Towards the Semantic Grid: Enriching Content for Management and Reuse

Feng Tao[+], Simon Cox[*], Liming Chen[+], Nigel Shadbolt[+], Fenglian Xu[*], Colin Puleston[#] and Carole Goble[#], Wenbin Song[*]
ft@ecs.soton.ac.uk
[+]Electronics and Computer Science & *School of Engineering Sciences, University of Southampton
[#]Dept of Computer Science, University of Manchester

Abstract

Knowledge and Semantic Web technologies are evolving the Grid towards the Semantic Grid [18] to facilitate knowledge reuse and collaboration within a community of practice. In the Geodise project we are exploring the application of a range of knowledge and Semantic Web technologies to assist users in solving complex problems in Engineering Design Search and Optimization (EDSO), in particular enabling semantically enriched resource sharing and reuse.

The target of content enrichment in Geodise ranges from command usage described in software manuals, a set of profile data, to a workflow customized to solve a particular problem. They become semantically enriched when their representations are delivered using a set of shared semantics which are well recognized in the domain. Knowledge acquisition and knowledge modelling (in particular ontology building) are the key steps to build these semantics.

The repository of semantically enriched content can be regarded as a resource based on which various knowledge services [4] are made available to and integrated into a Problem Solving Environment (PSE) to assist an engineer in design optimization routines. For example, when constructing a script to generate a computational mesh, an ontology assisted domain script editor can provide syntax highlighting and context sensitive knowledge-driven auto-completion and advice. A rule-based workflow advisor gives guidance on building a domain workflow by reasoning over semantically enriched system states [6]. The workflow construction process itself is driven by task ontology so as to guarantee that the resulting workflow instances are enriched with consistent semantics. In this paper we demonstrate how a number of technologies have been deployed in EDSO.

1. Introduction

The concept of Semantic Grid arises with the parallel development of Semantic Web and grid-computation, in particular the endeavor of applying the former on the latter. As was defined by Foster, Kesselman and Tuecke in [2], grid computation distinguishes itself from the traditional distributed computing by its extended emphasis on large scale coordinated resource sharing among dynamic collections of individuals, institutions and resources. The Semantic Web is an extension of the traditional web in which information is formally annotated using some commonly recognized metadata (e.g. an ontology). In this way, data on the web is defined and linked to facilitate more effective discovery, automation, integration and reuse across various applications and users. By analogy such techniques could be deployed to facilitate resource sharing in a grid environment. Furthermore, where activities may be domain knowledge intensive, they can provide methods to assist new users to exploit resources. In this paper, we address how this problem can be relieved by using Semantic Grid related technology, in particular, in the context of Geodise [5] project, content enrichment for knowledge reuse and management.

Geodise is one of the e-Science pilot projects which addresses multi-disciplinary scientific collaboration, data management and process enactment on a global scale. In particular, Geodise focuses on the domain of EDSO and aims to provide an integrated PSE that exploits grid-computation for simulation and optimization. We are also exploiting a range of knowledge technologies, such as ontologies and RDF, to facilitate knowledge reuse in EDSO processes. We have modeled knowledge either pre-acquired from domain experts or exposed as best practices during their interaction with the PSE. This knowledge is then transformed to shared and semantically enriched resources to enable global knowledge reuse and collaboration within an EDSO community of practice.

The challenge of content enrichment is set in the context of six key stages in the life cycle of management and engineering of knowledge. They are Knowledge Acquisition (KA), modelling, reuse, retrieval, publication and maintenance as specified in [1]. In this paper, we discuss its application and involvement in the first three stages through scenarios and working examples.

2. Preliminary

There are some preliminary steps which lay out the foundation to content enrichment and knowledge reuse.

2.1. Knowledge acquisition

Knowledge exists in various sources such as domain experts' heads or user manuals for domain applications. The role of knowledge acquisition is to access these sources and efficiently extract the key information so as to form a foundation for the following knowledge stages in the knowledge life cycle. Interview with domain experts is the most common technique [17] to acquire knowledge directly from the domain experts. The whole process is often audio-recorded so that it can be transcribed into texts for storage and further processing. In Geodise, audio recorded interviews have been carried out, transcribed and further processed by using the protocol editor in PC-PACK [13], which has been developed by Epistemics for knowledge acquisition purposes.

Similarly, key concepts can be identified and extracted from domain documents by using PC-PACK. As shown in Figure 1, the protocol editor enables key concepts to be extracted by allowing users to highlight text with markers of different colors, which represent different types of keywords.

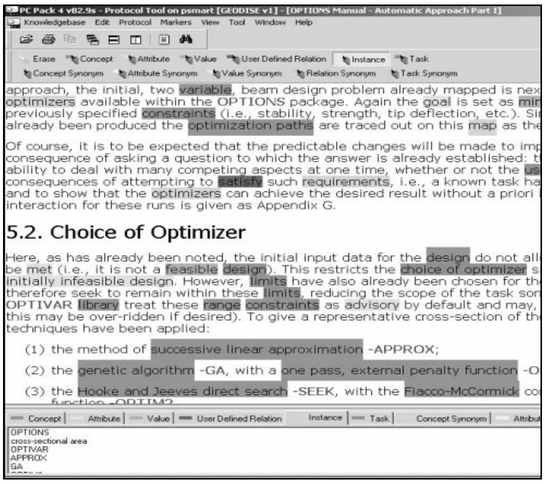

Figure 1 Knowledge acquisition using PC-PACK

The purpose of knowledge acquisition for content enrichment is to identify key concepts and relationships for a conceptualization of the domain in the form of an ontology. This ontology can be expressed in an explicit format and serves as a common grounding for enriching content in a later phase.

2.2 Knowledge modelling using Ontology

An ontology is a standard description of concepts and their relationships that are being used or shared in a specific domain.

For example, if we might consider the concept "parent" as a "person" who has at least one "child". This is part of an ontology. Now we have a person instance identified by his name, "Tom". If we say "Tom is a parent", then according to the ontology, we know the instance "Tom" is a "person" and has a property "has-children". Thus when we use this ontology to express the instance "Tom", the instance becomes semantically enriched with the ontology. Furthermore, this semantic enrichment of "Tom" allows understanding about "Tom" and could enable an agent to process instances automatically.

In Geodise, the purpose of knowledge engineering is to model knowledge and resources in a semantically rich manner so as to share them in the domain community. An ontology in this setting establishes the standard set of terminologies in the EDSO domain and makes sure that these terminologies (and their relationships) are always explicitly expressed so as to reduce ambiguity in knowledge reuse. This is particularly important when sharing resources across different applications for automating collaboration and computation on a wide scale. We demonstrate this through some examples and scenarios in section 3.2

3. Enriching content for knowledge reuse

The result of KA allows us to construct ontologies that contain conceptual vocabularies and underlying templates for the knowledge base. Using ontologies it is then possible to enrich content in a semantically consistent way. For example, a particular instance of a problem set up and optimization schedule is the result of content enrichment by using profile ontologies. Instances described in this way become semantically enriched with a shared and consistent set of ontologies, and can therefore be searched, reused and understood easily by various domain users and applications. This process is illustrated in Figure 2 and demonstrated in section 3.2

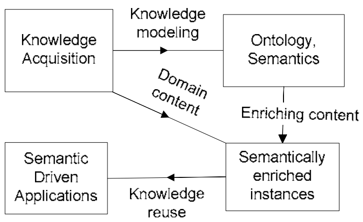

Figure 2 Enriching content for knowledge reuse

3.1 Key issues

There are several key issues in content enrichment and instance management:

1) Ontology can be built by various tools and expressed in various languages. In Geodise, ontologies have been built through Protégé 2000 [14] and OilEd [12] in RDF [15] and DAML+OIL language [3] respectively. More recently, OWL (Ontology Web Language) is being developed in W3C to assist easy publishing of ontology on the web [16]. OWL is developed as a vocabulary extension to RDF and is derived from DAML+OIL with more description power than RDF, such as disjoint and cardinality.

2) Instance generation - where we are concerned with configuring the system to tackle a particular engineering problem, or configuring the parameters for a specific optimization method. Several methods of generating instances have been investigated and experimented with. These include manually generating instances in protégé 2000 ontology editor; An XML template based approach where users fill out an XML-Schema backed template with concrete values and generate an XML instance that conforms to the XML-Schema. In the Semantic Web approach, ontologies can be used directly to drive automatic form generation. The Jena RDF framework [9] has been adopted for instance generation where each instance is an RDF file backed with its RDF schema (RDFS) as well as a DAML + OIL ontology as an semantic extension.

3) Instance reuse – the instances can be reused in different forms. Command syntax can be modeled as instances which are then loaded in a domain scripts editor to help user editing domain scripts through syntax highlighting and auto-completion; Workflow instances are available so that they can be loaded and modified accordingly to define a similar problem/solution schedule. We will demonstrate this in the next section.

4) Instance storage and querying – the generated instances can be stored in a repository as the knowledge content of the domain. We intent to adopt the triple store technology developed in the AKT project [1]. There are query languages especially designed for RDF and the DAML+OIL, a review of ontology storage and querying tools can be found at [11].

3.2 Application scenarios and examples

Ontology driven forms

In Geodise, the EDSO process begins with geometry design conducted by a CAD designer providing two files: a model file and a Matlab structure containing all necessary fields that describe the component or device (problem definition). A STEP/IGES file can be produced by running CAD on the model file. The STEP file is only readable by engineering software such as Gambit. The Matlab structure is designed to capture all necessary metadata about the problem and produce a high level human readable abstraction. As a problem pre-set, the Matlab structure (Figure 3) captures all problem definition related information embedded in the STEP file, such as all the possible design parameters which may be varied to modify the design.

```
       standard: [1x1 struct]
      modelname: 'NacelleRR'
        version: '0.80'
           type: 'cad'
 parameternames: {'SCARF_ANGLE'  'CENTER_OFFSET'  'REAR_EXTEND'}
  inputfilename: 'nacelle.inp'
 outputfilename: 'nacelle.stp'
      nooffiles: 3
        command: 'proe2001 '
      arguments: ' nacelletostp.txt -g:no_graphics'
        package: [1x1 struct]
      condorsub: [1x1 struct]
        comment: [1x57 char]
```

Figure 3 MatLab structure for Geometry

The analysts can also change the default value as necessary. The result is an instance of a problem setup in the problem profile. The analyst can also load existing instances from the problem profile repository to carry on analyzing work conducted previously.

description	Beam design problem
lastTimeUsed	2003-02-17T09:30:47
timeCreated	2001-12-17T09:30:47
user	barry
xmlns	http://www.geodise.org/knowledge
xmlns:*xsi*	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLoca...	http://www.geodise.org/knowledge D:\geodise\XML_Templates\problem_mo.xsd

designVariables (3)

	name	meaning	unit	limit	fi
1	width	beam width	mm	limit	false
2	breath	breath of the beam	mm	limit	false
3	material_type	beam material	null	limit	true

parameters (3)

	name	unit	meaning	value
1	length	mm	length of the beam	250
2	force	N	force on the tip of the beam	5000.0
3	density	kf/m3	density of the beam material	7850.0

objectiveFunction

objective	minimize the cross-section area of the beam
codeURL	d:\geodise\beam1s.exe
singleRunTime	PT30S
estimatedLowe...	20
estimatedUpper...	30

singlePeak	true
optimisation	not defined

Figure 4 XML Schema of EDSO problem setup

However, in a collaborative environment, different CAD designers may use different metadata to describe a device in the Matlab structure. This may cause confusion and inconsistency and inhibit sharing of previously generated CAD designs. An ontology can be used to allow for consistent description of components.

The Ontologies can be maintained separately at a centralized place, as demonstrated in Figure 4 and used in the construction of a Matlab structure to describe a device: CAD designers can interact with a set of ontology driven forms demonstrated in Figure 5 and Figure 6, which are automatically generated based on a controlled set of vocabularies and relationships specified in the ontology. Once the form is finished by the CAD designers, an instance of the component description is ready. This instance is passed to the following phases where it can be loaded again by analysts who, according to design requirements, can further specify the desired design variables by manipulating (e.g. checking off some parameters) the list of design parameters, or by changing the range and default value of some parameters, etc. We call this analyst operation "problem setup". These happen in a similar GUI and once this is finished, we have an instance that represents a particular concrete problem setup. Note that examples here are based on XML/Schema so far and are only for demonstrating the scenario. The auto-GUI rendering uses Jaxfront [8] and the semantics are expressed in XML schema using XML spy [19].

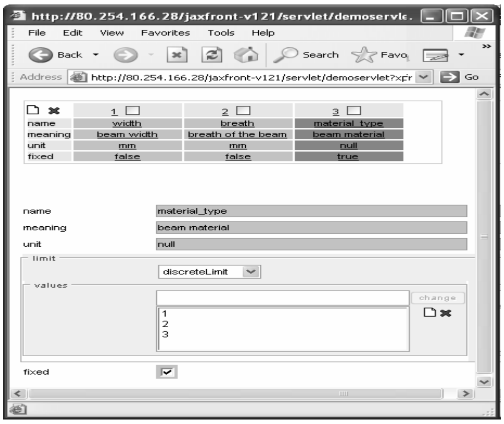

Figure 5 Instances of design variables

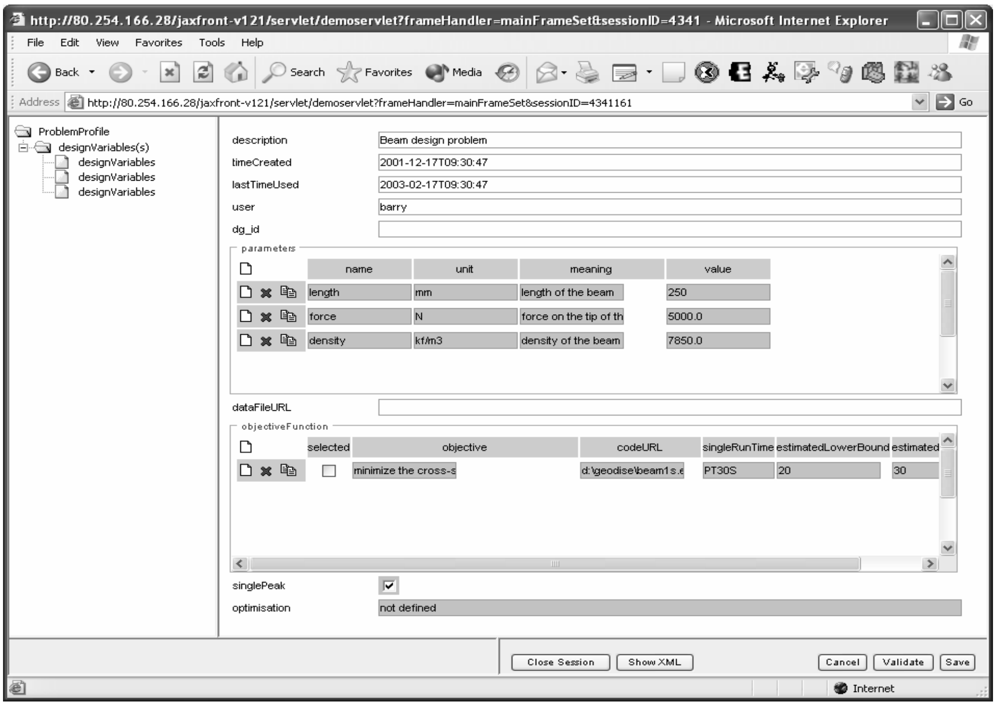

Figure 6 Instance of problem setup

Ontology assisted domain script editor

Another example of making use of the enriched content is the ontology assisted domain script editor. Content in the software manual for the script editor is processed and enriched using a pre-defined ontology. This is demonstrated in Figure 7 where instances of command usages are generated manually in Protégé 2000 based on the usage ontology and the corresponding usage entry in the Gambit command manual, which is a tool for generating meshes from a geometry. Each Gambit command can operate with a set of keywords and parameters in certain syntax and grammar. In Geodise, engineers need to edit these domain scripts frequently with the guidance from the manual to make sure that scripts are correct.

The ontology assisted domain script editor makes use of the pre-built command usage instances and colorizing the scripts syntax. It also provides real-time context sensitive hinting and auto-completion as illustrated in Figure 8. All these functionalities operate by consuming the semantically enriched content – the Gambit command usage instances.

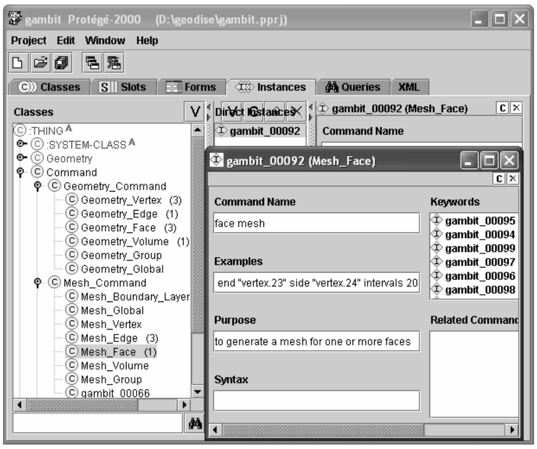

Figure 7 Building Gambit command ontology

Since the editor can load in any ontology, it is domain independent and has the potential to assist script editing in any other domain as long as the corresponding ontology is available.

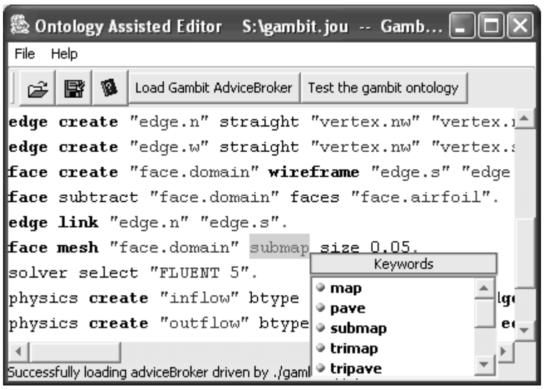

Figure 8 Ontology assisted Gambit script editing

We have also demonstrated not only hints on parameters to configure a command, but also horizontal suggestions of "next steps" based on expert knowledge. To develop this further would require additional knowledge capture.

Workflow editing and semantic instance generation

Scientific activities often involve constructing a workflow either manually or automatically to realize a particular experiment or series of computations. In the service-oriented Grid computing paradigm this process amounts to discovering resources (services) on the Grid and composing these 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 discovery of services at each point in the workflow often depends on the results of executing the preceding step. Moreover, the selection of a service from a set of competing services with similar capabilities 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 goal. 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.

We have developed a Workflow Construction Environment (WCE) as shown in Figure 9 for Geodise, which is intended to (1) exploit the semantically enriched services for semantic-based service discovery and reuse, (2) generate semantic workflows for the use of future problem solving, and (3) provide knowledge-based advice on service composition. The knowledge-based recommender system has been discussed in [7]; here we focus on the exploitation of semantic and instances for workflow construction.

Semantic service description is undertaken using ontologies accessed via the ontology services. As the DAML-S service ontology only provides the basic schema for describing a web service, it does not provide the vocabulary with which to describe specific services in different scientific domains. Therefore, domain specific ontologies are used to incorporate domain specific functions and terminology in creating semantic service descriptions. The process of specifying semantic service descriptions is carried out in two steps. Firstly, domain ontologies, such as the task ontology and the function ontology, are created. Then, the domain specific service ontology is built using concepts from the domain ontologies. The semantic descriptions of domain-specific services are actually instances of concepts from the service ontology. Semantic service descriptions are stored in the Semantic Service Description component.

The main components for semantic resource enrichment, discovery and reuse are the Component (Service) Editor (the middle right panel), Ontology Browser (the left panel) and the Workflow Editor (the middle panel). Each of them presents relevant structures and information via the control panel. The Component (Service) Editor is a frame-like data-storage structure. It is used to specify a service description for service discovery or to define a service directly by filling in the required data fields. The structure of the Component Editor is dynamically generated in accordance with the service ontology, thus semantically enriching the service when the service is defined. Service discovery is accomplished by the use of a semantic-based search engine. It is realized through reasoners such as FaCT or MatchMaker [10] acting on the semantic descriptions of services. The services that fulfills users' requirements will be returned to users as the basis for selection in the context of workflow specification. The Ontology Browser displays ontologies that provide service templates for workflow construction. Workflows are built in the Workflow Editor in which users either discover an appropriate service via semantic service matching or specify a semantically enriched service afresh. These services are connected in a semantic-consistent way to form a workflow.

Each time a workflow is constructed for a particular design problem, it can be archived to form a semantically enriched problem/solution within a knowledge repository. This facilitates the re-use of previous designs, while avoiding the overhead of manually annotating the solution.

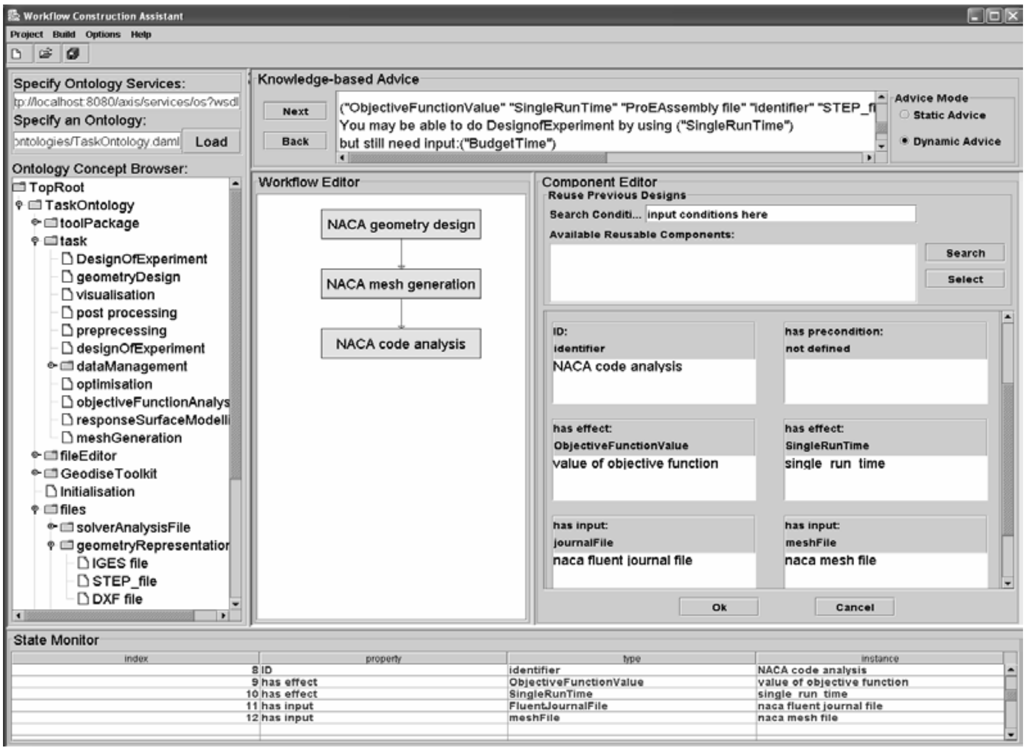

Figure 9 Knowledge guided workflow composer

4. Conclusion and future work

In this paper, Semantic Grid knowledge technologies have been discussed in the context of the first three steps of knowledge life cycles: Knowledge acquisition, modelling and reuse. We demonstrate through several examples and scenarios the content enrichment for knowledge management and reuse. We believe that these technologies are the first step towards extending the current grid in a way that information and resources are given well defined meaning to allow them to be transparently shared and re-used. In the future we plan to target more grid-enabled resources for content enrichment and reuse.

Acknowledgements

This work was undertaken as part of the EPSRC funded Geodise (GR/R67705/01).

References

[1] Advanced Knowledge Technologies http://www.aktors.org

[2] The Anatomy of the Grid: Enabling Scalable Virtual Organizations, I. Foster, C. Kesselman, S. Tuecke. International J. Supercomputer Applications, 15(3), 2001, http://www.globus.org/research/papers/ogsa.pdf

[3] DAML+OIL homepage, http://www.daml.org

[4] L.Chen, S.J.Cox, C.Gobel, A.J.Keane, A.Roberts, N.R.Shadbolt, P.Smart, and F.Tao, "Engineering Knowledge for Engineering Grid Applications" EuroWeb2002 - The Web and the GRID: from e-science to e-business, Oxford, UK, http://www1.bcs.org.uk/DocsRepository/03700/3775/chen.pdf, 2002, pp. 12-25

[5] GEODISE project homepage http://www.geodise.org

[6] Feng Tao, Liming Chen, Nigel Shadbolt, Graeme Pound, and Simon Cox, "Towards the Semantic Grid: Putting Knowledge to Work in Design Optimisation" , proceeding of the 3rd International Conference on Knowledge Management (I-KNOW '03), Graz, Austria, July, 2003, page 555-566

[7] Chen, L, Shadbolt, N.R, Goble, C , Tao, F., Cox, S.J., Puleston, C., and Smart, P. Towards a Knowledge-based Approach to Semantic Service Composition, 2nd International Semantic Web Conference (ISWC2003), 20-23 October 2003, Florida, USA

[8] Generating dynamic user interfaces based on XML Schema, http://www.jaxfront.com/pages/home.html

[9] An Introduction to RDF and the Jena RDF API, http://www.hpl.hp.com/semweb/doc/tutorial/RDF_API/index.html

[10] Paolucci, M. Kawmura, T., Payne, T. and Sycara, K.: Semantic Matching of Web Services Capabilities. In The First International Semantic Web Conference (2002)

[11] A. Magkanaraki, G. Karvounarakis, T.T. Anh, V. Christophides, D. Plexousakis, “Ontology Storage and Querying”, Technical report No.308, April, 2002

[12] Sean Bechhofer, Ian Horrocks, Carole Goble, Robert Stevens. OilEd: a Reason-able Ontology Editor for the Semantic Web. Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, September 19-21, Vienna. Springer-Verlag LNAI Vol. 2174, pp 396--408. 2001.

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

[14] Protégé homepage, http://protege.stanford.edu/index.html

[15] Graham Klyne and Jeremy J. Carroll, eds., Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Working Draft 23 January 2003. Latest version is available at http://www.w3.org/TR/rdf-concepts/

[16] Deborah L. McGuinness and Frank van Harmelen, eds., Web Ontology Language (OWL Lite and OWL) Feature Synopsis Version 1.0. W3C Working Draft 29 Jul 2002. Latest version is available at http://www.w3.org/TR/owl-features/

[17] G. Schreiber, H.Akkermans, A.Anjewierden, R.D.Hoog, N.Shadbolt, Van de Velde, and Wielinga, Chapter 8 (Knowledge-Elicitation Techniques), Knowledge Engineering and Management, The MIT Press, London, 2000.

[18] The Semantic Grid http://www.semanticgrid.org/

[19] XML spy homepage, http://www.xmlspy.com