Workflow Tool for Engineers in a Grid-Enabled Matlab Environment

Fenglian Xu and Simon J Cox

e-Science Centre,
School of Engineering Sciences,
University of Southampton
{F.Xu, sjc}@soton.ac.uk

Abstract

The Geodise Project aims to aid engineers in the design process by making available a suite of design search and optimisation tools and Computational Fluid Dynamics (CFD) analysis packages integrated with distributed Grid-enabled computing, databases and knowledge management technologies. Engineering Design Search and Optimisation (EDSO) is a long and repetitive process requiring a complex sequence of tasks to be scripted together. We have developed a visual workflow tool with a friendly graphic user interface (GUI) to assist in the generation and execution of these scripts. It provides a set of functions: workflow construction, resource validation, script generation, process automation, job state monitoring and visualisation. Our workflow tool generates scripts in Matlab, which is familiar to Engineers and allows them to be re-used. Future work will incorporate a range of knowledge support mechanisms into the Workflow tool to assist in workflow construction and configuration.

Keywords: Workflow, process automation, Engineering Design, Grid-enabled, Matlab.

1. Introduction

The Geodise project [1] is building a Grid-enabled Problem Solving Environment (PSE) to carry out Engineering Design Search and Optimisation (EDSO) involving computational fluid dynamics and brings together the collective skills of engineers and computer scientists. For design optimisation it is necessary to analyse a design and obtain a value of the objective function, which measures the performance of the design. The optimisation process usually starts with a parameterised geometry in a computer aided design (CAD) package. The geometry is then meshed for the analysis program. After the analysis the solution is post-processed to obtain the value of the objective function. This process is repeated to systematically improve the design as part of the optimisation procedure [2].

Feasibility studies on some workflow specifications such as WSFL [3], XLANG [4], XPBL [5] have shown that they are not suitable for engineers working in a Matlab environment. These specifications lead to scripts generated in XML format, which limits their subsequent re-use and editing by engineers. There were also no standards until IBM, BEA and Microsoft recently proposed WS-BPEL [6] as an open standard specification. It is designed for Web services choreography which requires that: the components are web services, and users have to understand XML in order to compose a workflow.

For engineers the scripts describing their workflows should be human-readable and editable outside any workflow tool. We have therefore focused on providing a workflow tool to generate scripts for the popular Matlab [7] PSE. Along with a sophisticated set of tools and post-processing facilities, Matlab is supported on a wide range of platforms and operating systems; therefore we have chosen it as an exemplar hosting environment for the EDSO process. The challenge is to provide a Grid-enabled PSE for workflow construction with knowledge support for novel users and resource sharing within a virtual organisation (VO).

2. Workflow System Architecture

The workflow architecture is designed for assembling components from Matlab components and/or legacy codes written in other languages. Fig.1 shows the workflow system architecture and illustrates how Grid technologies such as Globus [8] can be embedded into a workflow. This system provides a loosely coupled Grid-enabled environment so that any component can be easily plugged into the system.

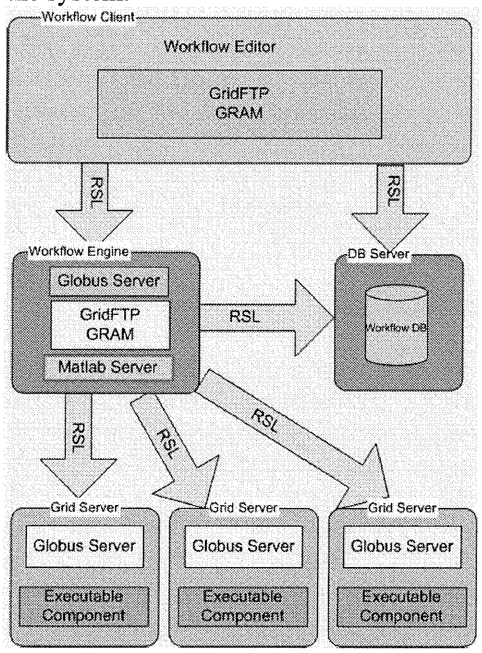

Fig.1 Workflow System Architecture

The Workflow Editor is a front end GUI tool mainly for workflow construction and runtime job management. GridFTP and GRAM are used for transferring files and submitting jobs to remote resources. The workflow engine is a Globus server where the Matlab program with user defined components is installed. To access user-defined components in a distributed system, GridFTP and GRAM are required in the workflow engine.

The Database server is where the workflows and their associated results are stored. The distributed Grid servers have hosted Globus servers and executable legacy codes. The implementation of communication is via the Resource Specification Language (RSL 1.0) [9].

3. Workflow Tool

The workflow tool and its deployment system requirements are described in this section.

3.1 Workflow GUI

Workflow Tool is a standalone GUI application which runs on any platform. It is implemented in Java language. The main frame (see Fig. 2) is split into three panels:

- Component View
- Workflow View
- Compute Resource View and Job Monitor View

The component view displays components loaded from an XML file, which may be dynamically generated. The components are defined as Matlab functions and are stored in a file system where Matlab is installed. The library includes Matlab scripts (.m files) for components and subcomponents. The subcomponents are functions to be called by the functions of the components. The components are laid out as a hierarchical tree structure. The tree leaves represent components and the other tree nodes define high level concepts for the leaf component. The concepts are used for retrieving information while obtaining knowledge advisory. The current version of the workflow tool only allows loading predefined components represented as an XML file. The XML has a list of functions that are defined by a function name, a list of inputs and a list of outputs. The inputs and outputs are specified by a type, a name

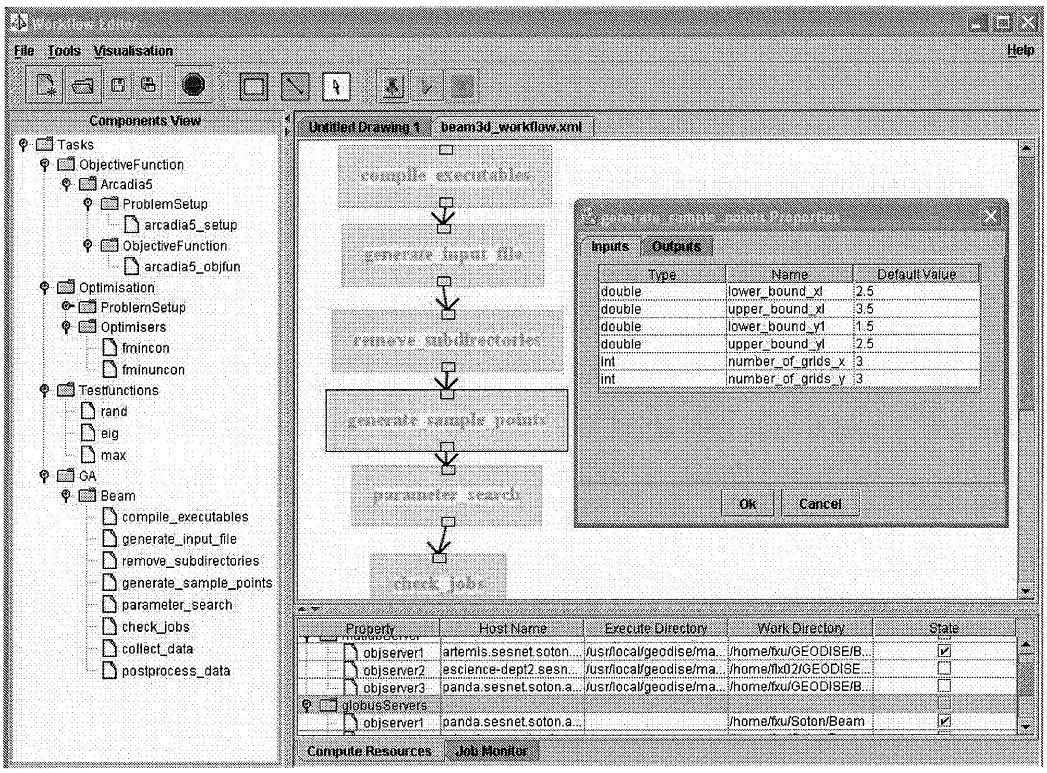

Fig.2 Workflow Tool

and an initial value. The leaf components can be dragged and dropped into the workflow view and they are displayed as text boxes. The labels of the text are the name of the components.

The workflow view is a multiple tabbed panel that allows users to work on multiple workflows simultaneously. The users can either create a workflow from scratch or open an existing workflow from a local file system or a database. The existing workflow can be modified and expanded using the same way as creating a new workflow. The panel is used for assembling components to form a workflow via drag and drop with connections between components representing dataflow. The input and output properties of a component are displayed in a property sheet that pops up while double clicking on a selected component with right mouse button. Each sheet has a table with three columns: type, name and value. The users can change the initial value from the table and the contents are saved after clicking on the OK button.

The compute resource view and job monitor view are in a shared space within a tabbed pane. The compute resource view shows all the Matlab servers and distributed servers which are predefined in a configuration file in XML format. The Matlab servers specify host names, directories where Matlab is located, and the users' working directories where the user defined Matlab components are located. The distributed servers specify host names and working directories. The users can check/uncheck the servers provided by the system to validate the workflow task components.

The job monitor view is used for displaying runtime job status, especially useful for a long time run, so that the users know the job status.

3.2 System Requirements

The software is implemented in Java as a standalone application and can be run on any platform. The workflow can be submitted to a remote Globus server which requires Java Cog [10][11] to be installed. The tool requires JSDK1.4 and the Java Cog 1.1 [10,11]. In addition, it requires a certificate authorised by a trusted organisation such as Globus Certification Authority (CA) or UK e-Science CA. The workflow engine requires: Matlab 6.5, Globus 2.2, Java Cog 1.1 (optional). The distributed servers require Globus 2.2.

4. Conclusions and Future Work

In this paper, the workflow system architecture and the workflow tool are presented for the EDSO process. The workflow tool enables the users to construct, validate, run and monitor workflows on a distributed Grid-enabled system. The user assembles a workflow in a drag and drop environment. The components of a workflow are validated before submitting to a remote Matlab server. The runtime job status can be monitored from the tool. This has provided a friendly GUI tool to users so that they can easily construct workflows. The script from the workflows is generated in Matlab and can be edited/ re-used. Future work will integrate knowledge support into the tool for EDSO. A decision component will be provided at decision points of a workflow and a parallel sequence will also be considered so that data can be passed from multiple components. We will also allow for dynamic integration of new components/ resources.

5. Acknowledgements

The work is supported by the Geodise e-Science pilot project (UK EPSRC GR/R67705/01). We are very grateful to the Geodise Team for many helpful discussions.

References

[1] The Geodise Project (2002) http://www.geodise.org/
[2] A Grid-enabled problem solving environment (PSE) for design optimisation within Matlab. Pound, G, Eres, H, Wason, J, Jiao, Z, Keane, A.J, and Cox, S.J. Proceedings of 17th IPDPS (2003). Nice, France.
[3] Web Services Flow Language (IBM) http://www.oasis-open.org/cover/wsfl.html
[4] Web Services For Business Process Design, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm
[5] Business Process Modelling Language (BPML) http://www.bpmi.org/
[6] Business Process Execution Language for Web Services (version 1.0), http://www-106.ibm.com/developerworks/library/ws-bpel/ (2003)
[7] Matlab 6.5, http://www.mathworks.com/ (2002)
[8] The Globus Project, http://www.globus.org/ (2002)
[9] Globus Resource Specification Languange (RSL) v1.0, http://www.globus.org/gram/rsl_spec1.html (2002).
[10] M. Parashar, G. von Laszewski, S. Verma, J. Gawor, K. Keahey, N. Rehn, A CORBA commodity grid kit, Concurrency and Computation: Practice and Experience 14 (13-15) (2002) 1057-1074.
[11] Commodity Grid Forum, http://www.globus.org/cog/ (2002)