Exploiting Semantics for e-Science on the Semantic Grid

Chen L.[1], Shadbolt N.R.[1], Tao F.[1], Puleston C.[2], Goble C.[2], Cox S.J.[3]
[1] *Department of Electronics and Computer Science*
University of Southampton, Highfield, Southampton, SO17 1BJ, U.K.
{lc, nrs, ft}@ecs.soton.ac.uk
[2] *Department of Computer Science*
University of Manchester, Oxford Road, Manchester, M13 9PL, U.K.
{carole, colin.puleston}@cs.man.ac.uk
[3] *School of Engineering Sciences*
University of Southampton, Highfield, Southampton, SO17 1BJ, U.K.
sjc@soton.ac.uk

Abstract

In this paper we address the problem of exploiting semantics for e-Science [1] *in the emerging future e-Science infrastructure - the Semantic Grid* [2]. *The discussion is taken in the context of Grid enabled optimisation and design search in engineering* ("*Geodise" project*) [3]. *In our work we have developed a semantics-based Grid-enabled computing architecture for Geodise. The architecture has incorporated a service-oriented distributed knowledge management framework for providing semantic and knowledge support. It uses ontologies as the conceptual backbone for information-level and knowledge-level computation. Geodise resources including computational codes, capabilities and knowledge are semantically enriched using ontologies through annotations, thus facilitating seamless access, flexible interoperation and resource sharing on the Grid. We describe ontological engineering work and various approaches to semantic enrichment in Geodise. The semantically enriched content together with the Semantic Grid paradigm have been used as the foundation for the development of an ontology-enabled Geodise problem solving environment prototype (PSE). We have partially implemented the workflow construction environment in the Geodise PSE in which semantics is exploited to describe, discover and compose engineering computation resources for engineering problem-solving.*

1. Introduction

E-Science [1] offers a promising vision of future large scale science over the Internet where the sharing and coordinated use of diverse resources in dynamic, distributed virtual organisations is commonplace. The Grid [4] has been proposed as a fundamental computing infrastructure to support the vision of e-Science, which enables flexible, secure, coordinated resource sharing among dynamic collections of individuals and institutions. Convergence between the Grid and recent developments in web service technologies [5] [11] [12] have seen Grid technologies evolving towards an Open Grid Services Architecture (OGSA) [6]. This sees the Grid as providing an extensible set of services and it enables rapid assembly and disassembly of such services into transient confederations in various ways so that tasks wider than that enabled by the individual components can be accomplished. At this time, a number of Grid applications is being developed [3] [7] [26] and there is a whole raft of middleware that provide core Grid functionality such as Globus [9] and Condor [10]. However there is currently a major gap between these endeavours and the vision of e-Science in which there is a high degree of easy-to-use and seamless automation and in which there are flexible collaborations and computations on a global scale. It has been commonly agreed [2] that the realisation of the e-science vision will rely on how the heterogeneous resources of the Grid, which include data, information, hardware (clusters, servers etc.), software (computation codes), capabilities, and knowledge on how to use these assets, can be effectively described, represented, discovered, pre/post-processed, interchanged, integrated and eventually reused to solve problems.

While a new research field of "Grid intelligence" is emerging to address the above issues, practical work has already been done under different banners [8]. One of them is the Semantic Grid [2], a future e-Science infrastructure that intends to bridge the practice and aspiration divide of the Grid. The Semantic Grid aims to support the full richness of the e-Science vision by considering the requirements of e-Science and the e-Scientist throughout their use of Grid resources (in the widest sense). The enabling technologies that evolve the Grid to the Semantic Grid are the Semantic Web [13] [14] and advanced knowledge technologies [15]. The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is the idea of having data on the Web defined and linked in a way that it can be used for more effective discovery, automation, integration, and reuse across various applications. Advanced knowledge technologies are concerned with the process of scientific knowledge management on the Grid in terms of a life cycle of knowledge-oriented activity that ranges over knowledge acquisition, modelling, retrieval, reuse, publishing and maintenance. It provides a knowledge infrastructure, i.e. tools and methods, to support the management and application of scientific knowledge. In the Semantic Grid knowledge technologies help achieve particular types of goals and objectives through the construction and exploitation of annotated knowledge content.

In this paper we will illustrate how Semantic Grid technologies are being exploited to assist engineers. Engineering design search and optimisation (EDSO) is the process whereby engineering modelling and analysis are exploited to yield improved designs [24]. It involves many tasks such as geometry design, mesh generation, code analysis, optimisation, etc., along with the use of distributed compute and data resources. Each of these tasks can be accomplished by one of a set of computation modules that have a similar function but different performance. Problems with different characteristics may be solved by different sets of tasks and very likely different computation modules. These modules are usually physically distributed under the control of multiple elements in the supply chain. The aim of e-Science in EDSO is to exploit the large-scale distributed computation and data resources on the Grid, which were not accessible before, for engineering design. From work carried out so far it is evident that for such ideas to be realised in practical applications EDSO resources should be described with common vocabulary and relevant metadata so that they can be shared and reused seamlessly. It also proves to be essential to provide knowledge support for domain-specific problem solving, such as the construction of workflows to support an optimisation search, due to the complexity and diverse characteristics of the EDSO domain.

This paper discusses the development and application of the Semantic Grid for e-Science in the context of the Geodise project [3]. We first introduce an integrated architecture for Grid-enabled design search and optimisation in engineering. The distinguishing feature of the architecture is the incorporation of the knowledge and ontology components, which migrates the Grid towards the Semantic Grid. Section 3 describes the core underlying technique of the Semantic Grid, i.e. ontology engineering work including ontology development, representation and deployment. Section 4 presents approaches and mechanisms for semantic enrichment. Section 5 concentrates on the exploitation of semantics for solving EDSO problems. We describe the development of an ontology-enabled Geodise problem solving environment (PSE) that aims to steer the user through the EDSO process such as setting up the problem, resource discovery, composition, executing and post-processing.. We also describe the implementation of the core component of the PSE: i.e. the workflow construction environment. Finally some initial conclusions are drawn from our work on Geodise.

2. The Semantic Grid Architecture for Engineering Design Search and Optimisation

Grid enabled optimisation and design search in engineering (Geodise [3]) is one of the UK e-Science pilot projects. It is intended to enable engineers to carry out engineering design search and optimisation by seamless access to a state-of-the-art collection of optimisation and search tools, industrial strength geometry modelling and meshing tools (ProE, Gambit) and analysis codes (FLUENT), and distributed computing and data resources on the Grid. In addition to this Geodise also aims to aid engineers in the design process by encapsulating and exploiting EDSO domain knowledge and valuable design expertise, thus enabling new designs to be developed more rapidly, or at lower cost. This requirement suggests that a knowledge infrastructure be developed to support the distributed management and application of scientific knowledge on the Grid. Figure 1 shows the Geodise architecture under the Semantic Grid paradigm. This architecture consists of four main components including the Geodise portal, the application service provider, and the optimisation and computation modules. The application service provider caters for both design and analysis tools integrated with support for databases that provide information about previous designs.

The optimisation component provides a variety of optimisation algorithms by which each design may be evaluated in terms of a selected objective function. The computation component calculates values for the objective function that is being optimised. All these components are viewed and implemented as web/Grid services and physically distributed. The user front end of Geodise is the Geodise portal, which allows users to locate and compose services they require, seeking advice as necessary.

Though the four modules described above form the main fabric of the Geodise architecture, i.e. the data, computation and applications, the components that are central to providing knowledge and intelligence for the Grid, and hence play a key role in the evolution of the Grid towards the Semantic Grid, are the ontology, the knowledge repository and the intelligent systems. The ontology component provides a shared, explicit specification of the conceptualisation for the EDSO domain. It consists of common vocabularies to represent domain concepts and the relationships between them. EDSO ontologies allow engineers to describe EDSO resources in a semantically consistent way so that they can be shared and processed by both machines and humans. Ontologies lay down the foundation on which seamless access to heterogeneous distributed resources on the Grid can be achieved. The knowledge repository component is intended to expose accumulated design expertise and/or practices to designers so that new design runs can be conducted based on previous design experience. The EDSO knowledge repository contains the intellectual and knowledge-based assets of the EDSO domain. These assets include domain dependent, problem specific expertise embodied in a set of semantically enriched resources which have been produced and archived by the EDSO designers during previous design runs and can subsequently be reused in various ways to enhance their design capabilities in the future. A typical example of such a repository is the EDSO workflow archive, which will store previous design practices such as problems solved, algorithms used and design solutions.

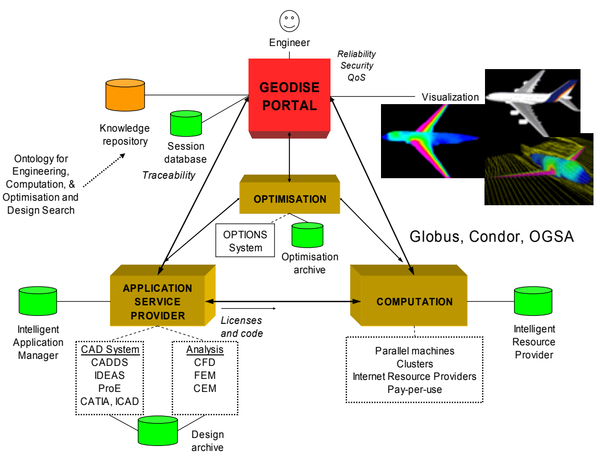

Figure 1: The semantic Grid architecture for engineering design search and optimisation

Intelligent systems aim to provide knowledge-based decision-making support for engineers to develop new designs. This may be done in the analysis codes and resources modules, for example, through an intelligent application manager that makes use of intelligence based on domain knowledge and an intelligent resource provider that makes use of intelligence on top of Grid infrastructure and/or middleware. In Geodise we initially concentrate on exploiting EDSO domain knowledge to facilitate problem solving. Knowledge-based support for decision-making can be provided at multiple knowledge intensive points of the design process and at multiple levels of granularity such as at the process level (what should be done next after a previous task), component level (if next task is optimisation, what methods or algorithm should be chosen from among a suite of 40+ optimisers) and parameter level (if a genetic algorithm optimiser is selected, how to set the control parameters such as population size).

To realise the three key components, i.e. the ontology, knowledge repository and intelligent systems that underpin the idea of the Semantic Grid, we have proposed and developed an integrated service-oriented framework for distributed knowledge management [27] as shown in Figure 2.

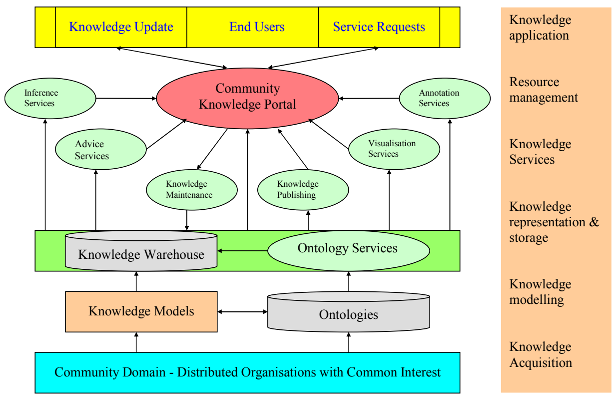

Figure 2: The service-oriented knowledge management framework

In this framework, knowledge about a specific domain is acquired, modelled and represented using a variety of techniques and formalisms. It includes ontologies, knowledge bases and other domain related information. This knowledge is then saved in a knowledge warehouse (or repositories). All activities related to knowledge consumption and supply are realised as knowledge services. Users are provided with a community knowledge portal as the entrance point. The knowledge portal facilitates the use of knowledge with different levels of access control. The framework has a layered modular structure with each component dealing with a specific aspect of the knowledge engineering process in a co-ordinated way. For example, ontologies can be built from knowledge acquisition, and further used to create knowledge bases or to do semantic annotation. These knowledge bases or their associated annotation archives, having been semantically enriched, can then be exploited by the services. These services have mechanisms for querying or searching semantic content so as to facilitate knowledge publishing, use/reuse and maintenance.

As the enabling knowledge infrastructure for the Semantic Grid, the service-oriented knowledge management framework covers all aspects of the knowledge management lifecycle. However, as can be seen from Figure 2, ontologies play a central role for the success of the Semantic Grid and its applications. It serves as a conceptual backbone for automated information access, sharing and reuse, and also enabling semantic-driven knowledge processing on the Semantic Grid [16]. Therefore this paper focuses on the ontological engineering and the use of semantics for e-Science in the context of Geodise. In Geodise ontologies have been created that capture the concepts and terms of the design process, in other words, the common vocabulary used by design engineers to describe what they do. In turn these ontologies are used to describe problem setup, database schemas, computation algorithms, design processes and design results with rich semantics. With the semantic information in place, there will be no communication barriers for people and soft agents. Resources will be transparent for authorised users so that they can be seamlessly shared and aggregated for use. The benefit of conducting EDSO on the Semantic Grid is that engineers, in particular designers wishing to leverage previous expert use of the system, are able to share not only computational resources but also the wider knowledge of the community.

3. Engineering Ontologies for Geodise

We have carried out extensive knowledge acquisition for the EDSO domain using the CommonKADS knowledge engineering methodology [25] and the PC PACK toolkit [31] [27]. The acquired knowledge is modelled as either ontologies or rules in knowledge bases. A set of ontologies has been built to conceptualise the characteristics of the EDSO domain using the OilEd ontology editor [29]. For example, Figure 3 shows the EDSO task ontology. The left panel displays the hierarchical structure of the EDSO tasks, plus all other information types that are relevant to EDSO tasks. The right panel is used to define an individual task by specifying its properties. The definition of a property is actually to establish relationships among concepts within one or multiple ontologies.

Since components in the Geodise architecture are web/Grid services, concepts in the task ontology are in fact different types of services. An instance of a task is a service specified for accomplishing a designated function. This makes us able to adopt DAML-S web services description framework to describe a (EDSO task) service's properties and functionality. To build EDSO-specific task ontology, we have specialised the high-level concepts of DAML-S with terms from EDSO domain ontologies while preserving the DAML-S service description structure such as service profile, service model and service grounding. This makes the EDSO task ontology consistent in both structural description and content semantics. It in turn guarantees that EDSO task ontology can be shared and understood in EDSO community, thus facilitating dynamic automated service discovery and composition.

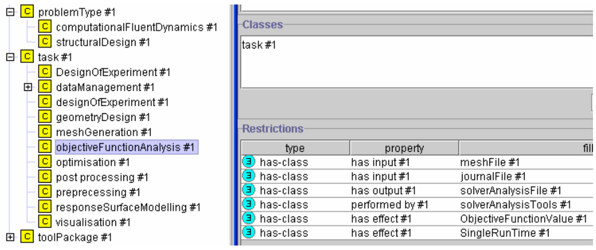

Figure 3: EDSO task ontology

EDSO ontologies are represented in an expressive markup language with well-defined semantics such as DAML+OIL [17] and OWL [18]. DAML+OIL takes an object-oriented-like approach, with the characteristics of the domain being described in terms of classes and properties. It builds upon existing Web standards, such as XML and RDF, and is underpinned by the expressive description logic. It supports the classification of concepts based on their property description - a description-based reasoning capability. Ontological reasoning can be used for subsumption and/or consistency checking. It can also be used as a concept match-maker. For instance, we can retrieve sets of ontological concepts matching some arbitrarily defined queries through classification and subsumption reasoning. Ontological reasoning provides a foundation for semantics-based service discovery as will be seen later when we use EDSO task ontology to perform service composition. Figure 4 shows a fragment of the EDSO task ontology in DAML+OIL.

We have developed ontology services to facilitate the deployment of the EDSO ontologies in Geodise. Ontology services are implemented as a typical SOAP-based web service independent of any specific domain. Therefore it can access any DAML+OIL ontology that is available over the Internet. The ontology service consists of four components: an underlying data model that holds the ontology (the knowledge model) and allows the application to interact with it through a well-defined API, an ontology server that provides access to concepts in an underlying ontology data model and their relationships, the FaCT reasoner [30] that provides reasoning capabilities and a set of user APIs that interface user's applications and the ontology. By using the service's APIs and the FaCT reasoner, common ontological operations, such as subsumption checking, retrieving definitional information, navigating concept hierarchies, and retrieving lexical information, can be performed when required.

```
- <daml:Class rdf:about="http://www.geodise.org/KnowledgeManagement/ontologies/TaskOntology.daml#geometryDesign">
    <rdfs:label>geometryDesign</rdfs:label>
  - <rdfs:comment>
      <![CDATA[  ]]>
    </rdfs:comment>
  - <oiled:creationDate>
      <![CDATA[ 2002-10-21T09:06:51Z  ]]>
    </oiled:creationDate>
  - <oiled:creator>
      <![CDATA[ le  ]]>
    </oiled:creator>
  - <rdfs:subClassOf>
      <daml:Class rdf:about="http://www.geodise.org/KnowledgeManagement/ontologies/TaskOntology.daml#task" />
    </rdfs:subClassOf>
  - <rdfs:subClassOf>
    - <daml:Restriction>
        <daml:onProperty rdf:resource="http://www.geodise.org/KnowledgeManagement/ontologies/TaskOntology.daml#performed by" />
      - <daml:hasClass>
          <daml:Class rdf:about="http://www.geodise.org/KnowledgeManagement/ontologies/TaskOntology.daml#geometryDesignTools" />
        </daml:hasClass>
      </daml:Restriction>
    </rdfs:subClassOf>
  - <rdfs:subClassOf>
    - <daml:Restriction>
        <daml:onProperty rdf:resource="http://www.geodise.org/KnowledgeManagement/ontologies/TaskOntology.daml#has input" />
      - <daml:hasClass>
          <daml:Class rdf:about="http://www.geodise.org/KnowledgeManagement/ontologies/TaskOntology.daml#geometryDefinitionFile" />
        </daml:hasClass>
      </daml:Restriction>
    </rdfs:subClassOf>
  - <rdfs:subClassOf>
    - <daml:Restriction>
        <daml:onProperty rdf:resource="http://www.geodise.org/KnowledgeManagement/ontologies/TaskOntology.daml#has output" />
      - <daml:hasClass>
          <daml:Class rdf:about="http://www.geodise.org/KnowledgeManagement/ontologies/TaskOntology.daml#geometryRepresentationFile" />
        </daml:hasClass>
      </daml:Restriction>
    </rdfs:subClassOf>
</daml:Class>
```

Figure 4: EDSO task ontology – the fragment of the geometry design task

As a standard web service, ontology service itself is a type of knowledge asset and can be accessed, shared, and reused using the service's WSDL. It has been developed using Java technologies and deployed using Apache Tomcat and Axis technologies.

4. Semantic Enrichment for EDSO Resources

Once ontologies are available, the question is how to add semantics to web pages, documents as well as computation resources. Currently there are a number of tools available for semantic enrichment [19] as well as different approaches to semantic resource (web/Grid services) descriptions [20] [21]. In Geodise we have adopted an annotation approach to conducting semantic enrichment for existing EDSO resources. Meanwhile we explore an alternative approach to automatically incorporating semantics into Geodise resources by means of DAML-S-based EDSO tasks.

4.1 Semantic Annotation

Engineering design has been practised for many years, which has accumulated huge, valuable resources including not only knowledge-rich design patterns (concept level information) but also specific design prototypes (individual level information). For example, industrial partners like BAE and Roll-Royce have all the previous designs stored in their own archives. New engines or ships are not designed from scratch but based on design data from previous practices. However, these archives are not, at least at the moment, web-accessible nor semantically enriched. To expose these resources it is necessary to semantically enrich these knowledge-rich design archives, so that previous design practices such as what tools, algorithms and/or control parameters are used can be examined and further explored.

In terms of the EDSO domain characteristics we have chosen a standalone annotation tool, OntoMat-Annotizer [32], to conduct semantic annotation. Initial work focuses on the annotation of engineering design workflow. A typical engineering design usually contains tool and/or algorithm information about the problem definition, mesh generation, code analysis, optimisation setup, the control parameters chosen and how they were specified, as well as the design's general information such as who did it, how long it took and the current status of the design. All of the above information may be derived from the log files that typical engineering design packages use to record a step-by-step activity of how the package was used for a given optimisation run. By semantically enriching these files using terms from the domain ontology, the knowledge contained in these log files can be used to answer such key questions as "what previous designs have been explored and how can one re-use them?".

Figure 5 shows a screenshot in which a design log file from the OPTIONS design package [33] is annotated. The right panel contains the specific design log file. The left panel consists of the EDSO ontology hierarchy, instances and attributes. To annotate, first a fragment of text in the log file is marked-up, then the text is copied and pasted as an instance of a corresponding concept in the ontology, and finally attributes are added to the instance by filling in the primitive data slots and/or referencing other instances. When finished, the selected text is linked to a concept as its instance and also other instances that are used as the restrictions of the selected text. Figure 6 shows the annotation result, in which the log file appears as an HTML document together with a block of semantic content in RDF format, which have been added by the annotation process. The semantically enriched log files can be built into a knowledge repository such as a RDF triple store, which can then be queried, indexed and reused.

4.2 A Mechanism for Automatic Semantic Enrichments

It is commonly accepted [34] that the bottleneck hindering the realisation of the Semantic Web, and hence also the Semantic Grid, is the generation of semantic content. In Geodise, on one hand we create semantic content through annotation. On the other hands, we develop mechanisms and tools for automatic semantic enrichment. The basic idea is that EDSO data/knowledge management infrastructures and design tools should have built-in capabilities that can not only exploit semantics but also support automatic semantic enrichment.

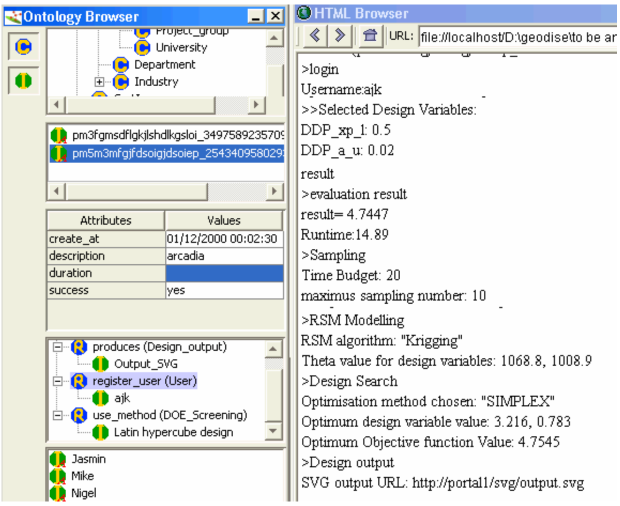

Figure 5: OPTIONS log file annotation using OntoMat

```
<html>
 <head>
  <!-- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#" xmlns="http://protege.stanford.edu/whole#">

  <Design_profile
rdf:about="http://yournamespace.org#pi5kdfkjglfdlkdxmapfsoo_443057830954353405730495709">
    <create_at>
      05/08/2002 15:30:45
    </create_at>
    <description>
      beam
    </description>
    <duration>
      5
    </duration>
    <problem rdf:resource="http://yournamespace.org#beam"/>
```

Figure 6: Semantically annotated log file

Rather than adding semantics to EDSO resources through annotation (the above approach to enriching legacy resources), an alternative approach is first to semantically describe EDSO resources so that when these resources are used to solve design problems the solution will be automatically semantically enriched. A typical application of this approach in Geodise is that we use the EDSO task and domain ontologies to describe EDSO tasks. When we construct an engineering design workflow by composing different EDSO tasks, the workflow is automatically semantically enriched.

An EDSO task is usually a type of function (service), which specifies what the task requires, what it can produce, how it works and also message invocation and binding mechanisms. When a task is semantically described, all such information can be extracted by following the ontological conceptual links. For example, the Code Analysis task in EDSO task ontology has a mesh file as its input and an objective function value as its output. The mesh file is an instance of the Mesh File concept in the ontology and is generated by the Mesh Generation task. The objective function value is an instance of a design variable specified in EDSO domain ontology. Figure 7 shows the semantic description for the Code Analysis task. By following the semantic links among concepts a task can be explicitly described and accessed using a shared vocabulary.

With semantic task descriptions we are able to create a semantics-enriched task archive so that previously performed tasks can be searched and reused in later design runs. Furthermore, by aggregating all semantic tasks at a process level a semantic-enriched design workflow can be achieved automatically as can be seen next section.

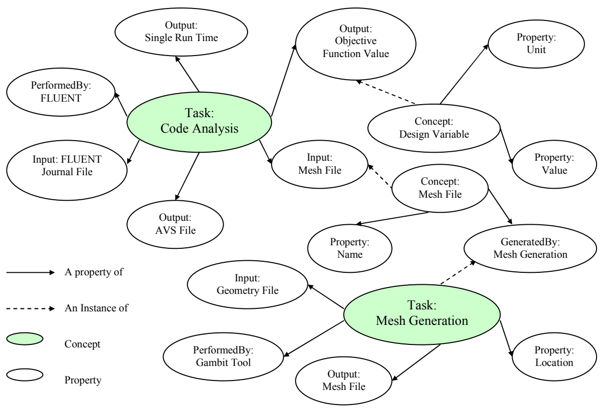

Figure 7: Example semantic task description

In Geodise we are also investigating using EDSO ontologies to create database schema for data/knowledge bases so that the stored data has clear semantics and hence can be shared and reused.

5. Ontology-enabled Geodise Problem Solving Environment (PSE)

A Grid-enabled problem solving environment is an approach to undertaking task specific reasoning on the Grid [35]. It tries to abstract the complexities of accessing the Grid by providing a complete suite of high level tools designed to tackle a particular type of problem. A PSE will allow users to solve particular problems in terms of domain knowledge without having to worry about the complexities of the Grid fabric management. In Geodise, we have developed an ontology-enabled Geodise PSE prototype so that semantics embedded in the semantically described EDSO resources can be exploited for design problem solving. While semantics can play roles in many areas such as service discovery, case-based reasoning and information/knowledge extraction [22] [23], we have focused on providing semantic support for workflow construction.

EDSO is a multi-step process. For example, a scenario for the design optimisation of a typical aero-engine or wing (see Figure 1) is as follows. It is necessary (1) to specify the wing geometry in a parametric form which specifies the permitted operations and constraints for the optimisation process, (2) to generate a mesh for the problem with mesh generation tools, (3) decide which code to use for the analysis, (4) decide the optimisation schedule and optimisation algorithm, and finally (5) execute the optimisation run coupled to the analysis code. Obviously a solution to a specific EDSO problem is a workflow. The problem solving process is actually a process of constructing and executing a workflow. For this reason, we have first of all developed an ontology-assisted workflow construction environment for the Geodise PSE and partially implemented it.

5.1 The PSE Architecture

The framework of the ontology-enabled Geodise PSE is shown in Figure 8. The Geodise Ontologies module contains a set of EDSO ontologies. The Ontology Services module provides a mechanism for users to access and use any ontology on the web. The Computational Web/Grid Services module refers to all EDSO computation tools and/or algorithms that can be used to accomplish a specific task. These resources should be described using EDSO task and domain ontologies, thus service discovery and matching can be achieved through the Semantic-based Web Search Engine.

The core component of the PSE is the Workflow Construction Environment in which users can construct a workflow to solve a specific EDSO problem. The environment consists of a set of tools, including an Ontology Concept Browser, a Workflow Editor, a Workflow Advisor, a Component Editor, an Ontological Reasoner and a State Monitor, to assist the workflow construction process. The Ontology Concept Browser presents the conceptual models of the EDSO tasks in a hierarchical structure. Every task is described with properties, which specify the relations among conceptual task models. Each task can be defined in the Component Editor and used as a primitive building block in the Workflow Editor. The Component Editor provides a dynamically-generated ontology-driven form. Each slot of the form represents a property of a task with an explicitly specified ontological concept type – the semantic link. A task can be defined by specifying every property following the ontological links. Alternatively users can specify the semantic description of a desirable task in the form and then submit it to the Semantic-based Web Search Engine. The Search Engine will return a set of similar tasks that have been performed before and available on the Grid. Users can choose an appropriate task in terms of such criteria as algorithm performance, run time or accuracy of these tasks.

Once a task is defined, it will be added to the workflow in the Workflow Editor. At the same time the task's input and output information is added to the State Monitor of the workflow construction environment. The current state space is then passed onto the Workflow Advisor in which ontological reasoning will be performed based on the task

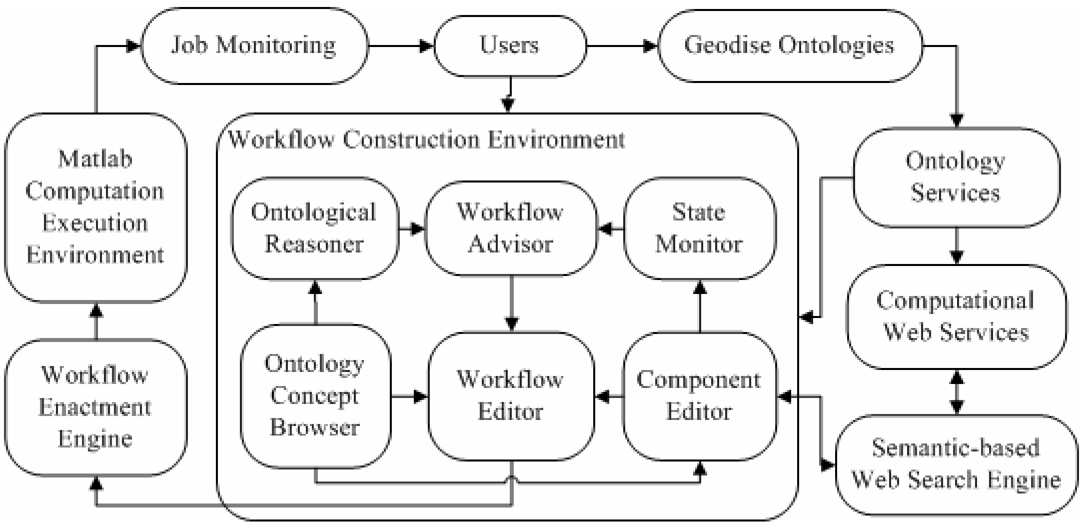

Figure 8: Ontology-enabled EDSO problem solving environment (PSE)

ontology and the state information. The Workflow Advisor will give advice on which task(s) should be undertaken next. Following the advice and repeating the task definition process a workflow can be built up in the Workflow Editor. The Workflow Editor provides editing functions such as modification and deletes functions as well as the graphical representation of tasks and workflow.

As a graphical representation of a workflow is constructed in the Workflow Editor, an underlying representation in appropriate workflow representation formalisms should also be constructed. Therefore, a Workflow Enactment Engine is needed to resolve an abstract specification of a task into a concrete task instance and to establish dynamic binding for service invocation. Apparently the selection of workflow representation formalism and the selection (or development) of workflow enactment engine are tightly coupled. In Geodise we have chosen the Matlab as the Computation Execution Environment [36] due to its popularity and familiarity in engineering community. This means that a workflow will eventually represented as a Matlab script. In such case an enactment engine will degrade to a simple intermediate mapping tool to convert an ontology-represented workflow to a Matlab script file.

The Geodise PSE comprises many other functions such as job management and mobile control through short messaging services. This is beyond the scope of the paper.

5.2 Implementation of the Workflow Construction Environment

The Workflow Construction Environment of the prototype Geodise PSE has been implemented in Java as shown in Figure 9. The left panel is used to specify ontology services and EDSO task ontology. It presents the task service hierarchy through the Ontology Concept Browser. The right panel is the Component Editor. The lower part of this form is used to specify the properties of a task service and the upper part is used to search for task services that match the semantic description defined in the lower part. As at the moment there are no semantic EDSO resources available on the Grid, the Component Editor has been implemented mainly to define a task service directly. The middle panel is the Workflow Editor where services are composed and edited into a workflow. The bottom panel is the State Monitor while the right top panel is used to display knowledge-based advice on service composition, which has been described in detail in [28] [37].

A workflow represents a design solution to a specific EDSO problem. The general procedure for composing services using the workflow construction environment is described step by step below. This process is also illustrated in Figure 9

a). Specify and load the Geodise task service ontology via ontology services in the left panel, and present the Geodise task service hierarchy in the Ontology Concept Browser.

To start a workflow construction process, users need to provide an initial description of the problem at hand, e.g., the problem type and its characteristics. The knowledge-based advice system can then give advice on what to do first to solve the problem via the advice panel. Alternatively a static knowledge support system will suggest to users that what should be done first.

b). Select a suitable primitive task service by navigating the service hierarchy utilising the initial advice, and drag and drop it into the Workflow Editor. A task service description form will appear in the Component Editor for specifying service properties.

c). Define a task service by filling in the property values of the task service description form. Users can follow the ontological concept links from the semantic task service description to define each property. For example, to define a mesh file for objective function analysis task the semantic link of the property "meshFile" will bring you to the "MeshFile" concept in the Geodise task service ontology. Dragging and dropping the concept into the property's input area will in turn open a concept definition dialog box for users to input relevant values. This process is demonstrated by the red dashed arrows in Figure 9 and in compliance with the semantic links depicted in Figure 7.

Alternatively users can partially describe the properties of a service using the form provided by the Component Editor. The semantic-based search engine (at the top of the Component Editor) will enable users to discover similar task services on the Internet; however, this feature has not been implemented at the present time.

d). Once a task service is defined or discovered and selected in the Component Editor, two key operations will follow. First, an instantiated task service with embedded semantics will be added to the Workflow Editor. It will form a single step of the workflow specified for the current problem. This is shown as a yellow box in Figure 9. Second the property information of the task service, in particular, the input, effect and output parameters, will be added to the state memory of the Workflow Construction Environment. These states are, in turn, passed on to the underlying advice system and displayed in the State Monitor. The recommendation on what one should/can do next is subsequently displayed in the knowledge advice panel. This advice guides the response choice of the user with respect to the selection of a suitable service fro the service hierarchy.

e). A database schema for any task service concept can be generated automatically by dragging and dropping the concept from the task service ontology. The instantiated service can then be archived in the database. By collecting all the services created for different problems a semantically-enriched knowledge base can be established over a period of time. This provides semantic content for the search engine to work on for future service discovery.

f). After an arbitrary number of loops, i.e. advising on required services, service discovery/configuration, and service composition, the user can construct a workflow that solves the specific problem. The generated workflow can be submitted to the underlying enactment engine where various resources will be bound together to form an executable. The executable will run in a domain specific execution environment. In Geodise, the executable is a

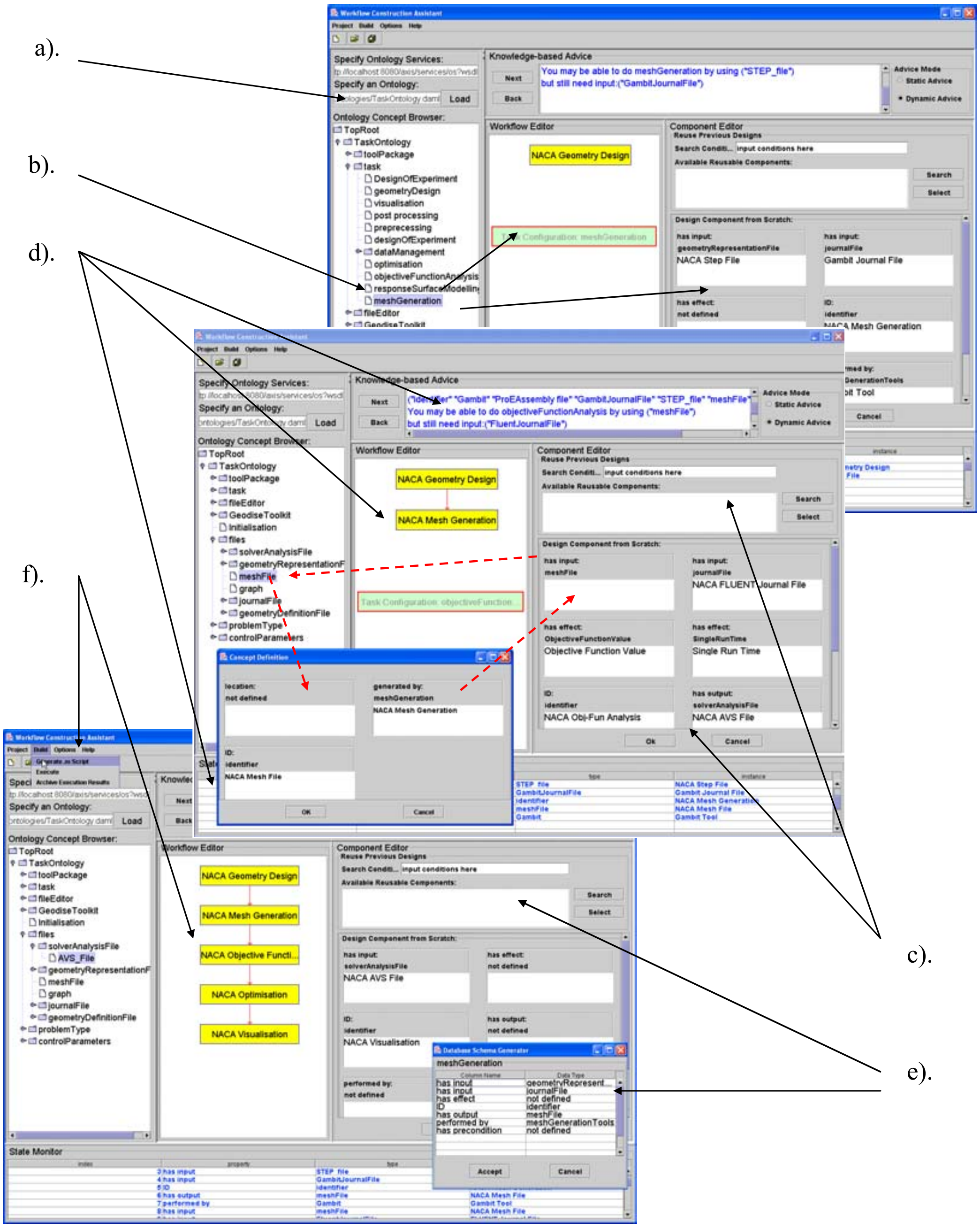

Figure 9: Screenshots of workflow construction environment

Matlab .m script and the execution environment is the Matlab environment. A full discussion of workflow enactment and execution issues is beyond the scope of this paper.

Each time a workflow is constructed and run successfully for a particular design problem, it can be archived in a knowledge repository as a semantically enriched problem/solution record. This facilitates the re-use of previous design results, while avoiding the overhead of manually annotating the solution with respect to semantic content. The Geodise PSE and a suite of high level Grid-enabled EDSO tools have been studied and developed by the Geodise team, which can be found in [3].

6. Conclusions

In this work, we have introduced the Semantic Grid architecture for engineering design search and optimisation. In particular, we present the integrated service-oriented distributed knowledge management framework, which migrate the Grid to the Semantic Grid. We have made full use of the latest semantic web technologies and developed mechanisms and tools to provide semantic support for EDSO. While the context of present research is design search and optimisation but the underlying infrastructure and approaches could be applied to many other types of Grid application. We believe that the Semantic Grid holds great promise for resource sharing, seamlessly automated and flexible collaborations on a widely distributed scale.

Up to now we have developed a number of ontologies that capture and model a substantial amount of EDSO domain knowledge, and ontology services that facilitate the use of ontologies. We have conducted semantic enrichment through annotations and developed mechanisms for automatic semantic enrichment, which are based on semantic service descriptions and ontology-driven data/knowledge management. We have developed a prototype for ontology-enabled Geodise problem solving environment with special emphasis being placed on the workflow construction environment. The implementation of the prototype has demonstrated that semantics can be exploited to facilitate resource description, discovery reuse and composition in EDSO, which enhances problem solving capabilities and also generates automatically more semantic content for future use.

Work on combining the Semantic Web, advanced knowledge technologies and the Grid towards the Semantic Grid and further towards the knowledge Grid is in its infancy. Developing technologies in each underlying field is difficult per se. The integration and synergy of these technologies is complex and challenging. Putting them into real applications like Geodise is even more painful. There are many topics that we have not touched upon such as semantic content integration and storage, semantic web/Grid mining, semantics-based knowledge extraction, the exact nature of Grid intelligence on top of the Grid infrastructure/middleware. Although our work is exploratory and preliminary, the approach, potentials and benefits of the Semantic Grid have been demonstrated through the Geodise example. It becomes clear through the work, that exploiting semantics is not only desirable but necessary and viable for e-Science on the Grid.

7. Acknowledgement

This work is supported by the EPSRC Geodise e-Science pilot (GR/R67705/01). The authors gratefully acknowledge the contributions from and discussion with EPSRC projects MyGrid (GR/R67743/01) and AKT (GR/N15764/01(P)).

References:

[1] Hey, T. and Trefethen, A.E. 2003. The Data Deluge: An e-Science Perspective. To appear in "Grid Computing – Making the Global Infrastructure a Reality", Wiley, January 2003.

[2] Roure, D., Jennings, N. and Shadbolt, N. 2001. Research Agenda for the Future Semantic Grid: A Future e-Science Infrastructure http://www.semanticgrid.org/v1.9/semgrid.pdf

[3] Geodise project http://www.geodise.org/

[4] Foster, I. and Kesselman, C. 1999. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann.

[5] Chinnici, R., Gudgin, M., Moreau, J. and Weerawarana, S.. (2002) Web Services Description Language (WSDL) 1.2, W3C Working Draft, 9 July 2002. http://www.w3.org/TR/wsdl12/.

[6] Foster, I., Kesselman, C., Nick, J. and Tucke S. 2002. The Physiology of the Grid. An Open Grid Service Architecture for Distributed Systems Integration. http://www.globus.org/ogsa/

[7] Allcock, W., Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. Journal of Network and Computer Applications, 23:187-200, 2001. http://grid.web.cern.ch/grid/

[8] Cannataro, M. and Talia, D. 2003, The Knowledge Grid, CACM Vol.46, No1, January 2003, pp.89 - 93

[9] The Globus Project. http://www.globus.org/

[10] The Condor Project. http://www.cs.wisc.edu/condor/

[11] UDDI. The UDDI technical white paper, 2000. http://www.uddi.org/

[12] SOAP 1.2 Working draft, 2001. http://www.w3c.org/TR/2001/WD-soap12-part0-20011217/

[13] Berners-Lee, T., Hendler, J. and Lassila, O. 2001. The Semantic Web, Scientific American.

[14] Goble, C. and Roure, D. D. 2002. Semantic Web and Grid Computing (preprint)

[15] Advanced Knowledge Technologies (AKT) project. http://www.aktors.org/

[16] Maedche, A., Motik, B., Stojanovic, L., Studer, R. and Volz, R. 2002. Ontologies for Enterprise Knowledge Management. IEEE Intelligent Systems, November/December 2002.

[17] Horrocks, I., Harmelen, F., Patel-Schneider, P., Berners-Lee, T., Brickley, D., Connoly, D., Dean, M., Decker, S., Fensel, D., Hayes, P., Hein, J., Hendler, J., Lassila, O., McGuinness, D. and Stein, L. A. 2001. DAML+OIL. http://www.daml.org/2001/03/daml+oil-index.html.

[18] Dean, M., Connolly, D., Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. and Stein, L. A. 2002. Web Ontology Language (OWL) Reference Version 1.0. W3C Working Draft. http://www.w3.org/TR/2002/WD-owl-ref-20021112/.

[19] Annotation and Authoring for Semantic Web. http://annotation.semanticweb.org/org/

[20] DAML Services Coalition. 2002. DAML-S: Web Service Description for the SemanticWeb. In The First International Semantic Web Conference (ISWC).

[21] Tuecke, S., Czajkowski, K. Foster, I., Frey, J., Graham S. and Kesselman C. 2002. Grid Service Specification Draft. http://www.globus.org/research/papers/gsspec.pdf

[22] Sirin, E., Hendler, J. and Parsia, B. 2002. Semi-automatic Composition of Web Services Using Semantic Descriptions. "Web Services: Modeling, Architecture and Infrastructure" workshop in conjunction with ICEIS2003.

[23] Gibbins, N., Harris, S. and Shadbolt, N. 2003. Agent-based Semantic Web Services. Accepted to The Twelfth International World Wide Web Conference, Budapest, Hungary.

[24] Siddall, J.N. 1982. Optimal Engineering Design: Principles and Applications. Marcel Dekker, Inc., New York and Basel.

[25] Schreiber S., Akkermans H., Anjewierden A., Hoog R., and Shadbolt N. 1999. Knowledge Engineering and Management. The MIT Press, London.

[26] The MyGrid project http://mygrid.man.ac.uk/index.shtml

[27] Chen, L., Cox, S.J., Goble, C., Keane, A.J., Roberts, A., Shadbolt, N.R., Smart, P., Tao, F. 2002. Engineering Knowledge for Engineering Grid Applications. Proceedings of Euroweb 2002 Conference, The Web and the GRID: from e-science to e-business, Oxford, UK, pp12-25.

[28] Tao, F., Chen, L., Shadbolt, N.R., Pound, G. and Cox, S.J. 2003. Towards the Semantic Grid: Putting Knowledge to Work in Design Optimisation. Proceeding of the 3rd International Conference on Knowledge Management I-KNOW '03, Industry meets Science, pp.551-562, Austria.

[29] Bechhofer S, Horrocks H, Goble C, Stevens R. 2001. OilEd: a Reason-able Ontology Editor for the Semantic Web. Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence. Springer-Verlag LNAI Vol. 2174, pp 396--408.

[30] Horrocks, I., Sattler, U. and Tobies, S. 1999. Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99), no.1705 in Lecture Notes in Artificial Intelligence, pp.161-180. Springer-Verlag.

[31] PC PACK Knowledge tool http://www.epistemics.co.uk/products/pcpack/

[32] OntMat-Annotizer Annotation Tool. http://km.aifb.uni-karlsruhe.de/annotation/ontomat.html

[33] Keane, AJ. OPTIONS Design Exploration System http://www.soton.ac.uk/~ajk/options/welcome.html

[34] Hendler J. 2003. Science and the Semantic Web. Policy Forum, Science, 24 Jan.

[35] Laszewski, G. von., Foster, I., Gawor, J., Lane, P., Rehn, N. and Russell, M. 2001. Designing Grid-based Problem Solving Environments and Portals. The 34th Hawaiian International Conference on System Science.

[36] Pound, G., Eres, H., Wason, J., Jiao, Z., Keane, A.J., and Cox, S.J. 2003. A Grid-enabled Problem Solving Environment (PSE) for Design Optimisation within Matlab. Accepted paper – International Parallel and distributed Processing Symposium IPDPS-2003, Nice, France.

[37] Chen, L., Shadbolt, N.R., Goble, C., Tao, F, Cox, S.J., Puleston, C. and Smart, P. 2003. Towards a Knowledge-based Approach to Semantic Service Composition. Submitted to the 2nd International Semantic Web Conference (ISWC2003), Florida, USA.