Semantics-assisted Problem Solving on the Semantic Grid

Chen L.[1], Shadbolt N.R.[1], Goble C.[2], Tao F.[1], Puleston 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.u*k

Abstract

In this paper we propose a distributed knowledge management framework for semantics and knowledge creation, population and reuse on the Grid. Its objective is to evolve the Grid towards the Semantic Grid with the ultimate purpose of facilitating problem solving in e-Science. The framework uses ontology as the conceptual backbone and adopts the service-oriented computing paradigm for information-level and knowledge-level computation. We further present a semantics-based approach to problem solving, which exploits the rich semantic information of grid resource descriptions for resource discovery, instantiation and composition. The framework and approach has been applied to a UK e-Science project - Grid Enabled Engineering Design Search and Optimisation in Engineering (GEODISE). An ontology-enabled Problem Solving Environment (PSE) has been developed in GEODISE to leverage the semantic content of GEODISE resources and the Semantic Grid infrastructure for engineering design. Implementation and initial experimental results are reported.

Key words: Semantic Grid, e-Science, engineering design, ontology, knowledge management, problem solving

1. Introduction

E-Science [Hey 2003] 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 [Foster 1999] 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 such as UDDI (http://www.uddi.org/), SOAP[1] and WSDL have seen Grid technologies evolving towards an Open Grid Services Architecture (OGSA) [Foster 2002]. 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

[1] SOAP, WSDL, RDFS and OWL are W3C standards, which can be found in http://www.w3c.org.

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 such as TereGrid (http://www.teragrid.org), AstroGrid (http://www.astrogrid.org) and DataGrid [Allcock 2001] and there is a whole raft of middleware that provide core Grid functionality such as Globus (http://www.globus.org) and Condor (http://www.cs.wisc.edu/condor). 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 that the realisation of the e-science vision will rely on how the heterogeneous resources of the Grid can be effectively described, represented, discovered, pre/post-processed, interchanged, integrated and eventually reused to solve problems.

To bridge the practice and aspiration divide of the Grid researchers have outlined a vision of the future e-Science infrastructure, the so-called Semantic Grid [De Roure 2003] [Goble 2002]. The Semantic Grid aims to support the full richness of the e-Science vision by considering the requirements of e-Science and the e-Scientists throughout their use of Grid resources in the widest sense. The enabling technologies of the Semantic Grid are the Semantic Web [Berners-Lee 2001] and advanced knowledge technologies (http://www.aktors.org). 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. On the Semantic Grid knowledge technologies help achieve particular types of goals and objectives through the construction and exploitation of annotated knowledge content.

Engineering Design Search and Optimisation (EDSO) is the process whereby engineering modelling and analysis are exploited to yield improved designs [Siddall 1982]. 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 may 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 of the Semantic Grid and its application for e-Science in the context of the GEODISE project (http://www.geodise.org). We first introduce a service-oriented distributed knowledge management framework, which evolves the Grid towards the Semantic Grid. Section 3 presents a semantics-based approach to problem solving on the Semantic Grid. Section 4 describes grid enabled optimisation and design search in engineering and an ontology-enabled PSE based on the Semantic Grid infrastructure, which aims to steer engineers through the EDSO process. Section 5 covers the implementation details of the Semantic Grid infrastructure, including ontology engineering, semantic enrichment and knowledge repository. We describe problem solving processes in GEODISE PSE and present initial experimental results in Section 6. Section 7 discusses related work. We conclude the paper and point out future work in Section 8.

2. Towards the Semantic Grid

We have proposed an integrated service-oriented framework for distributed knowledge management, as shown in Figure 1, with the ultimate purpose of providing information-level and knowledge-level computation to underpin the Semantic Grid. In this framework, information and knowledge about a specific domain are acquired, modelled and represented using a variety of techniques and formalisms, which may include ontologies, production rules and other domain related information. They are then saved in knowledge warehouses and information archives. A community knowledge portal is provided as an entrance point to facilitate knowledge retrieval 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. The compelling features of the framework are as follows:

- The framework deals with knowledge engineering and management in an integrated coordinated way. Knowledge engineering and management [Schreiber 1999] attempt to meet the six challenges of knowledge management life cycle – namely, those of acquiring, modelling, retrieving, reusing, publishing and maintaining knowledge. Although research has been carried out on each aspect of this life cycle, in the past each aspect of the life cycle was often developed in isolation from the others. For example, knowledge acquisition was often done with little consideration as to how it might be published or used. At the same time, knowledge publishing paid little attention to how knowledge was acquired or modelled.

Unlike traditional knowledge engineering practices that concentrate on separate capabilities, this framework tackles each individual issue in a much more coordinated way so that results from one piece of work can be used for another in an appropriate form. For example, ontologies from knowledge acquisitions can be used for annotation and/or ontology-enabled data mining. These ontologies and/or their associated annotations can be further exploited by other services such as inference services and recommender services.

- Ontology plays a central fundamental role in the interaction and interoperation of the knowledge management activities. It provides a common medium for inter-agent information transfer, which is applicable to both humans and machines. An ontology is actually a web/grid-oriented knowledge model for a specific problem domain. It can be

used for creating knowledge entities via semantic annotations, resource discovery and many other usages. Ontologies serve as a conceptual backbone for automated information access, sharing and reuse, and also enabling semantic-driven knowledge processing on the Semantic Grid [Maedche 2003].

- The framework adopts a service-oriented approach to distributed knowledge management. It implements all activities related to knowledge consumption and supply as knowledge services. This approach has two main advantages. First it fits naturally into the OGSA-based Grid infrastructure, thus easy and straightforward to evolve the Grid to the Semantic Grid. Secondly it makes both knowledge publishing and retrieval accessible on the Grid by wrapping up both legacy codes, standalone KBSs and new knowledge assets as web services, thus facilitating knowledge sharing and reuse.

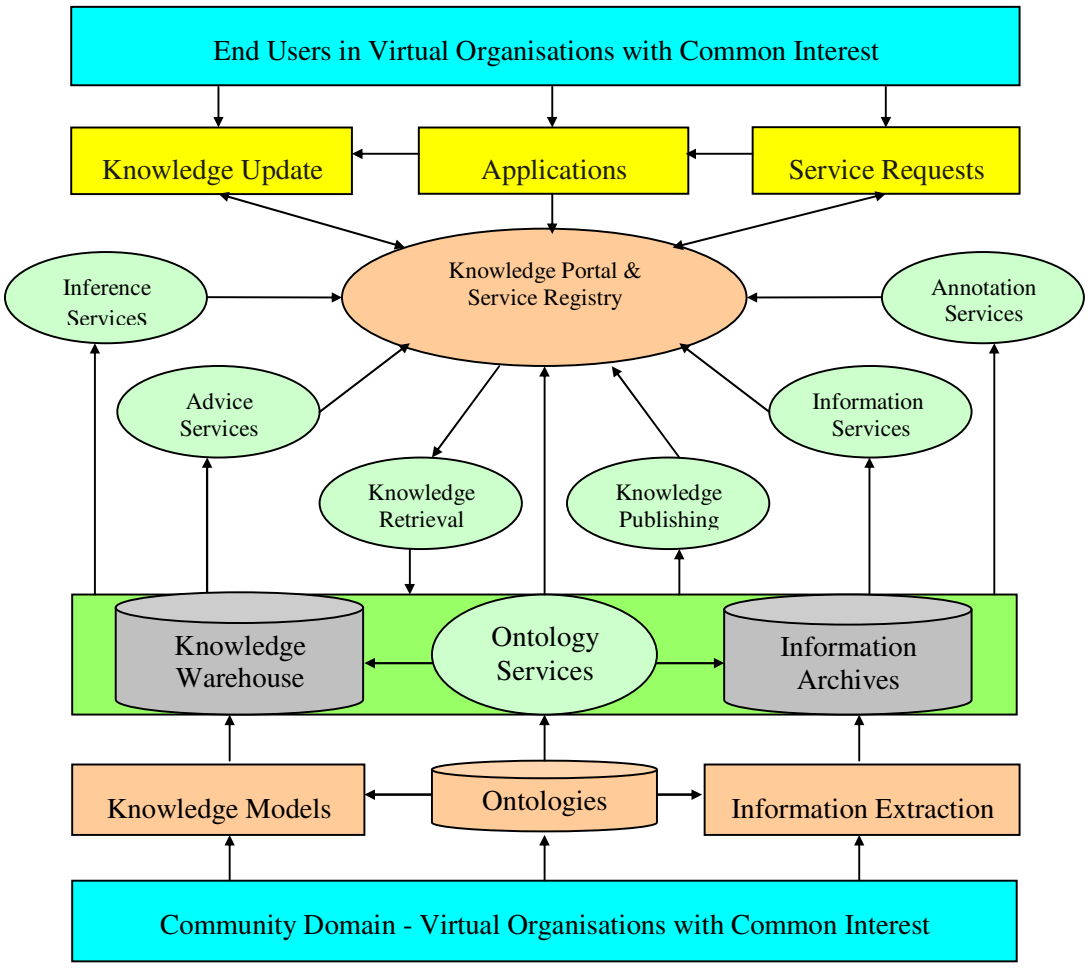


Figure 1: The service-oriented knowledge management architecture

- The layered modular framework makes grid-oriented knowledge management robust and scalable. While different knowledge management tasks are coupled together, their interactions are not hardwired. Each component deals with different tasks and can make use of different techniques and tools. Each of them can be updated whilst others are kept intact. This type of componentization makes the framework robust and scalable. It

means that new techniques and tools can be adopted at any time, and that the knowledge management system will continue working even if some of its components should fail or become unavailable. Knowledge can be added into knowledge warehouse at any time and so do knowledge services. For example, a data mining service may be added later for automated information extraction and dynamic update of knowledge repositories. This feature makes the framework flexible and extensible, thus ideal for supporting the realisation of the Semantic Grid on which resources in Virtual Organisations (VOs) are dynamically joining, aggregating, disbanding and disappearing.

As can be seen in Figure 1 the framework consists of a number of components, each fulfilling a specific functionality supporting the Semantic Grid. While new services can be added any time enabled by the extensible architecture, a set of core services have been identified indispensable for the realisation of the Semantic Grid. They are described as follows:

- Ontology services provide access to concepts, their properties and relationships in an underlying ontology data model. Services include extending the ontology and querying it by returning the properties and relationships (such as parents or children) of a concept. Ontology services are implemented as a typical SOAP-based web service that can be accessed using WSDL. As a result ontology services enable users to access and share ontologies available on the Grid no matter where it is.
- Annotation services associate grid resources with their semantic interface and metadata via ontologies in order to attach semantics to those resources. Annotation can be done either by describing resources with ontological concepts or by directly creating instances of an ontological concept if the resources of a specific domain can be classified into a concept hierarchy. As the essence of the Semantic Grid is to give grid resources well-defined meaning and populate the Grid with such semantically enriched resources, therefore, annotation is vital and often viewed as the bottleneck of the Semantic Web/Grid.
- Knowledge publishing and warehousing are services that publish and store knowledge assets for sharing and reuse on the Grid. Traditionally, knowledge bases have been small, standalone, locally stored, and consumed by the developer. Grid-oriented knowledge repository may be federated, distributed, developed and consumed by multiple grid users. It is necessary for knowledge to be first published on the Grid and then make use of the latest database technology for optimised management, which could include the use of ontologies for indexing the source databases.
- Inference services provide reasoning capabilities over various knowledge entities in the knowledge repository. It can be used together with ontology services to perform common ontological operations such as concept classification and subsumption checking. Reasoning is also needed in the semantic matching of resource interfaces, which can in turn be used for service classification, indexing, service discovery and composition.

In addition, the knowledge portal, the entrance point to the integrated knowledge management system, provides a security infrastructure for authentication and authorisation, so that knowledge can be used and updated in a controlled way. Knowledge publishing allows users to register new knowledge assets and service information via the portal. Through the knowledge portal knowledge can be accessed and retrieved in the same way we browse the Web.

3. A Semantics-based Approach to Problem Solving

Problem solving in science and engineering often involves composing various resources into a workflow either manually or automatically to realize a particular experiment or series of computations. In the service-oriented Grid computing paradigm resources including computation software, capabilities and knowledge are represented and exposed as web services. Therefore, problem solving on the Grid amounts to discovering services and composing them into a workflow. Some domains such as a supermarket demand-supply chain have a fixed flow of process and stationery bindings between services. However, for most scientific disciplines a workflow is both domain-specific and problem-dependent. The appropriate selection of services at a specific point of a workflow often depends on the results of executing preceding services. Moreover, the selection of a service from a set of competing services with similar capabilities is usually determined by the exact nature of a problem as well as the performances of available services. As a result, it is not practical to specify, a priori, the precise sequence of services for a problem. The successful orchestration of component services into a valid workflow is heavily dependent on bodies of domain knowledge.

We have proposed a semantics-based approach to problem solving on the Semantic Grid. It exploits the semantic content of resources' semantic descriptions for service discovery, composition and configuration during the process of workflow construction. The cornerstone of this approach is based on the built-in semantics inherited from the logical inference axioms of ontology representation languages such as RDFS and OWL. Using a very expressive Description Logic (DL) based inference engine, such as Racer system [Haarslev 2003], users can perform both terminological and assertion reasoning over very large scale knowledge bases. The semantics-based problem solving mechanisms are described below.

Service discovery: To build a workflow for a specific problem, users often need to know what resources are required and how to find them. These are not trivial problems, in particular, for novice users. Grid applications so far have proven that to discover the right resources is hard with regards to the heterogeneity of distributed dynamic grid resources. With resources semantically enriched users can perform semantic resource discovery based on resources' semantic metadata i.e. the conceptual links between services and their metadata. Semantic service discovery usually uses the semantic metadata of a service profile such as the author, organization, the algorithm used, version and usage to set up query criteria for the required resource. A DL-based reasoning engine will perform reasoning over semantic grid resources to discover those resources that are commensurate with their information processing goals.

Service composition advice: One frequently asked question during workflow construction is: what should be done next. As services can only be joined together to form a valid workflow when their interfaces' semantics matches each other, this indicates that the semantics of service interfaces can be used to guide service composition. When building a workflow, each time a service is added to the workflow users can obtain its outputs' semantic information. Using the outputs' type information users can use a DL reasoner to carry semantic matching and reasoning against the underlying service repository. The reasoner will return a list of semantically compatible services. Users can

examine these suggested services individually to get further information until an appropriate service is chosen.

Service configuration advice: It is often the case that there are many services available for a task and each service may have different performances with different configurations. For example, in the domain of engineering design search and optimization there are over a hundred different optimization methods, each of which is geared to solving a specific type of engineering problem. Even with a single method, different configurations of control parameters may produce very different results. Knowledge of selecting the correct method for a particular situation as well as the appropriate configuration of method parameters is an important feature of expert-level performance and a vital ingredient of problem-solving success.

Configuration knowledge about services can be captured, modelled and embedded in the service semantic descriptions during annotation. This may include some logical and mathematical rules that establish dependencies and relationships between configuration parameters and problem characteristics or the state variables of the workflow. When configuring a service users will be able to extract such configuration information by following ontological links. This includes, for example, what are the type and default values of a variable, what its dependency variables are and how it affects other variables. Users can simply use default values and/or the suggested values which are usually generated based on the problem characteristics and/or dynamic configuration of other services of the workflow. Configuration advice can be provided as just-in-time hands-on tips, auto-completion values and alternatives during service configuration.

In summary, semantics-based problem solving approach can help discover required services accurately and has the advantage of providing specific advice at multiple levels of granularity during service composition process. At the highest level, it can help determine what kind of service is required against a contextual backdrop that includes problem-solving goals and procedural knowledge. Once all the services that can fulfil the required functionality are discovered, semantics-based advice system can recommend an appropriate service, taking into account both problem characteristics and performance considerations. More specialized, in-depth advice can also be given, for example, how to initialize and configure the control parameters of a service. Such knowledge is usually only available from experienced users or domain experts.

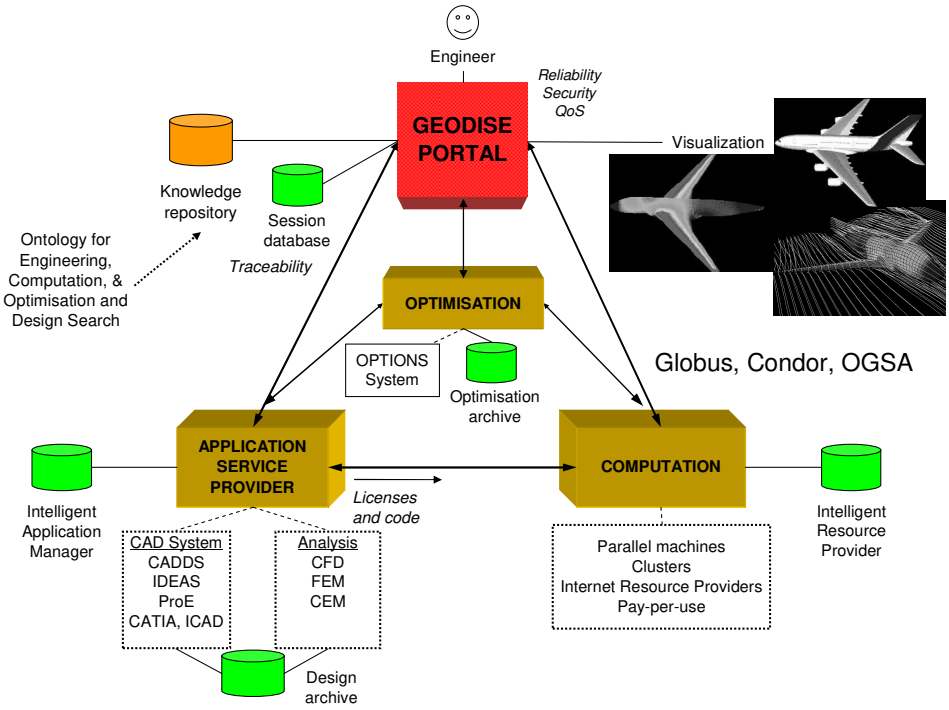
The semantics-based problem solving approach provides an effective way to reuse resources on the Semantic Grid, which can be applied to any application domain. Service discovery and advising on workflow construction is just some of many applications that benefit from the Semantic Grid infrastructure. Semantic metadata can be utilised for other purposes such as provenance and trust.

4. Ontology-enabled Problem Solving Environment (PSE) in GEODISE

4.1 GEODISE overview

Grid enabled optimisation and design search in engineering (GEODISE) 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

and analysis codes, 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. The dual objectives have determined that GEODISE will be based on the Semantic Grid paradigm as shown in Figure 2. 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.

Figure 2: GEODISE architecture on the Semantic Grid

The optimisation module provides a variety of optimisation algorithms by which each design can be evaluated in terms of a selected objective function. The computation module calculates values for the objective function that is being optimised. All these modules are viewed and implemented as Web/Grid services and geographically 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, the components that are central to the realisation and utilisation of the Semantic Grid, i.e. providing semantic and knowledge support for GEODISE, 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 their relationships. 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. These resources have been produced and archived by the EDSO designers during previous design runs so that they can subsequently be reused in various ways to enhance their design capabilities in the future.

Intelligent systems aim to provide semantics/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/or 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), 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).

4.2 GEODISE PSE

To exploit the rich semantic content of the semantically enriched EDSO resources we have developed an ontology-enabled PSE in GEODISE. A grid-enabled PSE is an approach to undertaking task specific reasoning on the Grid [Laszewski 2001]. 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 grid fabrics.

The ontology-enabled GEODISE PSE, see Figure 3, uses ontologies as the underlying modelling, representation and communication tokens, thus facilitating interoperability and enhancing resource sharing. The PSE consists of a number of functionality modules. Each module interacts with others and overall they form an integrated problem solving environment. The Ontologies module contains a set of EDSO ontologies, which are used for service classification, annotation and semantic reasoning. The Ontology Services module provides a mechanism for users to access and use any ontology on the Web. The Semantic Computational Services module refers to all type of EDSO resources such as computation tools and algorithms that can be used to accomplish a specific task. These resources are semantically enriched using EDSO ontologies. Service discovery is accomplished via the Semantics-based Web Search Engine (SWSE). It is realized through reasoners such as Racer or FaCT [Horrocks 1999] acting on the semantic descriptions of services. The services that fulfil users' requirements will be displayed in the Classified Service Browser for service composition.

The core component of the PSE is the Workflow Construction Environment (WCE) in which users can compose services into a workflow to solve a specific EDSO problem. The main components of WCE are:

- The **Classified Service Browser** presents a hierarchy of classified services in terms of EDSO service ontology, which comes either from the SWSE or a local knowledge repository.
- The **Workflow Advisor** provides recommendation on service composition and configuration based on semantic matching as described in Section 3. Users can obtain advice in two ways. First, a user may request advice according to his/her epistemic needs and requirements during workflow construction processes. Secondly, a software agent can be used to monitor a service composition process as it unfolds, and provides advice or recommendations along the way. Both approaches need to monitor the progress of workflow composition processes and collect snapshots of states when advice is requested.

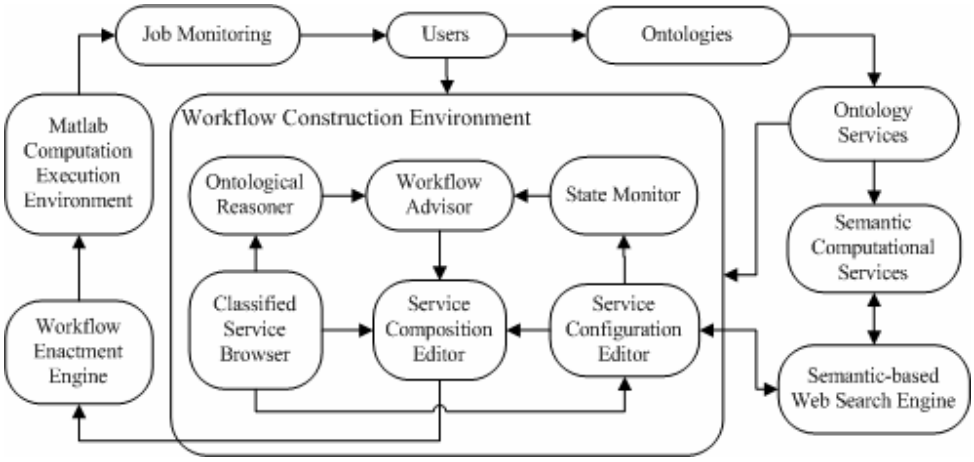


Figure 3: Ontology-enabled GEODISE PSE

- The **State Monitor** monitors workflow construction processes, and collects and maintains a workflow state space.
- The **Ontological Reasoner** performs terminological and assertion reasoning over knowledge repository for service discovery, composition and configuration, which is usually accomplished by reference to ontology and/or knowledge repository.
- The **Service Configuration Editor** is an ontology-driven dynamically-generated form that contains a service's attributes and their values. User can edit and modify these attributes' values either based on their own expertise or the advice.
- The **Service Composition Editor** is a graphical workspace for service composition. It provides various functionalities for service manipulation, which include to move, select, edit, delete and link services.

During workflow construction processes, once a service is discovered and added to a workflow in the Service Composition Editor, the State Monitor will collect and maintain the service's input and output information. The current state space of the workflow will be passed onto the Workflow Advisor. The Workflow Advisor will make use of the Ontological Reasoner to perform ontological reasoning against the underlying knowledge

repository. Relevant advice such as which service should be used next or how to choose a parameter for service configuration will be returned to WCE. Users can select one of the suggested services and further configure it according to recommendations. Repeating this process a workflow can be built quickly and requiring less domain expertise.

As a graphical representation of a workflow is constructed in the Service Composition Editor, an underlying workflow representation in appropriate representation formalisms should also be constructed. The Workflow Enactment Engine will resolve an abstract specification of a service into a concrete service instance and to establish dynamic binding for service invocation. Apparently the workflow representation formalism and the workflow enactment engine are tightly coupled. In GEODISE we have chosen the Matlab as the Computation Execution Environment [Eres 2004] 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 innovation of the GEODISE PSE is that it is based on the Semantic Grid infrastructure on which domain knowledge is modelled using web/grid oriented knowledge models – ontologies, and further embedded and published in resources' semantic descriptions. The use of ontology enhances resource interoperability and machine processing capability, thus enabling GEODISE PSE to access and reuse heterogeneous distributed grid resources. The power of the PSE comes from the semantics-based problem solving approach, which extracts and exploits the domain knowledge incorporated in the resources' semantic descriptions for service composition and configuration. The success of the GEODISE PSE is based on a number of tools and middleware that handle and manipulate semantic content on the Grid, which will be described in next section.

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. Implementation of the Semantic Grid in GEODISE

This section describes the implementation details of some key components of the Semantic Grid.

5.1 EDSO ontologies

We have carried out extensive knowledge acquisition for the EDSO domain using the CommonKADS knowledge engineering methodology [Schreiber 1999] and the PC PACK toolkit (http://www.epistemics.co.uk/products/pcpack) [Chen 2002]. The acquired knowledge is modelled as either ontologies or production rules and stored in a knowledge repository.

As GEODISE uses Matlab as its problem solving environment, the main EDSO resources are Matlab function scripts - a type of high-level computation programs that can accomplish various tasks for engineering design search and optimization by execution in Matlab environment. It has been identified that the key issues to achieving GEODISE objectives are (1) how to add rich metadata to these Matlab functions, (2) how

to semantically enrich them, and (3) how to allow sophisticated reasoning and query capabilities over them. For this purpose we have concentrated on the development of an EDSO function ontology based on the DAML-S (http://www.daml.org/services) ontology. We use function profiles to describe function metadata. Semantic descriptions are generated when linking resource metadata and interface (inputs/outputs) with underlying EDSO domain concepts [Chen 2004]. EDSO function ontology is represented in OWL.

Figure 4 shows the EDSO function ontology. The left hand panel displays the hierarchical structure of the EDSO function-related concepts. The right hand panel is used to define an individual function by specifying its properties. The definition of a property is actually to establish relationships among concepts within one or multiple ontologies.

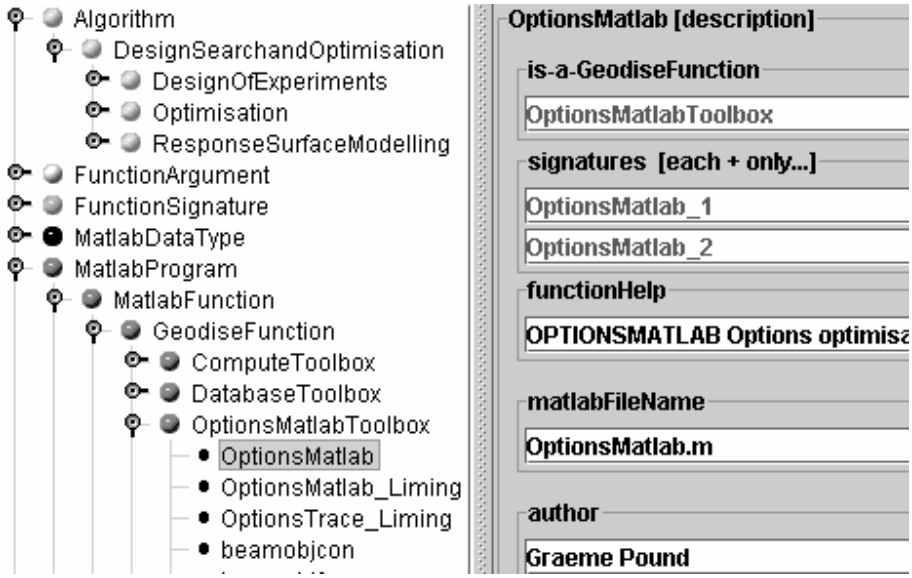


Figure 4: EDSO function ontology

5.2 Ontology services

We have developed ontology services to facilitate the deployment and access of ontologies in GEODISE. The ontology services consist 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 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.

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.

5.3 Annotation Services

We have developed a tool, called Function Annotator [Chen 2004], to attach metadata and semantics to a function's descriptions and publish them into a resource repository. Figure 5 shows the GUI of the Function Annotator, which consists of an Ontology Browser, an Annotation Palette and a Function Browser. The Ontology Browser in the left hand column contains a concept hierarchy (Panel 1) and a function hierarchy (Panel 2). The concept hierarchy presents the terms, relations and hierarchy of the function ontology. It is used for users to browse and choose suitable concepts for function descriptions. The function hierarchy displays available ontologically described functions under different function categories. These semantically enriched functions will be retrieved from a backend resource repository on a user's demand. They can be modified, edited and reused to generate new function descriptions.

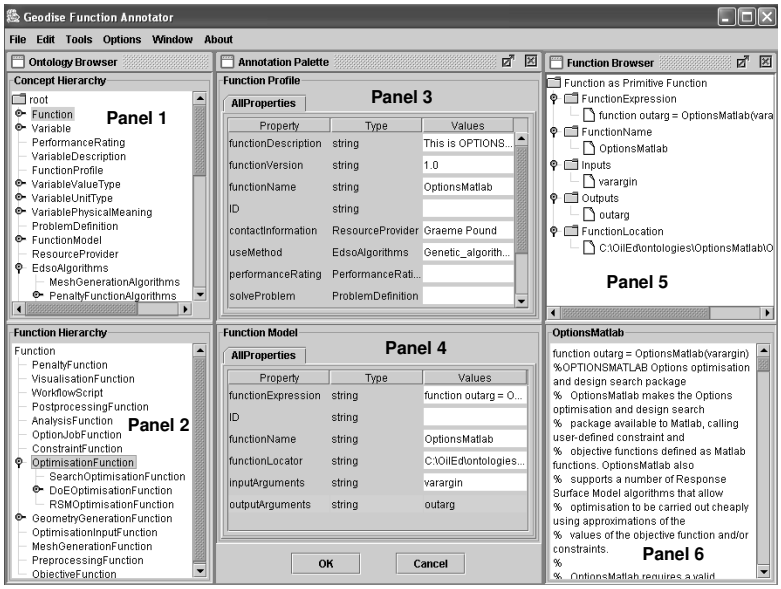


Figure 5. Function annotator interface

The right hand column of the GUI is the Function Browser, which is used to load Grid resources for semantic descriptions. In GEODISE the resources are mainly Matlab functions. We have provided a parsing capability to facilitate automatic information extraction based on the Matlab function interface and helper documentation. The extracted information, which includes a function's inputs/outputs as well as other metadata such as copyright, authors and summary, is listed in a tree structure in the top window of the Function Browser (Panel 5). The bottom window (Panel 6) displays the source code of a Matlab function that gives users more flexibility for annotation. In particular, for compound functions such as Matlab scripts users can markup, copy and paste specific information from the source into the Annotation Palette.

The Annotation Palette in the middle column of the GUI is where ontological description takes place. It consists of two panels, i.e. the Function Profile at the top

(Panel 3) and the Function Model at the bottom (Panel 4). Function Profile contains two types of metadata. One is about function metadata such as what a function does, what it requires from and provides for users as well as information about authors, version, used methods, required preconditions etc. These metadata are specified using formal ontological concepts. The second type of metadata is about function input/output interface such as type and default values. Function Model is used to hold information on how a function works and how it can be invoked. This includes input/output arguments, location and expression signatures. For scripts it could contain information on embedded functions as well as their sequential details.

To annotate a function users can fill in relevant ontology-driven forms in the Annotation Palette by selecting appropriate concepts from specific areas of the ontology (presented as either lists or sub-hierarchies), and where appropriate, specifying concrete values. For example an input could be specified as a high-level concept type, or an instantiated object or a concrete value. Once a user completes the forms, all inputs will be collected and combined with some additional automatically generated metadata, such as creator-name, date-of-creation and instance-ID. This information will form the semantic descriptions of the function and be saved in the function archive.

5.4 Knowledge repository

There are three different mechanisms to store semantic knowledge about a resource. First it can be added into the original resource with an embedded piece of descriptions. Semantic web community usually uses this approach to attach semantic information to web pages [Handschuh 2003]. Second, knowledge can be saved in a separate file in the same location as the resource. Thirdly knowledge can be archived in a central knowledge repository separate from resources. In the context of Grid computing resources are supposed to be owned by dynamic VOs and geographically located. The purpose of semantic descriptions for resources is to facilitate resource discovery, sharing and reuse. To some extent a knowledge repository should also be also a resource registry. From this perspective we decided to build a central knowledge repository for distributed resources.

When grid resources are modelled using ontologies and represented in OWL, knowledge objects will be generated as OWL individuals that are independent of the original resource formats and/or providers. In such cases both knowledge objects and knowledge structures will be save together in an ontology file that is actually a knowledge base. Applications can then consume knowledge by accessing the ontology file and carrying out DL reasoning over individuals. Unfortunately, existing technologies, either Racer's assertion reasoning or FaCT's terminological reasoning over pseudo-concepts, fail to scale up to the size of over 100,000s individuals that is usually required by real scientific applications on the Grid.

We have adopted the instance store technology [Horrocks 2004] to tackle this problem. The instance store uses a relational database such as MySQL and Oracle as permanent storage media and a DL reasoner to support reasoning. This means that assertions over individuals are stored in a database, together with information inferred using a DL reasoner over the position in the ontological taxonomy of their corresponding descriptions. The DL-based reasoner deals purely with terminological reasoning functionality. As terminologies are fairly restrictive there will be no size limitation

problem. Furthermore, pure terminological reasoning will significantly reduce reasoning cost while maintaining soundness and completeness. Retrieving individuals is then a combination of query against that database and subsumption and classification requests to the reasoner.

5.5 Service discovery and retrieval

Resource discovery and seamless access is the key to the success of the e-Science where the sharing and reuse of dynamic resources are commonplace. One appealing feature of the Semantic Grid is that it enables the semantic service discovery. Current WSDL-based web/grid service descriptions are more concerned with the signature of a service, i.e. the identifiers of the service and its parameters. It is usually impossible for software agents to figure out the precise meaning of service identifiers and functionality provided by the service. The ontological description of functions allows the definition of classes of related functions and can establish links to other classes that describe specific function types and their properties. This makes function discovery much easier in terms of the built-in links.

Figure 6 shows an ontology-powered query GUI, which can construct query criteria from ontological terms and perform semantic resource discovery. The left hand panel of the form is used for building up an overall query expression. It consists of an ontology-driven template together with logical operators. To define specific query criteria for a particular property, users can click on the "edit" button corresponding to individual property. This will brings up a sub-window, see Figure 7. In this window the left hand panel shows the query expression relating to that property, and allows the selection of different types of semantic information for this sub-expression. The right hand panel of the sub-window displays a hierarchy of concepts from the ontology and related instances from the instance store. Users can navigate the hierarchical structure to select appropriate attributes. Users can also create new instance-descriptions with arbitrarily complex semantic information and add them the sub-expression.

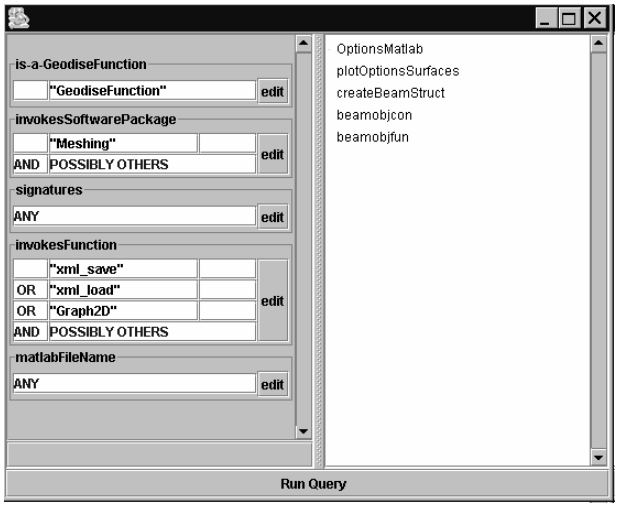


Figure 6. Top-level semantics-based query

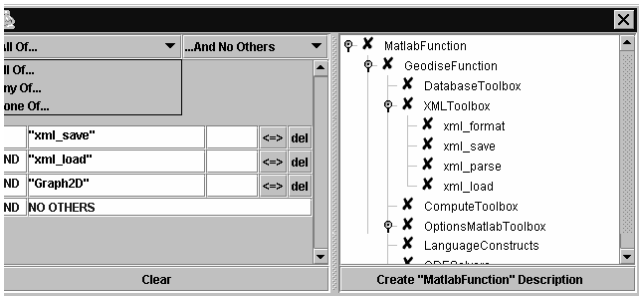


Figure 7. Property query construction form

When a query expression is built up and the "Run Query" button is clicked, the underlying reasoning engines will reason against the instance store to obtain a set of resources matching all of the specified criteria. The results are displayed in the right hand panel as shown in Figure 6. Users can then make an appropriate selection to build new workflows.

6. Integration and Problem Solving in GEODISE PSE

This section describes the integration of the Semantic Grid infrastructure and the semantics-based problem solving approach into the GEODISE PSE. It also presents the initial experimental results obtained by conducting semantics-assisted workflow construction.

Engineering design search and optimisation is a multi-step process. For example, a scenario for the design optimisation of a typical aero-engine or wing (see Figure 3) 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, special emphasis has been placed on the semantic support for the workflow construction environment of the GEODISE PSE.

We have incorporated the implemented Semantic Grid infrastructure into the GEODISE PSE, which includes an underlying knowledge repository, a description logic-based reasoning engine (Racer system), the GEODISE function ontology, the retrieval and query services of the knowledge repository and a workflow advisor. The knowledge repository contains both the function ontology and the enriched function instances. It is currently located in local machines. The reasoning engine runs locally in backend waiting for service calls. The resource retrieval and query as well as the workflow advisor are implemented as web services and deployed in a remote server.

Figure 8 shows the WCE GUI that consists of four main panels for various functionalities. The component view in the top left panel, corresponding to the Classified Service Browser in Figure 3, displays all annotated functions in a classified hierarchy in terms of function ontology. The multiple tabbed panels in the left bottom panel contain a property view, an advisor view and a document view. The property view, corresponding to the Service Configuration Editor, is used to display and edit the attributes of a selected function. The advisor view, corresponding to the Workflow Advisor in Figure 3, lists the functions that can follow or precede semantically the current function. The top right panel is the workspace for composing and editing resources, i.e. the Service Composition Editor in Figure 3. Users can drag and drop functions from the top left panel into the workflow workspace and further configure and edit them to build a workflow. The multiple tabbed panels in the bottom right panel show the compute resources and the run-time job status, i.e. the State Monitor in Figure 3.

The process of semantics-assisted service composition in the WCE is described step by step below.

a). Retrieve all functions from the Geodise knowledge repository and present them in a conceptual hierarchy in terms of function ontology as shown in the component view in Figure 8.

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

b). Select a suitable function by navigating the function hierarchy in the component view on your own or by following advice, and drag and drop it into the Service Composition Editor. A function description form will appear in the Service Component Editor in the bottom left panel for specifying function properties.

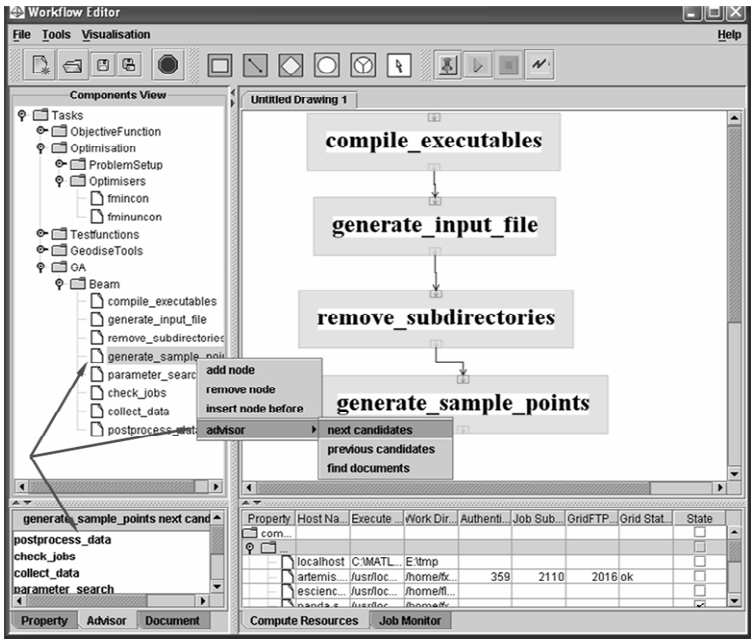


Figure 8: The workflow construction environment

c). Configure a function by filling in the attributes' values of the function property form. The Workflow Advisor can get advice about the function configuration by reasoning over ontological concept links from the semantic function descriptions. This could be the parameter's type, default value, dependency rules on other parameters, etc. Advice will be displayed as tool tips and recommended alternative values.

d). Once a function is configured and confirmed, it will be added to the Service Composition Editor as a single step of the workflow. The function component will be graphically represented as a yellow box as shown in Figure 8.

e). Each time a function is added into a workflow, its interface information, i.e. the input, effect and output parameters, will be added to the state memory of the WCE. The Workflow Advisor will collect the function interface and its semantics from the knowledge repository. It will then carry out semantic matching and reasoning against the

underlying function repository. The advisor will return a list of semantically compatible functions in the left bottom panel for users' selection.

Function composition advice can also be obtained directly from a function. Users can select a function in the top left panel, click on the right button and then select the Advisor menu item. An advice menu will pop up, displaying three choices of advising, including previous and next candidate functions and the documentation of current function. Choosing any one of them the advice will show in the knowledge advice panel in the bottom left panel. This process is shown in red arrows in Figure 8.

f). Repeat the above b, c, d, e steps, i.e. advising on required functions, function discovery, configuration, and function composition, users can construct a workflow that solves specific problems. The generated workflow will be submitted to Matlab environment for execution and computation results will return to the right bottom panel.

Once a workflow is constructed and run successfully, it can be archived in a knowledge repository with inherited functions' semantic metadata and its own metadata. This will generate a semantic workflow instance with little manual annotation, and at the end help create a semantic knowledge repository for workflows.

Semantics-assisted workflow construction in the GEODISE PSE has been tested and evaluated by specialised engineers and fresh PhD students. Initial experimental results have shown that (1) users of different levels of domain expertise can discover resources in a knowledge repository and effectively reuse them for engineering design; (2) problem solving takes less time. This is, in particular, apparent for less experienced engineers such as fresh PhD students who can use the workflow advisor to help select the right resources which is usually achieved only by domain experts. Formal complex evaluation is currently underway.

The GEODISE team has developed the GEODISE PSE and a suite of high level Grid-enabled EDSO tools. For details please refer to http://www.geodise.org.

7. Related work

The Semantic Grid is an emerging multidisciplinary research area. It is closely related to and technically underpinned by both the Semantic Web and Grid technologies. As such any significant development in these areas may have substantial impact on the Semantic Grid, for example the adoption of the service-oriented computing approach and the OGSA architecture. Although both the Semantic Web and the Semantic Grid face common issues such as semantic content management and reasoning, the targeted objects are different. The Semantic Web has focused on information and metadata while the Semantic Grid on resources and the use of more complex domain-related knowledge for problem solving.

Recently there are increasing numbers of endeavour that address the Semantic Grid, but most of them have concentrated on individual issues such as semantic grid service description [Johnston 2004] [Brooke 2004] [Sirin 2003]. Systematic research on the Semantic Grid is still rare at the moment. The most prominent one is the myGrid project (http://www.mygrid.org.uk), which tries to develop open source data-intensive bioinformatics applications on the Semantic Grid. The emphasis is on data integration, workflow, pernalisation and provenance. While myGrid and GEODISE share many commonalities, there are important differences among them. In particular GEODISE is

more concerned with (1) the architectural issues of the Semantic Grid, which has led to the service-oriented distributed knowledge management framework [Chen 2002a] [Goble 2004]; (2) the capture, modelling and representation of complex, unstructured domain knowledge [Chen 2002b]; and (3) the deep use of rich semantics and knowledge such as workflow recommender systems for problem solving [Tao 2003].

Research on the provision of information-level and knowledge-level support for Grid applications has also been active in the Web/Grid intelligence communities in a different guise – the Knowledge Grid. The Knowledge Grid is intended to provide an intelligent, sustainable application environment on the Internet that enables individuals and applications in the VOs to effectively capture, publish, share, and reuse explicit knowledge resources [Zhuge 2004]. It is realised by aggregating and synthesizing knowledge from data through mining and reference methods and enable search engines to make references, answer questions, and draw conclusions from masses of data [Berman 2001]. The Semantic Grid and the Knowledge Grid share a lot of commonality in their goals, underpinning technologies and realisation approaches. Both of them are built on the current Grid infrastructure [Cannataro 2003], make heavy use of the ontology technology and advanced knowledge technologies, and adopt a service-oriented approach. The difference seems that the Semantic Grid targets on the creation and population of semantic and knowledge rich content while the Knowledge Grid is more concerned with extracting, inferring and mining knowledge from the Grid. To some extent the Semantic Grid is a supporting layer for the Knowledge Grid. In applications, the Knowledge Grid takes a broad view of knowledge support in the context of Web/Grid intelligence, which includes social and economical applications. The Semantic Grid is at the moment concentrated on the knowledge intensive scientific Grid applications, i.e. the e-Science.

Semantics has been used for semantic service discovery and composition [Wroe 2004]. Ontologies have been used as a conceptual backbone for distributed knowledge management [Staab 2001] and domain dependent problem solving [Cannataro 2004]. The distinguishing feature of GEODISE PSE is that it makes deep use of semantic information via a workflow advisor rather than just ontology query and browser as in Proteus. Workflow advisor can extract knowledge not only from a resource classification but also its property descriptions for problem solving such as service configuration. GEODISE PSE is also different from others in the extensive use of description logic based reasoning capability.

8. Conclusions and Future work

Building a grid with rich and pervasive semantics and knowledge, i.e. the Semantic Grid, is the first step towards the realisation of the e-Science vision. Equally important are the capabilities that can consume such semantics and knowledge to ease real world problem solving. To realise them the key is the design, development and deployment of novel approaches, mechanisms, architecture and middleware.

In this paper, we have presented an integrated service-oriented distributed knowledge management framework, which aims to populate the Grid with ubiquitous semantic content. We have implemented the core components of the Semantic Grid infrastructure. While the framework focuses on knowledge supply a semantics-based approach to problem solving has been introduced for effective knowledge consumption. Both the

framework and the approach are based on the latest ontology, the Semantic Web and advanced knowledge technologies. These ideas have been integrated, implemented and further verified in the development and evaluation of an ontology-enabled PSE for engineering design search and optimisation. Through the GEODISE application example we have demonstrated the potentials and benefits of the Semantic Grid as well as its realisation and utilisation methodologies. The current research context is design search and optimisation but the underlying infrastructure and approaches could be applied to many other types of Grid application.

We are improving the distributed knowledge management framework in two aspects. One is to provide semantic support for workflows, which includes workflow annotation, management and the provenance of workflow result trails. The other focuses on the capture, modelling, representation and publishing of deep complex domain knowledge such as the procedural knowledge in the format of production rules. With rich knowledge, probably in new representation formalisms, available on the Semantic Grid we will explore novel approaches and mechanisms for knowledge consumption. At application level future work will concentrate on the issues of semantics/knowledge enabled trust and security.

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

Allcock, W., Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S. 2001. 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.

Berman, F., 2001. From TeraGrid to Knowledge Grid, Comm. ACM, vol. 44, no.11, pp. 27–28.

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

Brooke J., Fellows D., Garwood K. and Goble C.A. 2004. Semantic matching of Grid Resource Descriptions in Proceedings of The Second European Across Grids Conference, Nicosia, Cyprus

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

Cannataro, M., Comito, C., Lo Schiavo, F. and Veltri, P. 2004, Proteus, a Grid based Problem Solving Environment for Bioinformatics: Architecture and Experiments", IEEE Computational Intelligence Bulletin, Vol.3, No.1, pp7-18.

Chen, L, Cox, S.J, Goble, C, Keane, A.J, Roberts, A, Shadbolt, N.R, Smart, P, Tao, F. 2002a. Knowledge Services for Distributed Service Integration. In Proc. of first UK e-Science all-hands meeting , poster p8.

Chen, L., Cox, S.J., Goble, C., Keane, A.J., Roberts, A., Shadbolt, N.R., Smart, P., Tao, F. 2002b. Engineering Knowledge for Engineering Grid Applications. Proceedings of Euroweb 2002 Conference, The Web and the GRID: from e-science to e-business, pp.12-25.

Chen, L., Cox, S.J., Tao, F., Shadbolt, N.R., Goble, C., Puleston C. (2004), Empowering Resource Providers to Build the Semantic Grid. In Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence, pp271-278.

De Roure, D., Jennings, N. and Shadbolt, N. 2003. The Semantic Grid: A Future e-Science Infrastructure, Berman, F. et al ed. Grid Computing: Making the Global Infrastructure a Reality, ISBN 0470853190, pp. 437-470.

Eres, M.H., Pound, G.E., Jiao, Z., Wason, J.L., Xu, F., Keane,A.J., Cox, S.J. 2004. Implementation and utilisation of a Grid-enabled problem solving environment in Matlab. Future Generation Computer Systems, to appear.

Foster I, Kesselman C, Nick J, Tuecke S. 2002. Grid Services for Distributed System Integration. Computer, 35(6).

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

Goble C.A. and De Roure D. 2002. The Grid: an Application of the Semantic Web. ACM SIGMOD Record 31(4): pp.65-70.

Goble, A.C., De Roure, D., Shadbolt, N.R. and Fernandes, A.A.A., 2004. Enhancing Services and Applications with Knowledge and Semantics, Chapter in "The grid : blueprint for a new computing infrastructure, 2nd ed., Foster I. and Kesselman C., eds., pp. 431-458.

Haarslev, V. and Möller, R. 2003, Racer: A Core Inference Engine for the Semantic Web, Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), ISWC 2003, pp. 27-36.

Handschuh, S., Staab, S. and Volz, R. 2003. Annotation for the Deep Web, IEEE Intelligent Systems Vol.18, No.5, pp.42-48.

Hey, T. and Trefethen, A.E. 2003. The Data Deluge: An e-Science Perspective. Berman, F. et al ed. Grid Computing: Making the Global Infrastructure a Reality, ISBN 0470853190, pp.809-824.

Horrocks, I., Li, L., Turi, D. and Bechhofer, S. 2004. The instance store: DL reasoning with large numbers of individuals. In Proc. of the 2004 Description Logic Workshop (DL 2004), pp.31-40.

Horrocks, I., Sattler, U. and Tobies, S. 1999. Practical reasoning for expressive description logics. Lecture Notes in Artificial Intelligence (Springer-Verlag) LNAI 1705, pp.161-180.

Johnston, W. 2004. Semantic Services for Grid-Based, Large-Scale Science. IEEE Intelligent Systems, Vol19, No.1, pp.34-39.

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.

Maedche, A., Motik, B., Stojanovic, L., Studer, R. and Volz, R. 2003. Ontologies for Enterprise Knowledge Management. IEEE Intelligent Systems, IEEE Intelligent Systems 18(2): 26-33

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

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

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.

Staab, S. and Maedche, A. 2001. Knowledge Portals - Ontologies at Work, AI Magazine 21 (2). 2001.

Tao, F, Chen, L, Shadbolt, N,R. Pound, G, Cox, S.J. (2003) Towards the Semantic Grid: Putting Knowledge to Work in Design Optimisation. Journal of Universal Computer Science (J.UCS), Vol. 9, No. 6, pp551-563.

Wroe, C., Goble, C., Greenwood, M., Lord, P., Miles, S., Papay, J., Payne, T. and Moreau, L. 2004. Automating Experiments Using Semantic Data on a Bioinformatics Grid. IEEE Intelligent Systems, Volume 19, No.1, pp.48-55.

Zhuge, H., 2004. China's E-Science Knowledge Grid Environment, IEEE Intelligent Systems, 19 (1) 13-17.