A Semantic Web Based Approach to Knowledge Management for Grid Applications

L. Chen, N.R. Shadbolt and C.A. Goble

Abstract—Knowledge has become increasingly important to support intelligent process automation and collaborative problem solving in large-scale science over the Internet. This paper addresses distributed knowledge management, its approach and methodology, in the context of Grid application. We start by analysing the nature of Grid computing and its requirements for knowledge support; then we discuss knowledge characteristics and the challenges for knowledge management on the Grid. A Semantic Web based approach is proposed to tackle the six challenges of the knowledge lifecycle – namely, those of acquiring, modelling, retrieving, reusing, publishing and maintaining knowledge. To facilitate the application of the approach, a systematic methodology is conceived and designed to provide a general implementation guideline. We use a real world Grid application, the GEODISE project, as a case study in which the core Semantic Web technologies such as ontologies, semantic enrichment and semantic reasoning are used for knowledge engineering and management. The case study has been fully implemented and deployed through which the evaluation and validation for the approach and methodology have been performed.

Index Terms—Grid computing, methodology, knowledge management, Semantic Web, engineering design

◆

1 INTRODUCTION

Grid computing [1] is an emerging distributed computing paradigm that intends to enable flexible, secure and coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisations. It has been under vigorous investigation and has evolved rapidly [2] for the past several years, during which a whole raft of tools, middleware and infrastructure such as Globus (www.globus.org), OMII (www.omii.ac.uk) and UNICORE (www.unicore.org) has emerged, and a number of Grid applications across the whole spectrum of scientific disciplines such as myGrid (www.mygrid.org.uk), EUROGRID (www.eurogrid.org) and TERAGRID (www.teragrid.org) have been developed. One finding from these endeavours [3] is that the success of Grid applications depends on the effective Knowledge Management (KM) of Grid resources [4], [5].

The Semantic Web [6] 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 […] where data can be shared and processed by automated tools as well as by people." To achieve this vision, researchers in the Semantic Web community have developed core enabling technologies, APIs and tools, encompassing ontologies, ontology languages, annotation, semantic repositories and reasoning, which provide an infrastructure for distributed information and knowledge management based on metadata, semantics and reasoning.

In this paper we analyse the nature of Grid computing and identify its requirements for knowledge management. We further argue that an innovative and systematic approach to knowledge management on the Grid is required in order to help achieve the goal of the Grid. To this end, we propose a Semantic Web based approach to KM in which we use ontologies for knowledge acquisition and modelling, the web ontology language for knowledge representation and semantic-based reasoning for decision-making support.

Our contributions are three folds: Firstly, we propose the Semantic Web based approach to managing heterogeneous, distributed Grid resources for Grid applications. Secondly, we design an architecture to realize the proposed approach and conceive a methodology to addresses the complete life cycle of knowledge management. Thirdly, we apply the approach, concepts and methodology to a real world Grid application - Grid Enabled Optimisation and Design Search in Engineering (GEODISE) (www.geodise.org) in which domain knowledge and design expertise are exploited to assist Grid resource discovery and composition for problem solving. The implementation and deployment not only demonstrate the benefits of using KM in GEODISE, but also provides an infrastructure for applying this approach to other Grid applications.

The remainder of the paper is organized as follows: Section 2 discusses the motivation and challenges of knowl-

---

- *L. Chen is with the School of Computing and Mathematics, University of Ulster, Newtownabbey, Co. Antrim BT37 0QB, U.K. E-mail: l.chen@ulster.ac.uk.*
- *C.A.Goble is with the School of Computer Science, University of Manchester, Oxford Road, Manchester, M13 9PL, U.K. E-mail: carole@cs.man.ac.uk.*
- *N.R. Shadbolt is with the School of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, U.K. E-mail: nrs@ecs.soton.ac.uk.*

Manuscript received on 1 October 2005, revised on 14 April and 1 August 2006, accepted on 1 September 2006. Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.

edge management for the Grid, followed by an introduction to the Semantic Web based KM architecture. Section 3 describes the methodology for adopting and implementing the proposed approach. A case study is presented in Section 4, which describes the design rationale and implementation of the KM architecture in the context of GEODISE. In Section 5 we discuss our experience and lessons learnt from GEODISE. We present related work in Section 6 and conclude the paper in Section 7.

2 GRID, SEMANTIC AND KNOWLEDGE MANAGEMENT

2.1 Motivations

Grid computing was conceived to connote the idea of a "power grid": namely applications can plug into the Grid to draw computing resources in the same way electrical devices plug into a power grid to draw power. Analogous to a power grid, it views geographically distributed computing capabilities, storage, data sets, scientific instruments, knowledge and so on as utility resources to be delivered over the Internet seamlessly, transparently and dynamically as and when needed. The Grid is built upon two fundamental concepts: virtualisation, i.e. individuals and/or institutions with the required resources or common interests can dynamically form a virtual organisation (VO) that enables rapid assembly and disassembly of resources into transient confederations for coordinated problem solving, and dynamic provisioning, i.e. resources provision is transient, dynamic and volatile without guarantee of availability, central control for accessibility and prior trust relationships. Grid computing offers a promising distributed computing infrastructure where large-scale cross-organisational resource sharing and routine interactions are commonplace.

Grid applications usually refer to large-scale science and engineering that are carried out through distributed global collaboration enabled by the Grid. Typically such scientific enterprises are data-intensive and/or computation-intensive and/or collaboration-intensive, i.e. they require access to very large data collections, very large-scale computing resources and close collaboration among multiple stakeholders. This necessitates the interaction and sharing of various resources, in particular, domain-dependent application-specific resources, despite the heterogeneity of their respective policies, platforms and technologies, and their geographical and organisational dispersal.

It is envisioned that Grid applications would be carried out through flexible collaborations and computations on a global scale with a high degree of easy-to-use and seamless automation. However, the reality is some way off this vision. Current Grid infrastructure is difficult to install and manage, and requires continual nursing by trained administrators. For example, a typical installation procedure for the Globus toolkit – a very popular Grid middleware infrastructure, consists of twenty-five steps [7], and this does not include any sub-procedure required for accomplishing each individual step and any steps for fail-over or fault handling. Current Grid applications are usually based on bespoke solutions, i.e. resources are pre-located and/or prespecified with very limited support for dynamic resource provisioning. Cross-organizational resource sharing and effective reuse are constrained by the lack of interoperability and knowledge of configuration and usage. There is little automation for dynamic VO formation. Most configurations and operations are carried out manually with prior knowledge. As a result, the Grid and its applications are hard to reach and hard to use for ordinary scientists, engineers and researchers.

We contend that the Grid should use the metadata, semantics and knowledge of Grid resources in order to evolve the Grid beyond a manual plumbing practice [8]. More specifically, metadata provide rich descriptions for the states and properties of resources as well as their purposes and configurations for problem solving, thus facilitating resource discovery and sharing. Grid applications are usually knowledge intensive, and such knowledge resides implicitly in resource models and/or descriptions. Making domain knowledge explicit and understandable for third-party consumers can enhance effective resource reuse by providing well-informed decisions regarding when, where and how to use a resource. By enriching metadata and knowledge with semantics, the Grid can break down the barrier of heterogeneity and move to truly seamless access and cross-organisational resource sharing. Furthermore, semantics empowers machines or software agents to understand and process resources' metadata. Consequently, it will increase the level of automation and reduce the need of manual intervention.

2.2 Challenges

Metadata and knowledge pervade the Grid and some of them already exist on and within the Grid. For instance, a Grid resource published as a Grid/Web service exposes its metadata in its WSDL1 file, which include information about the service location, signature, input/output argument types and formats, interaction and invocation methods, etc. Given that the emphasis of the Grid is to share and reuse distributed resources in a VO for coordinated problem solving, it is reasonable to assume that domain-dependent application-specific scientific knowledge in a Grid application is also available. For example, in an engineering design search and optimization Grid application all design optimization algorithms and knowledge regarding their usage should already be there, though they might exist in a diversity of formats. While metadata and knowledge that are currently not available could be required in order to help realize the Grid vision, the key issues for using metadata, semantics and knowledge on the Grid are how (1) to acquire, formally model, explicitly represent, store, maintain and update them; and (2) to use them to support seamless resource sharing and interoperability, so as to achieve a high degree of automation.

Confronting these issues, we face a number of challenges: Firstly, metadata and knowledge of Grid resources are tacit, unstructured and largely in the province of human domain experts expressed in the medium of natural languages. They need to be captured and modelled in a way that facilitates knowledge preservation and reuse. Sec-

---

[1] WSDL, along with RDF, RDFS, OWL, SOAP, XML and HTTP mentioned later are all W3C standards. Detailed information can be found at the W3C web site - www.w3.org.

ondly, Grid resources are supposed to be used by others but their metadata and knowledge are usually only understandable by resource providers. This requires that metadata and knowledge be modelled and represented to support interoperability and mutual understanding between resource providers and consumers. Thirdly, to enable dynamic VO formation and resource provisioning with a high degree of automation rather than manual plumbing, machines and/or software agents should be able to understand and process resources' metadata and knowledge. This in turn requires resources' metadata and knowledge be given well-defined meaning, i.e. semantics. Fourthly, resource metadata and knowledge should be Web/Grid friendly, i.e. easy to publish collectively, store centrally or in a decentralised manner and retrieve on the Grid. Finally, knowledge modelling and representation should incorporate reasoning and inference capabilities so as to provide knowledge-based decision-making support for coordinated problem solving. For example, most Grid applications involve composing services into a workflow as a solution to a specific problem. By reasoning about resources' semantic metadata and knowledge, we can provide advice on which service should be selected and how it should be configured can be given.

Current knowledge engineering and management technologies [9] do recognize the importance of describing the competencies of knowledge models and of making them explicit. However, they do not consider the essentially distributed notion of KM implied by the Grid. For instance, traditional knowledge acquisition and modelling were usually done with little consideration as to how knowledge might be shared by third-party consumers (or applications) across organisations. Knowledge was usually captured and modelled for and used by a specific application. Knowledge publishing and representation paid little attention to how knowledge was interpreted by machine and/or humans because the interpretation was mostly hard-coded into the application system. Similarly, knowledge bases did not often concern themselves with how knowledge was accessed because it usually was used for local, standalone systems operating in a closed world. Traditional standalone, knowledge systems were often hand-crafted, involving lots of hardwiring and human intervention. To meet the challenges of the Grid knowledge lifecycle – namely, those of acquiring, modelling, retrieving, reusing, publishing and maintaining knowledge for Grid applications, we need to provide a Grid-oriented knowledge infrastructure based on innovative knowledge management architectures, methodologies, approaches, technologies, tools and APIs that are suitable for engineering and managing Grid resources' knowledge in the life cycle of Grid applications.

Metadata, semantics and knowledge can play many different roles for Grid applications. While our general objective is to develop an innovative KM architecture and methodology for Grid knowledge infrastructure, we particularly concentrate on the components and functions of the infrastructure that can provide semantic and knowledge support for resource discovery, orchestration and effective use for problem solving, which prove to be the key distinctives of a knowledge intensive Grid application. We distinguish Grid infrastructure-related resources, mainly Grid middleware such as clusters, storage and registry services, from application-level resources such as domain specific algorithms, tools and devices. As application-level resources contain rich domain knowledge critical for problem solving, our research is primarily concerned with engineering and managing knowledge for application-level Grid resources.

Recently the Grid has evolved from OGSA [1] to the Web Service Resource Framework (www.globus.org/wsrf) (WSRF) in which Grid resources have been wrapped and exposed as WSRF Grid services. WSRF services contain more metadata and information about service creation and lifetime management, state handling, retrospection and grouping. This leads to a demand for effective knowledge management. To facilitate this process we focus on two types of knowledge in a Grid resource: The first is high-level descriptive metadata describing a resource's signature, functionality, service quality, etc. The second is application-specific heuristics for using the resource such as parameter configuration rules and performance fine-tuning skills. The former is mainly used for service publishing and discovery, whereas the latter for service configuration and execution.

2.3 A Semantic Web based approach to knowledge management

We propose a Semantic Web based approach to engineering and managing Grid resources' knowledge for Grid applications, as shown in Fig. 1.

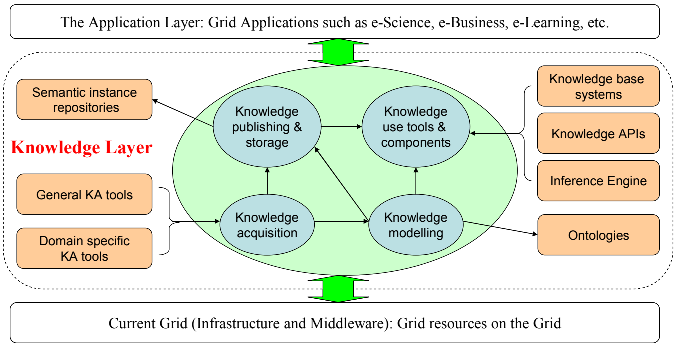


Fig 1. The Semantic Web based approach to knowledge management

The essence of the approach is to add a semantics-based knowledge layer between primitive Grid resources and Grid applications. In this layer, the Semantic Web technologies are used to carry out knowledge acquisition, modelling, representation, publishing, storage and reuse. Ontologies, an explicit, shared specification of the various conceptualisations in a problem domain, play a fundamental role in this approach. They are used to conduct knowledge acquisition through ontology modelling and semantic annotation. Ontology modelling provides conceptual structures for preserving knowledge. Semantic annotation captures metadata, generates semantic instances as knowledge entities and populates them into knowledge bases. Both ontologies and semantic instances are represented using the Web Ontology Language (OWL). OWL is built on top of previous ontology languages RDFS and description logic (DL), therefore, OWL-represented knowledge is Web-friendly and supports DL-based reasoning. Knowledge reuse is achieved by consuming semantic instances through a reasoner such as FaCT [10] and RACER [11]. Typically, knowledge-based decisions can be made via common ontological operations, such as subsumption, consistency checking, concept classification, navigation and retrieval.

The approach will adopt a service-oriented computing paradigm for knowledge management, i.e., knowledge components are implemented as Web/Grid services, known as knowledge services, which will be referred to as K-services hereafter. Each type of knowledge service provides users with a set of APIs that can be used to perform a variety of operations. For example, when using ontology services we can manipulate concepts and properties within an ontology in many different ways – e.g., asking for more general or specialised examples of a concept. This service-oriented approach enables the reuse and sharing of knowledge over the Internet.

3 THE METHODOLOGY

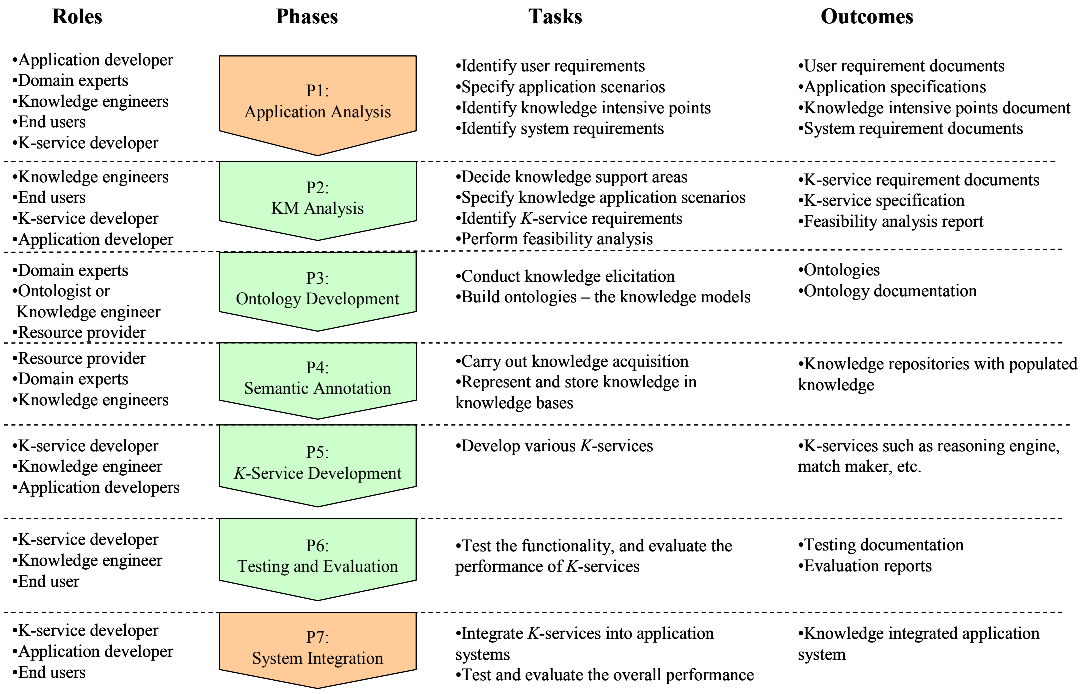
In order to support the application of the proposed approach, a methodology is needed to provide guidance for employing Semantic Web technologies for Grid knowledge management. Based on the general, widely used CommonKADS methodology [9], we develop a light weight, Web-oriented knowledge engineering and management methodology. Following this methodology, knowledge can be structured, represented and accessed on the Web/Grid in an appropriate way so that it becomes recognizable, sharable and reusable by both humans and machines.

The methodology is depicted in Fig. 2. Central to the methodology is a number of activities required for Grid knowledge management, which are organised in seven phases, i.e. Application Analysis, KM Analysis, Ontology Development, Semantic Annotation, Service Development, Testing and Evaluation and System Integration. The column to the right of the Phase column lists corresponding tasks performed in each phase, and the right column describes outcomes of each activity. The left column indicates corresponding roles involved in each phase. In the following, we shall describe in detail the methodology, including its applicability, roles and tasks at each phase.

3.1 Applicability

We scope the methodology by defining the use cases it applies to. Broadly, the methodology can be applied to two use cases: The first is the development of knowledge enabled Grid applications, i.e., Grid applications that intend to make use of knowledge and knowledge management infrastructure to support decision making or problem solving in their specific application scenarios. This use case involves not only the knowledge management lifecycle but also the development of application systems and the integration of K-services into the application systems. In this use case, the methodology will consist of all seven phases.

However, it is not always possible in Grid computing to determine how, who and in what context, resources will be used, therefore, the second use case of the methodology is for knowledge enabled Grid resource management. In this case, knowledge management focuses on the provision of knowledge models, knowledge bases and K-services. It will be the applications that determine how knowledge and knowledge services will be used in their scenarios. In this use case, the methodology will start from phase two and end at phase six.



Roles	Phases	Tasks	Outcomes
•Application developer •Domain experts •Knowledge engineers •End users •K-service developer	P1: Application Analysis	•Identify user requirements •Specify application scenarios •Identify knowledge intensive points •Identify system requirements	•User requirement documents •Application specifications •Knowledge intensive points document •System requirement documents
•Knowledge engineers •End users •K-service developer •Application developer	P2: KM Analysis	•Decide knowledge support areas •Specify knowledge application scenarios •Identify *K*-service requirements •Perform feasibility analysis	•K-service requirement documents •K-service specification •Feasibility analysis report
•Domain experts •Ontologist or Knowledge engineer •Resource provider	P3: Ontology Development	•Conduct knowledge elicitation •Build ontologies – the knowledge models	•Ontologies •Ontology documentation
•Resource provider •Domain experts •Knowledge engineers	P4: Semantic Annotation	•Carry out knowledge acquisition •Represent and store knowledge in knowledge bases	•Knowledge repositories with populated knowledge
•K-service developer •Knowledge engineer •Application developers	P5: *K*-Service Development	•Develop various *K*-services	•K-services such as reasoning engine, match maker, etc.
•K-service developer •Knowledge engineer •End user	P6: Testing and Evaluation	•Test the functionality, and evaluate the performance of *K*-services	•Testing documentation •Evaluation reports
•K-service developer •Application developer •End users	P7: System Integration	•Integrate *K*-services into application systems •Test and evaluate the overall performance	•Knowledge integrated application system

Fig. 2. The methodology for the Semantic Web based approach to knowledge management

It may be worth pointing out that at the time of writing the first use case is the dominant application scenario for the methodology. However, the second use case is gaining currency and may become the driving force to evolve the Grid towards the Semantic Grid and Knowledge Grid [12] where semantics and knowledge pervade and are easy to discover and use.

3.2 Roles

Managing knowledge for grid resources involves a variety of stakeholders; each of them has specific expertise and plays different roles in different phases. We can classify these stakeholders involved in the methodology according to their role in the knowledge management lifecycle. Briefly, the roles and their responsibilities are as follows.

A domain expert is an individual who has specific knowledge about a problem domain. Domain experts are responsible for providing raw knowledge in knowledge elicitation and validating knowledge after knowledge is captured and modeled.

A resource provider is responsible for the provision of Grid resources; it could be an individual or an organization. Resource provision has to expose relevant resource information in terms of the commonly endorsed standards.

A knowledge engineer is an individual who has specific expertise and skills, and is responsible, for knowledge elicitation, modeling and knowledge system design.

An ontologist is responsible for building ontologies for knowledge modeling. A knowledge engineer may assume the role of an ontologist.

A K-service developer is responsible for the design and implementation of knowledge services.

An application developer is responsible for the design and implementation of application systems, including the integration of K-services.

An end user is the user of the application systems as well as K-services, who is responsible for putting up user requirements and evaluating system performance.

In reality, an individual may assume more than one role. For example, a knowledge engineer could also take the role of a K-service developer; a domain expert may also be the resource provider. By defining roles and specifying the roles involved in each phase, as can be seen in Fig. 2, the methodology gives unambiguous advice on what kind of expertise is needed and who should be involved in each phase.

3.3 Activities and tasks

The methodology consists of seven activities, and each activity performs a number of tasks. Each task produces some outcomes that are used in the following activities. The methodology starts with the Application Analysis activity, which aims to collect user requirements and application system requirements, to specify application scenarios and to identify knowledge intensive points. In this phase, most of the roles are involved in order to get extensive inputs from multiple perspectives. The outcomes include user requirement documents, system requirement documents, application specifications and an analysis report about the knowledge intensive points of the application. The KM Analysis activity concentrates on the analysis of knowledge application and feasibility. It determines where knowledge support can be best provided and exploited. Based on the application specification, it specifies knowledge application scenarios and further identifies K-service requirements. In this phase, various risk factors and technical bottlenecks are analysed to evaluate the feasibility of knowledge application. The main players of this activity are knowledge engineers and K-service developers but end users and application developers will be consulted to give their comments on the proposed knowledge scenarios and services. The outputs of this phase include K-service requirement documents, K-service specifications and a risk evaluation report.

Following the formal analyses in the first two phases, knowledge elicitation and modelling will be carried out in the Ontology Development phase. Knowledge engineers or more specifically ontologists will work on knowledge sources, usually domain experts or resource providers, to elicit and capture actionable knowledge of the selected knowledge application areas. Ontologies can then be built to model a domain and application conceptualization. The results in this phase are mainly ontologies and the accompanying documentation. Once ontologies are available, knowledge population can be conducted in a Semantic Annotation phase. This is done by binding concrete resource information with conceptual knowledge structures, i.e. the ontology, to create instances of ontological concepts. All instances will form a knowledge base that can be used later by K-services. Semantic annotation can be performed either by knowledge engineers or resource providers with domain experts providing valuable information pertaining to resource discovery and reuse. The outcome of this phase is a number of knowledge repositories.

In the phase of Service Development, K-service developers design and implement knowledge services in terms of application specifications, knowledge application scenarios and K-service specifications. K-service developers need to interact closely with knowledge engineers and application developers so that K-services make best use of available knowledge and provide maximum intelligent support for problem solving. The outputs of this phase are a number of K-services that can serve as building blocks for Grid applications or more generally as the middleware for the Semantic Grid infrastructure.

In Phase 6, K-service developers will test the functionalities of K-services and evaluate their performance. Testing and evaluation results may be fed back to previous phases for further refinement. For example, new knowledge may need to be captured or modelled. The deliverables of this phase include K-service testing documentations, evaluation reports and software distribution documents. The methodology may end in this phase if the work focuses on the provision of a knowledge management infrastructure for Grid resources. If there is a concrete application, then the methodology will end in the System Integration phase, where K-services as well as knowledge repositories are integrated into application systems to facilitate Grid resource discovery and reuse. The integration involves end users, application developers and K-service developers. Testing and evaluation of the application system may also take place in this phase but they are not the focus of this methodology.

While tasks are performed in each phase by different roles, they are a coordinated and collaborative endeavour. The outcomes of one phase will be used by tasks in later phases. Results from previous activities are evaluated and validated by later tasks. Sometimes, it is necessary to iterate the refinement–evaluation cycle several times between phases before all requirements are met.

3.4 Discussion

The novelty of the proposed methodology is that it engineers and manages knowledge using a Semantic Web based approach within an integrated framework. This methodology integrates the various activities of knowledge management together. Knowledge management can be undertaken in a much more co-ordinated way so that results from one piece of work can be used for another in an appropriate form. For example, the ontologies from knowledge elicitation and modelling can be used to create knowledge bases via semantic annotation. In turn, these knowledge repositories can be exploited by K-services, which provide mechanisms for querying, searching or reasoning over semantic content so as to facilitate knowledge use.

As a general guidance, the methodology specifies activities and tasks that should be undertaken in order to manage Grid knowledge at conceptual level. It does not say how a task should be undertaken and what tools should be used. This gives the users of the methodology flexibility for implementation. Although different knowledge management tasks are coupled together in the methodology, their interactions are not hardwired. Each phase 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 componentisation makes the methodology robust.

4 CASE STUDY: KNOWLEDGE MANAGEMENT IN GEODISE

Scientific problem solving usually involves constructing a workflow either manually or automatically to realize a particular experiment or series of computations. Consider the design optimization of a typical aero-engine or wing in Engineering Design Search and Optimization (EDSO) [13] whereby engineering modeling and analysis are exploited to yield improved designs, it is necessary to (1) specify the wing geometry in a parametric form which specifies the permitted operations and constraints for the optimization process, (2) generate a mesh for the problem, (3) decide which analysis code to use and carry out the analysis, (4) decide the optimisation schedule, and finally (5) execute the optimisation run coupled to the analysis code. In this process, each of these tasks can be accomplished by one of a set of computation resources of a similar functionality. The set of resources may be geographically located and run on heterogeneous environments, and most probably, with different performance. Each resource requires specific domain knowledge for configuration and effective use. Design problems of different characteristics may be solved by different sets of tasks and most likely different resources, i.e. a different process should be constructed.

The Grid has been viewed as the underlying enabling infrastructure for scientific computing. In the service-oriented Grid computing paradigm the process of problem solving amounts to discovering services on the Grid and composing those services 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 each point in the workflow often depends on the results of the execution of preceding steps. Moreover, the selection of a service from a set of competing services is usually determined by the exact nature of the problem as well as the performances of the services available. As a result, it is not practical to specify, a priori, the precise sequence of steps for a problem. The successful orchestration of component services into a valid workflow specification is heavily dependent on bodies of domain knowledge as well as semantically enriched service descriptions. For this reason, we have applied our proposed knowledge management approach and its methodology to GEODISE to facilitate dynamic, intelligent workflow construction.

GEODISE, one of the UK e-Science pilot projects (www.rcuk.ac.uk/escience), is intended to enable engineers to carry out EDSO on the Grid by seamless access to Grid resources such as optimisation and search tools, geometry modeling and meshing packages, analysis codes and distributed computing and data resources. The nature of EDSO, i.e. the multiple choices of resources, the need for knowledge when discovering EDSO resources, the requirements of intensive data and computation, and the need for expertise in workflow construction and configuration, has determined that EDSO is a typical and "killer" application for both the Grid and knowledge management on the Grid.

In the following we briefly describe our efforts in engineering and managing EDSO resources' knowledge, which is guided by our methodology. We also present mechanisms for supporting problem solving in the application system.

4.1 Engineering EDSO resources' knowledge

Application analysis is carried out at the very beginning of the GEODISE project in which the involved roles perform the specified tasks as defined in the methodology. From the analysis, it has been decided that GEODISE will use Matlab (www.mathworks.com/products/matlab) as its execution environment, and a Problem Solving Environment (PSE) [14] will be constructed as its application system. The PSE will contain a Workflow Construction Environment (WCE) in which end users can drag and drop Grid EDSO resources to build EDSO workflows and then submit them to the underlying Matlab execution environment for execution.

In addition to Grid resources provided by Grid utility tools such as Globus and Java Cog [15], EDSO resources are mainly legacy EDSO algorithms such as those provided by the OPTIONS design exploration package [16]. These algorithms will be wrapped as Matlab functions so they can be used in PSE and executed in Matlab. Therefore, EDSO Grid resources in GEODISE are Matlab functions.

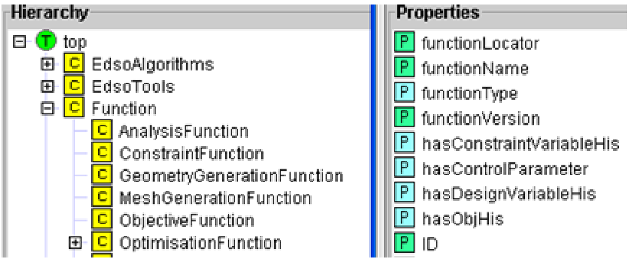
Application analysis also identifies that domain experts and software manuals will be the main knowledge sources. Initial knowledge elicitation with domain experts has concluded that EDSO engineers usually follow rules and design patterns in order to carry out EDSO. Patterns are captured and abstracted as an EDSO workflow flowchart. Further analysis has identified a number of knowledge intensive points where knowledge plays a key role for engineers to make design decisions.

KM analysis focuses on what knowledge support can and should be provided, and how, in terms of the results of application analysis. Knowledge engineers and K-service developers will choose a couple of areas from knowledge intensive points. For each area, they analyse the characteristics of raw knowledge, the viability of knowledge acquisition, modelling and representation, the impact of knowledge support, and also the benefit and cost ratio. The analysis will take into consideration end users' and application developers' opinions. Finally knowledge support scenarios are outlined.

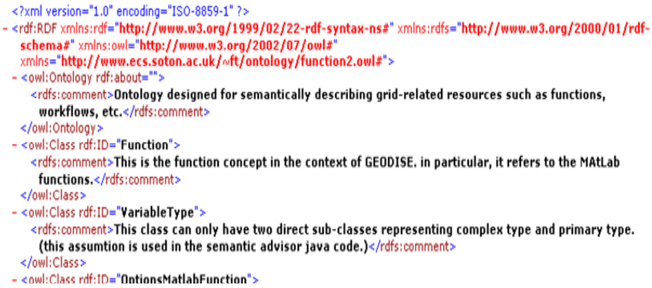
KM analysis in GEODISE has revealed two critical areas where knowledge support can make differences: The first is the use of knowledge for resource discovery, which aims to enhance EDSO resource interoperability and machine processability. The second area is concerned with decision support for resource configuration and effective reuse. It intends to enable novice engineers to conduct EDSO using previous design expertise. Further analysis has produced a knowledge application scenario, which include the following steps: (1) to capture rich metadata and in-depth usage heuristics about EDSO resources, (2) to create knowledge models, i.e., ontologies, for preserving such knowledge, (3) to populate knowledge repositories via semantic enrichment, (4) to develop K-services to run reasoning and queries over them, and (5) to integrate K-services in PSEs to enable semantic-based resource discovery, and provide advice on workflow construction and configuration.

Ontology development is, in essence, to create EDSO knowledge models through knowledge elicitation. We have concentrated on two types of EDSO knowledge: domain knowledge that includes fundamental EDSO concepts and principles, and resource knowledge that includes resource description information, configuration and execution heuristics. In GEODISE, knowledge acquisition is conducted by means of an integrated knowledge engineering toolkit PCPACK (www.epistemics.co.uk/products/pcpack/) in which a number of knowledge elicitation techniques, namely, interviews, protocol analysis, concept sorting, and other traditional methods are utilised. The acquired knowledge is modelled as ontologies and represented using OWL.

We have developed a suite of EDSO ontologies using the Protégé OWL Plugin (protege.stanford.edu/plugins/owl), which include a domain ontology, a task ontology and a function ontology. Fig. 3 shows the function ontology in which Fig. 3a displays all concepts and their hierarchical relationships; Fig. 3b lists the properties of a concept that are used to define the subject-predicate-object relationship among concepts. A fragment of the OWL representation is shown in Fig. 3c. A detailed description of the ontology engineering in GEODISE can be found in [17].



(a) Concept hierarchy (b) Concept properties



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
   schema#" xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://www.ecs.soton.ac.uk/~ft/ontology/function2.owl#">
 - <owl:Ontology rdf:about="">
    <rdfs:comment>Ontology designed for semantically describing grid-related resources such as functions,
     workflows, etc.</rdfs:comment>
   </owl:Ontology>
 - <owl:Class rdf:ID="Function">
    <rdfs:comment>This is the function concept in the context of GEODISE. in particular, it refers to the MAtLab
     functions.</rdfs:comment>
   </owl:Class>
 - <owl:Class rdf:ID="VariableType">
    <rdfs:comment>This class can only have two direct sub-classes representing complex type and primary type.
     (this assumtion is used in the semantic advisor java code.)</rdfs:comment>
   </owl:Class>
 - <owl:Class rdf:ID="OptionsMatlabFunction">
```

(c) OWL representation

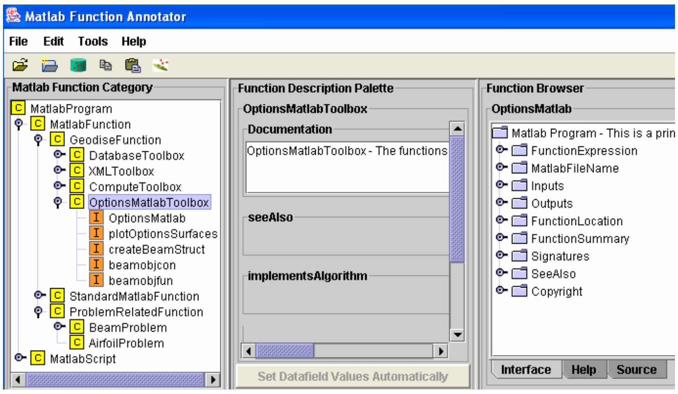
Fig. 3. Function ontology: concepts, properties and representation

Semantic annotation is the process to generate knowledge entities and populate knowledge repositories. Ontologies are knowledge models in which a concept is a structure for preserving knowledge. An instantiated concept, referred to as an instance, is a concrete piece of knowledge. Thus knowledge population is equivalent to instance generation, which is achieved by annotating the raw data source using pre-defined ontologies.

The Protégé OWL Plugin can be used to directly create instances. However, it is designed for knowledge engineers who usually perform several activities in one go, such as knowledge acquisition, ontology editing, knowledge population as well as knowledge base creation. This is a very complicated task and requires professional knowledge engineering expertise. In order to empower resource providers and/or domain experts to capture and model resources' knowledge, we have developed a lightweight knowledge acquisition tool, called Function Annotator, for domain scientists to carry out semantic annotation and create knowledge repositories.

Function Annotator provides intelligent support for knowledge acquisition and modelling, including automatic information extraction, classification and completion, to help create instances. Fig. 4 shows the front-end GUI of Function Annotator. To generate a function instance, a function script is first loaded into the right-hand panel, which can be viewed by clicking the Source tab. The source script is parsed and potential actionable information will be extracted and listed in the same panel by clicking the Interface tab. In terms of the content to be annotated, the Annotator can create an annotation panel (middle panel) automatically from a particular ontological concept (left-hand panel). Annotation is carried out by dragging relevant information from the function browser, dropping it into the annotation panel and filling out relevant fields. Generated instances are listed under the concept with a different symbol as can be seen in the left-hand panel in Fig. 4.

Function Annotator supports OWL representation and manipulation for both ontologies and instances. Generated instances are archived in backend knowledge repositories. Details about the Function Annotator can be found in [18].



Documentation: OptionsMatlabToolbox - The functions

Matlab Program - This is a prin

Fig. 4. Function Annotator

The two approaches to semantic annotation will produce the same knowledge content, which is interchangeable and interoperable, i.e., Function Annotator can load and edit function instances generated by Protégé OWL Plugin, and vice versus. The difference is that Function Annotator is specifically targeted at non-IT professionals, i.e., to enable domain experts and/or resource providers themselves to model and create knowledge for sharing and reuse. This is particularly important in Grid computing because it is not always possible to have knowledge engineers available to manage knowledge for a domain. To make the Grid full of knowledge, resource providers must be given tools to publish and populate knowledge by themselves.

Knowledge repositories are distributed knowledge bases that store EDSO services' semantic descriptions. In GEODISE we have adopted the Instance Store (IS) [19] technology to create EDSO functions' knowledge repository. The IS repository uses a relational database as its permanent storage media and the DL-based reasoner RACER [11] to support reasoning. A function's semantic descriptions, i.e. an ontological concept instance, is stored in the database together with information inferred using the RACER 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 functions' semantic descriptions is then a combination of query against the database and subsumption and classification requests to the reasoner.

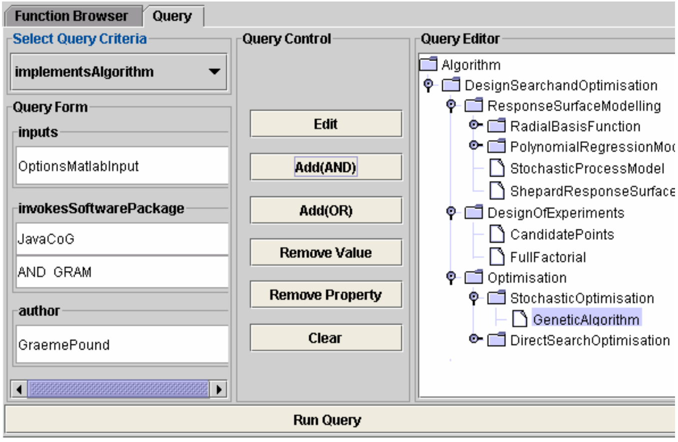
4.2 Knowledge service development

Once knowledge is captured, modelled, represented and stored into knowledge repositories, it can be used to provide knowledge-based support via reasoning and inference. We have developed three knowledge services in GEODISE to facilitate knowledge reuse.

4.2.1 Query services

EDSO functions contain rich metadata, including such information as version, applicability information, performance matrix, developer and organisation, input and output types. The function ontology defines classes of related functions and their properties. By semantic annotation, links are established among metadata, ontological concepts and functions, which facilitates service discovery in terms of the built-in links.

A query service is developed to perform semantic-based function discovery, which is carried out through semantic matching performed by a DL-based reasoner. Fig. 5 shows the query GUI, which is powered by the underlying ontology, i.e. the query criteria, query expression-building forms and the fillers' values of query criteria are all generated automatically from the ontology. The GUI consists of a dropdown list and three panels. The dropdown list contains all query criteria, which are actually the metadata types of resources. The left-hand panel is used for building up an overall query expression. Users can choose one or more criteria from the dropdown list to frame a query expression. The right-hand panel is used to show query results or display all available ontological concepts and/or instances for a selected metadata type. Users can assign a concept or instance to a query criterion. This panel can also be used as a textual editor for users to directly input values for a primitive data type. The middle panel contains a number of control buttons to facilitate query expression construction, including logical conjunction, disjunction, delete, add and ontology loading operations.



Select Query Criteria: implementsAlgorithm

inputs: OptionsMatlabInput

invokesSoftwarePackage: JavaCoG
AND GRAM

author: GraemePound

Fig. 5. Function Query GUI

When the "Run Query" button is clicked all query criteria will be collected and framed into a query expression. The expression will be initially represented as XML on the client side, passed onto the server side via HTTP and then transformed into OWL formats. The underlying DL reasoning engines will reason against the knowledge repository to obtain a set of entities matching the specified criteria. The results are displayed in the right-hand panel.

4.2.2 Advice services

During workflow construction, functions can only be assembled together if their interfaces semantically match each other to some extent, i.e., a function's input semantically consumes the output of another function. Engineers trying to build a workflow for EDSO often face such questions as what should be done next, which resource should be used, how to configure or tune the control parameter of a resource. Such knowledge has actually already been incorporated in the function semantic instances. By semantic reasoning and inference, suggestions about what to do, and how, can be deduced through the knowledge repository. This is especially useful when the function repository is dynamically updated or the number of functions is large, which is the case for Grid applications.

Function configuration advice provides automatically generated advice on function configuration, which suggests default values, ranges and types for functions' control parameters. Most importantly, pre-encoded knowledge such as rules and axioms can give dynamic advice during function configuration in terms of previous computation results and execution context. Recursive semantic decomposition might be used when a function parameter is a complex type, e.g., a structure that contains a list of fields which are either primary types or still complex types. In this case, the semantic interface can be expanded by decomposing this parameter and its subfields until there are no more complex types. This often yields richer semantic interfaces that contain more concepts and relationships for semantic matching.

Function assembly advice suggests which functions can be composed together according to semantic compatibility of their interfaces, which is provided during the process of assembling configured function instances into workflows. Fig. 6 shows the semantic interfaces of the FunctionInput and FunctionOutput of three functions, general_sample_points, parameter_search and check_jobs. The match of number_of_points between general_sample_points's output and parameter_search's input, as indicated by the link, means these two functions could be assembled to form part of a workflow.

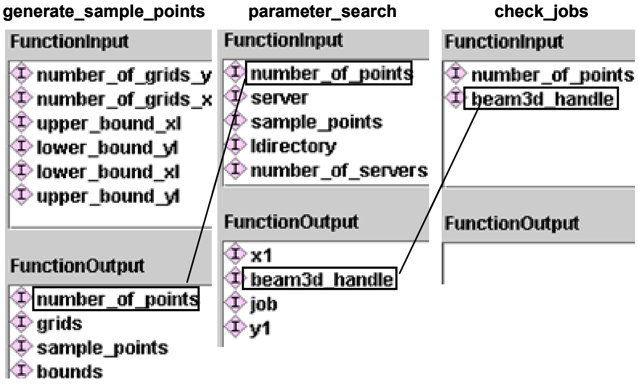


Fig. 6. Semantic matching for function assembly

The advice services involve ontology interpretation, semantic matching and reasoning/inference, which are implemented using Jena OWL ontology API (www.hpl.hp.com/semweb/jena.htm) and DL-based reasoner RACER.

4.2.3 Knowledge management infrastructure

Following the proposed Semantic Web based approach and architecture, we have developed technologies, tools, services and APIs in GEODISE as described in previous sections, which form the core components of a knowledge management infrastructure. Fig. 7 lists the key primitive functionalities that GEODISE knowledge infrastructure provides. Functionalities 1-4 and 9-0 are generic APIs for ontology interpretation and semantic consumption. Functionalities 5-8 are specially tailored to provide advice capabilities for EDSO. By combining different primitive functionalities complex knowledge support can be provided in terms of application context and requirements.

1	List all classes – (all classes defined in the ontology)
2	List subclass of a given class (as defined in the ontology)
3	List all individuals of a class (instances under a particular class, either direct or indirect)
4	List properties of a given individual (declared properties of a particular instance)
5	Expose semantic interface of a given individual function (an example of case 4 on a function)
6	Suggest contextual functions in a workflow
7	Expose in/output parameter individual of a given individual function
8	Decompose a particular parameter individual
9	Documentation (provide human readable comment on any semantic resources)
0	Individual exists? (Check instance existence)

Fig. 7. Key functionalities of knowledge management infrastructure

4.3 Knowledge support for application systems

We have integrated our knowledge management infrastructure into GEODISE PSE to facilitate service access, discovery and composition. Fig. 8 shows the deployment of GEODISE knowledge management system. As can be seen, the Server Side hosts the function ontologies, GEODISE KB repository and a DL-based reasoning engine. The Client Side includes the script-based Matlab execution environment and the GEODISE Workflow Construction Environment (WCE). Client-side applications access and manipulate function's semantic descriptions through GEODISE knowledge management middleware that comprises client-side tools, APIs and a number of KB Management Web Services.

KB Management Web Services are responsible for interacting with underlying knowledge or reasoning components and performing actions. For instance, Advice Service provides recommendation for service selection and configuration, and Query Service performs service discovery. Applications can use either client-side tools such as Function Browser or Query GUI to explore Grid resources directly or APIs to build such functionality in their systems for more complex functions. In GEODISE, we provide a particular type of high-level client-side tool called the KB management toolbox. The toolbox contains a number of Matlab functions that enable Matlab users to access a knowledge repository, retrieve functions and query semantic descriptions. In GEODISE, knowledge services are implemented using Apache Axis framework (ws.apache.org/axis). Client-side tools and APIs interact with knowledge services through KB Service Java Proxies that in turn communicate with knowledge services via SOAP messages.

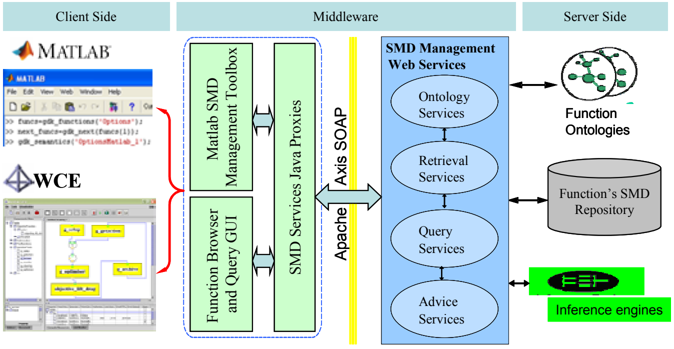


Fig. 8. The deployment architecture of knowledge management system

Semantic service discovery: one typical use of the GEODISE knowledge infrastructure and services is to perform semantic based service discovery. Engineers can use Query Services and its GUI to query the semantic descriptions in GEODISE KB repository and discover required services. Returned services will be listed in the left-hand bottom panel of WCE, see Fig. 9.

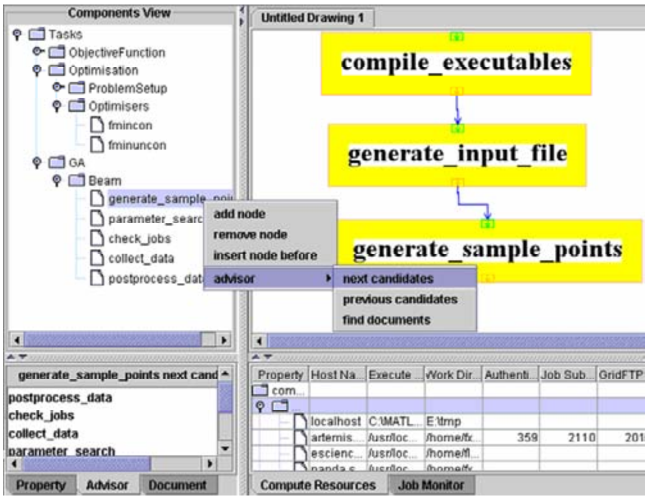
Recommender system for workflow composition: using Advice Services described in 4.2.2, the knowledge management infrastructure provides decision-making advice on selecting suitable EDSO functions and configuring the function's parameters during workflow composition, so that previous domain knowledge and valuable expertise is reused to solve complex problems quickly and by less experienced engineers.

In WCE as shown in Fig. 9a, all function instances are listed in the left-hand upper panel. When a function is dragged and dropped into the composition area in the right-hand upper panel, the advice service will be invoked to deduce its contextual functions. The advice service will then return a list of functions in the left-hand bottom panel that can be deployed before/after the previously selected function. When clicking on a function in the composition area, a form will appear to allow for the configuration of control parameters, here configuration advice will be provided. The WCE will generate a Matlab script, see Fig. 9b, for a workflow and submits it to a Matlab server for execution. PSE also takes care of the workflow management, monitoring and execution, which is beyond the scope of this paper, the interested reader can refer to [20] for further information.

GEODISE PSE together with the integrated knowledge services has been tested and evaluated by GEODISE domain experts and resource providers. Initial results demonstrate that knowledge support has greatly reduced the time for users who have never used the Grid to carry out Grid-enabled EDSO. It also lowers the requirements for users' expertise on EDSO, i.e., novice engineers can solve complex problems that could only be done before by domain experts. As the knowledge management infrastructure has hidden as much as possible the knowledge provision mechanisms and operations, engineers are happy that knowledge support is provided without them knowing the details of the support process.

5 Discussion

The case study has disclosed that initial collaborative analysis in phase one and two of the methodology is the key to the remaining KM activities and also to the final system delivery. Domain experts and/or resource providers



(a)

```
……
% Compile and transfer the beam3d executable to the client
compile_executables( 'blue02.iridis.soton.ac.uk', server,
number_of_servers, ldirectory )

% Generate the input file, and transfer it to the Globus servers
generate_input_file( server, number_of_servers, ldirectory )

% Clean-up. Remove all subdirectories starting with "job"
remove_subdirectories( server, number_of_servers )

% Generate sample points between lower and upper limits
 [ sample_point, number_of_points, bounds, grids ] =
generate_sample_points( 2.5, 3.5, 1.5, 2.5, 3, 3 )

……
```

(b)

Fig. 9. (a) The graphical user interface of the workflow construction environment and (b) the generated workflow script

have domain and resource usage knowledge. They also know what kind of knowledge support is desirable. But they do not have expertise in knowledge engineering. For example, they do not know how such unstructured knowledge can be abstracted, organized and modeled, nor do they know the tools and mechanisms by which such knowledge can be reused. The close interaction between knowledge engineers and domain experts can help domain experts identify what they exactly want and further recognize application and system requirements. Domain experts can help knowledge engineers decide and evaluate the knowledge application scenarios. Many of these points will be familiar to those who experienced the development of first and second generation knowledge-based systems two decade ago. They are no less relevant in the era of Grid computing.

While our methodology provides a generic guideline for carrying out KM activities, it is up to knowledge engineers to decide which technologies are used for individual activities. These choices are application-dependent and requirement-oriented. The right choice is critical, and to some extent, determines the performance of the final systems. For example, in GEODISE we require description based reasoning for function discovery, therefore, we select OWL as our ontology and instance representation language due to its expressive power and its support of DL-based reasoning. However, this comes with a sacrifice of performance. DL-based reasoning over large instances is slow compared to both RDF-based instance reasoning and traditional database systems. Detailed discussions about RDF and OWL based semantic storage and reasoning, performance matrix and future improvement is beyond the scope of this paper but can be found in [21], [18]. This observation suggests that if an application does not require DL-based reasoning, other ontology languages such as RDF could be adopted. This could also indicate the use of a different set of tools, including ontology editors, APIs and query languages. The lesson learnt is that technologies must be selected based on application charateristics and requirements.

We have developed knowledge services in GEODISE to support KM. The decision to adopt a service-oriented implementation was made based on several considerations. Firstly complex Grid applications for which KM have much more added value have increasingly adopted a service-oriented view for modeling and software engineering. Secondly a service-oriented implementation of a KM infrastructure makes it easy to integrate into an OGSA-based Grid framework, thus facilitating the adoption of the infrastructure in Grid applications. Finally a service-oriented KM infrastructure is easy for deployment in heterogeneous distributed environments, thus facilitating the access, sharing and reuse of knowledge.

The application of the proposed approach and methodology in GEODISE has reinforced our view that knowledge management is a co-ordinated endeavor among diverse roles rather than individual, isolated activities. While technologies in each separate area of knowledge management are important, their integration and the potential for synergy in successful problem solving depends on the nature of application, the characteristics of knowledge, the requirement of systems and the choices and tuning of individual technologies. A systematic approach supported with a methodology is not only necessary but indispensable for managing knowledge on the Grid. It also proves that a systematic methodology is very important for such complex activities that involve a multiple phase process and require coordination, collaboration and cooperation among different roles.

One finding from our work is that KM is no longer the job of knowledge engineers or knowledge system developers. Domain experts and resource providers should become key players and actively participate in the KM lifecycle. The Web and Grid are evolving towards globally interconnected knowledge bases. Users will not only obtain knowledge as consumers but will also contribute resources and knowledge as providers. This will see a shift of knowledge acquisition and population from a limited number of specialized knowledge engineers to a larger base of resource providers. This shift implies a need to provide easy-to-use tools for domain specialists. An analogous trend occurred in knowledge acquistion in the 1980s for stand alone knowledge-based systems.

Another finding is that end-users do not care how knowledge management activities are carried out, and what technologies are used. Their concern is with what they need to do in order to get knowledge support, and whether or not knowledge services are easy-to-use. This requires that knowledge services should hide as much of the technical details from end users as possible, so that the end users are able to exploit KM technologies for problem solving but without knowing about ontologies, reasoning, and a whole raft of KM terms and jargon.

6 RELATED WORK

Our work is inspired and underpinned by research results from the Semantic Web [6], [22], [23], but research issues in our work go beyond those addressed by the Semantic Web community. The Semantic Web focuses on adding meaning to documents, databases and web pages, and the emphasis is placed on information integration, search and retrieval [24]. Our work targets Grid resources, which, in addition to semi-structured documents and well-marked web pages, consist of computational systems, software codes, capabilities and storage. The objectives of our work are to capture knowledge of Grid resources, and make it explicitly represented and delivered. The ultimate goal is the seamless sharing and effective reuse of resources' knowledge for problem solving. This is even more challenging then the Semantic Web's ambition of global information sharing.

Nevertheless, Semantic Web services share some common goals with our work, e.g. using semantic metadata for service discovery, seamless access and automation. They have been investigated in two main initiatives. The first one is OWL-S [25], an upper level ontology for describing Web services, specified using OWL. The second one is the Web Service Modelling Ontology (www.wsmo.org), a conceptual framework for the semantic description of Web services, underpinned by the Web Service Modelling Framework (WSMF) [26] and specified using the Web Service

Modelling Language (www.wsmo.org/wsml). While some work on using semantic metadata, e.g. semantic service description [27], discovery [28] and composition [29], [30], has been conducted, it is often based on simple, sometimes artificial, application scenarios. Its purpose is to provide a proof of concept prototype rather than to solve real world problems. At present, Semantic Web services focus on the provision of superficial descriptive information for resource discovery. There is no mechanism in place to model detailed domain knowledge of services; nor is there much effort towards a systematic and integrated approach to streamlining the process of knowledge capture, modelling, archiving, manipulation, retrieval and reuse through ontologies and annotation.

The use of Semantic Web technologies for Web-oriented knowledge management has been investigated in the On-To-Knowledge (www.ontoknowledge.org) project, the HALO project (www.projecthalo.com) and the Advanced Knowledge Technologies (AKT) project (www.aktors.org /akt) and others [31]. On-To-Knowledge focuses on the use of ontologies to perform various tasks for information integration and mediation. AKT is more ambitious, aiming to develop and extend a range of methods and technologies using a Semantic Web infrastructure to provide integrated methods and services for the full lifecycle of knowledge management. All three projects have concentrated on the development of underlying knowledge infrastructure such as methods, tools and services. For example, On-To-Knowledge has produced OIL – the ontology inference layer, which evolved later into OWL. AKT has produced, amongst a list of methods, tools and techniques, the 3Store technology [21] that supports large-scale online semantic repositories of RDF. While the application focus of On-To-Knowledge is on knowledge management for large numbers of distributed semi-structured documents, AKT has looked to provide knowledge support for distributed large-scale information systems and was one of the inspirations for the GEODISE project. A number of other AKT spin-off projects have also looked at combining Grid and Semantic Web capabilities – for example the MIAKT project in the domain of e-Health [32].

Our work is contrasts to that in On-To-Knowledge in several aspects: Firstly, we focus on Grid applications such as e-Science, e-business and e-Health where resources are heterogeneous, diverse, domain-dependent, and most importantly knowledge intensive. Secondly, our research emphasis goes beyond information sharing to resource sharing, and in particular the use of knowledge for problem solving. This requires deep knowledge be captured and made available and reusable. Thirdly, our work concentrates on a co-ordinated and integrated approach to knowledge management rather than on individual components. We are especially keen on a methodology that helps not only IT professionals but, most importantly, scientists to use such an approach for Grid applications.

The myGrid project (www.myGrid.org.uk/) and METEOR-S project (lsdis.cs.uga.edu/projects/meteor-s) have explored the Semantic Web approach for managing knowledge of Web/Grid services. While we share common goals, we adopt different approaches in modelling, representing and using metadata and semantics. Both myGrid and METEOR-S projects deal with extra metadata and semantics by extending existing Web Service standards WSDL and UDDI (www.uddi.org). For instance, METEOR-S adds semantics to WSDL via annotation, and both of them extend a service's UDDI tModel to accommodate extra metadata. MyGrid also developed a semantic-enabled registry called GRIMOIRES [33] for semantic-based service discovery. Our approach is to have an extra knowledge layer on top of UDDI and WSDL, and it is based on OWL-S and has uniform OWL representation. The extra knowledge layer is flexible to handle rich domain knowledge, thus able to provide advice on service configuration and selection.

Our work is also distingushed from myGrid and METEROR-S with regard to semantic metadata representation and related query mechanisms. In METEROR-S metadata are embedded in WSDL files and UDDI data structures, and represented in XML syntax. The semantics of a service description is extracted by mapping data constructs in WSDL/UDDI to corresponding concepts of the ontologies. In myGrid all information of a service is represented in RDF and stored in a triple store. Interfaces have been provided to allow users to query the service directory using the RDQL query language. In GEODISE we use the OWL ontology language to represent semantic information. As OWL is built upon the constructs of RDF schema and description logic, it has more expressive power for modelling and representing knowledge. This feature is particularly useful for managing complex domain knowledge and providing knowledge-based decision-making support in terms of the built-in reasoning capabilities. GEODISE is also different from myGrid and METEROR-S by its focus on using domain knowledge to provide intelligent advice for problem solving such as workflow composition.

The closest research to our work is the ongoing EU OntoGrid project (www.ontogrid.net). OntoGrid addresses one of the grand challenges in Grid computing, i.e., the ability to explicitly share and deploy knowledge to be used for the development of innovative Grid infrastructure, and for Grid applications. It intends to leverage Semantic Web technologies to develop a technological infrastructure, and a sound methodology to facilitate the use of the approach and infrastructure. Our work has studied some of the issues that will be dealt with in OntoGrid. To some extent, our research results have informed and inspired the OntoGrid project.

7 CONCLUSIONS

In this paper, we have analysed the nature of Grid computing and its requirements for knowledge support. Following this, we have discussed the characteristics of knowledge management in distributed environments such as the Web/Grid, and in particular its distinction from traditional knowledge management practices. We then propose the Semantic Web based approach to knowledge management, which leverages the state of the art in ontologies, ontology languages and DL reasoning technologies. To support the adoption and implementation of the proposed approach, a methodology has been developed to provide guidance for

Grid practitioners to carry out knowledge management for Grid applications.

We have applied the proposed approach in GEODISE in which we have developed domain and function ontologies for modelling knowledge of EDSO resources. We have created knowledge repositories for semantic function instances; and key knowledge services have been developed to provide knowledge support. We have integrated the knowledge management infrastructure into GEODISE PSE. All these activities have been conducted by following the specification and principles of the conceived methodology step by step. The application systems have been used and evaluated. Initial results have demonstrated that the use of knowledge management in GEODISE makes Grid enabled EDSO easier, quicker and more efficient. This demonstrates that both the approach and the methodology are feasible and promising.

ACKNOWLEDGMENT

This work is supported by the UK EPSRC GEODISE e-Science pilot project (GR/R67705/01). The authors gratefully acknowledge the contributions from all members of the GEODISE team as well as discussion with EPSRC projects myGrid (GR/R67743/01) and AKT (GR/N15764/01(P)).

REFERENCES

[1] I. Foster and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure," 2nd ed., Morgan Kaufmann, 2004.

[2] I. Foster, C. Kesselman, J. Nick and S. Tuecke, "Grid Services for Distributed System Integration," Computer, vol.35, no.6, pp.37-46, 2002.

[3] G. Fox and D. Walker, "e-Science gap analysis, technical report," www.nesc.ac.uk/technical_papers/UKeS-200301/GapAnalysis30June03.pdf, 2003.

[4] D. De Roure, N.R. Jennings and N.R. Shadbolt, "The Semantic Grid: A Future e-Science Infrastructure," Grid Computing: Making the Global Infrastructure a Reality, F. Berman, et al eds., ISBN0470853190, pp.437-470, 2003.

[5] H. Zhuge, "Semantic Grid: scientific issues, infrastructure, and methodology," ACM Communication, vol.48, no.4, pp.117-119, 2005.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, Vol. 284, no.4, pp.34-43, 2001.

[7] M.H. Eres, G.E. Pound and S.J. Cox, "Globus Installation Procedure," http://www.geodise.org/files/ tutorials/Globus_Installation.pdf, 2004.

[8] C.A. Goble, D. De Roure, N.R. Shadbolt, and A.A.A. Fernandes, "Enhancing Services and Applications with Knowledge and Semantics," The Grid 2: Blueprint for a New Computing Infrastructure, I. Foster and C. Kesselman, Eds., Morgan-Kaufmann, pp. 431-458, 2004.

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

[10] I. Horrocks, U. Sattler and S. Tobies, "Practical reasoning for expressive description logics," Lecture Notes in Artificial Intelligence, No.1705, H. Ganzinger, D. McAllester, and A. Voronkov, eds, pp.161-180, 1999.

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

[12] H. Zhuge, "The Knowledge Grid," World Scientific Publishing Co., Singapore, 2004.

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

[14] M.H. Eres, G.E. Pound, Z. Jiao, J.L. Wason, F. Xu, A.J. Keane and S.J. Cox, "Implementation and utilisation of a Grid-enabled problem solving environment in Matlab," Future Generation Computer Systems, vol.21, no.6, pp.920-929, 2005.

[15] G. von. Laszewski, I. Foster, I. Gawor and P. Lane, "A Java Commodity Grid Kit," Concurrency and Computation: Practice and Experience, vol. 13, no. 8-9, pp. 643-662, 2001.

[16] A.J. Keane, "OPTIONS Design Exploration System," http://www.soton.ac.uk/~ajk/ options/welcome.html.

[17] L. Chen, S.J. Cox, C.A. Goble, A.J. Keane, A. Roberts and N.R. Shadbolt, "Engineering Knowledge for Engineering Grid Applications," Proc. Euroweb 2002 Conference, The Web and the GRID: from e-science to e-business, pp.12-25, 2002.

[18] L. Chen, S.J. Cox, F. Tao, N.R. Shadbolt, C.A. Goble and C. Puleston, "Empowering Resource Providers to Build the Semantic Grid," Proc. 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp.271-278, 2004.

[19] I. Horrocks, L. Li, D. Turi and S. Bechhofer, "The instance store: DL reasoning with large numbers of individuals," Proc. 2004 Description Logic Workshop, pp.31-40, 2004.

[20] Z. Jiao, J.L. Wason, W. Song, F. Xu and M.H. Eres, "Databases, Workflows and the Grid in a Service Oriented Environment," Proc. Euro-Par 2004, Parallel Processing, pp.972-979, 2004.

[21] S. Harris and N. Gibbins, "3store: Efficient Bulk RDF Storage," Proc. 1st International Workshop on Practical and Scalable Semantic Systems (PSSS'03), pp.1-15, 2003.

[22] A. Naeve, "The Human Semantic Web: Shifting from Knowledge Push to Knowledge Pull," J. Semantic Web and Information Systems (IJSWIS), vol.1, no.3, pp.1-30, 2005.

[23] N.R. Shadbolt, T. Berners-Lee and W. Hall, "The Semantic Web Revisited," IEEE Intelligent Systems, vol.21, no.3, pp. 96-101, 20006.

[24] I. Sheth, C. Ramakrishnan and C. Thomas, "Semantics for the Semantic Web: The Implicit, the Formal and the Powerful," J. Semantic Web and Information Systems (IJSWIS), vol.1, no.1, pp.1-18, 2005.

[25] D. Martin, "OWL-S 1.1 Release," www.daml.org/services/owl-s/1.1, 2004.

[26] D. Fensel and C. Bussler, "The Web Service Modeling Framework WSMF," Electronic Commerce: Research and Applications, vol.1, no.2, pp.113-137, 2002.

[27] W. Johnston, "Semantic Services for Grid-Based, Large-Scale Science," IEEE Intelligent Systems, vol.19, no.1, pp.34-39, 2004.

[28] E. Sirin, B. Parsia and J. Hendler, "Filtering and Selecting Semantic Web Services with Interactive Composition Techniques," IEEE Intelligent Systems, vol.19, no.4, pp.42-49, 2004.

[29] L. Chen, N.R. Shadbolt, C.A. Goble, F. Tao, S.J. Cox, C. Puleston and P.R. Smart, "Towards a Knowledge-based Approach to Semantic Service Composition," Lecture Notes in Computer Science, LNCS 2870, pp.319-334, 2003.

[30] I. Medjahed, A. Bouguettaya and A.K. Elmagarmid, "Composing Web services on the Semantic Web," Journal of VLDB, vol.12, no.4, pp.333-351, 2003.

[31] T. Hädrich and T. Priebe, "A Context-Based Approach for Supporting Knowledge Work with Semantic Portals," J. Semantic Web and Information Systems (IJSWIS), vol.1, no.3, pp.64-88, 2005.

[32] S. Dasmahapatra, D. Dupplaw, B. Hu, H. Lewis, P. Lewis and N.R. Shadbolt, "Facilitating multi-disciplinary knowledge-based support for breast cancer screening," J. Healthcare Technology and Management, vol. 7, no.5, pp. 403-420, 2006.

[33] S. Wong, V. Tan, W. Fang, S. Miles and L. Moreau, "Grimoires: Grid registry with metadata oriented interface: Robustness, efficiency, security," IEEE Distributed Systems Online, vol.6, no.10, 2005.

Liming Chen received the BSc and MSc degrees from Beijing Institute of Technology, China, in 1985 and 1988, respectively. He obtained the PhD degree in Artificial Intelligence from De Montfort University, United Kingdom, in 2002. He is currently a lecturer at the School of Computing and Mathematics, University of Ulster, UK. He previously worked as a senior research fellow in the School of Electronics and Computer Science at the University of Southampton, UK. He serves as a PC member for several international conferences. His research interests include the Semantic Web, Semantic Grid, Semantic Web Services, information/knowledge fusion and reasoning, intelligent recommender systems, agent technologies and their applications in e-science and e-business.

Nigel R. Shadbolt graduated with a BA from Newcastle upon Tyne University in England. He obtained his PhD from Edinburgh University's Department of Artificial Intelligence in 1984. He is currently Professor of Artificial Intelligence in the School of Electronics and Computer Science at the University of Southampton. He is a Fellow of the Royal Academy of Engineering, Fellow and President elect of the British Computer Society. He is the Director of the EPSRC Interdisciplinary Research Collaboration in Advanced Knowledge Technologies (AKT). From 2001 to 2004 he was Editor in Chief of IEEE Intelligent Systems and in 2005 he was appointed Emeritus Editor in Chief. He also serves as a senior member of various national and international advisory bodies. His research ranges from the Semantic Web to biologically inspired robotics.

Carole A. Goble is a Professor in the School of Computer Science in the University of Manchester. She is a co-director of the Northwest Regional e-Science Centre and the Information Management Group, and the director of UK e-Science myGrid project. She is also Editor-in-Chief and manager of the Journal of Web Semantics: Science, Services and Agents on the World Wide Web. She also serves as a senior member of various national and international organizations. Her research has been focused on the Semantic Web, e-Science, the Grid and in particular the Semantic Grid.