

# Applying the Semantic Web to Manage Knowledge on the Grid

Feng Tao, Colin Puleston, Carole Goble, Nigel Shadbolt, Liming Chen, Graeme Pound, Fenglian Xu, Simon Cox

University of Southampton  
University of Manchester

Presenter: Barry



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

# Overview

- Background and purpose
  - EDSO, Resource, knowledge support
- A layered semantic infrastructure
  - DL Ontology + Instance store, Ontoview, annotator and advisor
- Life cycle of semantic web base KM
  - Knowledge capture, binding and reuse
- Demonstrations of various tools
  - Ontoview, annotator, advisor, knowledge toolbox, etc.
- Summary and future work



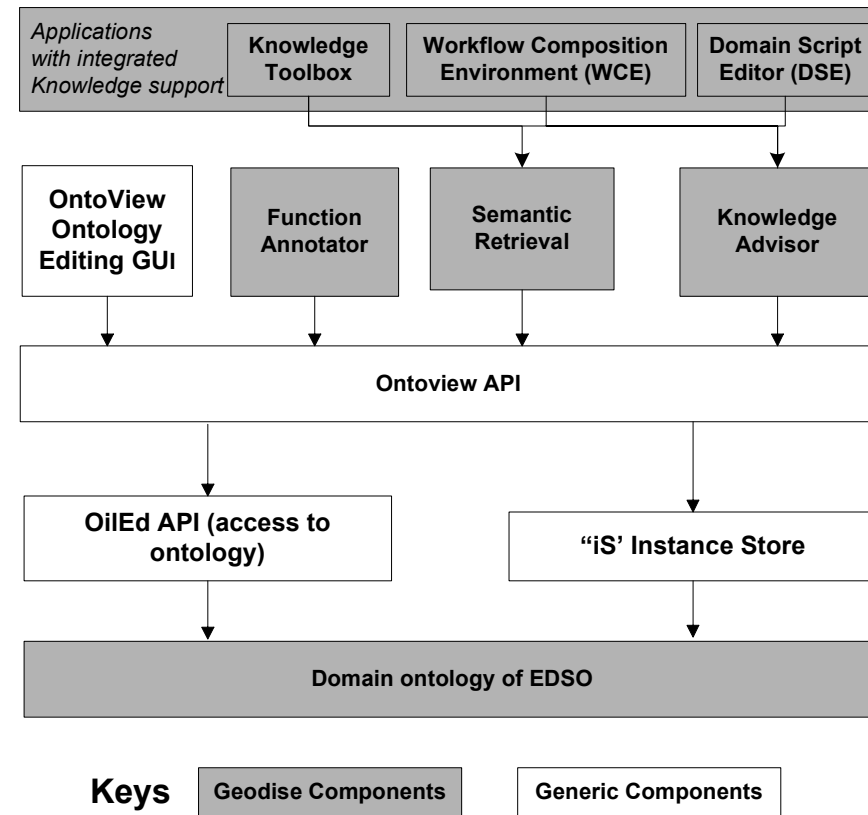
# Background and purposes

- Grid based EDSO
  - Engineering Design Search & Optimisation
  - Resources of distributed computation, distributed storage and distributed knowledge
  - Toolbox of Grid enabled Matlab functions
- Describe and share *resources* using Ontology and Semantic Grid technologies
  - Components: Matlab functions – the toolbox,
  - Domain knowledge: optimisation methods, valid configurations, etc.
- Provide *knowledge* through reusing the semantics
  - Retrieval of the semantics (direct use)
  - Advice (more advanced usage)
    - ❑ On function configuration
    - ❑ On workflow composition
  - Web services, distributed and service oriented architecture



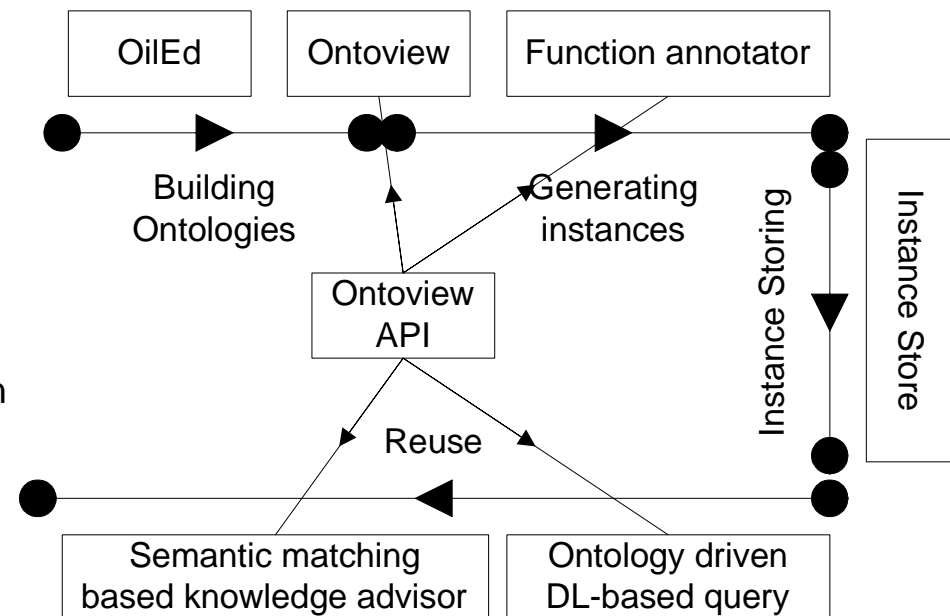
# Layered Semantic Web Infrastructure

- **EDSO Domain Ontology**
  - Concepts and relations in a domain
  - Obtained through KA
  - Represented in DAML+OIL
- **Generic Ontology manipulation and storing mechanism**
  - Ontoview API and editors
  - Ontology further refined and populated in Ontoview
- **Geodise knowledge services/demonstrators**
  - Function Annotator
  - Semantic Retrieval GUI
  - Knowledge advisor service
- **Geodise Apps integrated with knowledge**
  - Knowledge toolbox in Matlab
  - WCE standalone tool
  - DSE standalone tool



# Knowledge Life cycle

- **Knowledge Capturing**
  - Knowledge Acquisition, building Ontologies
- **Knowledge Binding**
  - Annotation : Creating instances of ontological concepts
  - Ontoview editor, function annotator
- **Knowledge Modeling**
  - Specification of useful knowledge: Semantic retrieval, advice on function configuration and assembly
- **Knowledge Reusing**
  - Semantic based function query and knowledge advisor

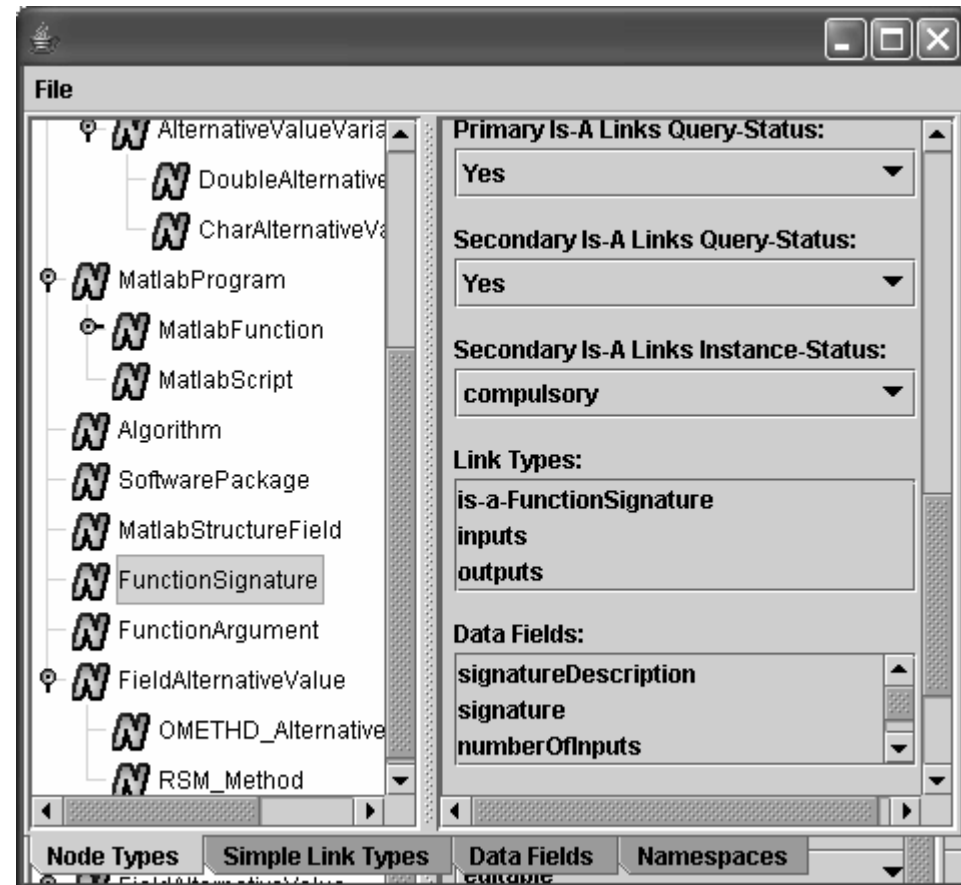


- Illustration follows



# Knowledge Capturing

- Knowledge Acquisition
  - From Domain experts and domain documents.
  - Build in OilEd and maintain in Ontoview)
- Ontology
  - Concepts
    - ☐ Node Type
  - Relationships
    - ☐ Hierarchy
    - ☐ Link Type & Data Field
- Result
  - DAML+OIL format



# Function Annotation (Knowledge Binding)

- Generating semantic instances
  - Binding ontology with semantics content
  - Populating the semantic web based knowledge base
- Supporting Tools
  - Through Ontoview editor
    - ☐ High flexibility (can generate instances of any ontology concept)
    - ☐ sometimes tedious
  - Through a customized function annotator
    - ☐ Automatic parsing
    - ☐ Lack of flexibility (only deal with functions at the moment)
  - We use both



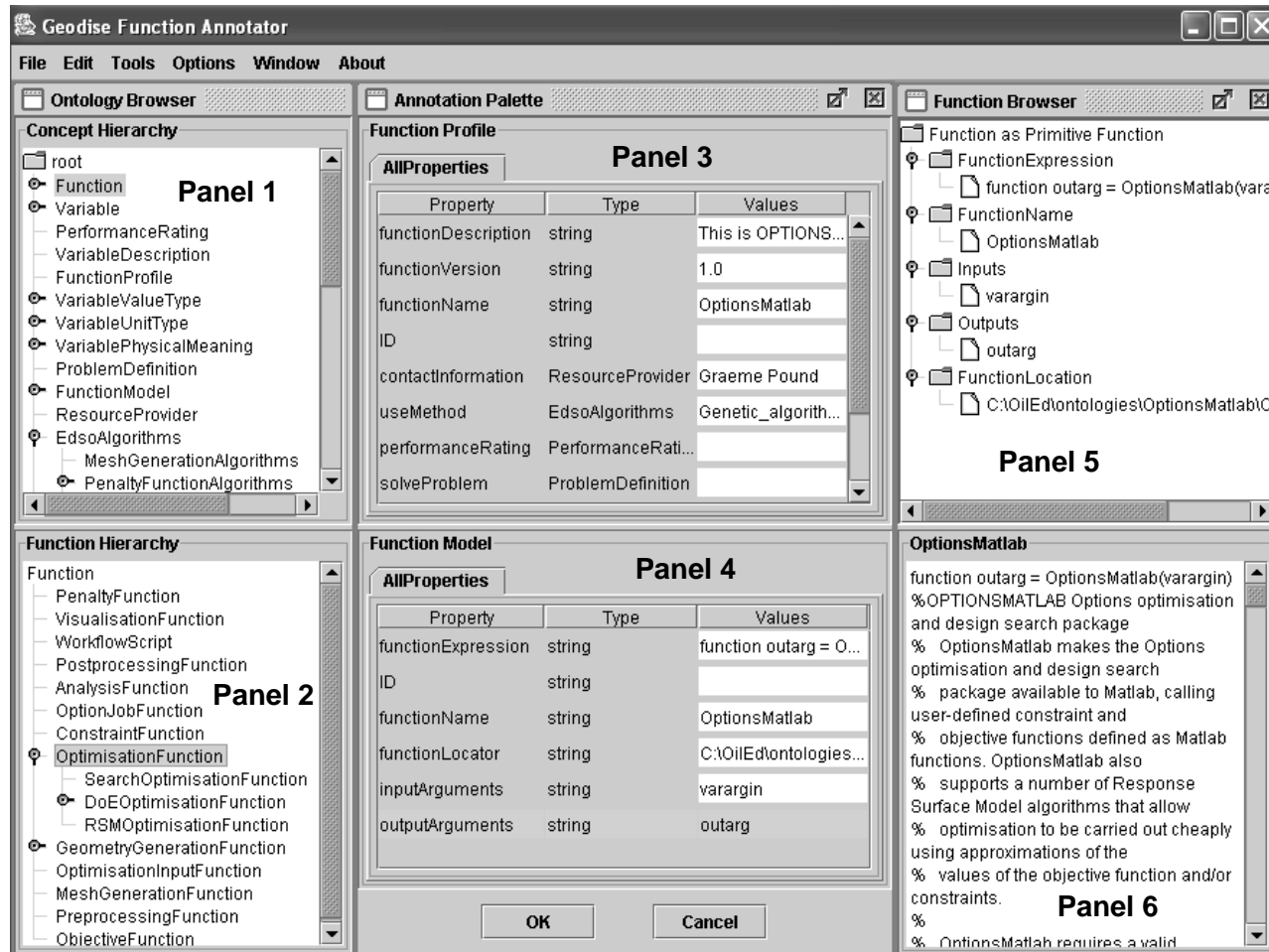
1

- 





# Function Annotator



Customised for Matlab functions

- Automatic parsing Matlab function source

Instantiating abstract nodes defined in ontology

Semi-automatic filling of the ontology driven forms

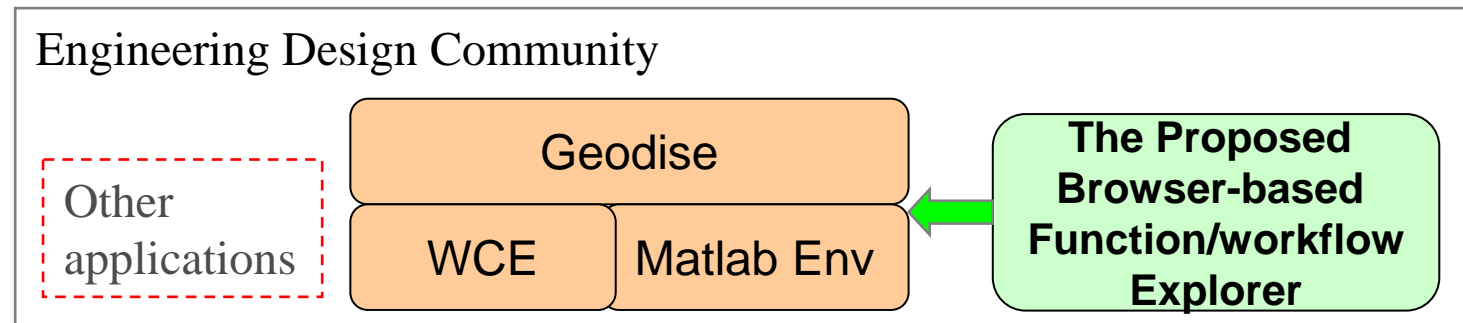


# Knowledge Reuse

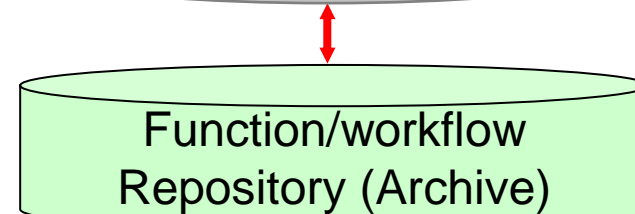
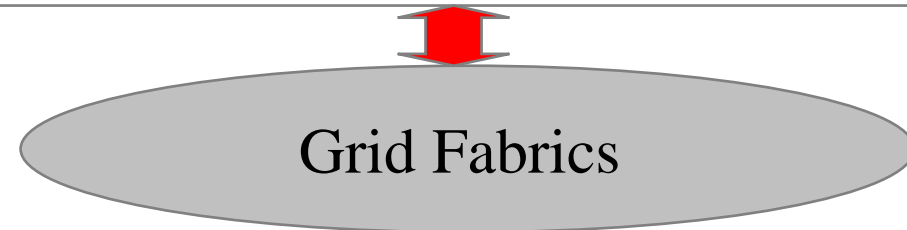
- **Function Query**
  - Identify functions based on their semantics criteria (Semantics → Functions)
- **Advice**
  - Retrieve semantics (Entity → Semantics)
  - Function assembly (Service composition)
  - Function configuration
- **Service Oriented Architecture**



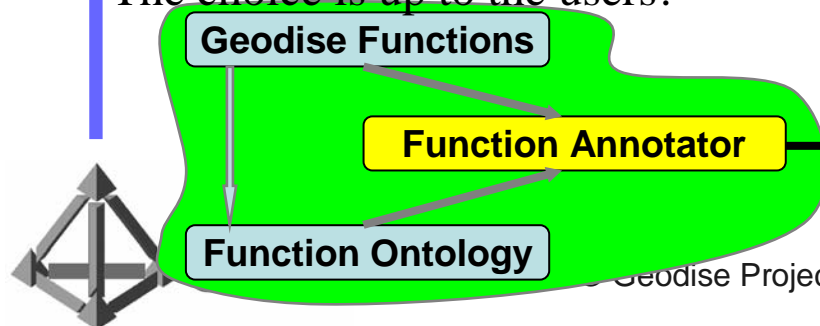
# Service-oriented Semantics Application Scenario



If we want to leverage and harvest the maximum benefit of Semantic web, i.e. effective discovery, machine-enabled (processable, understandable) interoperability and automation, and the Grid, i.e. resource sharing, this is one of the realisations.



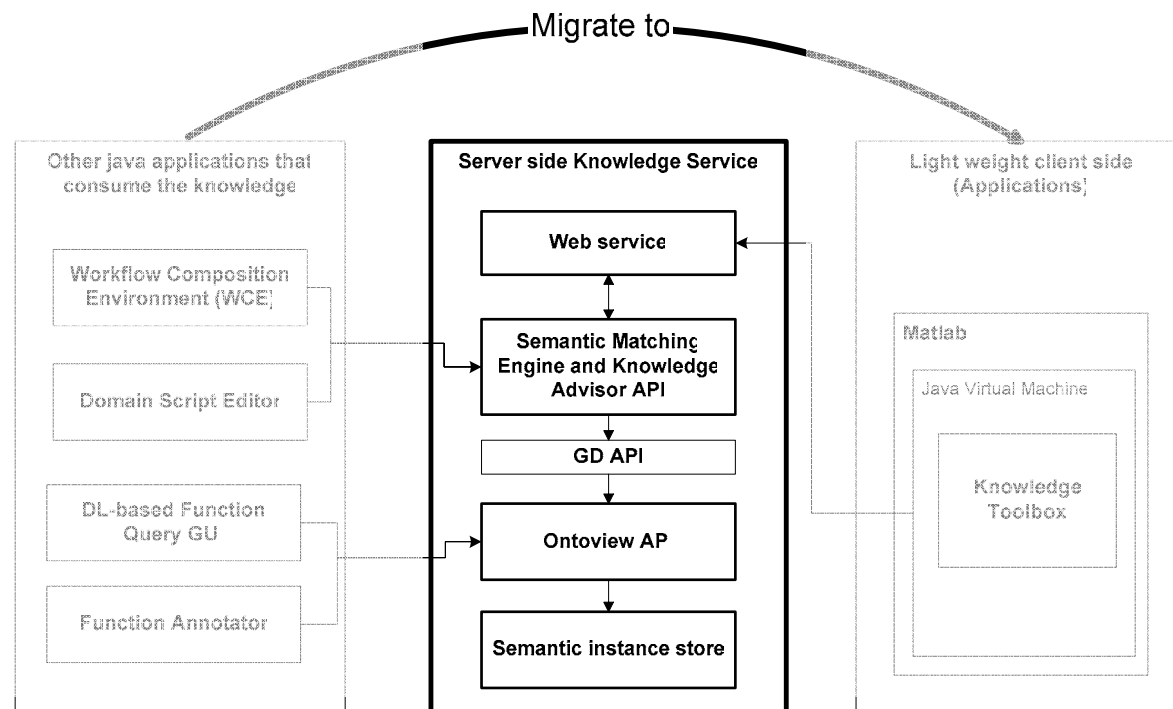
The choice is up to the users!



Geodise Project, University of Southampton, 2001-2004.  
<http://www.geodise.org/>

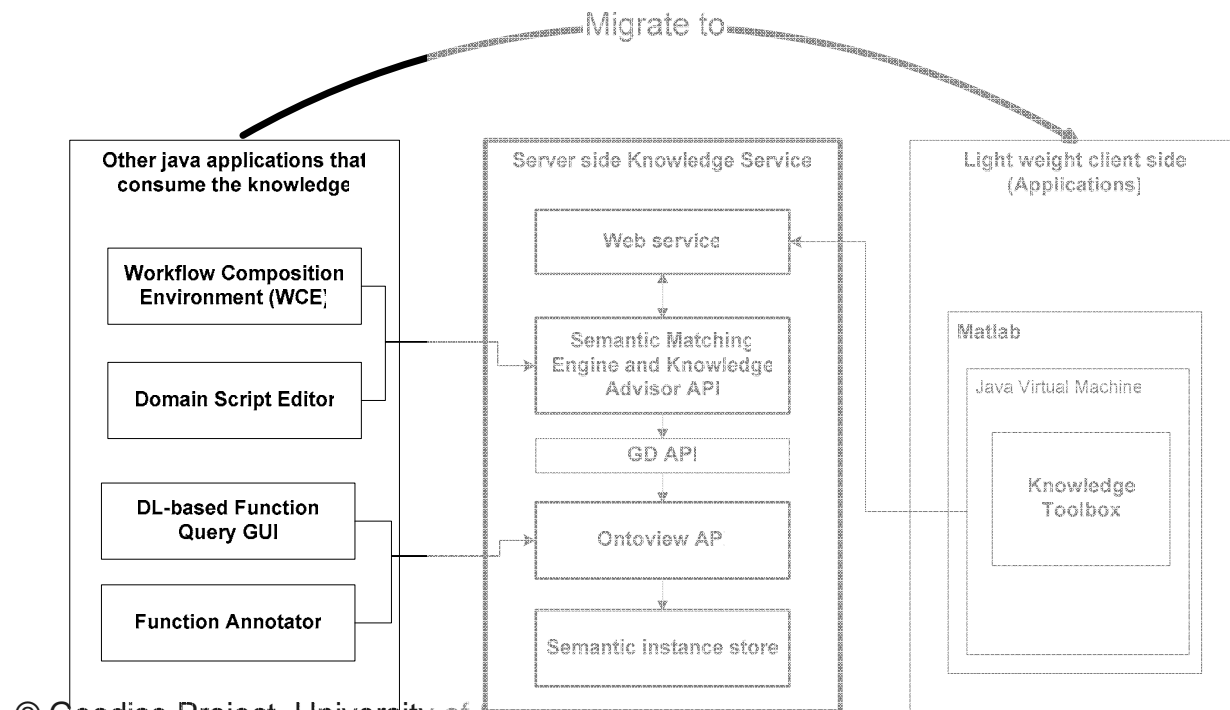
# Towards Service Oriented Paradigm

- Server (in the middle)
  - Semantic layer
  - Interfaces at different level
    - ❑ Web service
    - ❑ Java APIs



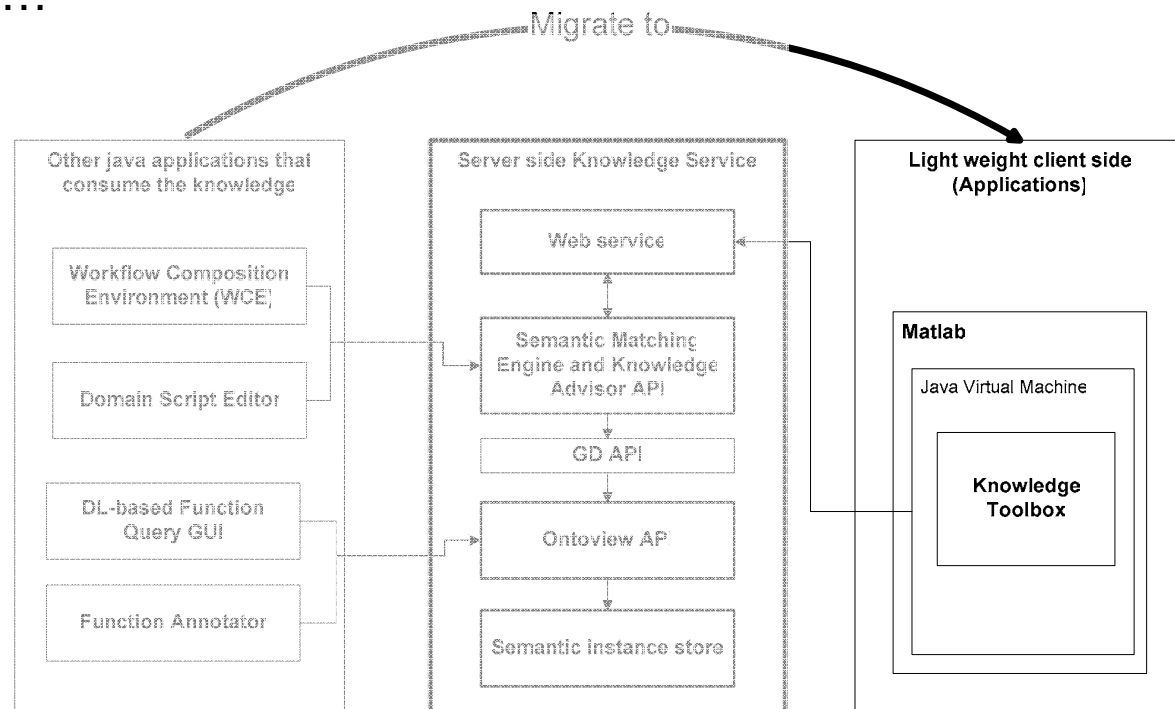
# Towards Service Orientated Paradigm

- Other java applications (to the left)
  - Consume the service via Knowledge service APIs
  - WCE, DSE, FA, DL-FQ
  - To be moved to client side in the right?



# Towards Service Orientated Paradigm

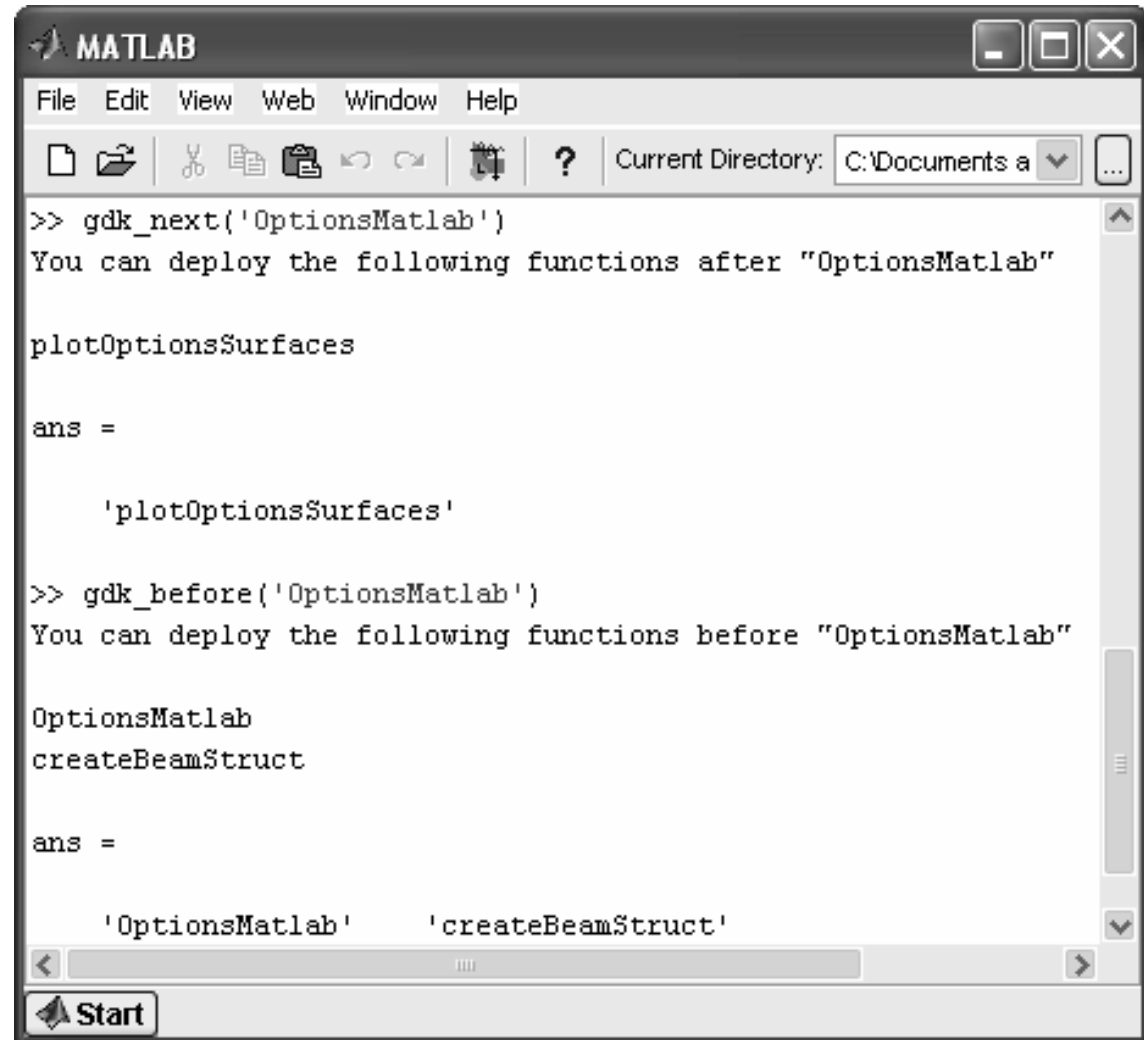
- Client (to the right)
  - Matlab PSE
  - Knowledge Toolbox
    - ❑ Matlab → Java
    - ❑ Web service consumer
- Illustrations follows ...



# Advice on Function Assembly

## (Integrated in Matlab – Knowledge Toolbox)

- **Goal**
  - Function assembly
  - What can be deploy next and before?
- **Mechanism**
  - Matlab → Java → WSDL → Web service
  - Function semantic interface
  - Semantic matching
- **Pre-requirements**
  - Function has been annotated
  - Semantics available in the instance store



The image shows a MATLAB window titled 'MATLAB' with a standard menu bar (File, Edit, View, Web, Window, Help) and a toolbar. The 'Current Directory' is set to 'C:\Documents a'. The command window displays the following text:

```
>> gdk_next('OptionsMatlab')
You can deploy the following functions after "OptionsMatlab"

plotOptionsSurfaces

ans =

    'plotOptionsSurfaces'

>> gdk_before('OptionsMatlab')
You can deploy the following functions before "OptionsMatlab"

OptionsMatlab
createBeamStruct

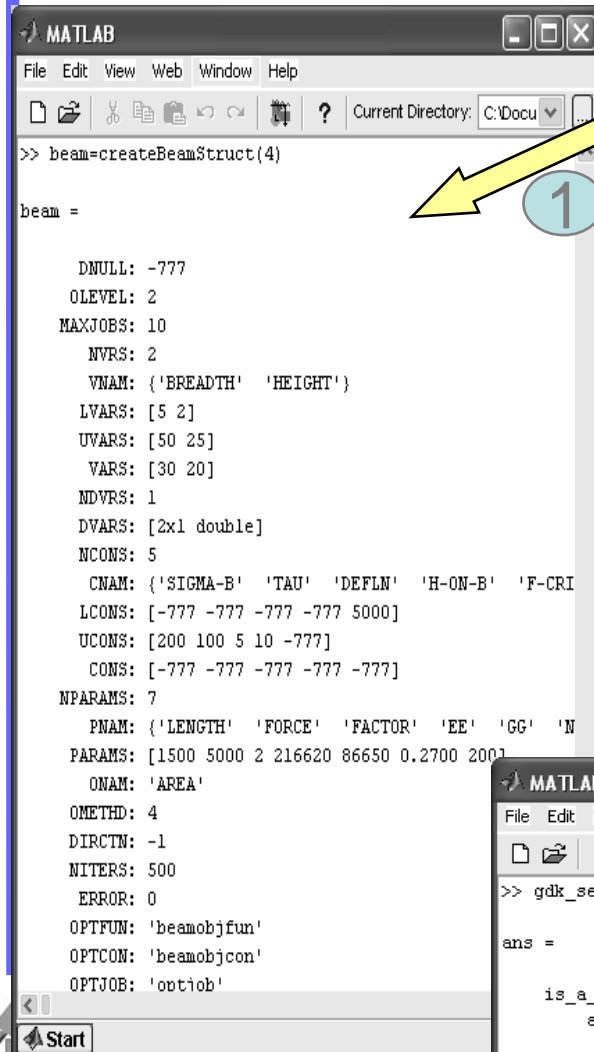
ans =

    'OptionsMatlab'    'createBeamStruct'
```

At the bottom of the window is a 'Start' button.



# Advice on Function Configuration



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\Docu...

>> beam=createBeamStruct(4)

beam =

    DNULL: -777
    OLEVEL: 2
    MAXJOBS: 10
    NVRS: 2
    VNAM: {'BREADTH' 'HEIGHT'}
    LVAR: [5 2]
    UVAR: [50 25]
    VAR: [30 20]
    NDVRS: 1
    DVAR: [2x1 double]
    NCONS: 5
    CNAM: {'SIGMA-B' 'TAU' 'DEFLN' 'H-ON-B' 'F-CRI
    LCONS: [-777 -777 -777 -777 5000]
    UCONS: [200 100 5 10 -777]
    CONS: [-777 -777 -777 -777 -777]
    NPARAMS: 7
    PNAME: {'LENGTH' 'FORCE' 'FACTOR' 'EE' 'GG' 'N
    PARAMS: [1500 5000 2 216620 86650 0.2700 200]
    ONAM: 'AREA'
    OMETHD: 4
    DIRCTN: -1
    NITERS: 500
    ERROR: 0
    OPTFUN: 'beamobjfun'
    OPTCON: 'beamobjcon'
    OPTJOB: 'optjob'
    
```

% get the default beam structure

**beam = createBeamStruct (4)**

% analyze the OMETH and advice on  
its additional control parameter (with  
default value)

**beamcontrol = gdk\_options(beam)**

% check semantics

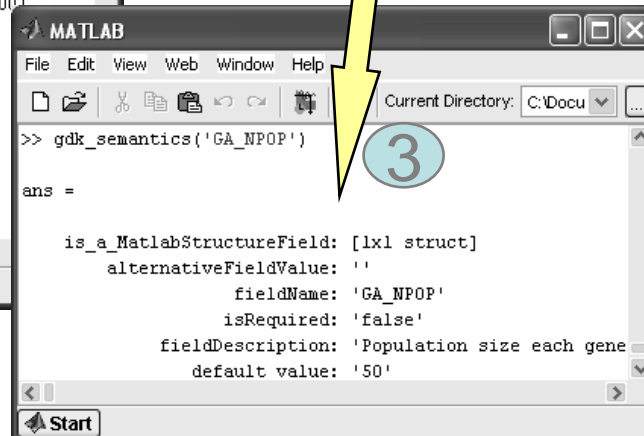
**gdk\_semantics('GD\_NPOP')**

% further configure these control  
parameters

...

% run options

**s = Option Matlab (beamcontrol)**



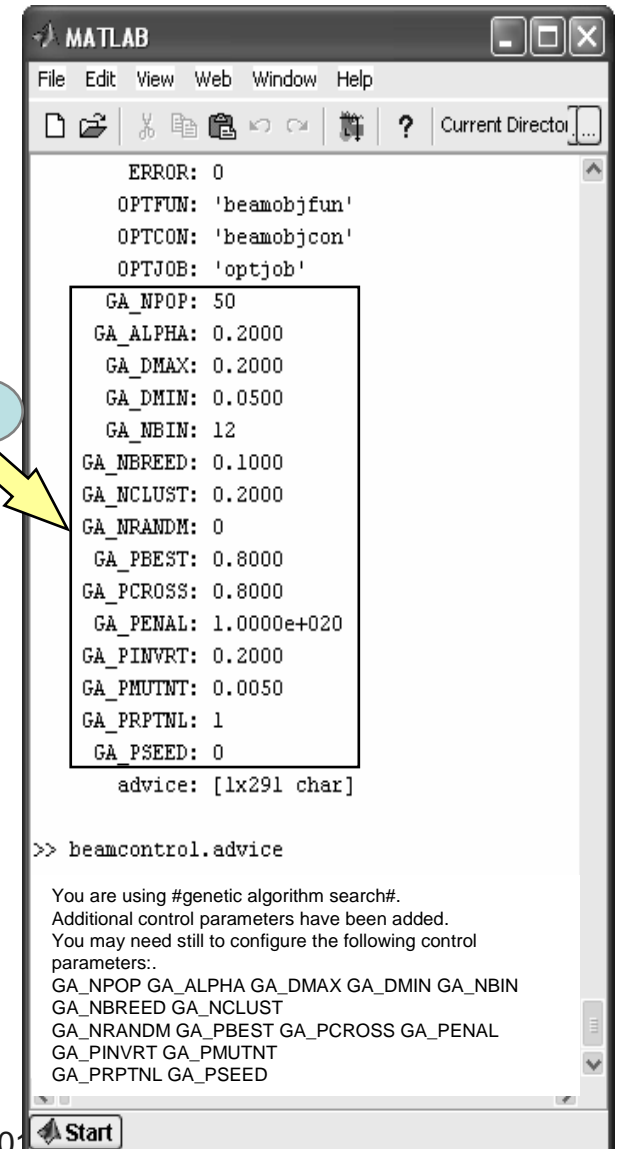
```

MATLAB
File Edit View Web Window Help
Current Directory: C:\Docu...

>> gdk_semantics('GA_NPOP')

ans =

    is_a_MatlabStructureField: [1x1 struct]
    alternativeFieldValue: ''
    fieldName: 'GA_NPOP'
    isRequired: 'false'
    fieldDescription: 'Population size each gene'
    default value: '50'
    
```



```

MATLAB
File Edit View Web Window Help
Current Director...

ERROR: 0
OPTFUN: 'beamobjfun'
OPTCON: 'beamobjcon'
OPTJOB: 'optjob'

GA_NPOP: 50
GA_ALPHA: 0.2000
GA_DMAX: 0.2000
GA_DMIN: 0.0500
GA_NBIN: 12
GA_NBREED: 0.1000
GA_NCLUST: 0.2000
GA_NRANDM: 0
GA_PBEST: 0.8000
GA_PCROSS: 0.8000
GA_PENAL: 1.0000e+020
GA_PINVRT: 0.2000
GA_PMUTNT: 0.0050
GA_PRPTNL: 1
GA_PSEED: 0

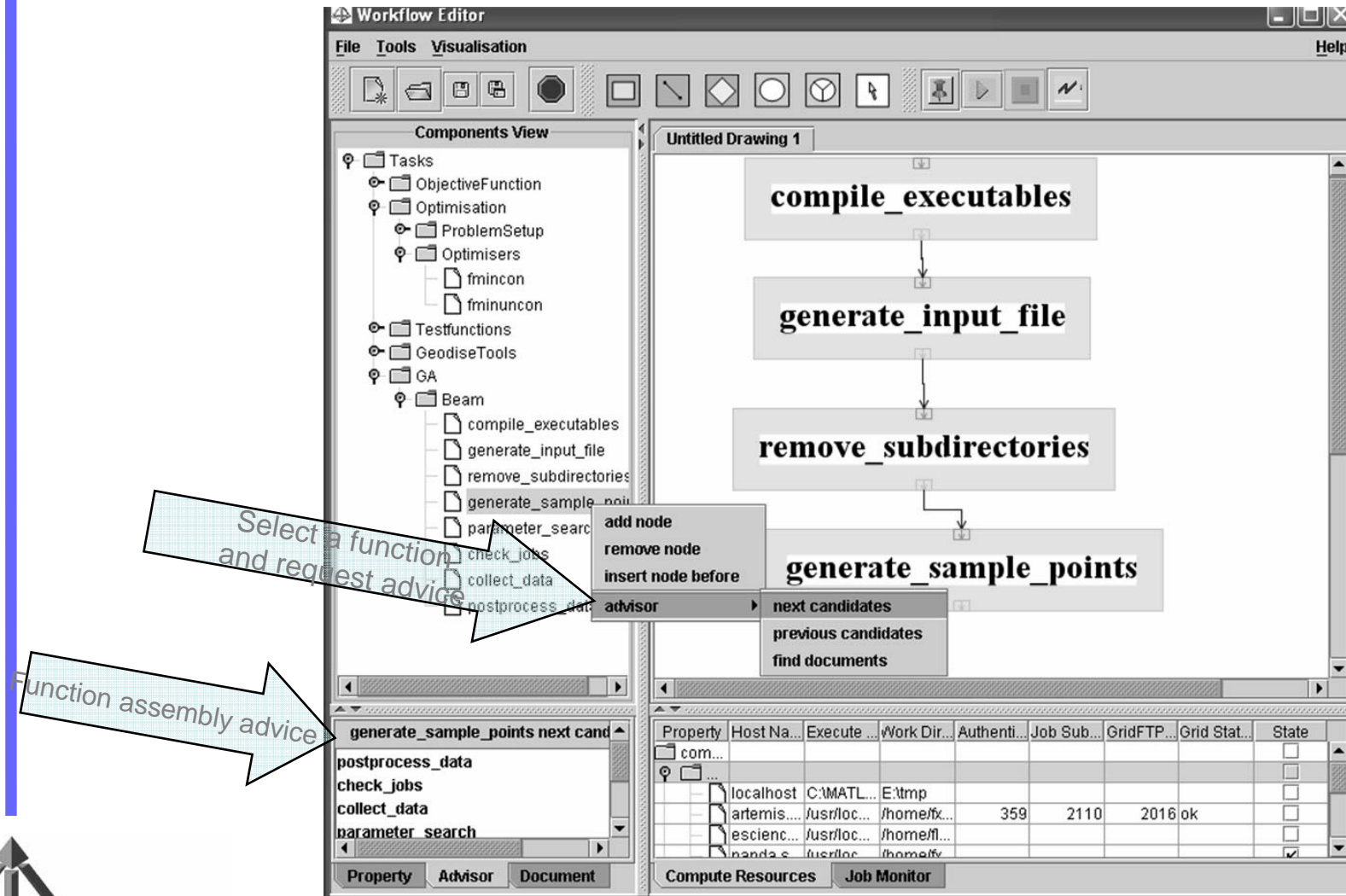
advice: [1x291 char]

>> beamcontrol.advice

You are using #genetic algorithm search#.
Additional control parameters have been added.
You may need still to configure the following control
parameters:
GA_NPOP GA_ALPHA GA_DMAX GA_DMIN GA_NBIN
GA_NBREED GA_NCLUST
GA_NRANDM GA_PBEST GA_PCROSS GA_PENAL
GA_PINVRT GA_PMUTNT
GA_PRPTNL GA_PSEED
    
```



# Advice on Function Assembly (Integrated in WCE – workflow editor)



# Advice on Function Assembly (Integrated in Domain Script Editor)

The screenshot displays the MATLAB script editor interface for a file named `.\matlab_script_samples\temp.m`. The interface is divided into several panes:

- Function Ontology (Left Pane):** A hierarchical tree structure showing the assembly of functions. The `Function` node is expanded, showing sub-nodes like `OptionsMatlabFunction`, `ApplicationRelatedFunction`, and `Hakki_prob_function`. The `generate_sample_points` function is highlighted under `Hakki_prob_function`.
- Scripts (Main Editor):** The central area for editing MATLAB code. It contains a script for setting up a server, generating sample points, and collecting data. The code includes comments and function calls like `generate_sample_points` and `collect_data`.
- Context commands (Bottom Left):** A list of commands that have been executed in the current session, including `generate_sample_points`, `postprocess_data`, `check_jobs`, and `collect_data`.
- Function configuration advice (Bottom Right):** A table providing configuration details for the functions used in the script. The table has two columns: the function name and its configuration parameters.

Annotations with arrows point to specific parts of the interface:

- ontology and semantics:** Points to the Function Ontology pane.
- Domain script editing area:** Points to the Scripts pane.
- Function configuration advice:** Points to the Function configuration advice table.
- Function assembly advice:** Points to the Context commands pane.

Function	Configuration
parameter_search	Hakki_prob_function
postprocess_data	Hakki_prob_function
problem_001	Problem
project	RSLstructDataEntry

# DL-Based Function Query GUI

The screenshot displays a graphical user interface for querying a database of MATLAB functions. The interface is divided into several sections:

- Search Criteria (Left Panel):** A series of input fields and buttons for defining search parameters.
  - is-a-GeodiseFunction:** Input field contains "GeodiseFunction", with an "edit" button.
  - invokesSoftwarePackage:** Input field contains "JavaCoG".
  - AND POSSIBLY OTHERS:** A button to expand search criteria.
  - signatures:** Input field contains "ANY".
  - implementsAlgorithm:** Input field contains "ANY".
  - invokesFunction:** Input field contains "ANY".
  - matlabFileName:** Input field contains "ANY".
  - author:** Input field contains "Graeme Pound".
  - dateWritten:** Input field contains "ANY".
  - authorInstitution:** Input field contains "University of Southampton".
  - version:** Input field contains "ANY".
- Search Results (Right Panel):** A list of functions found, categorized by type.
  - MatlabFunction:**
    - GeodiseFunction
      - DatabaseToolbox
      - XMLToolbox
      - ComputeToolbox
      - OptionsMatlabToolbox
    - StandardMatlabFunction
      - SpecializedMath
      - Graph2D
      - Graph3D
      - DataAnalysis
      - FileIO
      - NumericalLinearAlgebra
      - ElementaryMatrices
      - AudioSupport
      - LanguageConstructs
      - ODESolvers
      - Interpolation
      - GeneralPurpose
      - ElementaryMath
      - Operators
      - CharacterStrings
      - Graphics
      - SparseMatrices
      - TimeAndDates
    - ProblemRelatedFunction
      - BeamProblem
      - AirfoilProblem

At the bottom of the GUI, there are buttons for "Clear" and "Run Query".



GEODISE

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

# Summary & Conclusions

- Purpose and background of KM in EDSO
- A layered semantic infrastructure
  - DL Ontology + Instance store, Ontoview, annotator and advisor
- Life cycle of semantic web base KM
  - Knowledge capture, binding and reuse
- Demonstrations of various tools
  
- Long process
- Preparation of ontologies and semantics instances are important
- Integration is not easy
- Reusing in a smart way is the key (reuse in engineer's favorite PSE)



# Future Work

- Allow engineers to curate knowledge themselves in their favorite PSE (more integration)
  - ☐ WSE,
  - ☐ Matlab
- Synchronization
  - ☐ Engineers' need to Maintain local knowledge of their own
  - ☐ Selectively synchronize local knowledge with centralized knowledge
- Target more resources
  - Workflow
  - Grid fabrics
- More interfaces to the knowledge repository
  - More advanced advice on OptionsMatlab in Matlab
  - Function Browser

