Manager Service (WMS) instance back to the Portal.
5. The Portal then passes the XML encoded workflow to the identified WMS instance via one of its portType methods.
6. This WMS handles the progress of the workflow (or subsequently workflows) passed to it. Splits them into defined tasks and passes these to the Process Manager, whilst updating the Workflow Status Database with progress and results information.

3.4 GeneGrid Process Manager (GPM)

The process manager is the 'engine room' of the GeneGrid system. It resides in a world governed and driven by tasks and does not have any interest in the bigger workflow picture. It implements some internal rules, gathering information from tasks and other system elements to determine how they should be applied. The following list summarises its main functions:

- Accept tasks from one or more GWMs, and ensure they are efficiently executed.
- Extract instructions and data from the XML encoded tasks received.
- Implement a set of task execution rules, governing priority schemes and resource mapping.
- Handle the scheduling of both tasks and resources.
- Maintain up to date links with the resource management, GAM and GDM components to have an accurate picture of resource availability.
- Dispatch, in accordance with pre-defined rules, tasks to the appropriate (most efficient) resources for execution.
- Receive task success, failure, completion and results information, report this back to the GWM that sent the task and 'owns' its parent workflow.

The Grid Service architecture depiction of how the GPM functionality structure will be implemented is shown by figure 6 Elements are again included in this diagram to provide linkage to the previously discussed figures in this document. The following is an explanation of

actions for a GAM service (applies similarly for a GDM service):

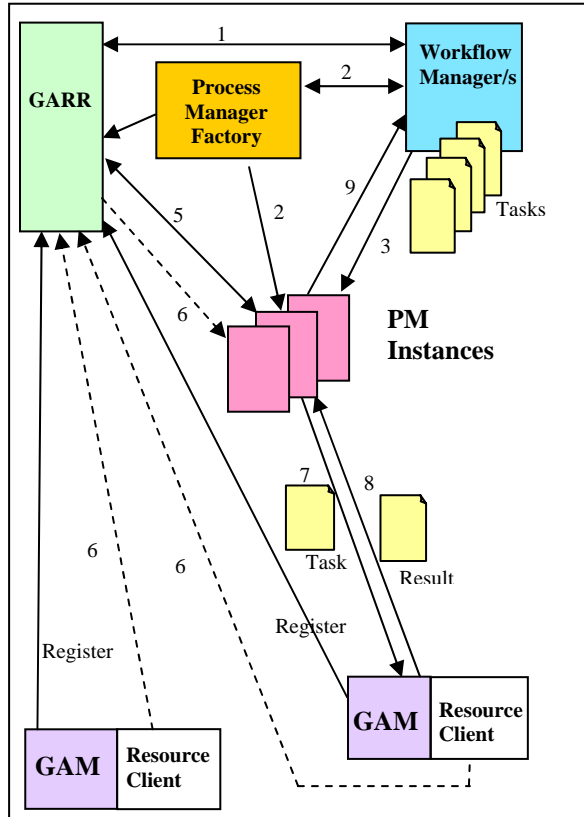


Figure 6 – PM Grid Service Structure

1. A GWM instance executing a workflow contacts the GARR to get the location of the Process Manager Factory.
2. The GWM then requests the GPM factory to create a Process Manager instance which it subsequently contacts.
3. Tasks are then passed to the GPM instance by the GWM, these are XML files generated from specific task sections of the larger 'workflow' XML file.
4. The GPM extracts the service request information from the task XML description.
5. The GARR is then interrogated for registered services that can satisfy the task request. These are subject to the application of execution rules embedded in the GPM code.
6. Service resource information (e.g. location, load average of host machine) is held by the GARR. A GPM can access this and use to map a task to the most appropriate service, and if necessary schedule its execution.
7. The GPM dispatches the task to the selected service whose location was provided by the GARR.

8. When the service has completed execution of the task a results file (XML) is returned to the GPM that dispatched it. This contains success or failure indication and information on any results produced.
9. Using identification information on the task held in the XML file, the GPM identifies the GWM from which the task came, and forwards the XML results file unmodified to the appropriate GWM.

4 Conclusion: The Road Ahead

The GeneGrid project is approximately half way through its development cycle. Prototype GWPM services have been deployed on the GeneGrid test bed since March 2004. This initial test bed is comprised of a number of heterogeneous resource platforms. Services were deployed to Globus or Apache Tomcat [20] containers running under, Linux on standard desktop PCs or, Solaris on a Sun SMP host. These tests with 'release 1' of GeneGrid have produced encouraging results [21]. Work on the majority of the GWPM functionality in this paper is ongoing. Features like task scheduling and prioritisation will be further addressed as the development matures. Practical 'use case' scenarios and very positive feedback have resulted from stakeholder reviews. Planned migration to GT4 will hopefully add increased functionality, ease of integration and improved core stability. Scalability has been tested by the deployment of services integrating a number of disparate (local and international) resources including a 32 node cluster at BeSC. Overlaid on top of these are a number of administrative organisational domains. Further opportunities exist to develop brokerage and optimisation features for the GWPM component in line with the already accepted need for enhanced resource management capability. Other areas for exploration are the dynamic graphical construction and submission of workflows (e.g. myGrids Taverna workbench [22]), and the development of a GeneGrid ontology or support for some of the other more comprehensive and complex workflow description languages. International links have already been established with the EOL project [23] around synergies with their iGAP development, and it is anticipated that this will lead to further future collaboration.

5 References

- [1] Donachy P., Harmer T.J., Perrott R.H. *et al.* (2003) Grid Based Virtual Bioinformatics Laboratory. *Proceedings of the UK eScience All Hands Meeting 2003*, 111-116.
- [2] The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. I. Foster, C. Kesselman, J. Nick, S. Tuecke, *Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.*
- [3] Open Grid Services Infrastructure (OGSI) Version 1.0. S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt, D. Snelling; *Global Grid Forum Draft Recommendation, 6/27/2003.*
- [4] Global Grid Forum <http://www.ggf.org/>
- [5] OGSA Data Access & Integration project <http://www.ogsadai.org.uk/>
- [6] Kelly N., Jithesh P.V., Simpson D., Donachy P. *et al.* (2004) Bioinformatics Data and the Grid: The GeneGrid Data Manager. *Proceedings of the UK eScience All Hands Meeting 2004.*
- [7] Jithesh P.V., Kelly N., Simpson D., Donachy P. *et al.* (2004) Bioinformatics Application Integration and Management in GeneGrid: Experiments and Experiences. *Proceedings of the UK eScience All Hands Meeting 2004.*
- [8] GridSphere <http://www.gridisphere.org>
- [9] Globus Security <http://www-unix.globus.org/toolkit/docs/3.2/security.html>
- [10] WfMC <http://www.wfmc.org/>
- [11] <http://www.e-workflow.org/>
- [12] <http://www.ebpml.org/bpel4ws.htm>
- [13] Prof. Dr Frank Laymann IBM (2001) WSFL 1.0 <http://www-306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>
- [14] XLANG http://www.gotdotnet.com/team/xml_wss_pecs/xlang-c/default.htm
- [15] Stephen Mock *et al.* NMI Workflow Architecture SDSC <http://kbi.sdsc.edu/events/SWF-11-03/nmi-workflow-arch.pdf>
- [16] myGrid <http://www.mygrid.org.uk/>
- [17] GridCast <http://www.qub.ac.uk/escience/projects/gridcast/>
- [18] Globus Toolkit <http://www-unix.globus.org/toolkit/>
- [19] XML Schema <http://www.w3.org/XML/Schema>
- [20] Apache Tomcat <http://jakarta.apache.org/tomcat/>
- [21] <http://www.qub.ac.uk/escience/projects/genegrid/testbed>
- [22] Taverna <http://taverna.sourceforge.net/>
- [23] EOL <http://eol.sdsc.edu/>