

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*

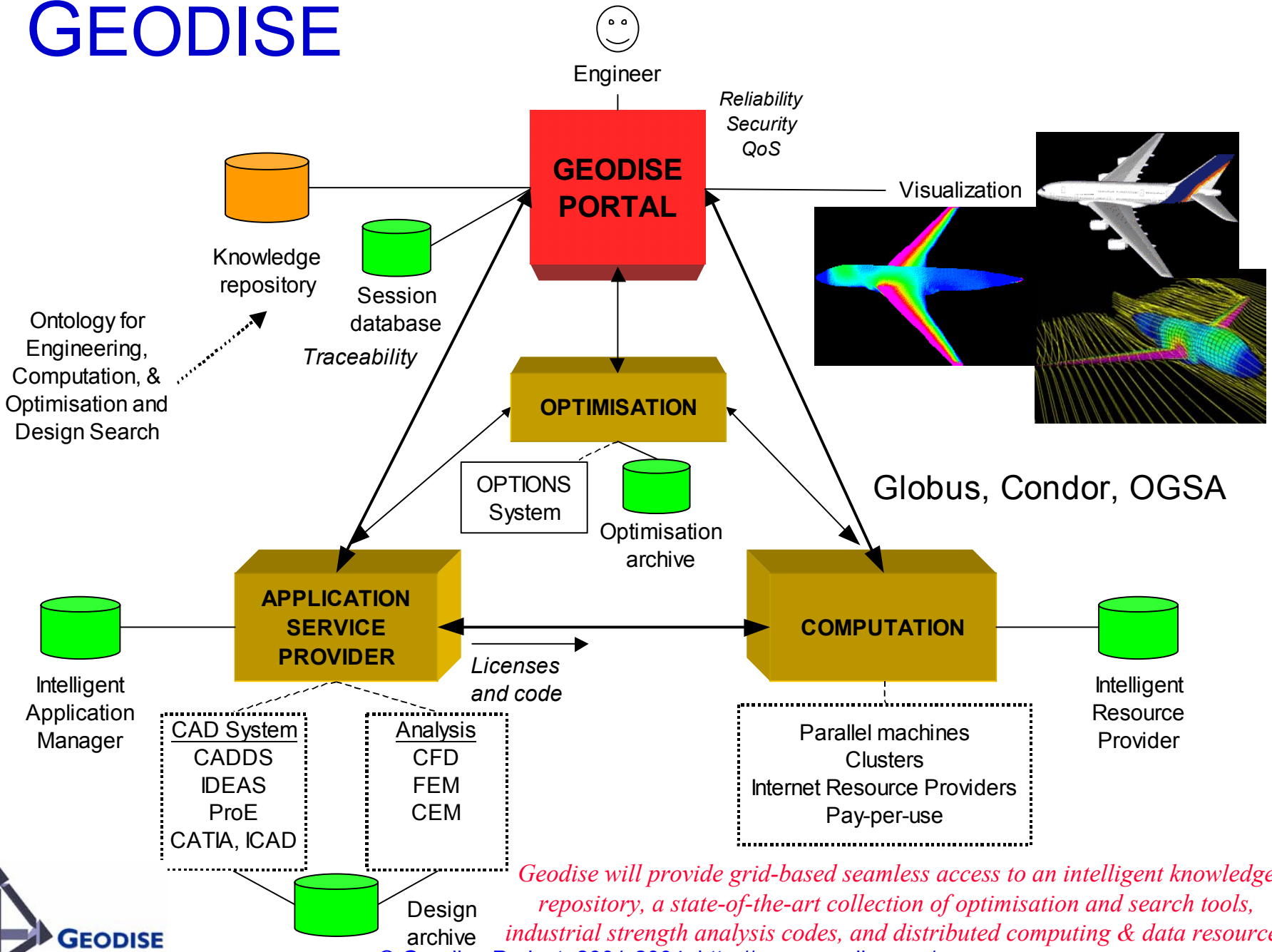


The GEODISE Team ...

- 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



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

© Geodise Project, 2001-2004. <http://www.geodise.org/>

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?

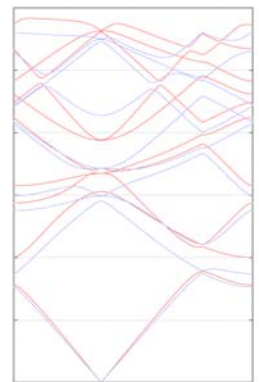
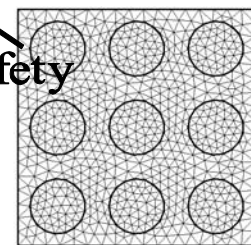
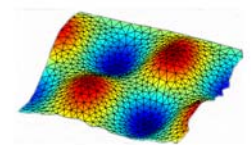
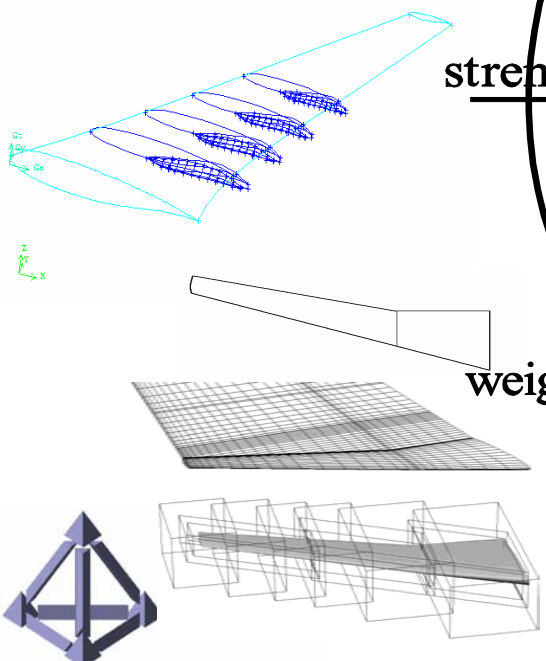
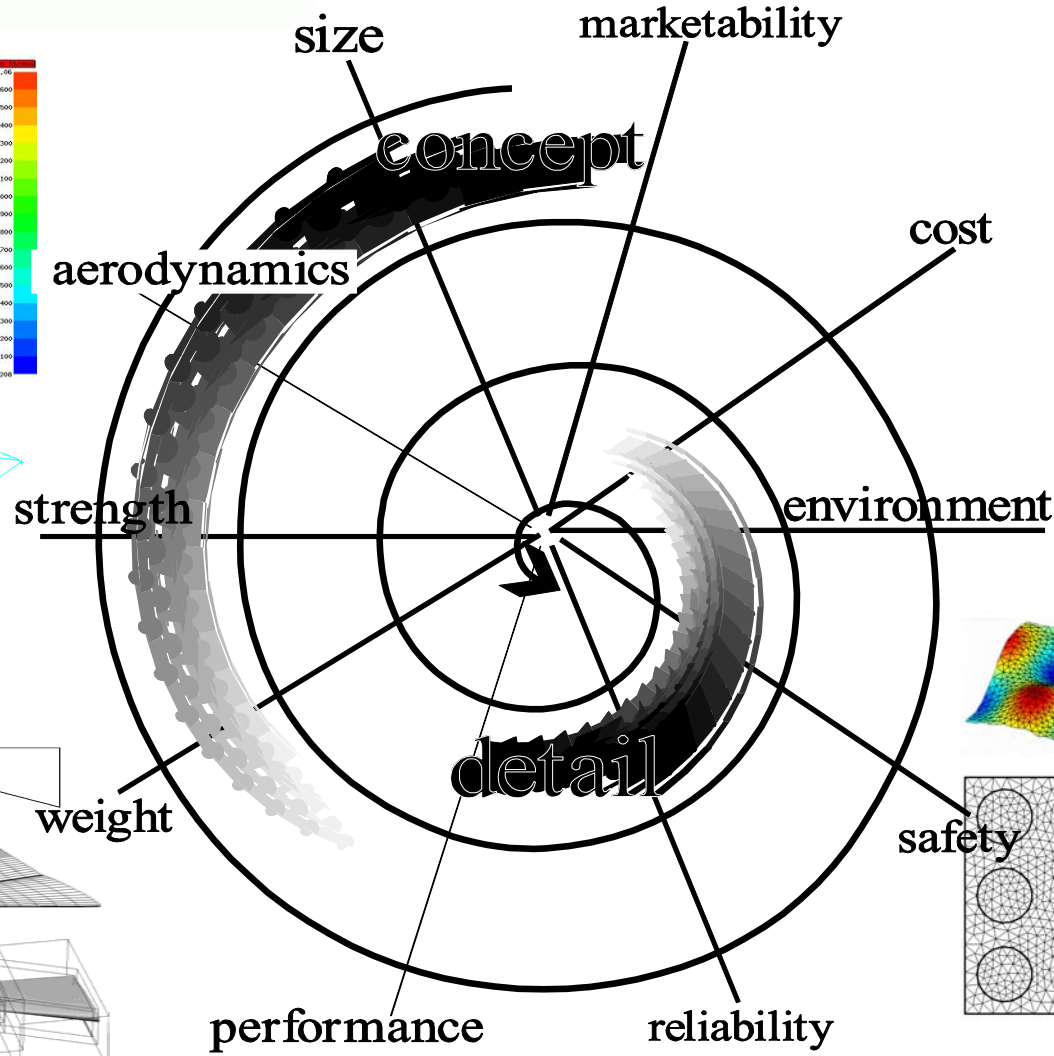
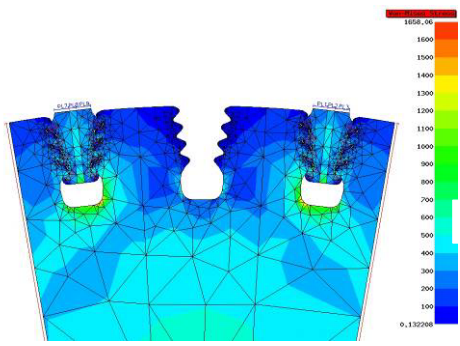
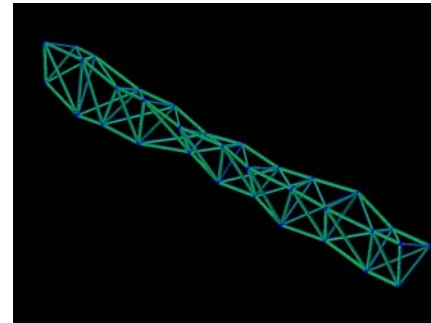
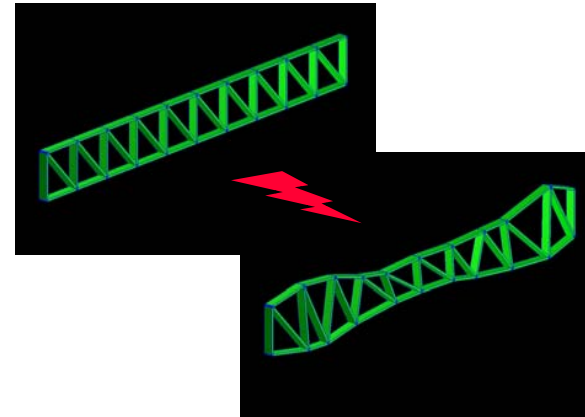
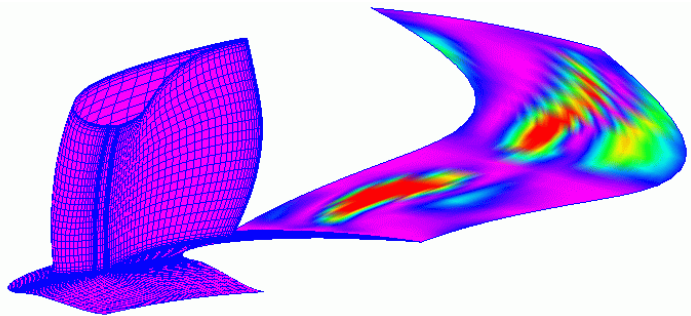
- Flexibility
- High-level functionality
- Quick application development
- Extend the user's existing PSE



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')
```


Design



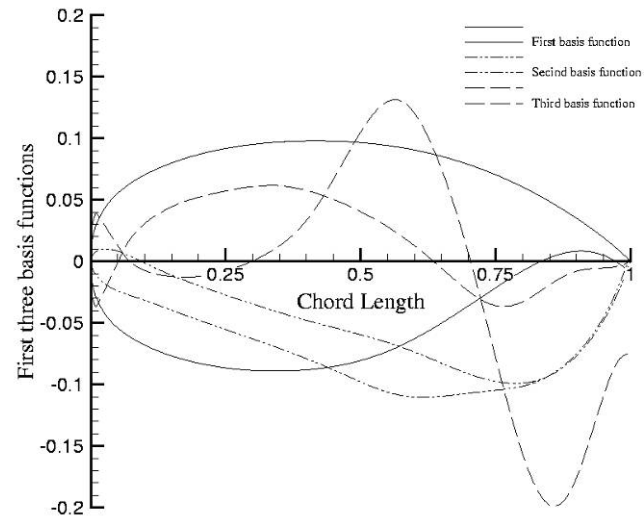
Aerodynamic Shape Optimisation using CFD

- 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

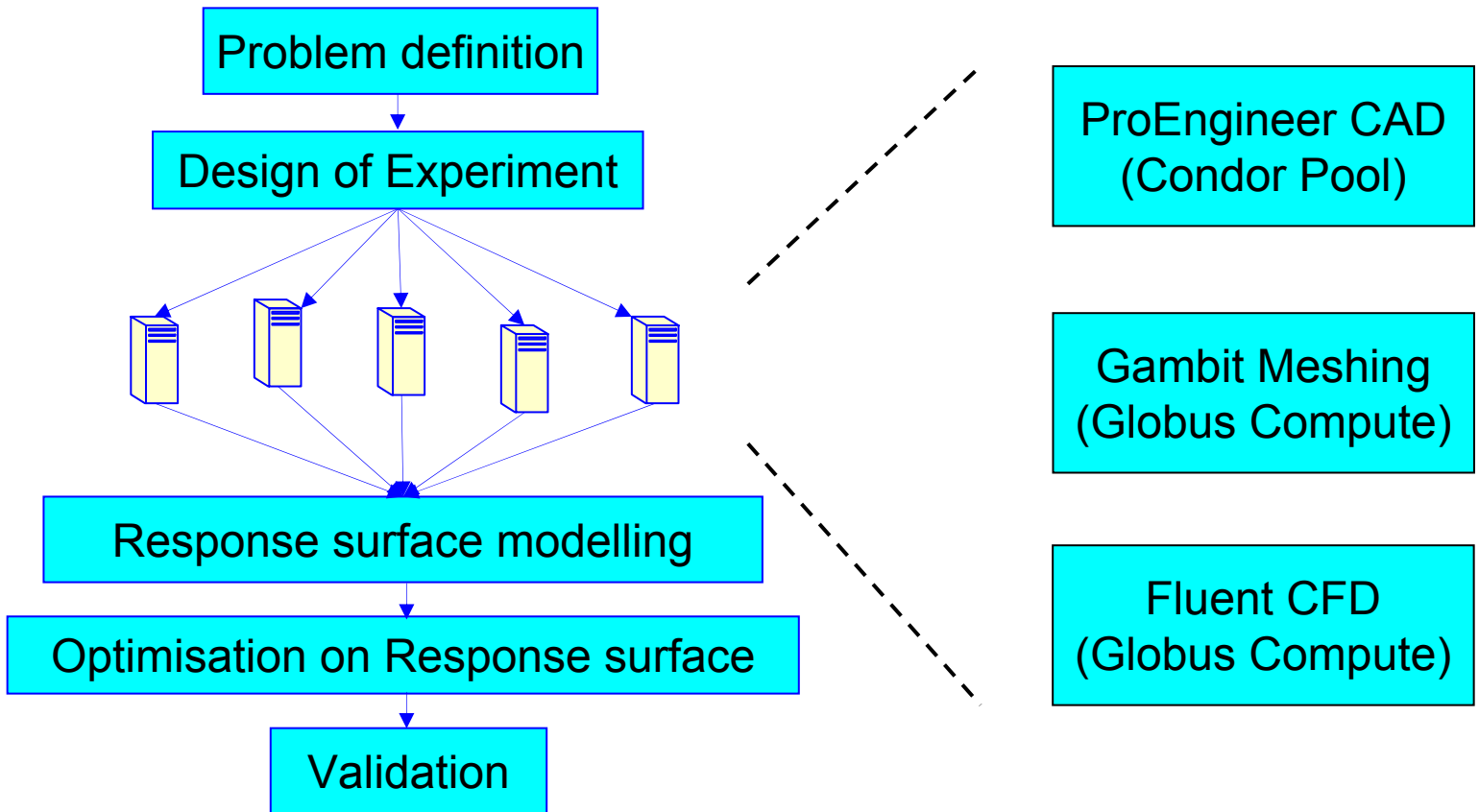


Robinson, G.M. and Keane, A.J., “Concise Orthogonal Representation of Supercritical Airfoils”, Journal of Aircraft 38(3) (2001) 580-583.

Optimisation Workflow and Results

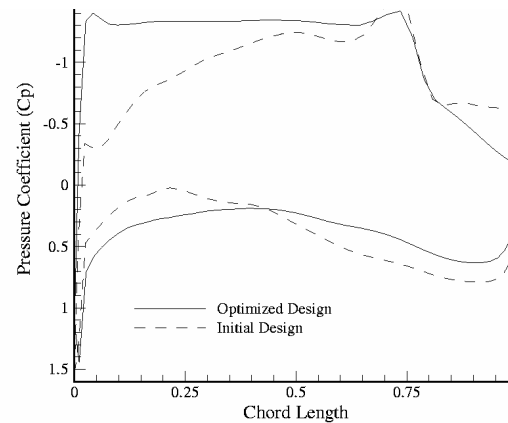
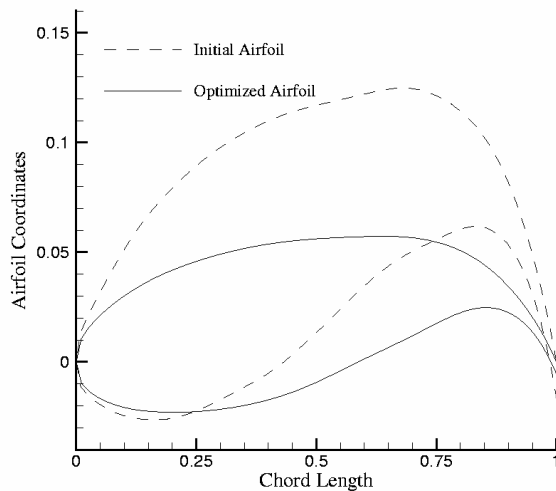
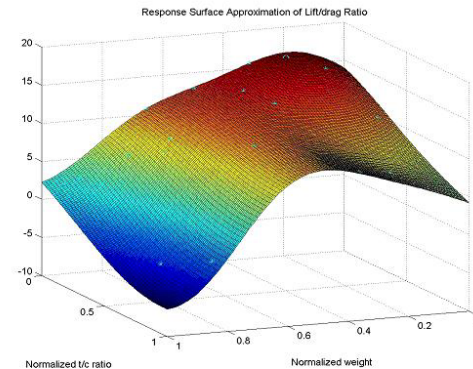
- Optimisation strategy:
 - Single optimisation method proved to be inefficient for practical problems, more complex strategies are required
 - Two-stage hybrid approach (Genetic algorithm + Gradient search)
- Surrogate modelling
 - CFD runs on complex configurations is too expensive (24hrs/run)
 - Surrogate modelling methods
 - Polynomial curve fitting
 - Stochastic method (DACE or Kriging)
 - Neural network
 - High-dimensional design space
- Combining response surface modelling (RSM)
/two-stage approach

Workflow for aerodynamic shape optimisation using CAD, Gambit, and Fluent



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

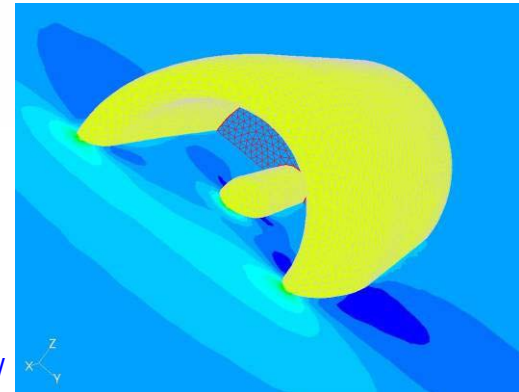
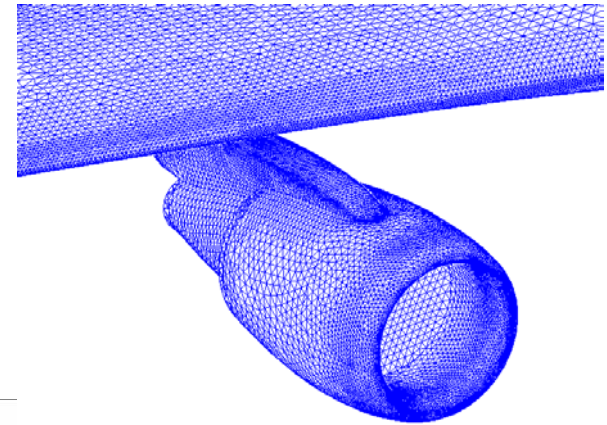
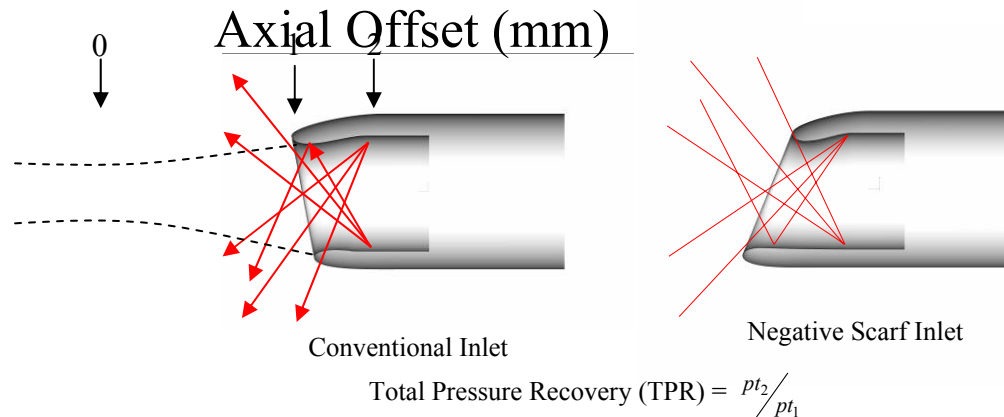
(problem definition)

Assumption: Noise radiated to ground reduces with increasing scarf angle

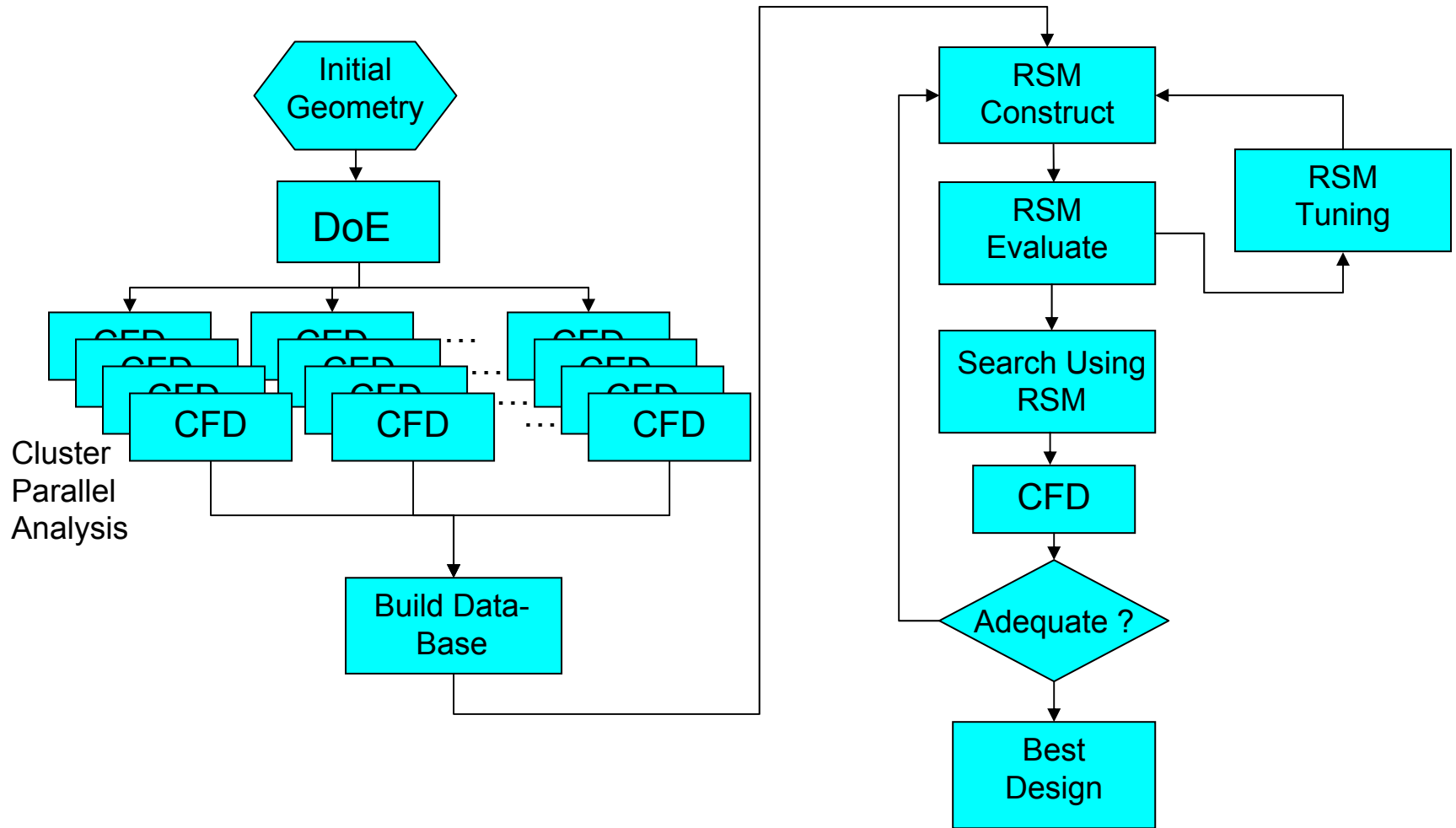
Objective function:

Total Pressure Recovery (p_{t2}/p_{t1})

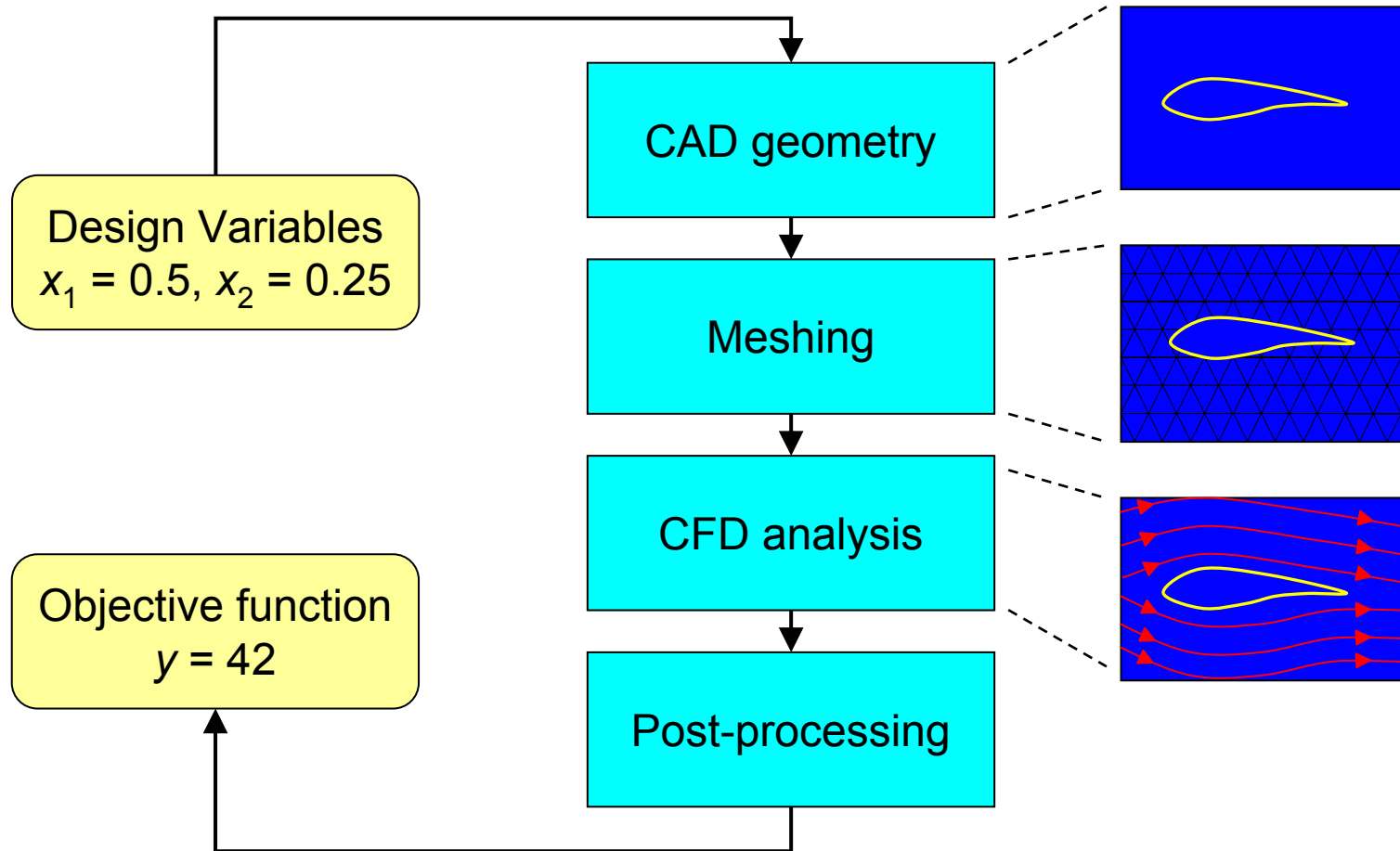
Design variables: Scarf Angle (degrees)



Design of Experiment & Response Surface Modelling



Defining the Objective Function

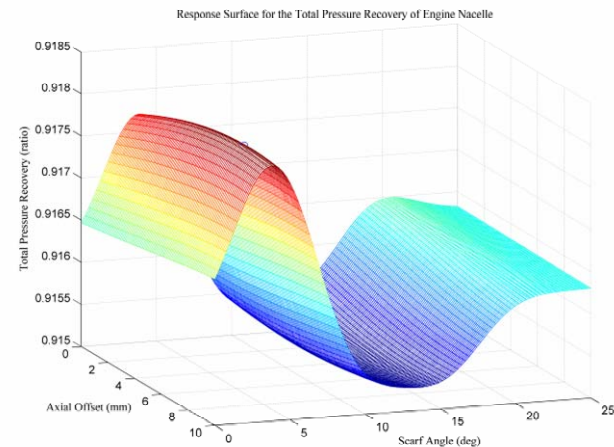
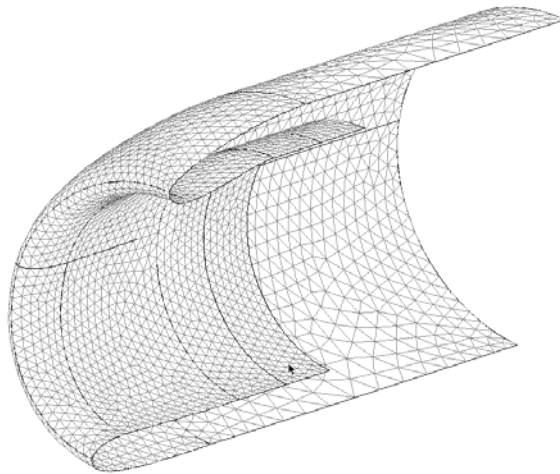


***“Bigger workflows are made from little workflows,
Little workflows are made from littler workflows,
And so on...”***

Engine Nacelle Optimisation (3D)

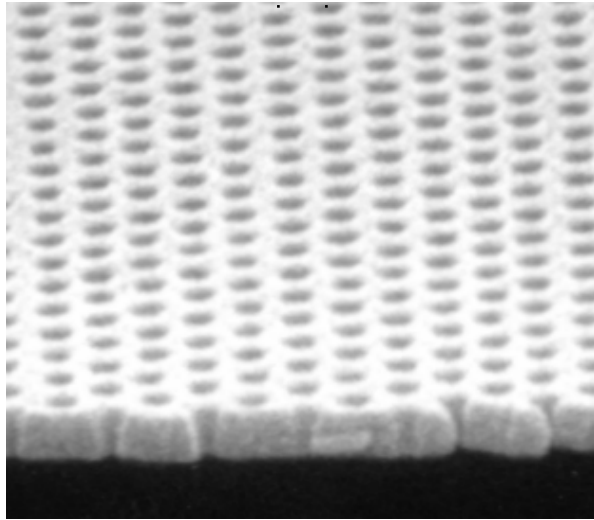
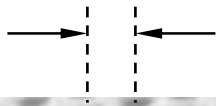
(some results)

- 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

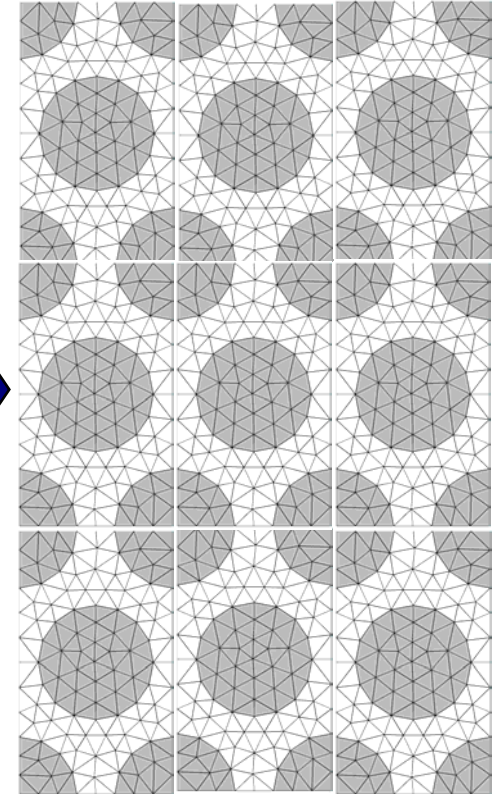
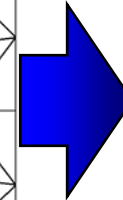
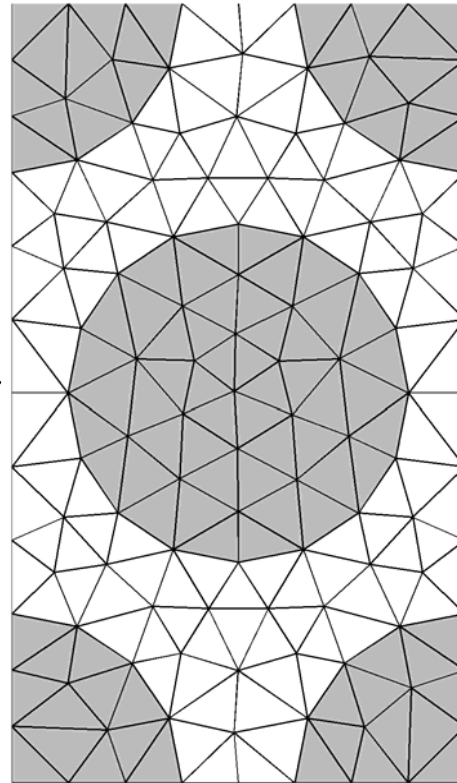
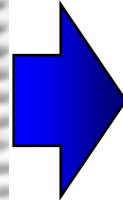


Photonic Device Modelling

pitch=300nm



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



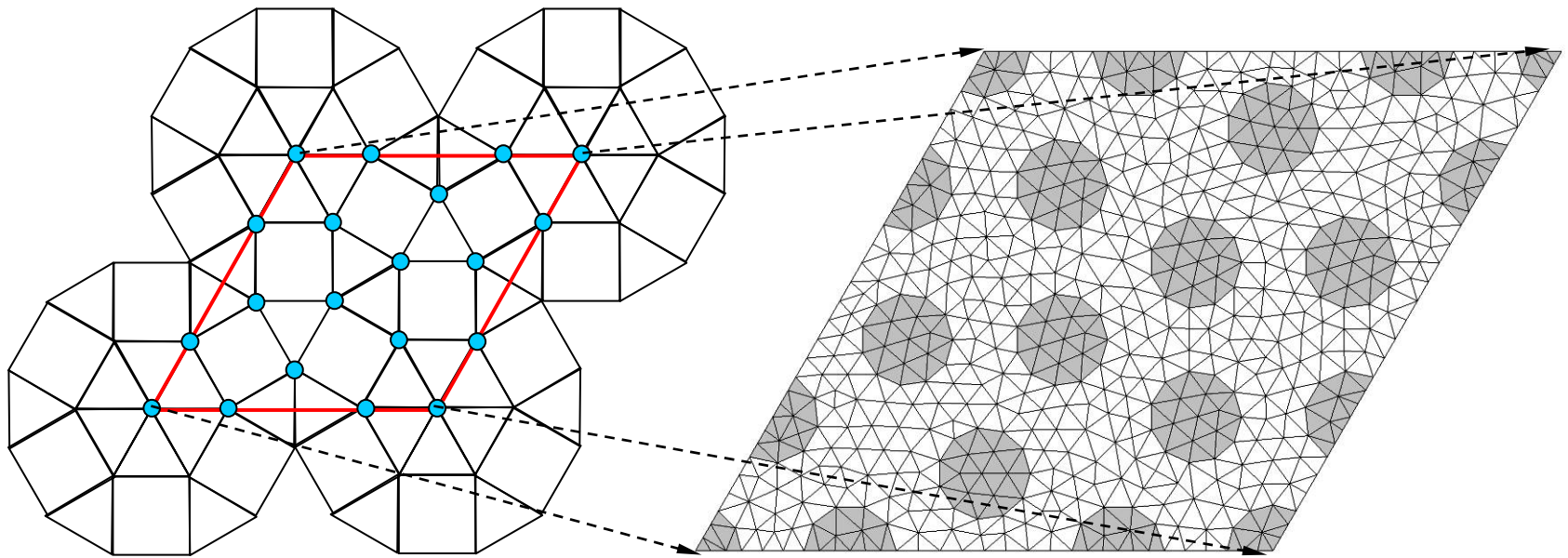
REAL-THING
(photo)

UNIT-CELL

PERIODICALLY
TILED UNIT-CELLS

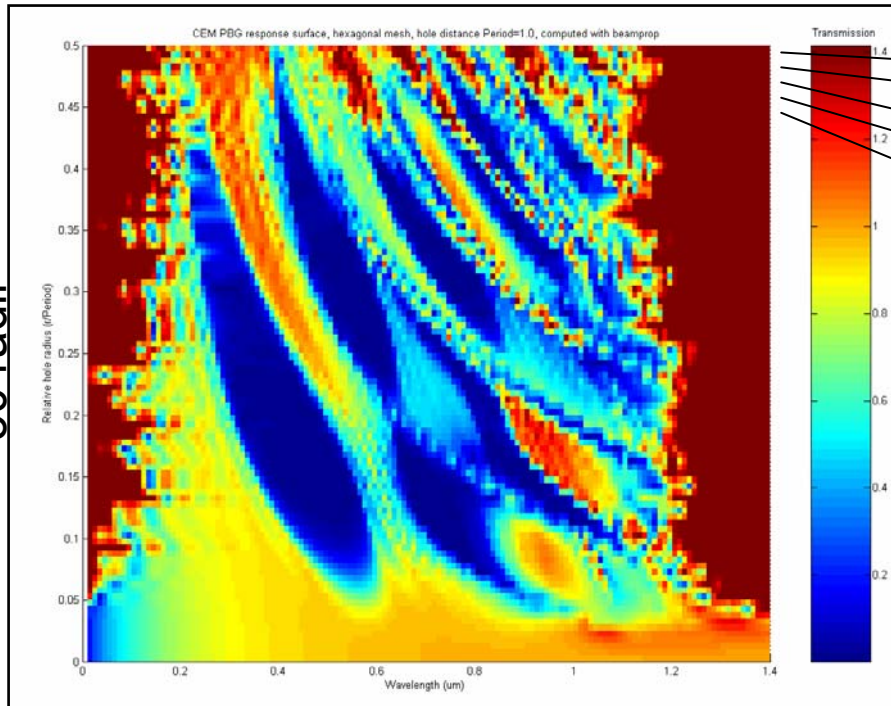
12-fold Symmetric Quasicrystals

- 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

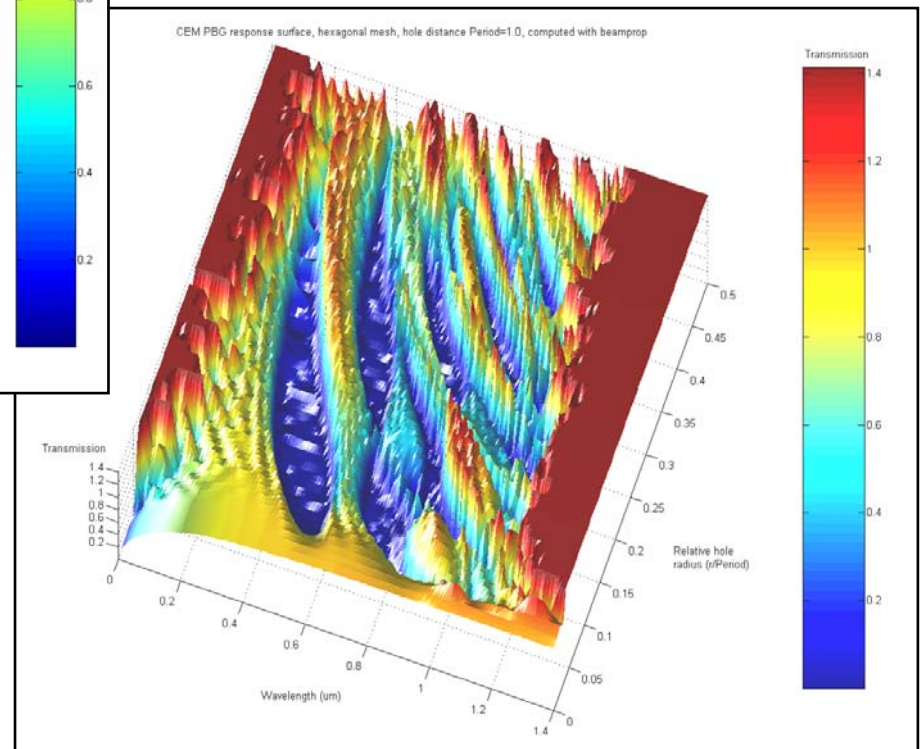
50 radii



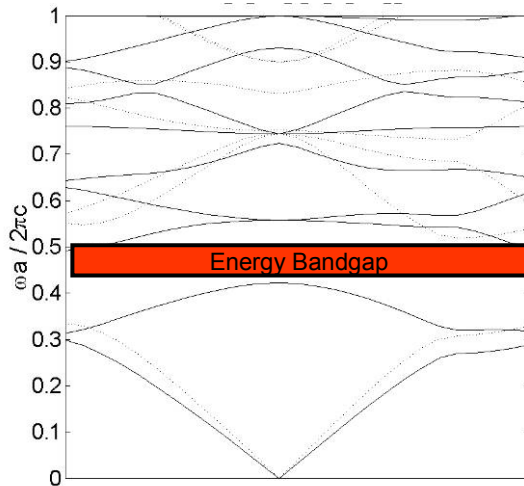
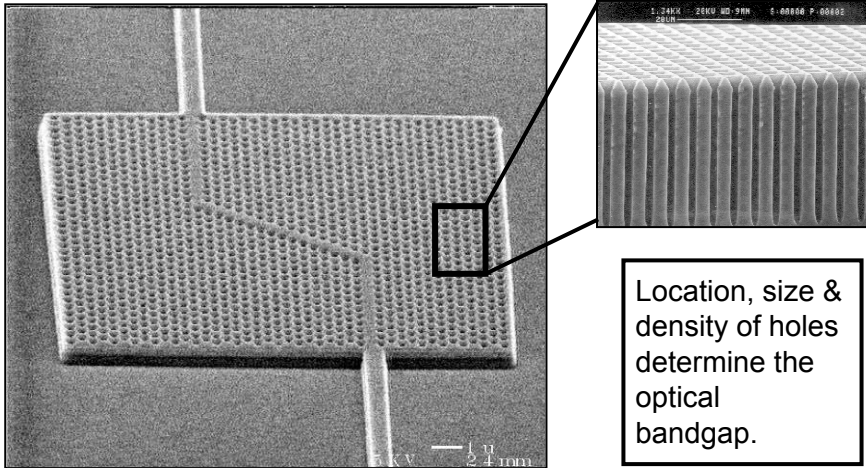
- CompResource.1
- CompResource.2
- CompResource.3
- CompResource.4
- CompResource.1
- ...


50 frequencies

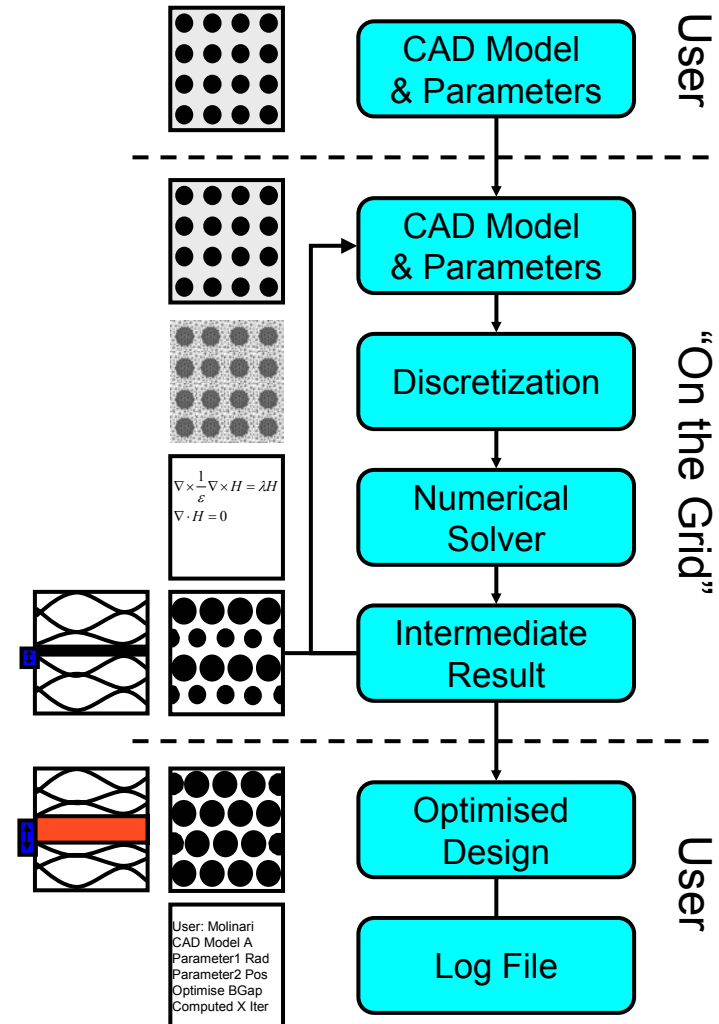
Photonic Crystal Response Surface /
Photonic Band Gap Map



Photonic Crystal - Optimisation

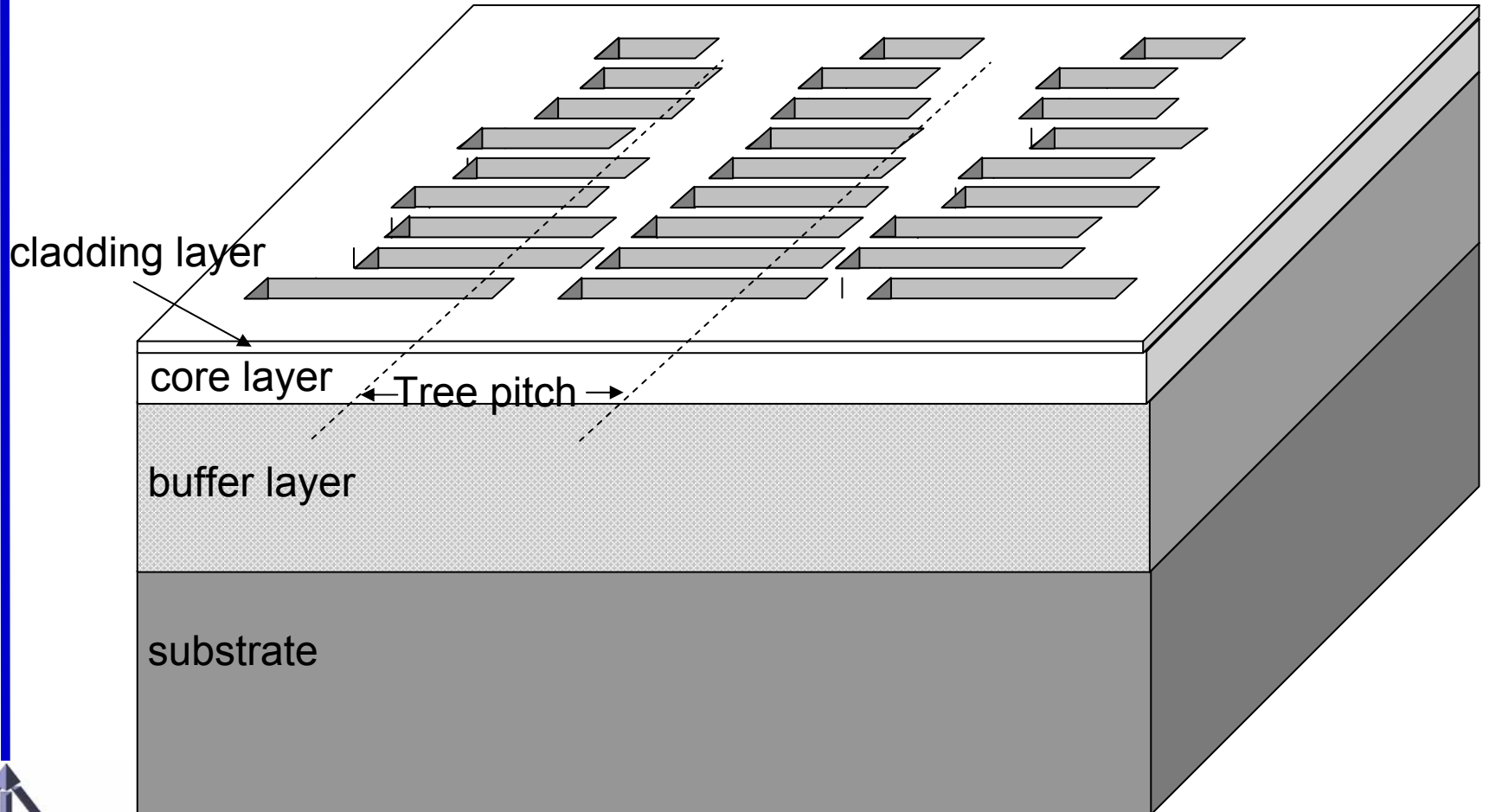
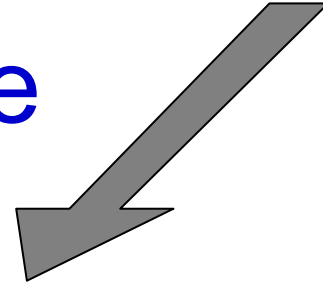



 The aim of this design optimisation process is to find a configuration which maximises this bandgap and minimizes energy loss.



The Device

light coupled into
the waveguide core



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

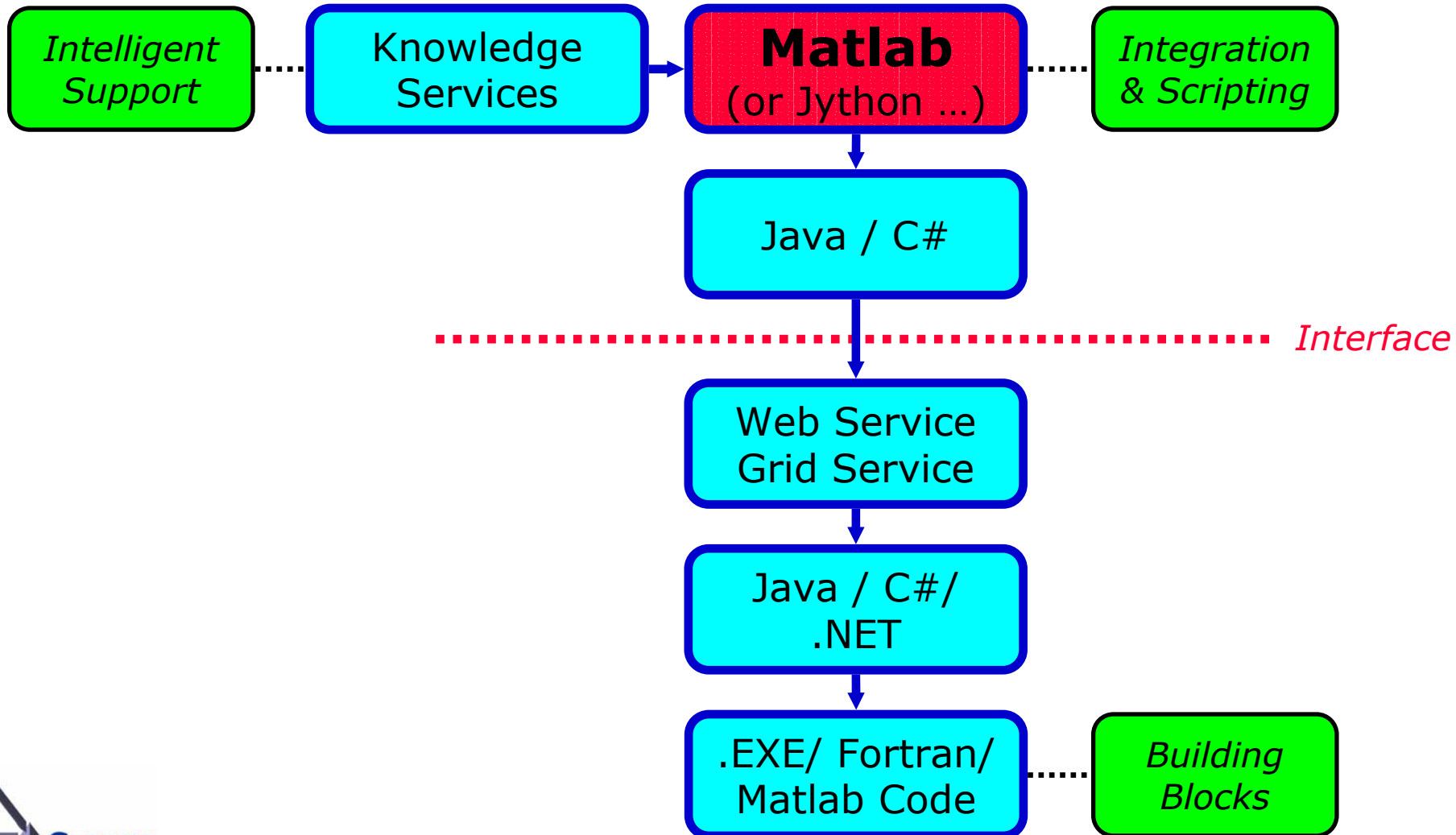


Matlab Grid-Enabled Scripting Environment

- 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



Grid-Enabled Toolkits in Matlab

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

Pound, G.E., Eres, M.H., Wason, J.L., Jiao, Z., Cox, S.J., and Keane, A. J., "A Grid-enabled Problem Solving Environment (PSE) for Design Optimisation within Matlab", 17th International Parallel and Distributed Processing Symposium (IPDPS 2003) 22-26 April 2003, Nice, France, 2003

Scripting the optimisation workflow within Matlab

```
MATLAB
File Edit View Web Window Help
Current Directory: D:\geodise2\orthfoil

>> % import java class
import org.geodise.optimisation.*;
% problem definition
nvars = 2;
lvars = [-1, 0.06];
uvars = [1, 0.18];
vars = [0.3, 0.12];
% Options data structure for orthfoil problem
orthfoilRSM = JavaOptData(nvars, vars, lvars, uvars);
% JavaOptions
opt = JavaOptions;
opt.Load(orthfoilRSM);
niters = 15;
ncons = 1;
vars = [0.154185145, 0.146553658, -0.251986064, 0.134932023, 0.538988307, 0.177126053, 0.044716965, 0.062852334, -0.541704384, 0.122527654, 0.941380805, 0.165200035, -0.455053, 9.736681192, 10.74828461, 1.096400541, 12.85513197, 10.61839788, 5.582960856, 12.07712295, 8.01420247, 3.852191454, 13.66563896, 8.578822294, 5.875477296, 12.71259167, ...];
objs = zeros(1,15);
lcons = -1*ones(1,15);
ucons = ones(1,15);

% fill in the data
opt.optfill(niters, nvars, ncons, vars, cons, objs, lcons, ucons);
% find hyperparameters
orthfoilRSM.objmod=4.1;
orthfoilRSM.conmod=2.1;
opt.optrss(orthfoilRSM);
% plot the response surface
npoints=100;
orthfoilRSM.update=0;
orthfoilRSM.omethd=0;
orthfoilRSM.niters=1;
x1=[orthfoilRSM.lvars(1):(orthfoilRSM.uvars(1)-orthfoilRSM.lvars(1))/(npoints-1):orthfoilRSM.uvars(1)];
x2=[orthfoilRSM.lvars(2):(orthfoilRSM.uvars(2)-orthfoilRSM.lvars(2))/(npoints-1):orthfoilRSM.uvars(2)];
for i=1:npoints
    orthfoilRSM.vars(1)=x1(i);
    for j=1:npoints
        orthfoilRSM.vars(2)=x2(j);
        opt.Search(orthfoilRSM);
        obj(j,i)=orthfoilRSM.objfn;
    end
end
mesh(x1,x2,obj);
orthfoilRSM.niters=100;
orthfoilRSM.omethd=4.0;
opt.Search(orthfoilRSM);

Start
```

Workflow

Workflow Editor

```

<?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>
  
```

The Workflow Editor interface consists of several panels:

- Components View:** A tree view on the left showing a hierarchy of tasks and sub-tasks, including 'Tasks', 'ObjectiveFunction', 'Optimisation', and 'Beam'.
- Workflow Graph:** A central area showing a sequence of tasks connected by arrows: 'compile_executables', 'generate input file', 'remove subdirectories', 'generate sample points', 'parameter search', and 'check jobs'.
- generate_sample_points Properties:** A dialog box with 'Inputs' and 'Outputs' tabs. The 'Inputs' tab shows a table of parameters:

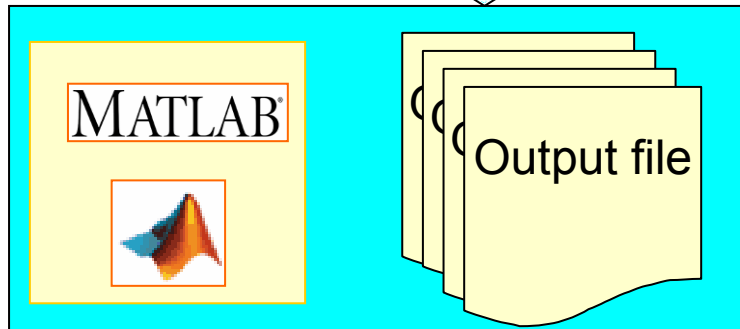
Type	Name	Default Value
double	lower_bound_xd	2.5
double	upper_bound_xd	3.5
double	lower_bound_yd	1.5
double	upper_bound_yd	2.5
int	number_of_grids_x	3
int	number_of_grids_y	3

The 'connection12 Properties' dialog box has a 'Mapping' tab with three columns:

- inputs:** sample_point, number_of_p, bounds, grids
- outputs:** server, number_of_s, number_of_p, sample_point, ldirectory
- preferred outputs:** sample_points=s, number_of_point

Navigation arrows are present between the columns, and 'Ok' and 'Cancel' buttons are at the bottom.

Submit to Matlab



Workflow Tool (Part i)

Retrieve

Projects

The screenshot shows the Workflow Editor interface. On the left is the 'Components View' tree with categories like Tasks, Optimisation, Testfunctions, and GA. The main 'Composition Area' displays a workflow graph with four steps: 'generate_sample_points', 'parameter_search', 'check_jobs', and a partially visible 'collect_data' step. Below the graph is a table of 'Compute Resources'.

Property	Host Name	Execute Directory	Work Directory	State
globoServers				
objserver2	artemis.sesnet.soton.ac.uk		/home/eres/Soton/Beam	✓
objserver3	escience-dept2.sesnet.soton.ac.uk		/home/eres/Soton/Beam	✓
objserver4	panda.sesnet.soton.ac.uk		/home/eres/Soton/Beam	✓
objserver5	pablo.sesnet.soton.ac.uk		/home/eres/Soton/Beam	✓
objserver6	pacific.iridis.soton.ac.uk		/home/eres/Soton/Beam	✓
objserver7	blue02.iridis.soton.ac.uk		/scratch/eres/Soton/Beam	✓
objserver8	blue07.iridis.soton.ac.uk		/scratch/eres/Soton/Beam	✓
objserver9	tempo.amtp.cam.ac.uk		/home/eres/Soton/Beam	✓
objserver10	herschel.amtp.cam.ac.uk		/home/eres/Soton/Beam	✓
objserver11	pioneer.lesc.doc.ic.ac.uk		/homes/mhe/Soton/Beam	✓
objserver12	viking.lesc.doc.ic.ac.uk		/homes/imha/Soton/Beam	✓

Composition Area

Execution - Mapping to Resources

Functions

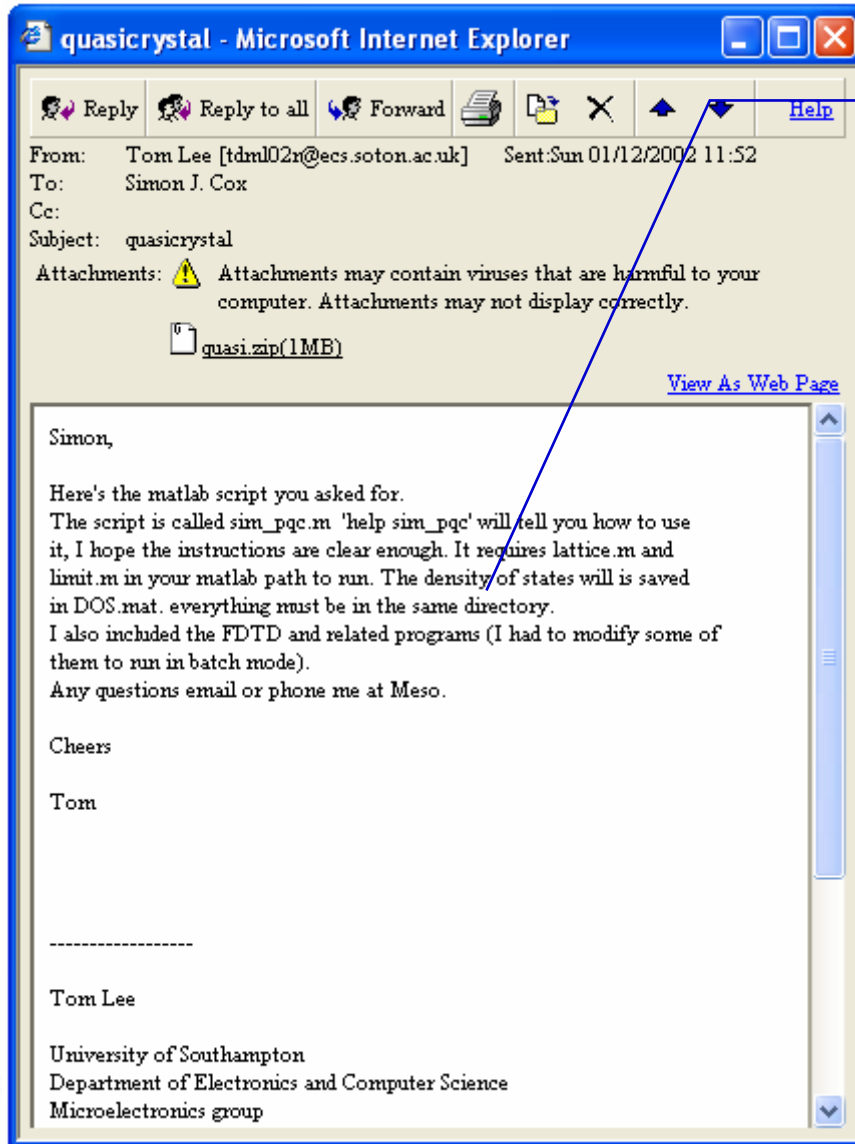
Monitoring and Steering



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

- ✓ Create
- ✓ Retrieve
- ✓ Cut 'n' Shut
- ✓ Configure
- ✓ Execute
- ✓ Monitor
- Share
- Steer
- Dynamically modify

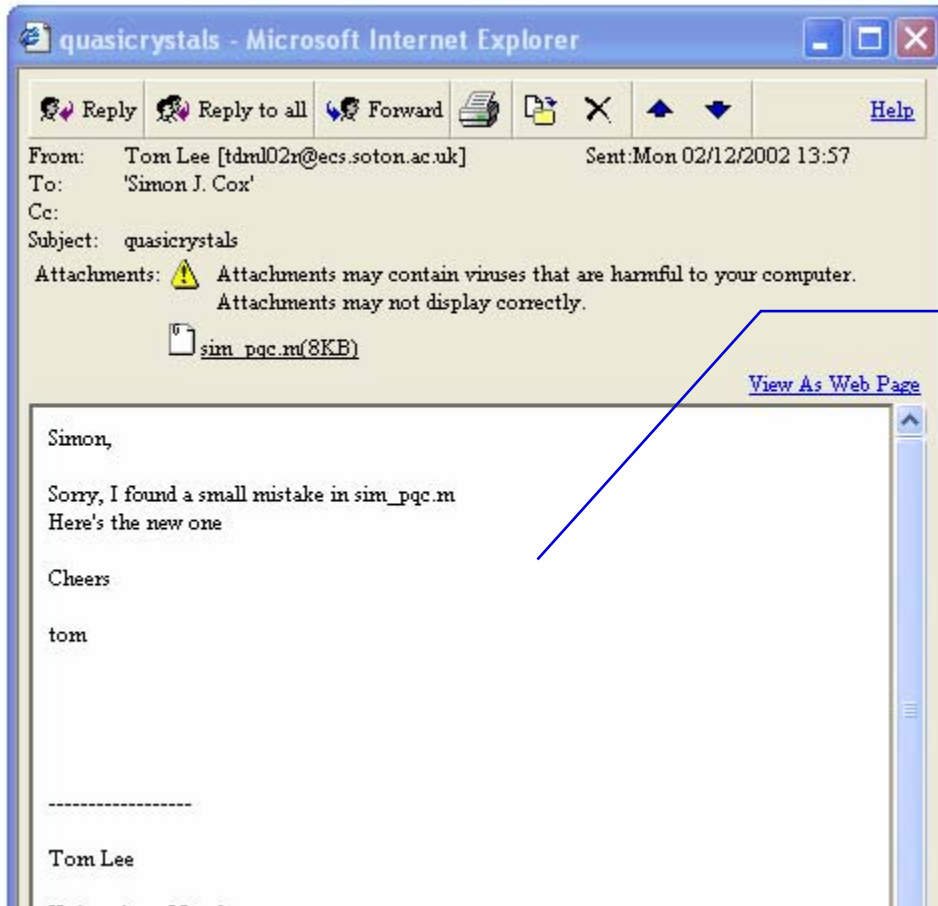
Sharing (i)



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 be 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.



Sharing (ii)



Simon,

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

Cheers

tom

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

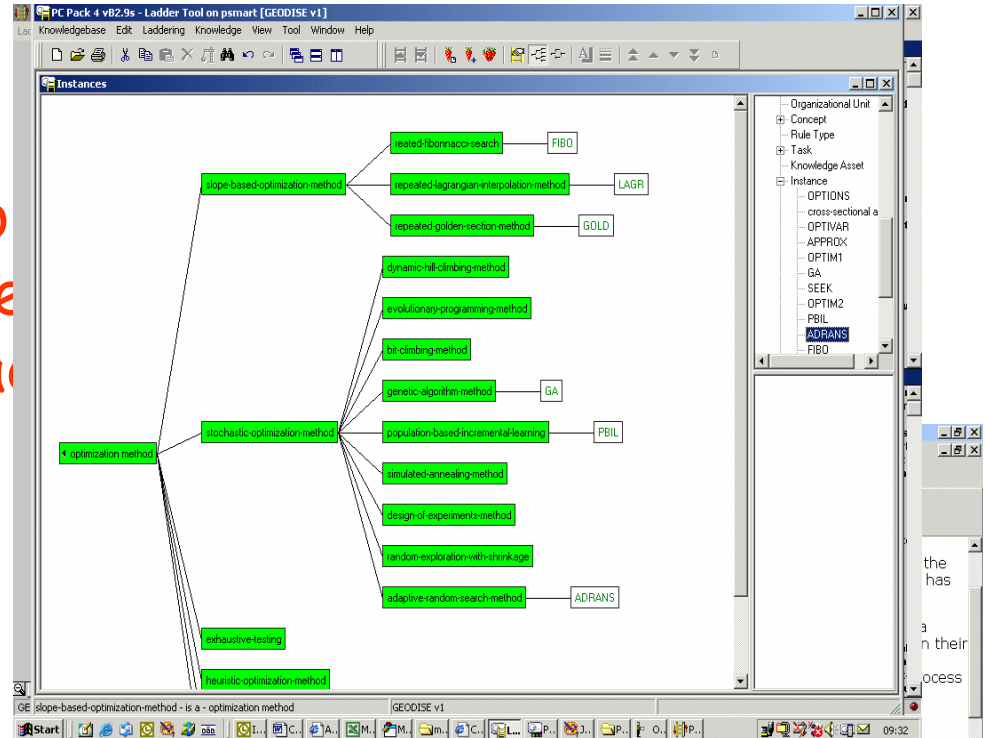
PC-PACK integrated knowledge engineering toolkit

Knowledge acquired

EDSO domain knowledge, EDSO processes and problem definition

Co
hie
Lac

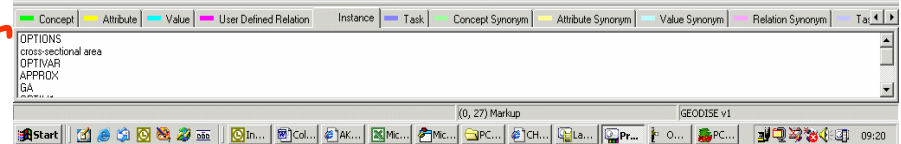
Concep
up in Pr
Editor



5.2. Choice of Optimizer

Here, as has already been noted, the initial input data for the design do not allow all the criteria used as constraints to be met (i.e., it is not a feasible design). This restricts the choice of optimizer since some do not work when given an initially infeasible design. However, limits have also already been chosen for the design variables and the optimizers will therefore seek to remain within these limits, reducing the scope of the task somewhat (some of the methods within the OPTIVAR library treat these range constraints as advisory by default and may, therefore, explore beyond their limits; this may be over-riden if desired). To give a representative cross-section of the methods available the following techniques have been applied:

- (1) the method of successive linear approximation -APPROX;
- (2) the genetic algorithm -GA, with a one pass, external penalty function -OPTIM1 and
- (3) the Hooke and Jeeves direct search -SEEK, with the Blacoo-McCormick combined external and internal penalty function -OPTIM2



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

ODISE\models\GEODISE Knowledge Model\index.htm

CS LTD
business

Concept: genetic-algorithm-method

The genetic-algorithm concept represents types of genetic a

Domain Schema
design-optimization-and-search-schema

Supertypes
• *stochastic-optimization-method*

Subtypes

```

<description>The computational-fluid-dynamics-schema describes knowledge constructs relevant to the domain of computational fluid dynamics.
This domain schema is to be extended and refined in future knowledge acquisition initiatives.</description>
</terminology>
</construct-information>
- <concept id="concept-13" name="CFD-code">
- <construct-information>
- <terminology>
<description>The CFD-code concept represents a self-contained piece of computer code that is specifically developed for use in the domain of
computational fluid dynamics.</description>
</terminology>
</construct-information>
- <abstraction-relationships>
- <super-types>
- <super-type id="concept-2">analysis-code</super-type>
</super-types>
</abstraction-relationships>
</concept>
- <concept id="concept-79" name="fluid-flow-feature">
- <construct-information>
- <terminology>
<description>The fluid-flow-feature concept represents features of the fluid flow in the domain of computational fluid dynamics. Examples of
flow features include vortices, wakes, etc. The detailed modelling of these constructs is reserved for further knowledge engineering work in
the domain of computational fluid dynamics.</description>
</terminology>
</construct-information>
- <abstraction-relationships>
- <super-types>
- <super-type id="concept-196">thing</super-type>
</super-types>
</abstraction-relationships>
</concept>
    
```



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

Composition Advice

Composition Area

Advice in Domain Editor

Functions

State Monitor

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

The screenshot displays the Protégé Editor interface. The top menu bar includes 'Classes', 'Slots', 'Forms', 'Instances', and 'Queries'. The main window is titled 'Linear_programming' and shows a class hierarchy on the left, a central 'Hierarchy' view, and a right-hand 'Documentation' and 'Restrictions' panel.

Protégé Editor (indicated by red text) is the primary tool shown. The 'Linear_programming' class is highlighted in the hierarchy, and its 'Abstract' property is visible in the right panel. The 'Documentation' panel contains the text: 'optimisation methods' and 'Objective function analysis is to execute the analysis code to return objective function.'

OilEd Editor (indicated by red text) is also visible, showing a 'Restrictions' table with the following data:

type	property	filler
has-class	has input #1	meshFile #1
has-class	has input #1	journalFile #1
has-class	has output #1	solverAnalysisFile #1
has-class	performed by #1	solverAnalysisTools #1
has-class	has effect #1	ObjectiveFunctionValue #1
has-class	has effect #1	SingleRunTime #1

DAML+OIL (indicated by green text) is the language used for the ontology. The bottom of the screenshot shows a snippet of the ontology code:

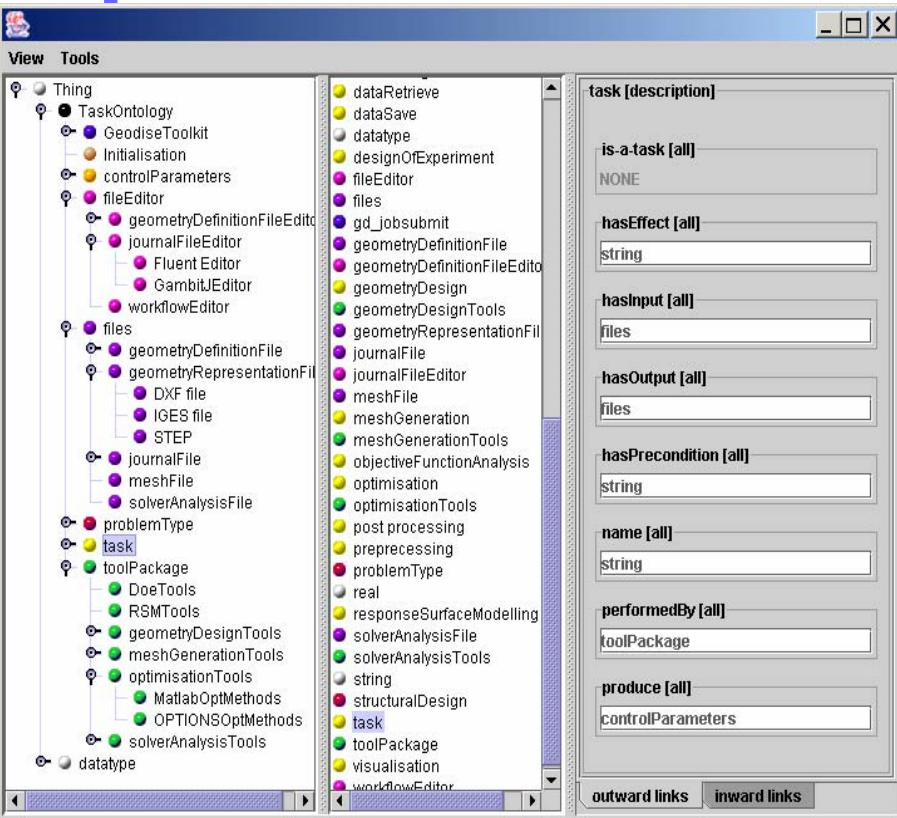
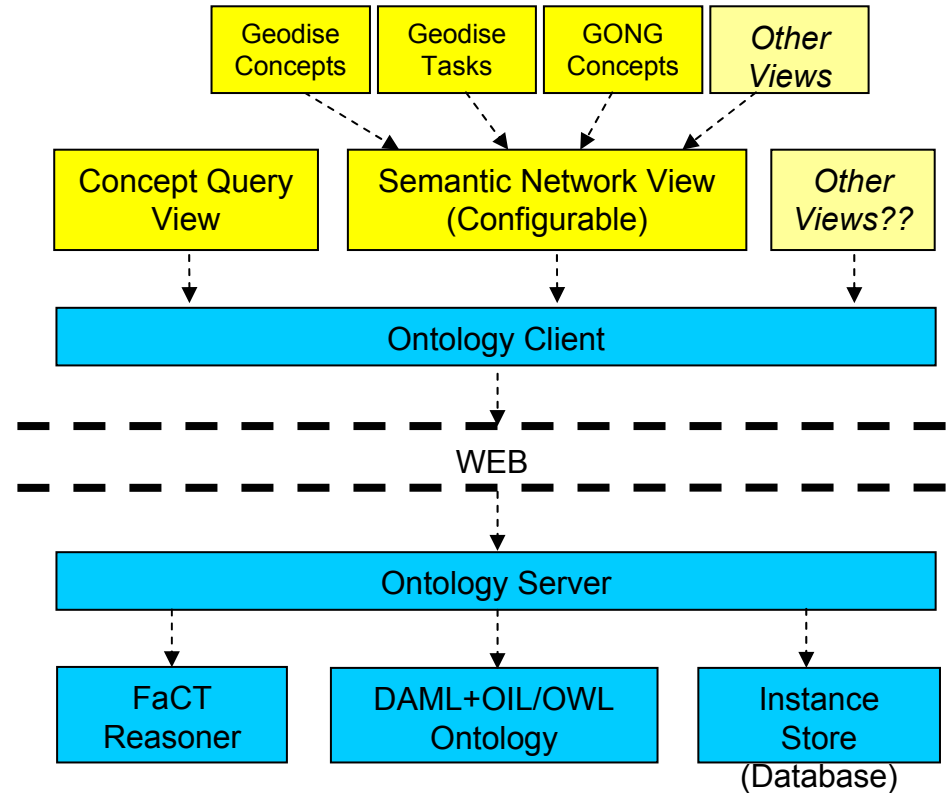
```

<rdf:label>geometry
<rdf:comment>
<![CDATA[ ]]>
</rdf:comment>
<!-- creationDate -->
<![CDATA[ 2002-10-
</rdf:comment>
<!-- creator -->
<![CDATA[ 1c ]]>
</rdf:comment>
<!-- subClassOf -->
<daml:Class rdf:about=
</rdf:subClassOf>
<!-- subClassOf -->
<!-- Restriction -->
<daml:Restriction
<daml:hasClass>
</daml:hasClass>
</daml:Restriction>
</rdf:subClassOf>
<!-- subClassOf -->
<!-- Restriction -->
<daml:Restriction
<daml:hasClass>
</daml:hasClass>
</daml:Restriction>
</rdf:subClassOf>
<!-- Restriction -->
<daml:Restriction
<daml:hasClass>
</daml:hasClass>
</daml:Restriction>
</rdf:subClassOf>
<!-- Restriction -->
<daml:Restriction
<daml:hasClass>
</daml:hasClass>
</daml:Restriction>
</rdf:subClassOf>
</daml:Class>
    
```



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

Ontology Services

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://zeus.sesnet.soton.ac.uk:8080/>

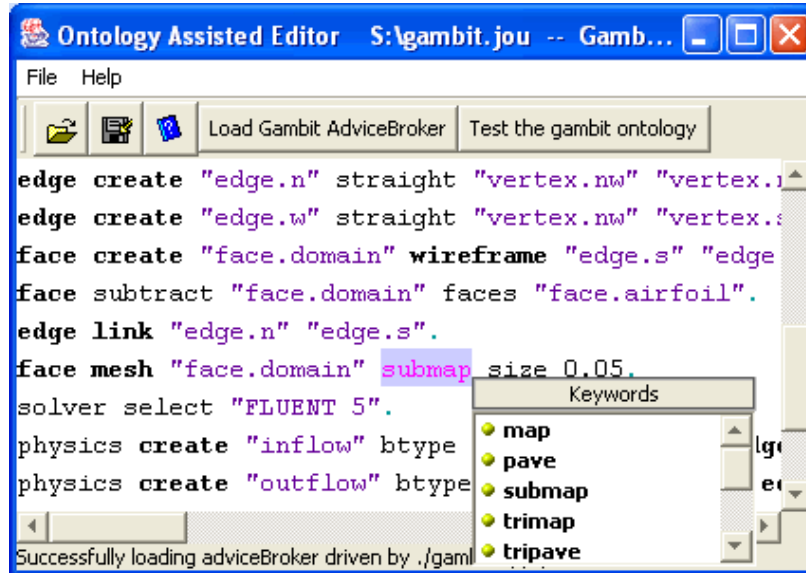
SOAP interface:
<http://zeus.sesnet.soton.ac.uk:8080/axis/services/os?wsdl>

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

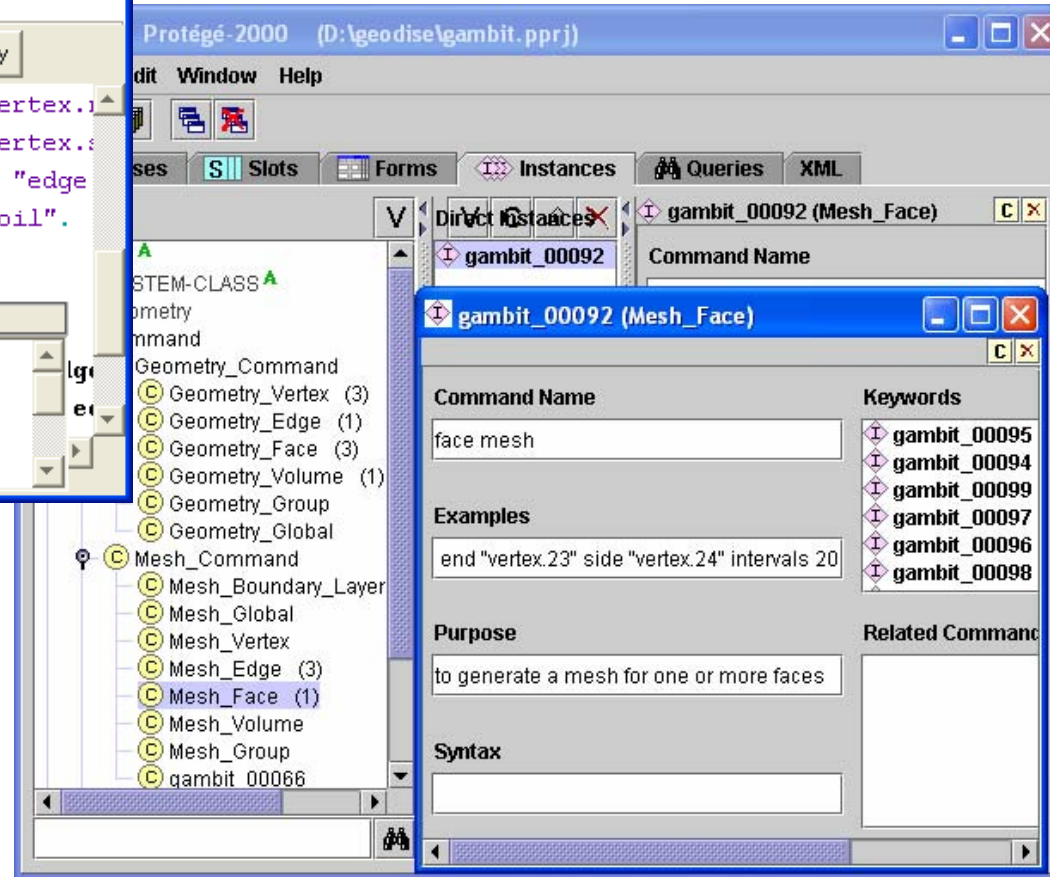


```
edge create "edge.n" straight "vertex.nw" "vertex.
edge create "edge.w" straight "vertex.nw" "vertex.
face create "face.domain" wireframe "edge.s" "edge
face subtract "face.domain" faces "face.airfoil".
edge link "edge.n" "edge.s".
face mesh "face.domain" submap size 0.05.
solver select "FLUENT 5".
physics create "inflow" btype
physics create "outflow" btype
```

Keywords

- map
- pave
- submap
- trimap
- tripave

Successfully loading adviceBroker driven by ./gambit



Protégé-2000 (D:\geodise\gambit.pprj)

Classes: Geometry_Verx (3), Geometry_Edge (1), Geometry_Face (3), Geometry_Volume (1), Geometry_Group, Geometry_Global, Mesh_Command, Mesh_Boundary_Layer, Mesh_Global, Mesh_Verx, Mesh_Edge (3), Mesh_Face (1), Mesh_Volume, Mesh_Group, gambit_00086

gambit_00092 (Mesh_Face)

Property	Value
Command Name	face mesh
Keywords	gambit_00095, gambit_00094, gambit_00099, gambit_00097, gambit_00096, gambit_00098
Examples	end "vertex.23" side "vertex.24" intervals 20
Purpose	to generate a mesh for one or more faces
Syntax	

- 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)

The screenshot shows a web browser window displaying the Geodise interface. The address bar shows the URL: `http://80.254.166.28/jaxfront-v121/servlet/demoservlet?frameHandler=mainFrameSet&sessionId=4341161`. The page is titled "ProblemProfile" and contains the following fields and sections:

- description:** Beam design problem
- timeCreated:** 2001-12-17T09:30:47
- lastTimeUsed:** 2003-02-17T09:30:47
- user:** barry
- dg_id:** (empty)
- parameters:** A table with columns: name, unit, meaning, value.

name	unit	meaning	value
length	mm	length of the beam	250
force	N	force on the tip of th	5000.0
density	kg/m3	density of the beam	7850.0
- dataFileURL:** (empty)
- objectiveFunction:** A table with columns: selected, objective, codeURL, singleRunTime, estimatedLowerBound, estimated.

selected	objective	codeURL	singleRunTime	estimatedLowerBound	estimated
<input checked="" type="checkbox"/>	minimize the cross-s	d:\geodise\beam1.s	PT30S	20	30
- singlePeak:**
- optimisation:** not defined

Buttons at the bottom: Close Session, Show XML, Cancel, Validate, Save.

The screenshot shows a web browser window displaying the Geodise interface. The address bar shows the URL: `http://80.254.166.28/jaxfront-v121/servlet/demoservlet?xcr`. The page displays a detailed configuration for a parameter:

- name:** material_type
- meaning:** beam material
- unit:** null
- limit:** discreteLimit (dropdown menu)
- values:** A list box containing 1, 2, 3. A "change" button is next to it.
- fixed:**

Ontology Driven Forms in Geodise -2

- Configuring tasks in Workflow Composing Environment

The screenshot displays the Workflow Construction Assistant interface, which is used for configuring tasks in a workflow composing environment. The interface is divided into several panels:

- Specify Ontology Services:** Shows the URL `tp://localhost:3080/axis/services/fo?wsdl` and a "Load" button.
- Specify an Ontology:** Shows the path `hntologies/TaskOntology.daml` and a "Load" button.
- Ontology Concept Browser:** A tree view showing the ontology structure, including `TaskOntology`, `toolPackage`, `task`, `DesignOfExperiment`, `geometryDesign`, `visualisation`, `post processing`, `preprocessing`, `designOfExperiment`, `dataManagement`, `optimisation`, `objectiveFunctionAnalysis`, `responseSurfaceModell`, `meshGeneration`, `fileEditor`, `Geodise Toolkit`, `Initialisation`, `files`, `solverAnalysisFile`, and `geometryRepresentation` (with sub-items `IGES file`, `STEP_file`, and `DXF file`).
- Knowledge-based Advice:** Displays a message: `("ObjectiveFunctionValue" "SingleRunTime" "ProEAssembly file" "Identifier" "STEP_f... You may be able to do DesignofExperiment by using ("SingleRunTime") but still need input:("BudgetTime")`. It includes "Next" and "Back" buttons.
- Workflow Editor:** Shows a workflow diagram with three tasks: "NACA geometry design", "NACA mesh generation", and "NACA code analysis", connected by arrows.
- Component Editor:** A dialog box for configuring a component. It includes a search field, "Search" and "Select" buttons, and a table of properties:

ID:	has precondition:
identifier	not defined
NACA code analysis	
has effect:	has effect:
ObjectiveFunctionValue	SingleRunTime
value of objective function	single run time
has input:	has input:
journalFile	meshFile
naca fluent journal file	naca mesh file
- State Monitor:** A table showing the state of the workflow:

index	ID	property	type	instance
8	ID		identifier	NACA code analysis
9	has effect		ObjectiveFunctionValue	value of objective function
10	has effect		SingleRunTime	single run time
11	has input		FluentJournalFile	naca fluent journal file
12	has input		meshFile	naca mesh file

Exploiting Knowledge in Geodise



Knowledge Application 1: Create Semantic Content

The screenshot shows a desktop environment with several windows open. On the left is the 'Ontology Browser' window, displaying a tree view of an ontology with categories like 'Single_variable_minimisation', 'Stochastic_methods', 'User_defined_methods', 'heuristic-optimization-met', 'slope_based_optimization', 'PenaltyFunctions', 'Performance', 'Problem', 'ResponseSurfaceModelling', 'UnitType', 'User', and 'ValueType'. Below the tree is a table with 'Attributes' and 'Values' columns, showing 'complexity' as 'simple', 'description' as 'design of nacelle', and 'problem_ID'. At the bottom of the browser is a list of ontology elements: 'constraint_function (Constraint_F...', 'objective_function (Objective_Fu...', 'problem_designer (User)', 'ajk', 'variable (VariableType)', 'DDP_a_u', and 'DDP_xp_J'. In the center is the 'HTML Browser 1' window, showing a URL 'file://localhostD:\geodise\to be annotated\demo2\log.html' and displaying the content of a log file. On the right are two 'Notepad' windows. The top one is 'annotatedGeodise2wf.html', showing XML code with various namespaces and elements like 'beam', 'RSM option', 'build', 'problem', 'select', 'Breath', 'Height', 'not selected', 'evaluation', 'Objective', 'Runtime', 'Sampling', 'Time Budget', 'maximum', 'Sampling', 'estimated', 'DOE algorithm', 'Computation', 'RSM Model', 'RSM algorithm', 'Theta value', 'Design', 'Optimization', 'Optimum', and 'Optimum'. The bottom Notepad window is 'geodise2wf.htm - Notepad', showing the original XML code before annotation.

• Goals

- Machine understandable information
- Facilitate sharing & reuse

• Techniques & tools

- OntMat-annotizer
- Geodise Ontologies

• Example

- OPTIONS log-files annotation

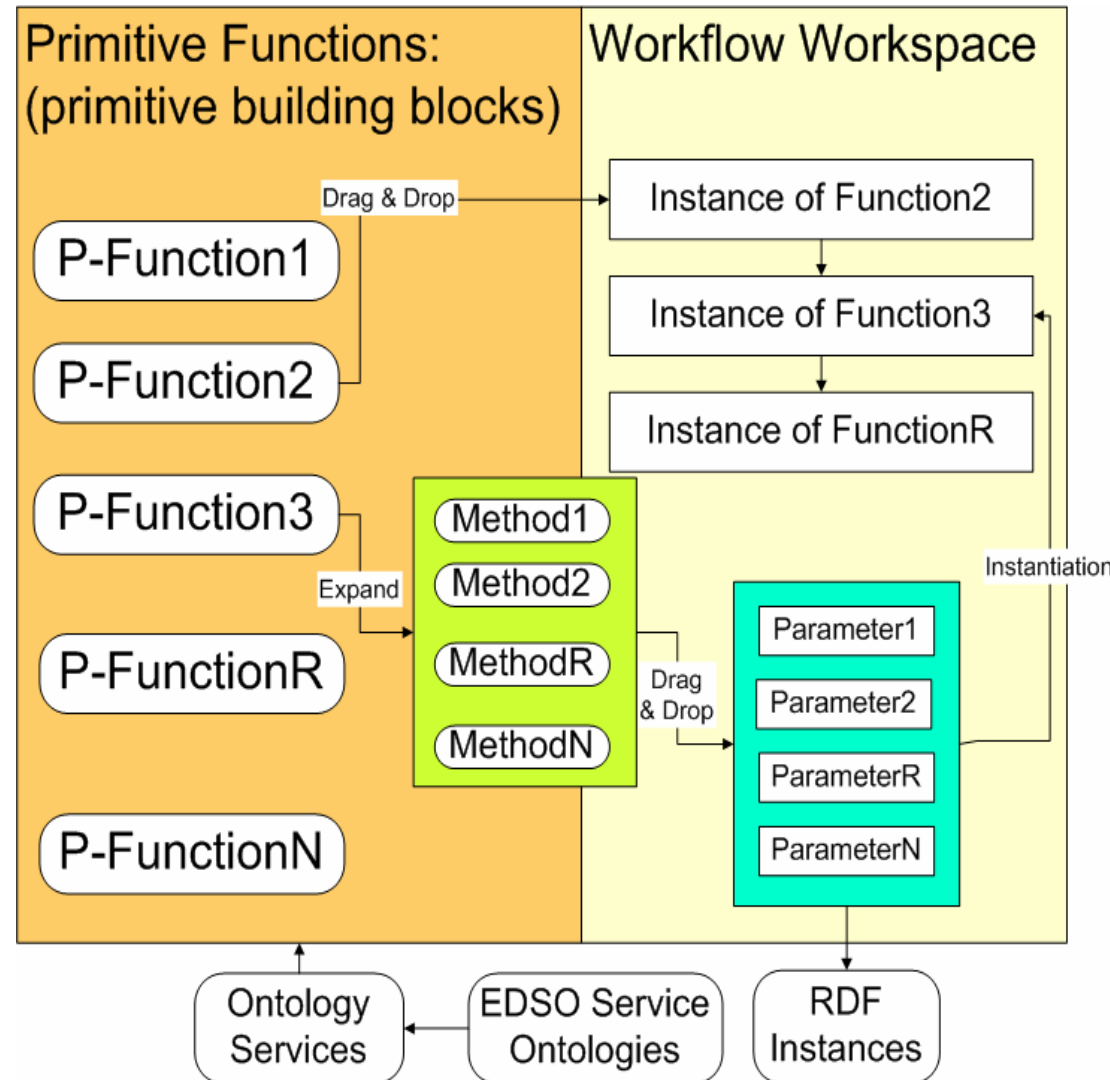
Ontology-assisted Workflow Management

- **Features:**

- Function selection
- Function instantiation
- Database schema
- Semantic instances
- Semantic workflow

- **Technologies:**

- EDSO ontologies & ontology services
- Java JAX-RPC, DOM/SAX



Knowledge Application 3:

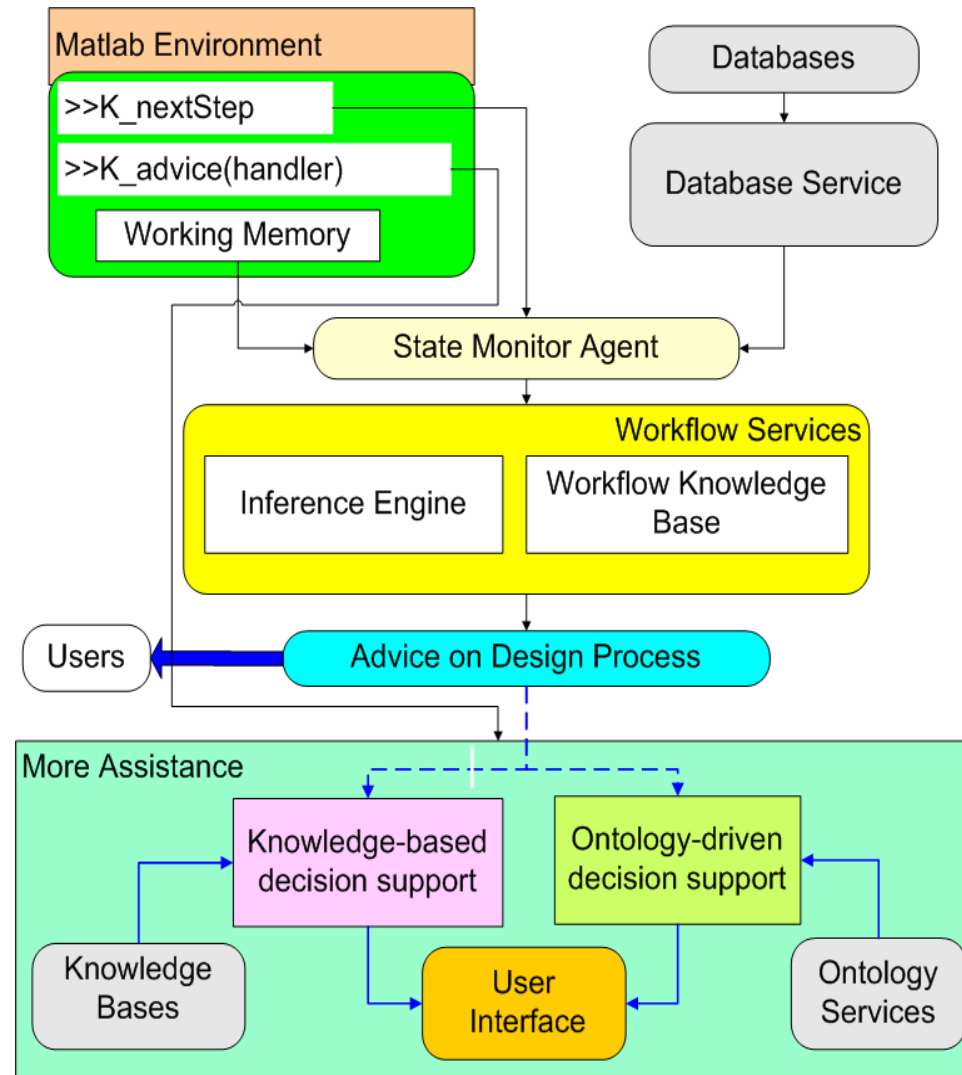
Knowledge-based Design Advisor

- **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')
```

