Proceedings of the UK e-Science All Hands Meeting 2004, © EPSRC Sept 2004, ISBN 1-904425-21-6

Workflow Support for Advanced Grid-Enabled Computing

Fenglian Xu[*], M. Hakki Eres[*], Feng Tao[+], and Simon J. Cox[*]

[*] School of Engineering Sciences, University of Southampton
Southampton SO17 1BJ
{fxu, eres, sjc}@soton.ac.uk

[+] School of Electronics and Computer Science, University of Southampton
Southampton SO17 1BJ
ft@ecs.soton.ac.uk

Abstract

The Geodise project brings computer scientists and engineer's skills together to build up a service-oriented computing environment for engineers to perform complicated computations in a distributed system. The workflow tool is a front GUI to provide a full life cycle of workflow functions for Grid-enabled computing. The full life cycle of workflow functions have been enhanced based our initial research and development. The life cycle starts with a composition of a workflow, followed by an instantiation of the workflow, further processing to validate the workflow, and it finally ends with a submission of the workflow to a Grid-enabled Matlab enactment engine. Further more, the workflow tool has the following features: debugging, and run-time monitoring and steering. The workflow tool integrates computation, database, knowledge, and Grid technologies to share hardware, software and data resources within and across boundaries of organisations.

1. Introduction

The Geodise project [1] aims to build a service-oriented computing environment, which enables users to perform complicated computations within and across their network boundaries. The engineering design search and optimisation (EDSO) is a long and repeated process, which involves generating geometry using computer aided design (CAD) packages, followed by meshing the geometry and analysis of the meshed geometry to find objective function values which measure the performance of the design. The motivation of this research is to provide an integration environment for engineers to perform a Grid-enabled computing with minimum programming requirements and with their own familiar language in an efficient and productive way. The integration of database management, knowledge support and Grid-enabled computing tool would enable the engineers to share data resources, software resources and compute resources with an efficient knowledge advisor. The database management is integrated to archive and retrieve workflows and run-time data or data structures as well as their associated metadata to/from a database. The knowledge support in Geodise aims to annotate compute components and published them into a library so that they can be retrieved late by providing a key word. The knowledge support also provides the advisor during a workflow construction. The compute tools are set of Matlab functions which allow the computing to be performed from remote computers where the Globus servers have been installed. The run-time results can be transferred from remote compute resources back to the Matlab server via GridFTP and can be further transferred back to the workflow tool for visualisation or monitoring.

Therefore a workflow tool is required to provide a GUI environment, which assists users quickly, and easily to construct new workflows or load existing workflows to perform a complicated service-oriented computing over the Grid [2]. The execution engine is the Matlab software [3], which is a very popular computing, and scripting environment for engineers. The Matlab scripts representing workflows are more readable and editable for engineers than other languages such as XML. The workflow tool also decouples the workflow orchestration and the process execution environments. This would allow many orchestration clients to a process execution server. This design has not only reduced the cost of expensive Matlab server, but also allowed multiple users to orchestrate the workflows from different platforms: Windows, Linux or Unix.

There are numerous workflow execution and problem solving environments available in both academic and commercial domains. For example, iSIGHT [10] enables users to rapidly integrate simulation based design processes, such as commercial CAD/CAE software, in-house software, and spreadsheets. This integration process yields automatic execution of simulation programs and easy exploration of the design space. Another commercial tool is called Fiper [11] also integrates various workflow processes in a graphical environment and allows them to be executed on distributed systems. ModelCenter [12] is also a graphical problem solving environment which specializes in process integration and design optimisation. Triana [13] is an open source problem solving environment which includes a library of analysis tools and it also allows users to use their tools.

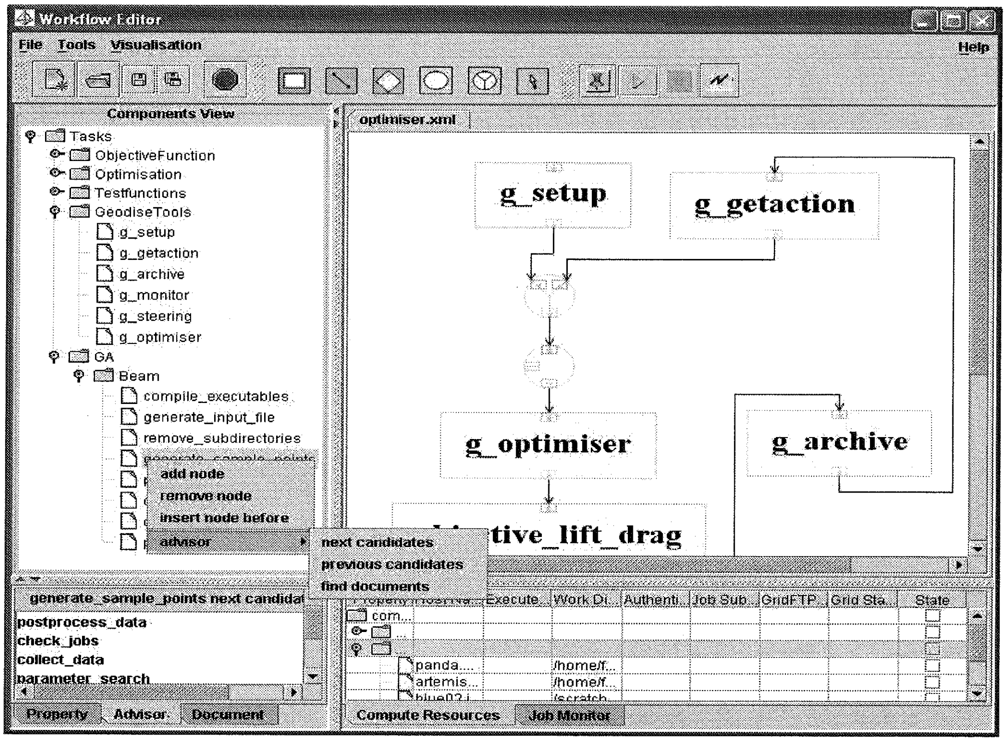


Figure 1 The main frame of the workflow tool.

The remainder of the paper is arranged as follows: The features of the workflow tool are presented in Section 2. Experiences of users are discussed in Section 3, conclusions and future work is made in the final section.

2. Workflow Tool Features

The workflow tool provides the following functions:

- Construction
- Instantiation
- Validation
- Submit to workflow engine for running
- Debugging
- Run-time monitoring and steering

Most of the features have been developed in the workflow tool; the monitoring and steering are under development. These features are discussed in details in the following sections. Figure 1 shows the main frame of the workflow tool.

The main frame is consisted of a menu bar, a tool bar and a main panel. The main panel is split into four panels: a components view and a multiple tabbed panels to show the property, advisor and the document of the selected components in the components view are on the left side and a workspace for composing and editing a workflow and a multiple tabbed panels to show the compute resources and the run-time job status are on the right side.

The menu contains a File menu, Tools menu, a Visualisation menu and a Help menu. The File menu mainly includes the submenus: new, open, save, save as, import from DB and export to DB, etc. The Tools menu mainly contains tools for managing proxy certificates, computer resources checking and run-time management. The visualisation tool provides a set of functions to visualise both static and run-time graphic results.

The tool bar consists of a group of tool buttons and some of them have the same functions as the submenus. The drawing buttons are used for drawing different workflow nodes. The workflow nodes include task nodes, conditions nodes, loop nodes, and join nodes and connection nodes. As it can be seen at Figure 1, each node has at least one input port and one output port for storing the input and output parameters, and for making connections.

The components view loads components from an XML format file and shows the components in a hierarchical tree structure. It can be seen from the top left panel, the tree leaves, such as gd_archive are Matlab components under the GeodiseTools node. The ascend nodes of the leaf nodes represent a hierarchical structure with semantic knowledge information.

In order to provide more knowledge of components, the bottom left panel used for showing the properties of the selected components, the detailed documentation is also available for annotated components via knowledge advisor. To assistant the user to compose a complicated workflow, the next and previous candidates of the selected component can be retrieved via the knowledge support. This feature has enhanced Geodise workflow tool comparing with the other tools.

The top right panel is a workspace for the users to construct and edit workflows. The workflows can be saved as XML format file and it can be loaded or retrieved from both a local and a remote file system.

A configured workflow needs to be submitted to a specified Matlab server and can be further deployed on to the Grid level for computation. Therefore the compute resources management is required for a particular user. The Matlab server is used for an execution engine, as the components of workflow are Matlab scripts or functions. The Matlab is also a very familiar problem-solving environment for engineers. The users can benefit from the large number of numerical and plotting library functions.

2.1 Workflow Composition

Workflows normally consist of a set of task nodes linked by connection nodes. They may include some control nodes: condition nodes, loop nodes and join nodes. The task nodes are components from the components tree. The components in the tree are executable Matlab functions, which can either be native Matlab library functions, or user defined Matlab functions. The user can construct workflows by using one the following methods:

- construct workflows from scratch with knowledge support,
- browse existing workflows from a local file system,
- search existing workflows in the database,

Three different ways are designed for different purposes. The first one is to assist the user to quickly create new workflows with knowledge support [3] which lists all possible pre- or post-candidates for a selected component in the tree view. Figure 2 shows how the knowledge advisor can be obtained during the workflow construction. After the `generate_sample_point` has been linked to the workflow, the user needs to get advisor to know which are the next possible candidates. The next candidates are listed in the advisor panel. The user can then select one of the candidates as the next task node. The pre- or post- candidates can be found by matching functions semantic interfaces that are pre-acquired during function annotation process. The function annotation process is a custom process which annotates the Matlab components and is used by the component developers. The annotated Matlab components can then be loaded into the workflow tool to provide the advisor. The details of the function annotation process can be found in the paper [6]. The user can drag a component from the components view and drop it into the workspace; a visual task node is formed in the workspace.

The connection node can be drawn from one output port of a node to an input port of another node. The connected nodes can be task nodes, or any of control nodes but not the connection nodes. The design of one input port and one output port for a task node over one input parameter with one input port is to avoid making over heavy connections especially for the function components with lots of input or output parameters.

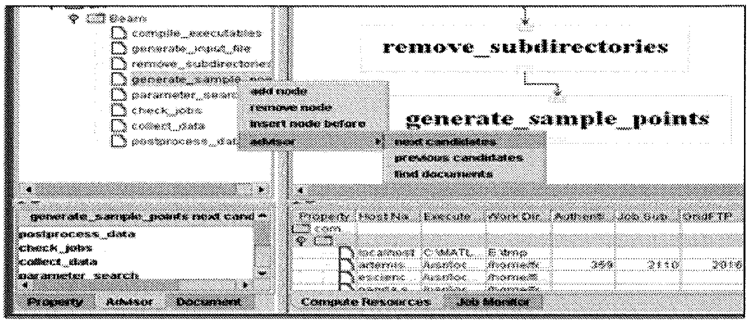
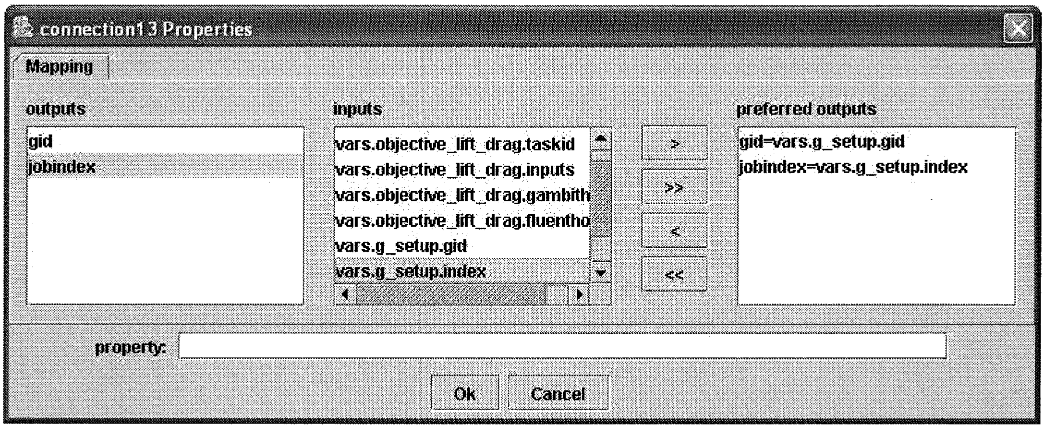


Figure 2 Knowledge advisor during a workflow construction.


connection13 Properties

Mapping

outputs

- gid
- jobindex

inputs

- vars.objective_lift_drag.taskid
- vars.objective_lift_drag.inputs
- vars.objective_lift_drag.gambith
- vars.objective_lift_drag.fluentho
- vars.g_setup.gid
- vars.g_setup.index

preferred outputs

- gid=vars.g_setup.gid
- jobindex=vars.g_setup.index

property:

Figure 3 Connection node configuration window of the workflow editor.

The second method enables the user to reuse their previous workflows which have been saved in the local file system. The user can browse to a certain directory to load an existing workflow for running or editing. This has increased the efficiency of productivity as construct and instantiate a workflow can be very time consuming. However this method has limited the user who can only reuse their previous workflows.

The last one allows the user to reuse and share workflows with other users within or across virtual organisations via the database management [7][8]. In order to share their data resources with the others, the workflow tool has enabled the user to export their workflows, Matlab script file, run-time data and their associated with meta data into the database system. The workflow tool has also enabled the user to import these files from the database to the local file system for a reuse.

2.2 Workflow Instantiation and Validation

A constructed workflow is a concept workflow, which has not been necessarily instantiated. Only the instance of the concept workflow can be executed in a run-time environment. To ensure the instantiated workflow can be run on a specified Matlab server, the instance of workflow has to be validated before a submission. This is because the workflow task nodes may be unavailable on the specified Matlab server.

The workflow instantiation involves an initialisation of the workflow nodes and a deployment of the workflows to the computer resources. The initialisation of the workflow includes configuring the workflow nodes and mapping dataflow between connected nodes. Simply filling the property sheet of each node can configure the workflow nodes. The main configuration work is on mapping dataflow between the two connected task nodes. This version of the workflow tool has enhanced the feature of dataflow which can be any data from all previous linked task nodes to the next connected task node. Figure 3 shows how to configure dataflow for the current connection node. The configured workflow can be deployed to a set of Grid-enabled computer resources, such as the UK e-Science Grid. The compute resource tool also needs the user to select a Matlab server to execute the workflow. The details of many other features of the compute resource tool can be found elsewhere [9].

The validation of a workflow checks availability of task nodes against the selected Matlab server. After the validation, a visual feedback is presented to the user by the workflow tool. We choose the green colour to represent a valid node and the red colour for an invalid node. When the user click on the validation button on the tool bar, a run-time configuration window is popped up which enables the user to configure run-time mode:

- run the workflow on a local Matlab without a distributed mode
- run the workflow on a local Matlab with a distributed mode
- run the workflow on a remote Matlab without a distributed mode
- run the workflow on a remote Matlab with a distributed mode

If the user selects the fourth configuration mode, more computer resources have to be chosen to perform the distributed computation. These computer resources have to have the Globus server installed on. The further distributions processes can be handled by the custom specified Matlab components or by the Geodise compute tools.

The run-time data can be archived into the database and retrieved later for monitoring or steering. During configuration of the input/output parameters of the workflow nodes, the user can select the parameters to be archived. This would enable the selected parameters to be archived into a group, which is associated with the workflow.

2.3 Submission and Monitoring

A valid workflow instance can be submitted to a remote or a local Matlab server for running. This process first generates a Matlab script of the workflow and then submits the script to the remote Matlab server for running via the Globus job submission. The run-time process messages and the output results can be monitored from the workflow tool. The final results can also be available for a view.

The workflow tool is designed for being able to run multiple threads. Two threads are in charging of a job submission and a job monitor. While the job submission thread producing and transferring the output file to the local file system, the monitor thread is responsible for displaying the process messages and the run-time results to the monitor window, which is shown in the bottom window in Figure 4.

It is necessary for the user to monitor the long process so that the user can make a decision to stop or carry on the current process. The run-time plot monitoring function is still under implementation.

2.4 Compute Resource Management

The compute resource component of the workflow tool manages all registered shared compute resources (Figure 5). The Matlab servers are specified by host names, directories indicating the executable Matlab location, and the users' working directories where the user defined Matlab components are located. The distributed servers are specified by host names and working directories. In addition to the above server information, all of the servers have included Grid resource checking property: authentication time, GRAM job submission time, GridFTP time and Grid resource status. The Grid resource status can be either "ok" or "failed". If the authentication, job submission, and GridFTP functionality from the client to Globus servers are tested successfully, the Grid resource checking is passed. The compute resource management enables users to add and remove a server from the tree table. It also enables the users to check the Grid computing resource individually before using it. The tests are measured as authentication time, GRAM job submission time, and GridFTP time. These measurements give the users an indication of the status of Grid resources prior to a real computation.

2.5 Debugging

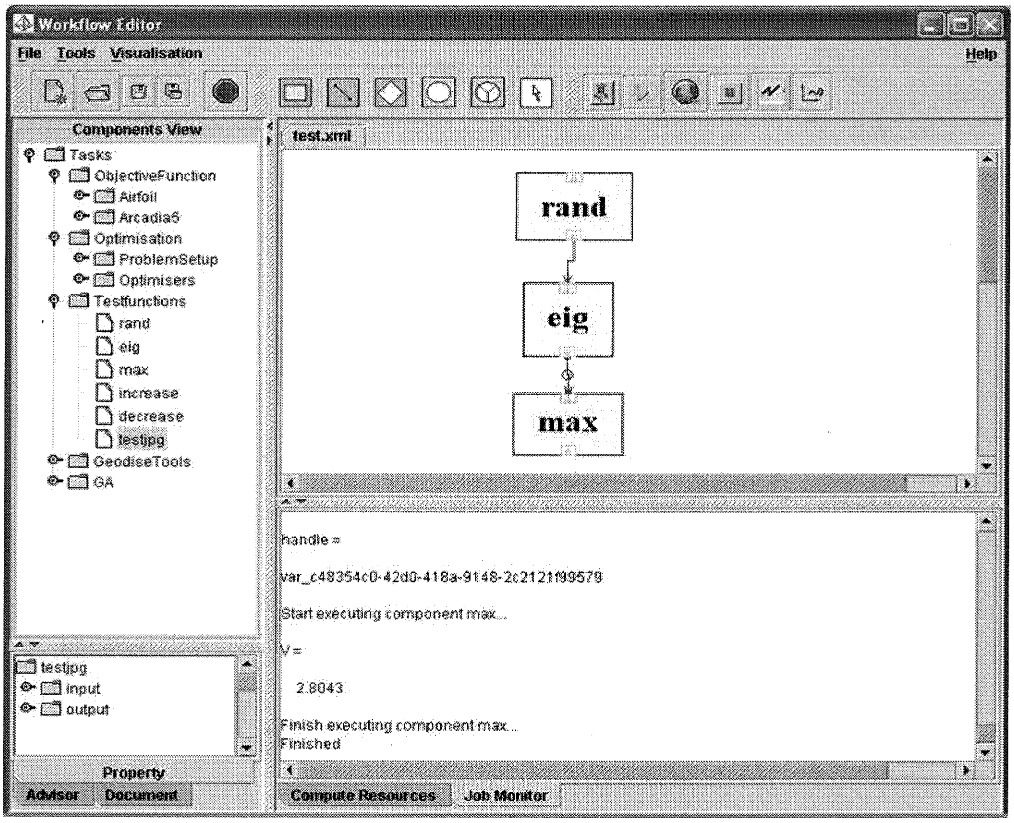
The debugging function is used for checking the run-time results step by step. Due to decouple of orchestration and the execution engine, it is difficult to pause the running job and resume it again. Submitting a sub workflow to the remote

Matlab server and examining the outputs of the sub-workflows can resolve the problem. A start node and an end node can define the sub-workflow. The start node is selected by clicking the right mouse button and the end node is selected by a combination of the *shift* key and the right mouse button.

The debugging feature helps the user to find out where the problem is immediately so that they can try with new input values.

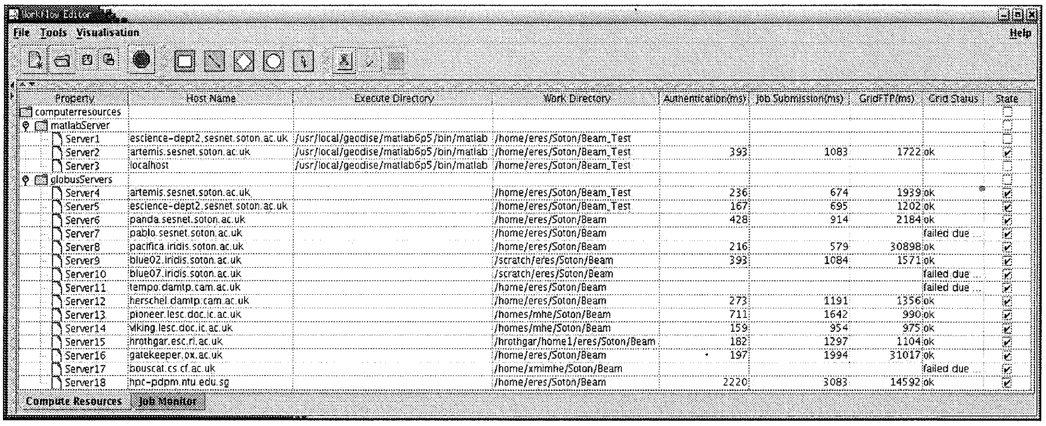
3. Experiences of users with different tools

The workflow tool has been released to the Computing Engineering Design (CED) group in School of Engineering at University of Southampton. The postgraduates, who have experience in using other commercial optimisation tools, such as iSIGHT [11] and ModelCenter [12], are using these tools to perform their design search and optimisation computations. Most commercial tools allow users to integrate computational components to their environment to a certain level, however, in some cases users require more flexibility for their specific research purposes. The Geodise toolboxes proved to be more flexible and powerful as they rely on open standards and Grid technologies.



```
handle =

var_c48354c0-42d0-418a-9148-2c2121f99579

Start executing component max...

V =

   2.8043

Finish executing component max...
Finished
```

Figure 4 Run-time job monitor window of the workflow tool showing real-time results of an executing workflow.



Property	Host Name	Execute Directory	Work Directory	Authentication(ms)	Job Submission(ms)	GridFTP(ms)	Grid Status	State
computerresources								□
matlabServer								□
Server1	escience-dept2.sesnet.soton.ac.uk	/usr/local/geodise/matlab6p5/bin/matlab	/home/eres/Soton/Beam_Test					□
Server2	artemis.sesnet.soton.ac.uk	/usr/local/geodise/matlab6p5/bin/matlab	/home/eres/Soton/Beam_Test	393	1083	1722	ok	☑
Server3	localhost	/usr/local/geodise/matlab6p5/bin/matlab	/home/eres/Soton/Beam_Test					□
globusServers								□
Server4	artemis.sesnet.soton.ac.uk		/home/eres/Soton/Beam_Test	236	674	1939	ok	☑
Server5	escience-dept2.sesnet.soton.ac.uk		/home/eres/Soton/Beam_Test	167	695	1202	ok	☑
Server6	panda.sesnet.soton.ac.uk		/home/eres/Soton/Beam	428	914	2184	ok	☑
Server7	pablo.sesnet.soton.ac.uk		/home/eres/Soton/Beam				failed due ...	☑
Server8	pacifica.iridis.soton.ac.uk		/home/eres/Soton/Beam	216	579	30898	ok	☑
Server9	blue02.iridis.soton.ac.uk		/scratch/eres/Soton/Beam	393	1084	1571	ok	☑
Server10	blue07.iridis.soton.ac.uk		/scratch/eres/Soton/Beam				failed due ...	☑
Server11	tempo.damtp.cam.ac.uk		/home/eres/Soton/Beam				failed due ...	☑
Server12	herschel.damtp.cam.ac.uk		/home/eres/Soton/Beam	273	1191	1356	ok	☑
Server13	pioneer.lesc.doc.ic.ac.uk		/homes/mhe/Soton/Beam	711	1642	990	ok	☑
Server14	viking.lesc.doc.ic.ac.uk		/homes/mhe/Soton/Beam	159	954	975	ok	☑
Server15	hrothgar.esc.rl.ac.uk		/hrothgar/home1/eres/Soton/Beam	182	1297	1104	ok	☑
Server16	gatekeeper.ox.ac.uk		/home/eres/Soton/Beam	197	1994	31017	ok	☑
Server17	bouscat.cs.cf.ac.uk		/home/xmimhe/Soton/Beam				failed due ...	☑
Server18	hpc-pdpm.ntu.edu.sg		/home/eres/Soton/Beam	2220	3083	14592	ok	☑

Figure 5 Compute resources management panel of the workflow tool.

4. Conclusions and Future Work

Geodise workflow tool has provided a GUI for the engineers to perform their complicated computing tasks with minimum requirements of the programming skills. This tool has provided an efficient and economical way for the engineers to perform their optimisations. The knowledge advisor enables the user to quickly construct a workflow or search for an existing workflow with a key word within or across networks. The database system has allowed the user to archive the workflows and together with the associated data for storing and to retrieve them for a reuse. The Grid technology has granted a complicated computation to be performed in a distributed environment.

Future work will include the implementation of run-time plot monitoring and steering. This would enable the user to steer the process such as sending commands to carry on or to stop a long time process or even to change the design variables according to the monitoring results for a long time process. Further research can be done on exploring to generate BPEL4WS [15] as an additional feature of the workflow tool.

References

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

[2] M.H. Eres, G.E. Pound, Z. Jiao, J.L. Wason, F. Xu, A.J. Keane, and S.J. Cox. Implementation and Utilisation of a Grid-enabled Problem Solving Environment in Matlab, Future Generation Computer Systems (In Press).

[3] Matlab 6.5, http://www.mathworks.com/

[4] F. Tao, N. Shadbolt, L. Chen, F. Xu, S. Cox. Semantic Web based Content Enrichment and Knowledge Reuse in e-Science. Submitted to ODBASE, Larnaca, Cyprus, October 25-29, 2004.

[5] L. Chen, S.J. Cox, F. Tao, N.R. Shadbolt, C. Goble, C. Puleston. Empower Resource Providers to Build the Semantic Grid. Accepted to the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI'04), Beijing September 2004.

[6] Z. Jiao, J. Wason, M. Molinari, S. Johnston, and S.J. Cox. Integrating Data Management into Engineering Applications. Proceedings of UK e-Science All Hands Meeting 2003, pp. 687-694.

[7] Z. Jiao, J.L. Wason, W. Song, F. Xu, A.J. Keane, and S.J. Cox. Databases, Workflows and the Grid in a Service Oriented Environment.

[8] J.L. Wason, Z. Jiao, W, Song, and S.J. Cox. Deployment and Exploitation of Grid-enabled Data Management for Engineers. Accepted to the e-Science AHM 2004, Nottingham, 31 August - 3 September 2004.

[9] F. Xu, M.H. Eres, D.J. Baker, and S.J. Cox, Tools and Support for Deploying Applications on the Grid. Accepted to the IEEE SCC 2004.

[10] iSIGHT http://www.engineous.com/

[11] Fiper http://www.engineous.com/

[12] ModelCenter http://www.phoenix-int.com/products/ModelCenter.html

[13] Triana Project http://www.triana.co.uk/

[14] F. Tao, S.J. Cox, L. Chen, N.R. Shadbolt, F. Xu, C. Puleston, and C. Goble. Applying the Semantic Web to Manage Knowledge on the Grid. Accepted to the e-Science

AHM 2004, Nottingham, 31 August - 3 September 2004.

[15] Business Process Execution Language for Web Services (version 1.0), http://www-106.ibm.com/developerworks/library/ws-bpel/