Applied Workflows in Geodise

e-Science Workflow Services Workshop
Edinburgh (Dec 3rd – 5th 2003)

Dec 4th 2003
Prof Simon Cox

Computational Engineering & Design Group
School of Engineering Sciences
University of Southampton
Grid Enabled Optimisation and Design Search for Engineering (GEODISE)

Southampton, Oxford and Manchester

Simon Cox- Grid/ W3C Technologies and High Performance Computing
Global Grid Forum Apps Working Group

Andy Keane- Director of Rolls Royce/ BAE Systems University Technology Partnership in Design Search and Optimisation

Mike Giles- Director of Rolls Royce University Technology Centre for Computational Fluid Dynamics

Carole Goble- Ontologies and DARPA Agent Markup Language (DAML) / Ontology Inference Language (OIL)

Nigel Shadbolt- Director of Advanced Knowledge Technologies (AKT) IRC

BAE SYSTEMS- Engineering
Rolls-Royce- Engineering
Fluent- Computational Fluid Dynamics
Microsoft- Software/ Web Services
Intel- Hardware
Compusys- Systems Integration
Epistemics- Knowledge Technologies
Condor- Grid Middleware

The GEODISE Team ...

- Richard Boardman
- Sergio Campobasso
- Liming Chen
- Mike Chrystall
- Trevor Cooper-Chadwick
- Simon Cox
- Mihai Duta
- Clive Emberey
- Hakki Eres
- Matt Fairman
- Mike Giles
- Carole Goble
- Ian Hartney
- Tracey Hunt
- Zhuoan Jiao
- Andy Keane
- Marc Molinari
- Graeme Pound
- Colin Puleston
- Nicola Reader
- Angus Roberts
- Mark Scott
- Nigel Shadbolt
- Wenbin Song
- Paul Smart
- Barry Tao
- Jasmin Wason
- Fenglian Xu
- Gang “Luke” Xue

Geodise will provide grid-based seamless access to an intelligent knowledge repository, a state-of-the-art collection of optimisation and search tools, industrial strength analysis codes, and distributed computing & data resources.

A few of my favourite things to do with workflows

- Create
- Retrieve
- Cut ‘n’ Shut
- Configure
- Execute
- Monitor
- Share
- Steer
- Dynamically modify
When is a script not a script?

... when it’s a “workflow”
Scripting languages

Why use scripting languages?
• Flexibility
• High-level functionality
• Quick application development
• Extend the user’s existing PSE
Example Script

hostname = 'pacifica.iridis.soton.ac.uk'
jobmanager = [hostname,'/jobmanager-fork']
rsl = '&(executable=/bin/date)(stdout=remote.txt)'

%Create a proxy certificate
gd_createproxy

%Submitting a globus job and returning handle
handle = gd_jobsubmit(rsl,jobmanager)

%Polling the job
gd_jobpoll(handle)

%Getting the standard output
gd_getfile(hostname,'remote.txt','local.txt');

%Print the output to screen
type('local.txt')
Aerodynamic Shape Optimisation using CFD

- Integration of CAD, mesh generation, and solver
  - Direct API access to CAD models
  - third-party standards based data exchange

- Robust mesh generation
  - Automatic mesh generation
  - Control over mesh properties

- Multi-fidelity models
  - Euler solver
  - Navier Stokes solver
Orthogonal Basis Function for Airfoil Design

- Parameterisation methods for airfoil:
  - mathematical functions
  - empirical basis functions
  - control points based curve fitting

- Orthogonal basis functions
  - unique mapping from parameter space to design space
  - for preliminary wing design
  - fewer number of design variables
  - different set of basis functions for different design task

Optimisation Workflow and Results

- Optimisation strategy:
  - Single optimisation method proved to be inefficient for practical problems, more complex strategies are required
  - Two-stage hybrid approach (Genetic algorithm + Gradient search)

- Surrogate modelling
  - CFD runs on complex configurations is too expensive (24hrs/run)
  - Surrogate modelling methods
    - Polynomial curve fitting
    - Stochastic method (DACE or Kriging)
    - Neural network
  - High-dimensional design space

- Combing response surface modelling (RSM)
  /two-stage approach
Workflow for aerodynamic shape optimisation using CAD, Gambit, and Fluent

- Problem definition
- Design of Experiment
- Response surface modelling
- Optimisation on Response surface
- Validation

ProEngineer CAD (Condor Pool)
Gambit Meshing (Globus Compute)
Fluent CFD (Globus Compute)
Response surface model and two-stage hybrid search using GA/Local tuning

- Response surface model
- two-stage hybrid search methods
- Comparison of Airfoil shape and pressure distribution
Engine Nacelle Optimisation

(problem definition)

Assumption: Noise radiated to ground reduces with increasing scarf angle

Objective function:
Total Pressure Recovery \(\left(\frac{p_{t_2}}{p_{t_1}}\right)\)

Design variables: Scarf Angle (degrees), Axial Offset (mm)

Conventional Inlet
Total Pressure Recovery (TPR) = \(\frac{p_{t_2}}{p_{t_1}}\)

Negative Scarf Inlet
Design of Experiment & Response Surface Modelling

Initial Geometry

DoE

CFD

Cluster Parallel Analysis

Build Database

RSM Construct

RSM Evaluate

Search Using RSM

Adequate?

Best Design

RSM Tuning

Defining the Objective Function

Design Variables
\[ x_1 = 0.5, \quad x_2 = 0.25 \]

Objective function
\[ y = 42 \]

“Bigger workflows are made from little workflows,
Little workflows are made from littler workflows,
And so on…”

Engine Nacelle Optimisation (3D)
(some results)

- Typical unstructured mesh used in the problem (left)
- Response surface model built for two design variables (right)
- The effect of other geometry parameters need to be investigated
Photonic Device Modelling

pitch=300nm

Bridge waveguide structure courtesy of Martin Charlton, Southampton Microelectronics Research Group.

REAL-THING (photo)

UNIT-CELL

PERIODICALLY TILED UNIT-CELLS

12-fold Symmetric Quasicrystals

- Based on tiling of dodecagons composed of squares and equilateral triangles
- Possesses 12 fold rotational symmetry
- Leads to a highly homogeneous band gap
CEM Simulation Results

50 radii

CompResource.1
CompResource.2
CompResource.3
CompResource.4
CompResource.1
...

50 frequencies

Photonic Crystal Response Surface /
Photonic Band Gap Map

The aim of this design optimisation process is to find a configuration which maximises the bandgap and minimizes energy loss.

Location, size & density of holes determine the optical bandgap.
The Device

- Core layer
- Buffer layer
- Substrate
- Cladding layer

Light coupled into the waveguide core

Tree pitch
The Script

geometry_optimise_EDIT.m
Scripting languages

Why use scripting languages?
- Flexibility
- High-level functionality
- Quick application development
- Extend the user’s existing PSE
Grid-Enabled Scripting Environment

- **Motivations:**
  - Flexible, transparent access to computational resources
  - Easy to use for engineers (and in widespread use)

- **Our Approach**
  - Matlab chosen as the hosting environment
    - Extend the user’s existing PSE
    - High-level functionality
    - Quick application development
    - … is our execution/enactment engine too
  - Computational resources exposed in the form of Matlab functions
    - Job submission to Globus server using Java Cog
    - Job submission to Condor pool via Web services interface
  - Integration of CAD, Mesh generation, and Fluent solver via the use of intermediate data format, often standard-based, or package-neutral
  - Hybrid search strategies to make the best use of different search methods

- Can also use Python, Jython, etc.
Geodise Architecture

Intelligent Support → Knowledge Services → Matlab (or Jython ...) → Integration & Scripting

Java / C# → Web Service Grid Service → Java / C#/.NET → .EXE/ Fortran/ Matlab Code

Building Blocks

Interface

# Grid-Enabled Toolkits in Matlab

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>gd_createproxy</td>
<td>Creates a Globus proxy certificate from the user’s credentials</td>
</tr>
<tr>
<td>gd_proxyinfo</td>
<td>Returns information about the user’s proxy certificate</td>
</tr>
<tr>
<td>gd_proxyquery</td>
<td>Queries whether a valid proxy certificate exists</td>
</tr>
<tr>
<td>gd_certinfo</td>
<td>Returns information about the user’s certificate</td>
</tr>
<tr>
<td>gd_destroyproxy</td>
<td>Destroys the local copy of the user’s Globus proxy certificate</td>
</tr>
<tr>
<td>gd_jobkill</td>
<td>Terminates the GRAM job specified by a job handle</td>
</tr>
<tr>
<td>gd_jobstatus</td>
<td>Returns the status of the GRAM job specified a job handle</td>
</tr>
<tr>
<td>gd_jobpoll</td>
<td>Queries the status of a Globus GRAM job until complete</td>
</tr>
<tr>
<td>gd_jobsubmit</td>
<td>Submits a GRAM job to a Globus server</td>
</tr>
<tr>
<td>gd_getfile</td>
<td>Retrieves a file from a remote host using GridFTP</td>
</tr>
<tr>
<td>gd_putfile</td>
<td>Transfers a file to a remote host using GridFTP</td>
</tr>
<tr>
<td>gd_archive</td>
<td>Stores a file in repository with associated metadata</td>
</tr>
<tr>
<td>gd_query</td>
<td>Retrieves metadata about a file based on certain criteria</td>
</tr>
<tr>
<td>gd_retrieve</td>
<td>Retrieves a file from the repository to the local machine</td>
</tr>
<tr>
<td>gd_sendtext</td>
<td>Sends a SMS text message to the specified mobile phone number</td>
</tr>
</tbody>
</table>

Scripting the optimisation workflow within Matlab
Workflow
<?xml version="1.0" encoding="UTF-8"?>
...
<Functions>
  <Testfunctions>
    <rand type="Instance">
      <inputs>
        <input1 type="int" name="n" value="5"/>
      </inputs>
      <outputs>
        <output type="matrix" name="A" value=""/>
      </outputs>
    </rand>
  </Testfunctions>
</Functions>
Workflow Tool (Part i)

- Retrieve Projects
- Composition Area
- Execution - Mapping to Resources
- Functions
- Monitoring and Steering

A few of my favourite things to do with workflows- review

- ✓ Create
- ✓ Retrieve
- ✓ Cut ‘n’ Shut
- ✓ Configure
- ✓ Execute
- ✓ Monitor
- Share
- Steer
- Dynamically modify
Here's the matlab script you asked for. The script is called sim_pqc.m. 'help sim_pqc' will tell you how to use it, I hope the instructions are clear enough. It requires lattice.m and limit.m in your matlab path to run. The density of states will is saved in DOS.mat. everything must be in the same directory. I also included the FDTD and related programs (I had to modify some of them to run in batch mode). Any questions email or phone me at Meso.

Cheers

Tom

----------

Tom Lee
University of Southampton
Department of Electronics and Computer Science
Microelectronics group
Sharing (ii)

Simon,

Sorry, I found a small mistake in sim_pqc.m
Here's the new one

Cheers

tom

P.S. “Provenance” too
Semantics & Knowledge
Semantic Grid in e-Science

• Bridging the gap:
  – Grid: seamless access to distributed computation and data resources
  – e-Science: distributed collaboration and reuse of knowledge and resources
  – Semantic Grid: Semantic Web technology applied on Grid application

• Building ontology and semantically enrich resource for reuse and management
  – Resource is semantically meaningful
  – Expressed using a standard conceptualization
  – Which is well recognized within a specific community of practice.
Knowledge Technologies

Six challenges define the Life Cycle:

Acquire • Model • Reuse • Retrieve • Publish • Maintain knowledge
Knowledge Acquisition (KA)

Knowledge sources
Domain experts, software manuals & textbooks.

KA techniques
Interview, protocol analysis, concept sorting etc.

Tools used
PC-PACK integrated knowledge engineering toolkit

Knowledge acquired
EDSO domain knowledge, EDSO processes and problem definition
Knowledge Modelling

Techniques
CommonKADS knowledge engineering methodologies.

Knowledge models
Organization, agent & task templates, domain schema inference rules.

Tools used
PC-PACK integrated knowledge engineering toolkit

Deliverables
Knowledge web in HTML, XML and UML, Conceptual task model, EDSO process flowchart
Ontologies

- common conceptualisation of a domain -
Semantic Workflow support in Geodise

• Ontology modelling
  – Definition: Domain conceptualisation that collect a controlled set of vocabulary and their relationship through hierarchy and explicitly expressed properties.
  – Examples: User profile ontology, Problem profile ontology, Task ontology, etc.

• Instance generation
  – Definition: Semantic enriching instances by referencing to ontology files
  – Methods: annotation content with ontology, populating ontology with content

• Semantic consumption
  – Ontology driven instance querying
  – Ontology driven from generation
    - task configuration
    - Problem setup
  – Ontology assisted domain script editing
  – Service oriented workflow composing – querying semantic enriched service component.
Workflow Tool
(… with added semantics)
Ontology Development (1)

- Tools
  - Protégé & OilEd Editor
- Representation
  - DAML+OIL & CLIPS
- Deliverables
  - EDSO domain ontology
  - EDSO task ontology
  - Mesh generation tool (Gambit software) ontology
  - User-profile ontology
Ontology Development (2)

- Ontology Views
  - DL ontologies (DAML/OWL)
  - Simplified views
  - Tailored to specific domains

- Ontology Views
  - Underlying complexity hidden

- Ontology editing by...
  - Knowledge engineers
  - Domain experts
Ontology Services

- Facilitating ontology sharing & reuse
  - Ontology service APIs
- Domain independence
  - DAML+OIL/OWL standards
- Soap-based web services - WSDL
- Java, Apache Tomcat & Axis technologies

This page gives information about the ontology services (OS), including the purpose of the OS, interface specification, where it is and how to use it in your applications.

The purpose of Ontology Services

The ontology services provide Java API tools for common ontological operations, such as subsumption checking, retrieving definitional information, navigating concept hierarchies, and retrieving lexical information. By adopting the emerging web ontology standard, DAML+OIL as its underlying representation language, the services can be used with anybody else's ontologies and metadata repositories as long as they're accessible via URLs. As a result of this, the ontology services become perhaps more of an ontology gateway.

Ontology Service API Specification

Ontology service APIs include CGI interface and SOAP interface. They are described in detail at OS API Specification.

Where it is

Ontology services are housed temporarily in the following server:

- CGI interface and tester page: http://zuu304 requesting_host.ac.uk:8080/
- SOAP interface: http://zuu304 requesting_host.ac.uk:8080/axis/services/axis?wsdl

How to use it

SOAP-based ontology services can be used in the same way as any web service by creating proxy classes and then calling its functions. For further information about web service, please refer to relevant websites.
Ontology Assisted Domain Script Editor

- Pre-defined command syntax ontology with Gambit command syntax instances
- Semantic rich instances being consumed in the editor
- Syntax colorizing and auto-completion
Ontology Driven Forms in Geodise -1

• Setting up problems (a scenario using JaxFront)
Ontology Driven Forms in Geodise -2

- Configuring tasks in Workflow Composing Environment
Exploiting Knowledge in Geodise
Knowledge Application 1:

Create Semantic Content

• **Goals**
  - Machine understandable information
  - Facilitate sharing & reuse

• **Techniques & tools**
  - OntMat-annotizer
  - Geodise Ontologies

• **Example**
  - OPTIONS log-files annotation
Knowledge Application 2: Ontology-assisted Workflow Management

• Features:
  – Function selection
  – Function instantiation
  – Database schema
  – Semantic instances
  – Semantic workflow

• Technologies:
  – EDSO ontologies & ontology services
  – Java JAX-RPC, DOM/SAX
Knowledge Application 3: Knowledge-based Design Advisor

• **Features**
  – Context-sensitive advice
  – Advice at multi-levels of granularity (process, task …)
  – KBSs as knowledge services

• **Technologies**
  – Knowledge engineering
  – EDSO ontologies
  – Rule-based reasoning techniques
Intelligent Workflow Monitoring and Advice
(“rule-based to case-based in real-time”)

• Updating constructed workflow using rule-base
  – At run-time:
    - find ‘similar’ workflows to the one constructed
    - is this one performing ‘as expected’? Might a different workflow outperform current one?
  – Resolution: Perhaps problem is anomalous?
    - Change method/ modify workflow?
    - Feedback to expert … update rule-base?

• Exploiting new components in workflows
  – Example
    - New optimisation method added in semantically consistent way
    - Workflows constructed (by expert) with new method, … and then:
  – ‘similar workflow’ search above will find workflows with new method in:
    - Might they outperform the currently constructed workflow?
    - Substitute new method into constructed workflow?
    - Feedback to expert … update rule-base?
A few of my favourite things to do with workflows- review (ii)

- ✓ Create
- ✓ Retrieve
- ✓ Cut ‘n’ Shut
- ✓ Configure
- ✓ Execute
- ✓ Monitor
- ✓ Share
- ✓ Steer
- ✓ Dynamically modify
Questions
Example Script

hostname = 'pacifica.iridis.soton.ac.uk'
jobmanager = [hostname,'/jobmanager-fork']
rsl = '&(executable="/bin/date")(stdout="remote.txt")'

%Create a proxy certificate
gd_createproxy

%Submitting a globus job and returning handle
handle = gd_jobsubmit(rsl,jobmanager)

%Polling the job
gd_jobpoll(handle)

%Getting the standard output
gd_getfile(hostname,'remote.txt','local.txt');

%Print the output to screen
type('local.txt')