Proceedings of the UK e-Science All Hands Meeting 2005, ©EPSRC Sept 2005, ISBN 1-904425-53-4

GeodiseLab: Making the Grid Usable

Graeme Pound, Jasmin Wason, Marc Molinari, Zhuoan Jiao & Simon Cox

School of Engineering Sciences, University of Southampton, UK
{gep, j.l.wason, m.molinari, z.jiao, sjc}@soton.ac.uk

Abstract

The GeodiseLab project is supported by the OMII managed programme to extend and improve the Geodise toolboxes which provide Grid client functionality to Problem Solving Environments (PSEs) used by engineers and scientists. The three key areas of development in the Geodise toolboxes have been: extending the functionality of the toolboxes, supporting additional environments, and hardening the toolboxes for public release. The Geodise computational toolboxes now support additional Grid technologies, including Condor and the OMII_1 platform. Recent enhancements to the Geodise Database and XML Toolbox are described. The Jython scripting environment is now supported in addition to the Matlab technical computing environment. We also discuss the software engineering process used to harden the toolboxes for public release, and the usability issues highlighted by feedback from users. Finally, a new application of the Geodise toolboxes in the domain of Electric Impedance Tomography is presented.

1. Introduction

The Geodise toolboxes provide Grid client functionality to Problem Solving Environments (PSEs) in widespread use by engineers and scientists [1]. The toolboxes include client functionality to a managed data archive and to computational resources exposed via a range of Grid middleware. Our focus has been to provide simple yet robust functionality driven by the scientific challenges that we and our users face.

The GeodiseLab project is supported as part of the managed programme of the UK Open Middleware Infrastructure Institute (OMII) [2] to enhance the robustness and functionality of the Geodise toolboxes by extending their applicability to a wider community of users. The GeodiseLab toolboxes are freely available to the e-Science community from a repository maintained by the OMII. In this paper we motivate and discuss a number of key enhancements we have made.

Geodise Compute Toolbox
Geodise CondorWS Toolbox
Geodise Condor Native Toolbox
Geodise OMII_1 Toolbox
–
Geodise Database Toolbox
Geodise Database Server
–
Geodise XML Toolbox

Figure 1 GeodiseLab toolboxes for Matlab and Jython (*July 2005*)

The Geodise toolboxes subject to development include the Geodise computational toolboxes, Geodise Database Toolbox and XML Toolbox. The products currently available are listed in Figure 1.

The three key areas of development in the Geodise toolboxes have been: extending the functionality of the toolboxes, supporting additional environments, and hardening the toolboxes for public release. These developments meet a number of requirements raised as scientists and engineers exploit Grid technologies in more challenging or entirely new application scenarios.

Many improvements to the toolboxes have been in response to, or informed by, feedback sought during user workshops. In particular several usability issues have been highlighted by the experiences of engineers exploiting the Grid for the first time.

In section 2 we will describe the Grid technologies supported by the Geodise computational toolboxes, and recent enhancements to the Database and XML Toolboxes. In section 3 we discuss support for the Jython scripting environment. Section 4 outlines the software engineering process used to deliver toolboxes for public release, and our response to user feedback. Finally we demonstrate the use of two of the Geodise toolboxes in the application domain of Electrical Impedance Tomography.

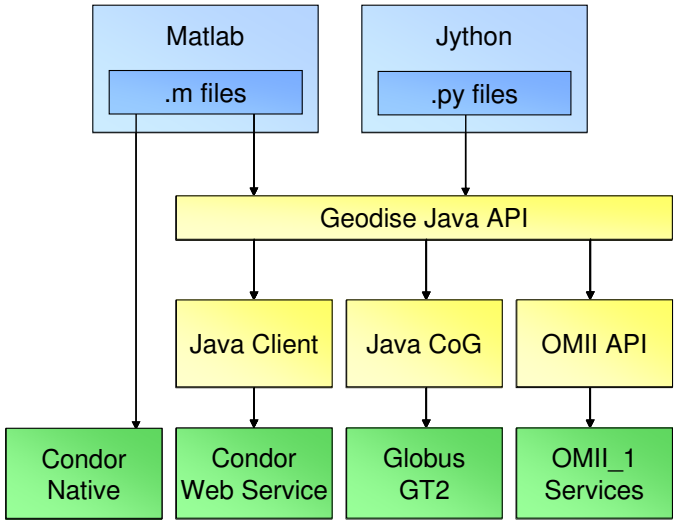

Figure 2 GeodiseLab Compute Toolboxes

2. Extending Functionality

2.1 Compute toolboxes

The original Geodise Compute Toolbox provides client functionality to Grid resources exposed via Globus GT2 [3]. The Globus GT2 middleware is a stable product that is a core technology of the UK National Grid Service (NGS) [7].

However, a user's choice of Grid resource may be motivated by any number of factors, including hardware and software requirements, availability and cost. Indeed it is apparent that users may wish to exploit a heterogeneous collection of resources when scripting workflows that involve a complex assembly of applications.

For example, an engineer performing design optimization involving the fluid dynamic properties of a design may need to invoke three separate applications to calculate the design geometry, computational mesh and CFD solution. These applications may require specific hardware, or resources that have the necessary software licenses.

We have developed additional computational toolboxes to support resources exposed by middleware other than the prevalent Globus software. By providing client functionality to alternative Grid middleware within the same PSE the engineer is able to marshal all of the resources that they require with a single script.

We have developed toolboxes to enable access to previously unsupported computational resources including Condor pools, and those exposed via the OMII_1 platform (Figure 2). These technologies provide differing, yet complementary capabilities. The popular Condor software specializes in the utilization of transient resources (frequently unused Windows workstations) for high throughput computing. The OMII_1 platform allows specific applications to be hosted and invoked as Grid services in an accountable manner.

The OMII_1 platform is a lightweight infrastructure for building Grid applications based upon Web services [2]. OMII_1 provides a number of sophisticated functions that support the operation and management of *Virtual Organisations*, including an accounting model and the ability for users to share resources between themselves.

The Geodise OMII Toolbox provides a suite of functions that enable users to stage data to and from OMII_1 resources, and to invoke applications that are hosted as Grid services. Utility functions are provided to manage OMII_1 accounts and resource allocations. Wherever possible the underlying details are concealed. For example, every effort is made to streamline the configuration and management of the user's credentials in a Java keystore. However we believe that these scriptable

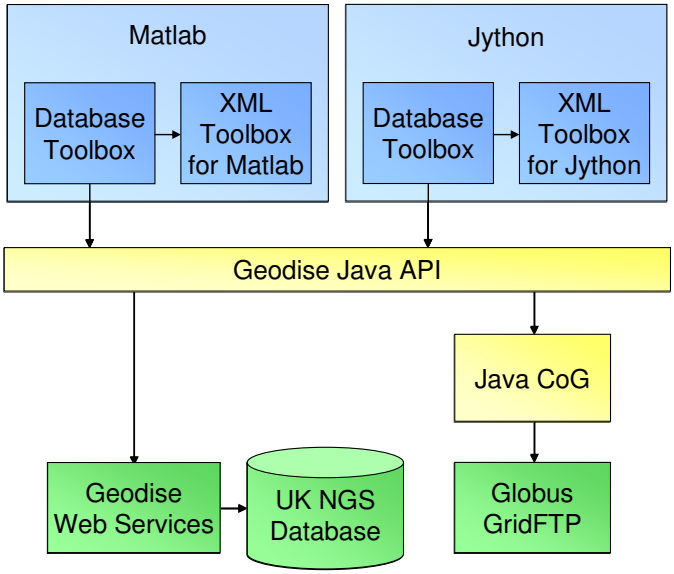

Figure 3 Geodise Database and XML Toolboxes

functions retain flexibility and power that is harder to achieve with a bespoke GUI interface.

Condor [4] is a mature workload management system for high throughput computing that supports cycle scavenging from idle workstations. Providing native and remote Condor pool interfaces to our software platform enables users to almost effortlessly make use of the additional compute power on offer and to integrate this into their daily workflows.

The Condor Native Toolbox provides a lightweight set of functions to programmatically manage the submission of ensembles of Condor jobs from the Matlab environment. The CondorWS Toolbox allows remote access to a Condor pool via a Web service developed at the Southampton eScience Centre. This overcomes a problem commonly experienced by Condor users; namely remote access to a Condor pool across an institutional firewall.

Because of differences in the capabilities of the alternative Grid technologies we provide client functionality in a series of toolboxes each containing commands specific to that middleware. Whilst there are common themes between the toolboxes (e.g. job submission and file transfer), it would be difficult to reflect the full functionality of each technology within a single toolbox.

For example, the OMII_1 platform provides the facility for users to manage the permissions of other members of a *Virtual Organization* to access jobs, data and resource allocations that are shared upon a resource. This functionality is not present in the Globus GT2 and Condor platforms, but is supported by the Geodise OMII_1 Toolbox.

As part of our future work we intend to develop a suite of high level commands that aggregate the common functionality provided by these toolboxes. These functions would allow users plug-and-play resources into a single workflow, perhaps replacing a Globus resource with an OMII_1 resource, or *vice versa*, where appropriate.

We hope to meet the needs of our users by supporting new Grid technologies as required. We anticipate that demand will grow for support for WSRF resources [5] as WSRF Grids are developed.

2.2 Database Toolbox

The Geodise Database Toolbox consists of client and server tools which enable distributed users to easily manage, share and reuse their data from within a PSE. Users with no database experience can integrate data management into their applications by calling the archive, query and retrieve functions provided by the toolbox. Any data files or variables can be stored in the

Geodise archive. These may be augmented with user defined data structures that specify additional descriptive metadata, which can be queried to locate data of interest more easily. Using an Oracle database we store non-standard user defined metadata as unstructured XML for flexibility alongside standard relational metadata for performance.

The Database Toolbox has been extended to support new query commands and administrative functions requested by users of the toolbox. Users can now generate summaries of their metadata using aggregation functions, and return distinct or ordered results. The aggregation functions available are `count`, `max`, `min`, `avg` and `sum`, with an optional group by clause. An example query would be to count the number of archived files with metadata that meets certain criteria and group by user ID, so that there would be a count total for each user. Another example is to find the minimum value of metadata field *a* for each value of *b*.

The ability to remove unwanted data matching certain criteria was frequently requested. To avoid accidental loss of important results we allow users to mark their data for deletion, so that it is hidden from ordinary queries but remains in the database until removed by an administrator. This provides an opportunity for recovery using special query and unmark functions.

The database and its secure Web service interface comprise the Geodise Database Server. Preparation for release of these server components to the OMII has included significant query performance improvements, configuring support for Oracle 9i and 10g, producing server installation documents and build scripts, and deployment testing at different sites (e.g. by the GENIE project [6]).

We have found that Oracle's query execution plan is not always optimal when there is a mixture of relational and XML data. We have therefore improved the performance of our queries by providing hints to the optimiser so that expensive XML searches are only attempted once the more efficient relational conditions and joins have been executed. Performance has also been improved by making use of some XML bug fixes available in Oracle 10g and by not performing authorisation checks when a user is explicitly querying their own data (certificate based authentication is always performed).

To further productionize the software, we have also deployed a Geodise database on the National Grid Service (NGS) [7] and public Web services at Southampton University for users within the UK e-Science community.

2.3 XML Toolbox

The XML Toolbox for Matlab allows the transparent integration of the XML data format into the Matlab PSE. The XML Toolbox allows Matlab variables to be converted to and from an XML based format using a clear and concise syntax. Over 1000 downloads in the first 14 months with feedback from many users have allowed us to improve both the stability and functionality of the toolbox.

The latest version allows a user to easily deal with almost any XML formatted document by transparently converting it to a data structure within their more familiar problem solving environment without having to become an XML expert. This feature was frequently requested by users of the XML Toolbox as they encountered third party XML documents with increasing frequency.

The XML Toolbox is also used by the Database Toolbox to facilitate the storage and retrieval of variables and user defined metadata structures to and from a commercial XML enabled database.

3. Additional Environments

The Geodise toolboxes seek to provide Grid client functionality to environments familiar to scientists and engineers. We initially focused on the Matlab technical computing environment [8][9] as a flexible and powerful PSE which was already widely employed by our users.

We now seek to further our aim by supporting additional PSEs and development environments, such as Jython, that may be preferred by some users to script engineering or scientific tasks on the Grid.

We provide support for the Python scripting language via the Jython implementation of the language [10]. To achieve this we have ported the functionality of the Compute, Database and XML toolboxes into Python by converting the

Matlab

```
>> a.b = 3.1415926535897;
>> a.c = 'a character string'

a =

    b: 3.1416
    c: 'a character string'

>> xml_save( 'demo.xml', a )
>> type demo.xml

<?xml version="1.0"?>
<!-- Written on 01-Jul-2005 13:54:45 using the XML Toolbox for
Matlab -->
<root xml_tb_version="3.1" idx="1" type="struct" size="1 1">
  <b idx="1" type="double" size="1 1">3.1415926535897</b>
  <c idx="1" type="char" size="1 18">a character string</c>
</root>
```

Jython

```
>>> from gdxml import xml_load
>>> v = xml_load( 'demo.xml' )
>>> print v
{'b': 3.1415926535897, 'c': 'a character string'}
```

Figure 4 Interchange of data between Matlab and Jython using the XML Toolbox

syntax of the original Matlab functions into a syntax familiar to users of the Python language.

Our choice of the Jython implementation of Python was based upon its seamless support of third party Java libraries. In this way we have been able to mirror the original architecture of the Geodise Compute and Database toolboxes for Matlab. In this architecture much of the client logic is contained within Java classes, including third party APIs wherever possible, which are accessed via a thin wrapper in the target scripting language (Figure 2). This architecture allows code reuse where-ever possible, thus limiting the cost of maintenance over several environments and reducing the potential for bugs to be introduced to the software.

For example the Geodise Compute Toolbox utilizes the client functionality to Globus resources provided by the Java CoG API [11]. Additional logic is provided by the Geodise Java API that is invoked directly from the Matlab and Jython environments. Most of the logic performed by the wrapper functions in the target scripting language is concerned with validating user input, converting arguments to and from data types in the target language, and handling error conditions appropriately.

Effort was required to provide an interface that is intuitive and appropriate for users in the target language. Additionally function documentation, tutorials and help files are provided in the syntax of the target language.

The XML Toolbox for Jython extends the functionality already present in its Matlab counterpart. It provides a similarly concise syntax to allow the user to serialize and deserialize variables from the Jython workspace.

In addition, it will allow the transparent exchange of scientific data between the Matlab and Jython environments and thus contribute to a PSE-independent collaborative environment for users of different application scripting software Figure 4. Furthermore the interchangeable XML format used by the XML Toolboxes mean that variables archived to the Geodise Database by a user from the Matlab environment can be retrieved by a colleague to the Jython workspace, or *vice versa*.

Agreeing upon a format for data exchange between PSEs would otherwise be a non-trivial task. By providing a common set of tools the engineer is free to choose the PSE that they most prefer.

4. Hardened for Release

4.1 Software Engineering

To provide tools suitable for public release the software engineering process used to develop the Geodise Toolboxes has been standardised to the requirements of the OMII. Mainstream software engineering methodologies and technologies have been leveraged to deliver a quality product.

This process has involved using Ant scripts to automate release builds from CVS. Unit test suites for the target scripting languages are used to test the toolboxes in each of the supported PSEs. Each toolbox is supplied with step-by-step installation instructions for all supported platforms.

The provision of quality documentation is essential to ensure that the toolboxes will be of lasting value to users, therefore in addition to the existing function documentation we have provided high level tutorials. These tutorials describe in detail the basic concepts which engineers and scientists should become familiar with in order to exploit Grid resources.

The toolboxes undergo a Quality Assurance audit before release from the Southampton eScience Centre, and upon receipt by the OMII. Following release the popular Bugzilla tool is used to record and track the progress bugs and feature requests that are reported.

This standardised release process has facilitated the development process for the Geodise team, and has improved the quality of the released toolboxes.

4.2 User feedback

We have also sought to support early adopters of the Geodise toolboxes by co-developing and debugging scripts with users at user-workshops. This process gives us a better understanding of the problems faced by engineers attempting to use the Grid in their research.

Feedback from users of the toolboxes, and from third party testers during the Quality Assurance process, has been invaluable in highlighting bugs, usability issues and raising feature requests.

Recent enhancements to the Geodise Database Toolbox have arisen from feature requests during user-workshops. For example several users were concerned about the original *write-once read-only* philosophy of the data archive, having found that they frequently archived extraneous data that polluted subsequent queries to the database. The *mark for deletion* feature represents a compromise that allows unwanted data to be removed from the archive in a managed fashion.

Documentation is a key usability issue for the toolboxes, and user feedback has been useful in raising issues that require more detailed documentation. User certificates and Public Key Infrastructure are concepts that are new to many users, and frequently represent a significant barrier to adoption. Step-by-step instructions for users encountering this technology for the first time are essential. Thankfully, once correctly configured user certificates often pose few further problems to users of the toolboxes (for the 12 months until renewal at least).

Another usability issue clarified by user feedback is *"what happens when things go wrong?"* Grid resources may be more or less robust, and failures are part of life. A Java stack trace is often intimidating and hard for users to decrypt. It is therefore vital that messages are clear and relevant to the user's problem. Improvement of the clarity of error messages in the target scripting language is part of the ongoing development of the Geodise toolboxes.

Documentation is also valuable to determine and troubleshoot the cause of a failure. The tutorial provided with the Geodise Compute Toolbox also provides suggestions about how users can best cope with failures on the Grid by using exception handling in their scripts.

4.3 Applications

The Geodise toolboxes have been used by researchers in the fields of earth science [12], computational electromagnetics [13], and engineering design search and optimisation [14]. In this paper we describe a new application of the toolboxes in the domain of Electrical Impedance Tomography (EIT).

Electrical Impedance Tomography is an imaging method which infers the distribution of conductive materials inside an opaque volume from measurements of electric potentials on its surface. This non-invasive technique has

```
% create a number of 3D head models, distorted by a small factor
distortion_factor = {0.0, 0.01, 0.02, 0.03, 0.04, 0.06, 0.08, 0.10, 0.12};
for i=1:9
    model = cfg_load( 'default_head_3D.cfg' );
    model = stretch( model, distortion_factor{i} );
    inputfilename = generate_inputfile( model );
    metadata.distortion_factor = distortion_factor{i};
    metadata.model_name = 'head';
    gd_archive( inputfilename, metadata );
end

...

% find all previous runs in database which match range
models = gd_query( [' model_name=head & distortion_factor>=0 & '...
                    ' distortion_factor<0.20 '] );

for c = 1:length(models)
    % download file to local model directory
    filename = ['./model_', num2str(c), '.geom'];
    gd_retrieve( models{c}.standard.ID, filename );
    % create submission information
    job = condor_job( 'beginner' );
    job.executable = 'reconstruct3d.exe';
    job.arguments = filename;
    jobhandles{c} = condor_submit( job );
end

% wait for results
condor_waitfor( jobhandles );

% extract reconstruction errors
ErrList = extract_image_errors3d( './model_*', 'default_head_3D.cfg' );
```

Figure 5 An Electrical Impedance Tomography design search utilising the Geodise Database and Condor Toolboxes

applications in the fields of medical imaging and industrial material characterisation.

When applied to complex 3D problems such as the geometry of a human head, the computational demands of the EIT image reconstruction algorithm can be very high. The large computational and memory requirements make Grid technology well suited for this problem area. In addition, parametric design searches, which can be performed in parallel on Grid resources, allow researchers to quickly establish the effects of the many control parameters of an EIT algorithm upon the quality of the reconstructed image and its medical relevance.

In this example, we will describe the application of the Geodise Condor Native Toolbox and the Geodise Database Toolbox to meet the computational and data management requirements of an EIT parameter study.

Here the EIT reconstruction algorithm was available as a standalone application that the researcher wished to invoke multiple times concurrently. A local Condor pool was available to the researcher, who used the Geodise Condor Native toolbox to automate the submission of the computational jobs.

The Geodise Database Toolbox was used to store the numerous input and output data files from the multiple invocations of the EIT reconstruction algorithm. Files and workspace variables can be annotated with appropriate metadata and archived to the Geodise Database. The numerous files corresponding to a single invocation of the algorithm may be associated into datagroups. Queries across the metadata can be used to locate and retrieve files and variables from the archive.

The Matlab script in Figure 5 is a simplified version of a workflow used to explore the effects of nine control parameters upon the output of an EIT reconstruction algorithm. The

first section of the script generates input files for a model problem in which a single control parameter, the *distortion_factor*, is varied. These input files are then archived to the Geodise database with appropriate metadata using the function `gd_archive`.

The second section of the script which may be run at a later date invokes `gd_query` to return the metadata of all of the input files in the Geodise Database that meet the search criteria. These files, identified by the file ID contained in the metadata, are then downloaded to the current directory using `gd_retrieve`.

The function `condor_job` is used to create a Matlab structure that is used to describe the properties of a Condor job. This structure is then passed to `condor_submit` which submits the EIT reconstruction algorithm to a Condor pool, along with the executable and input file.

Once the EIT reconstruction jobs have been submitted the function `condor_waitfor` is used to block the script until all of the jobs have finished. Upon completion Condor will transfer all of the files produced by a job to the local filesystem.

Output files are then parsed to determine the errors resulting from the influence of the *distortion_factor* parameter. These errors can be plotted and viewed in the Matlab environment. Additionally the output files may be archived to the Geodise database to produce a record of the experiment.

In this way the researcher can easily explore the effects that different parameters have upon the performance of EIT reconstruction algorithms. The high throughput computing methodology can scaled to cope with the large numbers of simulations required to explore the interactions of multiple control parameters.

5. Conclusions

The GeodiseLab project has sought to improve and extend the Geodise toolboxes to meet the requirements of engineers and scientists wishing to use the Grid. Through a standardised development process and attention to user feedback we have attempted to deliver quality tools that improve the usability of the Grid.

References

[1] Geodise Project. http://www.geodise.org/

[2] Open Middleware Infrastructure Institute. http://www.omii.ac.uk/

[3] The Globus Alliance. http://www.globus.org/

[4] Condor Project. http://www.cs.wisc.edu/condor/

[5] WS-Resource Framework http://www.globus.org/wsrf/

[6] Grid Enabled Integrated Earth system model. http://www.genie.ac.uk/

[7] National Grid Service. http://www.ngs.ac.uk/

[8] Eres, M.H., Pound, G.E., Jiao, Z., Wason, J.L., Xu, F., Keane, A.J., Cox, S.J. Implementation and utilisation of a Grid-enabled problem solving environment in Matlab. Future Generation Computer Systems. (2005) 21: 920-929.

[9] Matlab. http://www.mathworks.com/

[10] Jython. http://www.jython.org/

[11] von Laszewski, G., Foster, I., Gawor, J., Lane, P. A Java Commodity Grid Kit. In: Concurrency and Computation. (2001) 13: 643-662.

[12] Price, A. R., *et al.* 2004, Tuning GENIE Earth System Model Components using a Grid Enabled Data Management System. In: Proceedings of the UK e-Science All Hands Meeting 2004, Nottingham, EPSRC (2004) 593-600.

[13] Molinari, M., Thomas, K.S., Cox, S.J. Electromagnetic design search and optimisation of photonic bandgap devices on distributed computational resources. In: Proceedings of the Fifth International Conference on Computation in Electromagnetics. (2004) 103–104.

[14] Song, W., Keane, A.J., Cox, S.J. CFD-based shape optimisation with Grid-enabled design search toolkits. In: Proceedings of the UK e-Science All Hands Meeting 2003, Nottingham, EPSRC. (2003) 619–626.