Engineering Knowledge for Engineering Grid Applications

L. Chen, N. R. Shadbolt, F. Tao
Department of Electronics and Computer Science, University of Southampton, Highfield, Southampton, SO17 1BJ, U.K.
lc@ecs.soton.ac.uk, nrs@ecs.soton.ac.uk, ft@ecs.soton.ac.uk

S. J. Cox, A. J. Keane
School of Engineering Sciences, University of Southampton, Highfield, Southampton, SO17 1BJ, U.K.
sc@ecs.soton.ac.uk, ajk@soton.ac.uk

C. Goble, A. Roberts
Department of Computer Science, University of Manchester, Oxford Road, Manchester, M13 9PL, U.K.
carole@cs.man.ac.uk, robertsa@cs.man.ac.uk

P. Smart
Epistemics Ltd., Strelley Hall, Nottingham NG8 6PE, U.K.
paul.smart@epistemics.co.uk

Abstract

Computing increasingly addresses collaboration; sharing; and interaction involving distributed resources. This has been fuelled in part by the emergence of Grid technologies and web services. Drawing on our expertise in the Geodise project[1]. We argue that there is a growing requirement for knowledge engineering methods that provide a semantic foundation for such distributed computing. Such methods also support the sharing and coordinated use of knowledge itself. In this paper we introduce a service-oriented knowledge engineering approach that seeks to provide knowledge orientated support for distributed grid-based computing. This approach has been implemented in a generic integrated architecture. The application context is the process of design search and optimisation in engineering. It demonstrates how knowledge has been captured and modelled, as well as illustrating how ontologies have been developed and deployed. The knowledge acquired has been made available and accessible through a portal that invokes a number of basic services.

Keywords: Knowledge Engineering, Ontologies, Grid Computing, Semantic Web, Geodise[1], Design Search, Optimisation

1. INTRODUCTION

Grid technologies [1] have been developed and are now being adopted as a fundamental computing infrastructure for 21st century science and engineering. Grid technologies are meant to support a vision of e-Science where the sharing and coordinated use of diverse resources in dynamic, distributed virtual organisations is commonplace. Recent developments in web service technologies [2] driven by industrial and commercial demands have seen Grid technologies evolving towards an Open Grid Services Architecture (OGSA) [3] (a service-oriented distributed computing metaphor). This sees the Grid as providing an extensible set of services that virtual organisations can aggregate in various ways. Whilst a number of grid applications [4] [5] are being developed and progress has been made on the construction of such an infrastructure, it is apparent that there is currently a major gap between these endeavours and the vision of the Grid that is easy to use, seamlessly automated and in which these flexible collaborations exist on a widely distributed scale. We contend that the Grid ought to exploit knowledge engineering techniques and the emerging metadata infrastructure found in the semantic web community [6] in order to work with heterogeneous information across multiple domains.

Knowledge is information applied to solve particular tasks and achieve specific goals. Effective use, in particular the sharing of knowledge among organisations within a domain, will support faster decision-making, increase productivity and lower costs. Knowledge engineering and management [7] attempt to meet the six

challenges of the knowledge lifecycle - namely, those of acquiring, modelling, retrieving, reusing, publishing and maintaining knowledge. Whilst research has been carried out on each aspect of this lifecycle, in the past each facet of the lifecycle was often developed in isolation from the others. For example, knowledge acquisition was often done with little consideration as to how it might be published or used. At the same time, knowledge publishing paid little attention to how knowledge was acquired or modelled.

Furthermore, with the rapid development of the Internet and Internet-based applications, it has become apparent that research is needed into how to best exploit knowledge in a distributed environment.

Recently work in the area of knowledge technologies [8] [9] has tried to bring together methods, tools, and services to support the complete knowledge lifecycle. One important emerging technology is the development and exploitation of ontologies [10]. These fulfil the important job of providing a common language that computers can understand for particular and generic domains. Ontologies have been deemed as vital to the success of the semantic web [4], and to the goals of information sharing and automated information processing.

A significant number of Grid applications have been initiated under the UK e-science initiative (for example Geodise [11] and MyGrid [12]). An obvious problem faced by these applications is how to integrate knowledge engineering activities into the Grid implementation infrastructure, in particular, the OGSA paradigm. How do we acquire, model, publish and reuse knowledge content?

In this paper we first introduce an integrated service-oriented architecture for distributed knowledge engineering over the Internet. We believe this can meet the requirements of distributed computing for knowledge support. In section 3 we describe in detail each component of the architecture and show how the proposed knowledge engineering approach works by demonstrating how we have used it when engineering knowledge for a Grid application in the area of design search and optimisation. Section 4 presents a simple, practical example of knowledge use. Finally we discuss the initial conclusions from our work on Geodise and point out where this work is likely to go in the future.

2. A KNOWLEDGE INTENSIVE SERVICE-ORIENTED ARCHITECTURE

We have proposed and developed an integrated service-oriented architecture for distributed knowledge management as shown in Figure 1. This broadly fits into the paradigm of the Open Grid Service Architecture (OGSA) and, more importantly, can be seen to underpin the ideas to be found regarding the Semantic Grid [13]. In this framework, knowledge about a specific domain is acquired, modelled and represented using a variety of techniques and formalisms. This knowledge is then saved in a knowledge warehouse that includes ontologies, knowledge bases and other domain related information repositories. A community knowledge portal is provided as an entrance point for users. This is meant to facilitate use of knowledge with different levels of access control. As can be seen from Figure 1, the architecture has a layered modular structure with each component dealing with a specific aspect of the knowledge engineering process.

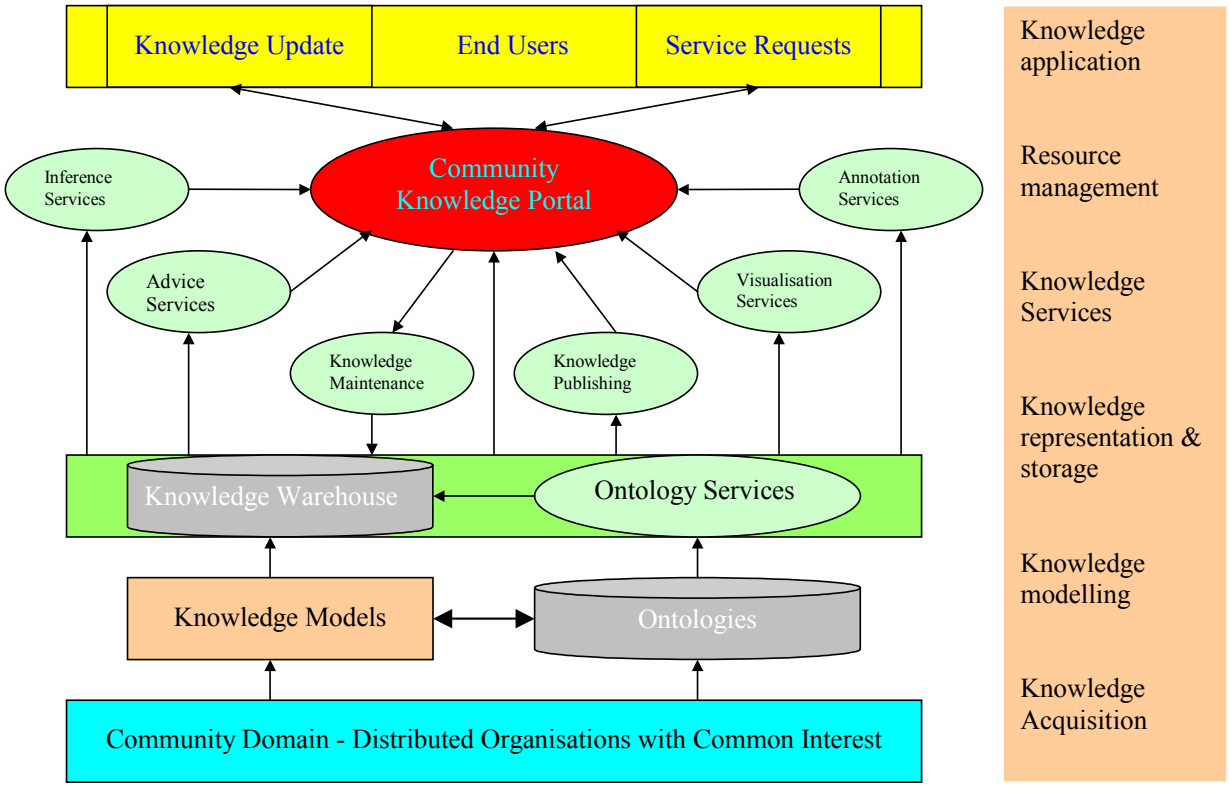

FIG 1: The general architecture

The novelty of the proposed architecture is that it engineers knowledge using a service-oriented approach within an integrated framework. Unlike traditional knowledge engineering practices that concentrate on separate capabilities, this architecture integrates the various activities of knowledge engineering together. Knowledge engineering 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 acquisition can be used to create knowledge bases or to do annotation. These knowledge bases or their associated annotation archives, having been semantically enriched, can be exploited by the services. These services have mechanisms for querying or searching semantic content so as to facilitate knowledge publishing, use and maintenance.

While different knowledge management tasks are coupled together in the architecture, their interactions are not hardwired. Each component deals with different tasks and can make use of different techniques and tools. Each of them can be updated whilst others are kept intact. This type of componentisation makes the architecture robust. It means that new techniques/tools can be adopted at any time, and that the knowledge management system will continue working even if some of its components should fail or become unavailable. Knowledge can be added into the knowledge warehouse at any time. It is only necessary to register the knowledge with the community knowledge portal. After registration all of the services such as publishing and inference can be used to expose the new knowledge for use. Knowledge services can be added in the same way. For example, a data mining service may be added later for automated knowledge acquisition and dynamic update of knowledge repositories. This service-oriented feature makes the architecture flexible and extensible.

The approach is to implement knowledge services as web services and/or Grid services. Each type of knowledge service provides users with a set of APIs that can be used to implement a variety of operations. For example, when using ontology services we can manipulate concepts and properties within ontology in many different ways - e.g., asking for more general or specialised examples of a concept. This service oriented approach should make it easier to reuse and share knowledge resources over the Internet.

3. ENGINEERING KNOWLEDGE FOR GRID APPLICATIONS - GEODISE

The service-oriented architecture for distributed knowledge management provides a generic framework and guidance for engineering knowledge as well as supporting knowledge reuse for Grid applications. Whether this intended goal can be achieved or not depends very much on the methodology and technologies that the modules of the architecture exploit. While new technologies for knowledge engineering, in particular, in the context of WWW and semantic web, are emerging or under development, traditional methodologies and techniques for knowledge engineering are still effective. The overall approach to putting the architecture into practice involves applying and extending the CommonKADS knowledge engineering framework [7]. In the following we describe in more detail the main components of the architecture, their functions and roles, the technologies each of them uses and the inter-play among components. We shall do this with reference to our exemplar problem of design search and optimisation.

Grid enabled optimisation and design search for engineering (Geodise) is one of a new breed of EPSRC projects funded by the UK e-science initiative [14]. Geodise's objective is to utilise Grid technologies, design optimisation techniques [15], knowledge management technologies, web services and ontology techniques to build a state of the art knowledge-intensive design tool based on the OGSA infrastructure. In Geodise, knowledge engineering focuses on encapsulating and exploiting knowledge so that new designs of, for example, aero-engine components, can be developed more rapidly, and at a lower cost.

3.1 Knowledge acquisition

Knowledge acquisition (KA), located at the lowest level of the architecture, is the starting point of a knowledge management lifecycle [16]. Useful information and patterns can be captured through KA processes. Actually it is an indispensable and most important step for any knowledge engineering practice, without which other knowledge engineering work at high-levels of abstraction cannot be carried out.

With the WWW rapidly evolving into a global knowledge base, new KA techniques such as information extraction (IE) and data mining have been emerging. However, for scientific knowledge such as that involved in design search and optimisation, this knowledge is still largely the province of human domain experts expressed in the medium of natural languages. This means that formal knowledge capture processes are invaluable for editing and structuring domain knowledge.

In the context of the Geodise domain, we utilise a number of knowledge acquisition techniques, namely, interviews, protocol analysis, concept sorting, and other traditional methods. The acquired knowledge is represented using a predefined set of modelling formalisms. Knowledge elicitation and modelling was undertaken with the help of PCPACK [17]. For example, one of the components of the toolkit, the PC-PACK Protocol Editor is used to enumerate key concepts by marking up text-based knowledge source as shown in Figure 2. Another tool from PCPACK, the PC-PACK Laddering Tool, is used to organise these concepts into hierarchies so that relationships can be explicitly expressed and refined as shown in Figure 3. Space does not permit us to detail further KA techniques but we have now captured a rich variety of design search and optimisation domain knowledge, including key concepts, properties and aspects of the design search and optimisation workflow. Even though much more KA is needed to capture the full richness of this task the knowledge captured to date has made other knowledge engineering activities possible.

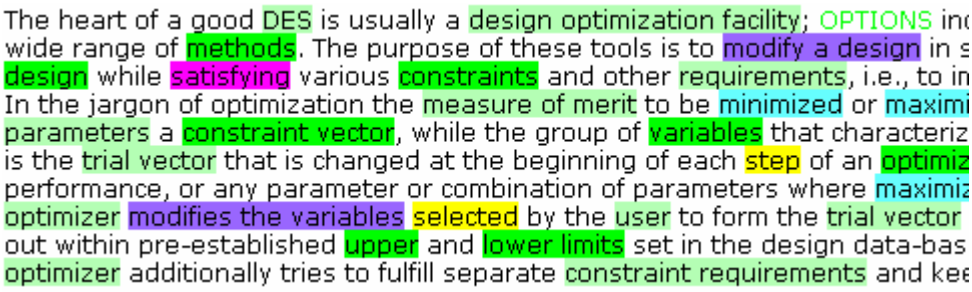

FIG 2: Concept mark-up in the PC-PACK protocol editor

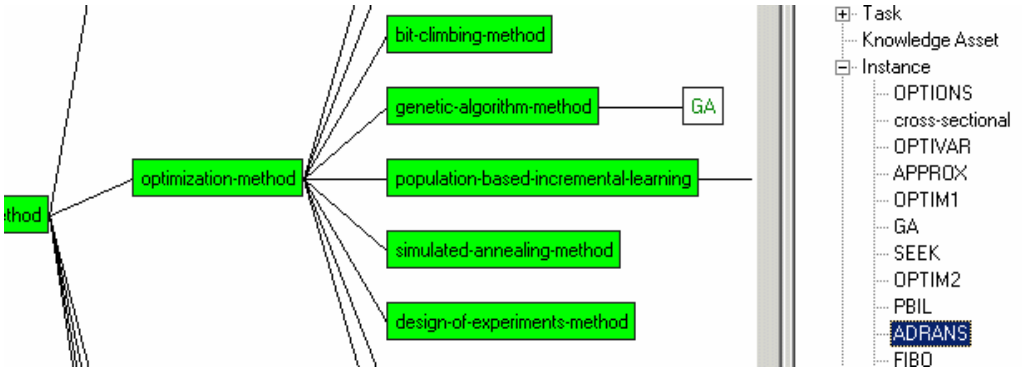

FIG 3: Building concept hierarchies with the PC-PACK laddering tool

3.2 Knowledge modelling

Knowledge modelling aims to provide a structured description of the knowledge infrastructure of an application domain within which knowledge can be most usefully held, and reasoned with. It bridges the gap between knowledge acquisition and knowledge use. Knowledge modelling must be able both to act as a straightforward placeholder for the acquired knowledge, and to represent the knowledge so that it can be used to realise problem-solving objectives.

CommonKADS, one of the most commonly used knowledge engineering methodologies, provides an effective knowledge modelling technique. With it a variety of models are built to capture how knowledge is deployed in a problem-solving context. These models include organizational templates, communication protocols, agent decompositions, task subtask breakdown, detailed domain schema and sets of canonical inferences made in the domain. A guiding principle is that of structure preserving design in which the structure of the knowledge models is preserved as far as possible in the final operational implementations.

Although CommonKADS knowledge models have been successfully used in a number of knowledge engineering and management initiatives, serious problems are encountered when it is applied to the proposed architecture for distributed knowledge management as shown in Figure 1. The fundamental question is how the knowledge templates/structures and the knowledge they contain are represented in an appropriate way. How are the components of knowledge recognised by other users, thus facilitating knowledge reuse and sharing in the Grid environment. To resolve this issue, ontologies are widely exploited in the architecture.

An ontology is an explicit, shared specification of the various conceptualisations in a problem domain. It contains a shared vocabulary used to describe domain concepts and the relationships among them. This knowledge is independent of any representation, i.e., an ontology defines the semantics of terms at the conceptual level. It seeks to abstract information from its syntactic and representational forms, which, for example, can be implemented in XML[2], and database views.

Ontologies play a fundamental role in the architecture outlined in Figure 1. They provide a common medium for inter-agent information transfer, which is applicable to both humans and machines. For example,

ontologies are used as the conceptual framework when building and maintaining a knowledge portal. Ontologies facilitate the retrieval and structuring of information in a comprehensive way. They serve as the conceptual backbone for every task in the knowledge management lifecycle.

In Geodise, we have developed knowledge models for design search and optimisation using the CommonKADS methodology and exploiting ontological engineering techniques. By using the CommonKADS knowledge modelling formalism we have built a Geodise knowledge model with the PCPACK toolkit, which includes domain schemas, tasks, concepts, relations and so on. This knowledge model is represented and published through the knowledge web as discussed in the next section. The ontological engineering of the engineering design search and optimisation (EDSO) ontology has been undertaken using the Protégé [18] and OilEd [19] ontology editors. Figure 4 shows part of the ontology developed in Protégé. The left pane displays the key concepts of an ontology while the right pane is used to edit the attributes and relations of a concept. An advantage of the Protégé editor is that the generated ontologies can manipulate instances, thus facilitating the creation of knowledge bases - knowledge bases can be regarded in part as populated ontological schema. The OilEd ontology editor has a similar graphical interface to the Protégé editor. But it uses the web ontology language, DAML+OIL [20] as its underlying representation language. DAML+OIL supports reasoning, based on a mapping to Description Logic. This can be used to help query the ontology, and also to classify new concepts, assisting with ontology authoring.

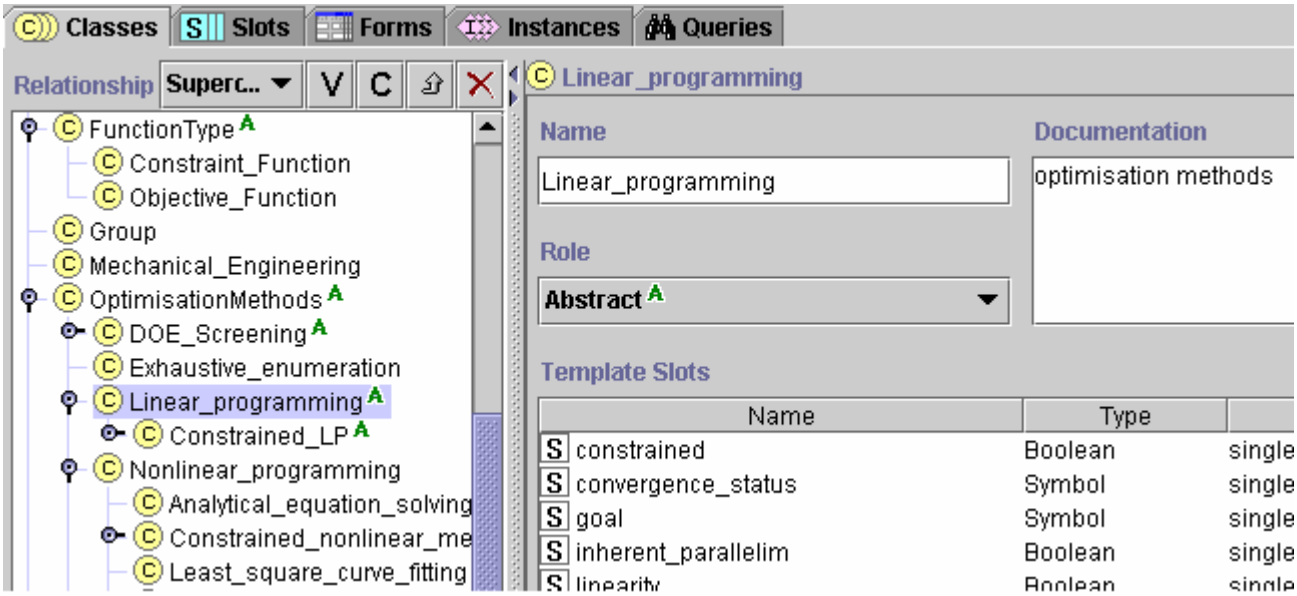

FIG 4: Ontology development using the Protégé ontology editor

The EDSO ontology acts as a shared understanding of the concepts for all of the various grid-enabled components in Geodise. It characterises core optimisation methods, the typical parameter templates of these methods, the tasks and subtasks that these methods contribute to and much more. As the backbone supporting knowledge sharing and re-use within the architecture, the EDSO ontology should and will have many uses. For example, it has been applied to enrich EDSO workflows with semantically meaningful annotations, which will be discussed later.

3.3 Knowledge representation, publishing and storage

Knowledge representation schemes aim to organise content in structured forms that can be accessed and processed by machines or agents. A wide variety of knowledge representation formalisms are available such as production-rules, frame-like structures and varieties of probabilistic representation. It is commonly agreed that the choice of formalisms to use depends on the characteristics of the application domain, the demands of knowledge processing, and the type of user interaction required. For Grid applications, an extra issue needs to be dealt with, that is how to publish and retrieve knowledge through the Web. To this end, we have turned to Web technologies such as HTML and XML to make knowledge available and accessible to any Grid-based

application.

In Geodise the captured knowledge is represented in several formats in order to meet different requirements. One of them is a hypertext format specified in HTML, called the knowledge web (KW), which is illustrated in Figure 5. The advantage of the KW is that it is easy to access and understandable for humans. The KW is organised around the CommonKADS knowledge models. This makes it easier to update and maintain. Another form of knowledge representation adopted is as an XML format as shown in Figure 6, which can be processed by machines and software agents. We also use other traditional methods to represent some types of knowledge. For example, a flow chart is used to represent the design search and optimisation workflow, which will later be represented in machine-understandable formats such as planning operators or a state transition calculus.

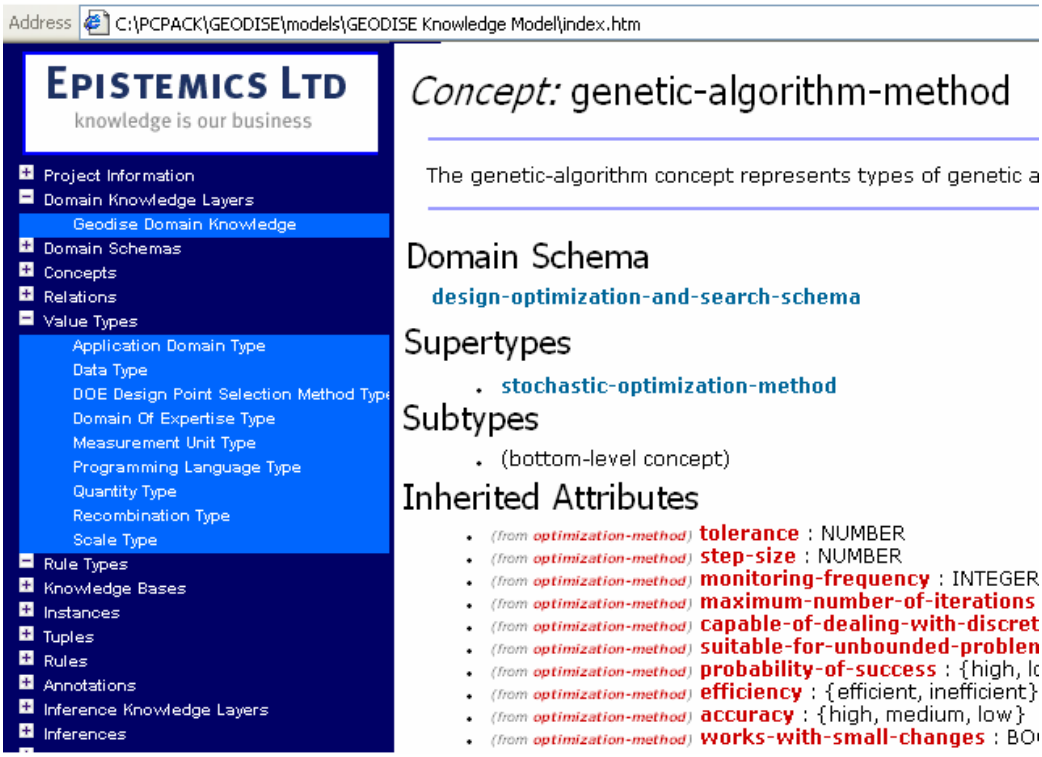

FIG 5: Concept schema in the knowledge web

```
</concept>
- <concept id="concept-85" name="genetic-algorithm-method">
  - <construct-information>
    - <terminology>
        <description>The genetic-algorithm concept represents types of genetic algorithm
          optimization method.</description>
      </terminology>
    </construct-information>
  - <abstraction-relationships>
    - <super-types>
        <super-type id="concept-192">stochastic-optimization-method</super-type>
      </super-types>
    </abstraction-relationships>
  - <attributes>
    - <attribute id="attribute-86" name="nbreed">
      - <type-range>
          <primitive-type data-type="number" />
```

FIG 6: Domain knowledge in XML

In the service-oriented architecture, knowledge that is acquired, modelled and represented is stored in a

knowledge warehouse or repository. The knowledge warehouse differs from traditional, stand-alone knowledge bases. It comprises distributed, federated knowledge repositories that include facts, metadata about documents and ontologies, processes and tasks. They can reside in different locations, run on different platforms, have different representations and may well possess different retrieval and reasoning mechanisms. They are accessed through knowledge service APIs via a knowledge portal as is discussed in the next section.

3.4 Knowledge portal

One key module of the aforementioned architecture is the knowledge portal, which provides views onto domain-specific information on the WWW, thus facilitating the retrieval and exchange of relevant, domain-specific knowledge. As the entrance point to a distributed knowledge management system, the knowledge portal has three basic functions. The first is that of knowledge location, i.e. the registration and management of various knowledge components in various formats in distributed knowledge repositories. The second is knowledge dissemination, which is concerned with how to expose knowledge through the portal and in what ways. The third function is about security issues - i.e. the infrastructure for authentication and authorisation, so that knowledge can be provided, used and updated in a controlled way.

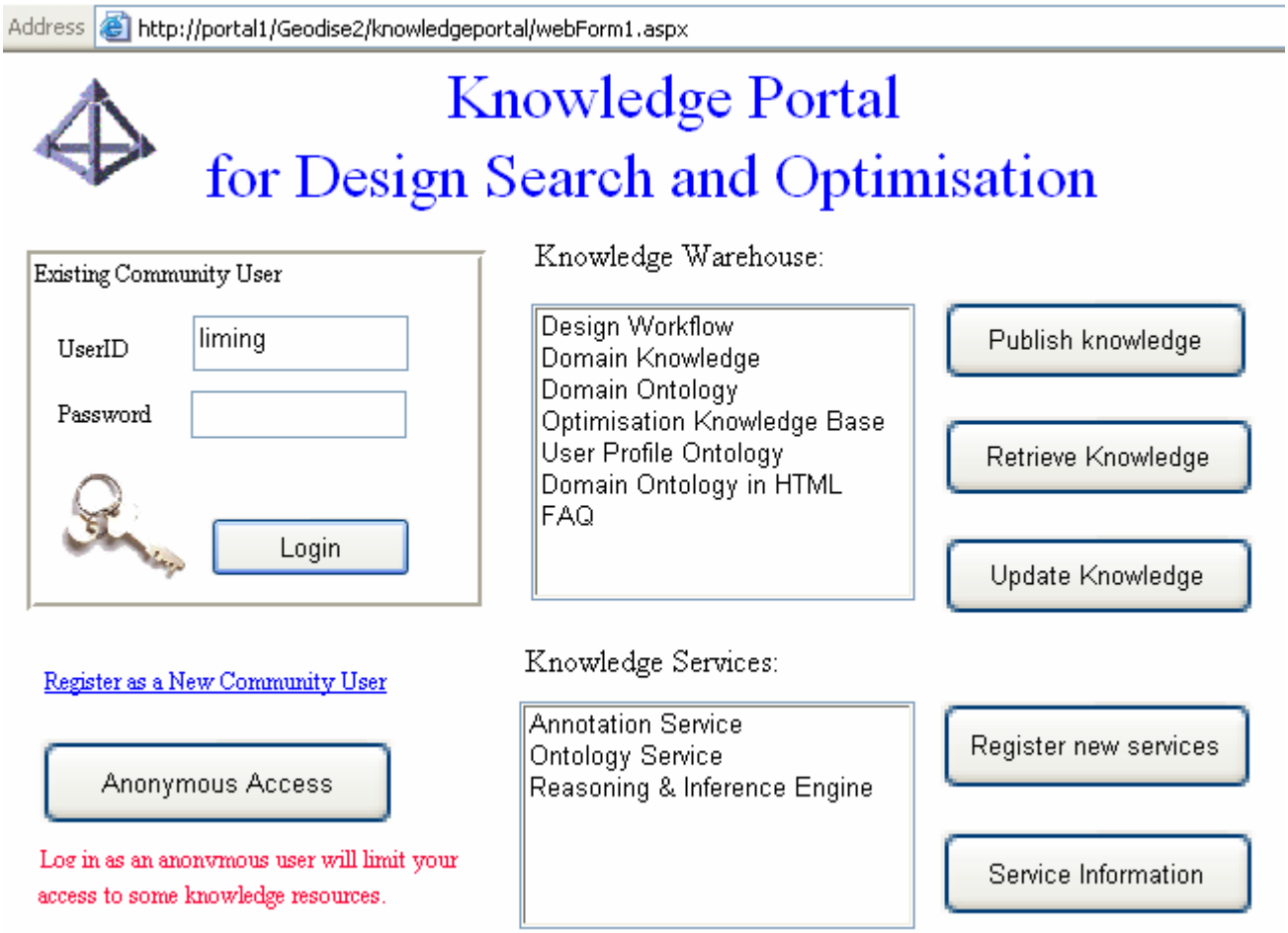

FIG 7: The knowledge portal for Geodise

Figure 7 shows the browsing interface of the knowledge portal we developed for design search and optimisation. On the left of the figure is the registration and security mechanism. In the middle are the knowledge resources that the portal provides. Currently, it includes a set of knowledge repositories and several knowledge services. Information about knowledge resources is stored and maintained in relevant

databases. On the right the buttons are indicative of the functions that have been implemented up to now, which include knowledge publishing, retrieval and update, and service registration and service information provision. Knowledge publishing allows users to register and disseminate new distributed knowledge services. The access and retrieval of knowledge and service information is approached in the same way as we browse the WWW as long as the resources have been registered with the portal.

3.5 Knowledge use and reuse through knowledge services

Thus far we have discussed knowledge acquisition, modelling, publishing and storage within the architecture. However, end knowledge users are principally concerned with how knowledge is accessed and used. Traditionally, knowledge intensive systems are constructed afresh. There is little reuse of existing domain content or problem solving experience. While researchers are investigating approaches to facilitate knowledge reuse, for example, to develop extensive libraries of reusable problem-solving components [21] [22] or to exploit ontologies for knowledge reuse and fusion, a new question arises for knowledge use and reuse in Grid applications, that is how knowledge is shared and reused among virtual organisations over the Internet rather than locally.

To tackle this issue we can derive inspirations from the Grid application paradigm itself, which is aimed at creating virtual computing systems from geographically distributed components in the form of web/Grid services by sharing and using diverse resources in dynamic, distributed virtual organisations. We argue that knowledge can be managed and used by building knowledge services in an integrated knowledge engineering architecture. By adopting the service-oriented paradigm, activities relevant to the supply and consumption of knowledge can be realised through various knowledge services implemented using web services, thus making knowledge accessible and available for any virtual organisation. One extra benefit of such an implementation metaphor is that domain and operational knowledge can be separated into different knowledge services. This means that specific knowledge can be used for different purposes by applying different operational knowledge and one type of operation knowledge can be used to manipulate knowledge from different domains. This provides great flexibility for knowledge reuse and also facilitates knowledge maintenance.

Knowledge services should cover all subtasks of the knowledge engineering management lifecycle. When the architecture is fully implemented, knowledge engineering could be carried out not only using traditional methods but also through web services [23] [24]. For example, knowledge acquisition could be done through a data mining or knowledge discovery service. As ontology plays a core role in knowledge modelling, information sharing, legacy knowledge reuse through annotation and also in semantic web development, we have placed great emphasis on developing ontologies and ontology services.

In Geodise, after domain knowledge is captured and modelled as ontologies, we first develop an ontology service so that both successive knowledge engineering work and other Geodise work can be done with embedded domain semantics. In the following we describe in detail an example knowledge service - the ontology service and illustrate show how knowledge services work in the architecture.

3.5.1 Ontology services

Ontology services aim to improve both the accessibility and easy of use of an ontology. In accordance with the principle of separating domain and operational knowledge, ontology services are built independent of any specific domain. This provides complete access to any DAML+OIL ontology that is available over the Internet.

The ontology services are implemented as a typical SOAP-based web service, which consists of four components: an underlying data model that holds the ontology (the knowledge model) and allows the application to interact with it through a well-defined API, an ontology server that provides access to concepts in an underlying ontology data model and their relationships, the FaCT reasoner [25] that provides reasoning capabilities and a set of user APIs that interface user's applications and the ontology. They have been developed using Java technologies and deployed using Apache Tomcat and Axis technologies.

As a standard web service, ontology services can be accessed using WSDL. As a result an ontology available on the web no matter where it is can be accessed through the ontology services. By means of the service's

APIs together with the help of the FaCT reasoner, common ontological operations, such as subsumption checking, retrieving definitional information, navigating concept hierarchies, and retrieving lexical information, can be performed when required.

Table 1 lists some APIs of the ontology services and their parameters. Optional parameters are marked (*). A call to an API has three possible parameters: service, ontology and service argument. The service argument may be a URI identifying a concept, or an RDF model.

service	Ontology	arg	Explanation
ontologies			Provides a list of ontologies currently held by the server.
load	URL	URL(*)	Loads the given ontology, using the given lexicon if specified.
connect			Connect to the reasoner by using the host and port as specified in the configuration file.
allconcepts			Provides a list of all the concepts in a given ontology.
lookup	URL (*)	String	Looks for a concept corresponding to the given string. If no ontology is specified, will look in all ontologies.
supers	URL	URL/RDF	Returns the supers of the given concept (w.r.t. the given ontology). If the reasoner is connected, the server can deal with arbitrary concept descriptions presented in RDF.
subs	URL	URL/RDF	As supers, but returns subs.
render	URL	URL	Returns a rendering for the concept.
restrictions	URL	URL	Returns restrictions in a concept definition

TABLE 1: Ontology service APIs

Ontology services have been successfully used by other Geodise services. For example, a user profile ontology has been used in Geodise authentication and authorisation mechanisms. Database services have been developed using ontologies and ontology services to generate database schema dynamically and thus ensure that all archived data is semantically enriched using the ontological schemas as metadata. Additional work is being carried out to move both OilEd and the Geodise ontology services to use the emerging standard OWL Web Ontology Language [26], as their representation language.

4. KNOWLEDGE APPLICATION EXAMPLES

While Geodise aims to exploit knowledge in a diversity of areas such as developing an intelligent design system and a design advisor, the first serious use of knowledge is to semantically enrich engineering design workflows through annotation - the latest technique to add semantic content to documents or websites [27], so that workflows can be reused in later designs.

One key question that Geodise should be able to answer is: what previous designs have been explored and how can one re-use them? A typical engineering design usually contains information about the problem definition (the geometry), the tools used for meshing and analysis, the algorithms used for optimisation, what control parameters are chosen and how they are specified as well as the designer's information, time used and the status of the design. All of the above information may be derived from the log files which typical engineering design packages use to record a step by step activity of how the package was used for a given optimisation run. In order to re-use the knowledge contained in these log files we need first to semantically enrich these files using terms from the domain ontology.

Figure 8 shows a screenshot in which a design log file from the OPTIONS design package [15] is being annotated using the OntoMat [28] annotation tool and the ontologies we developed for the Geodise domain. The right pane contains the specific design workflow for annotation. The left panel has several panes containing ontology the hierarchy, instances and attributes. To annotate, first markup a fragment of text in the log file and then create an instance of the corresponding concept from the domain ontology, and finally add attributes to the instance. Thus the selected text can be linked to any information added during the

annotation. Figure 9 is the result after annotation, in which the log file appears as an HTML document with blocks of semantic content in RDF format, which have been added by the annotation process.

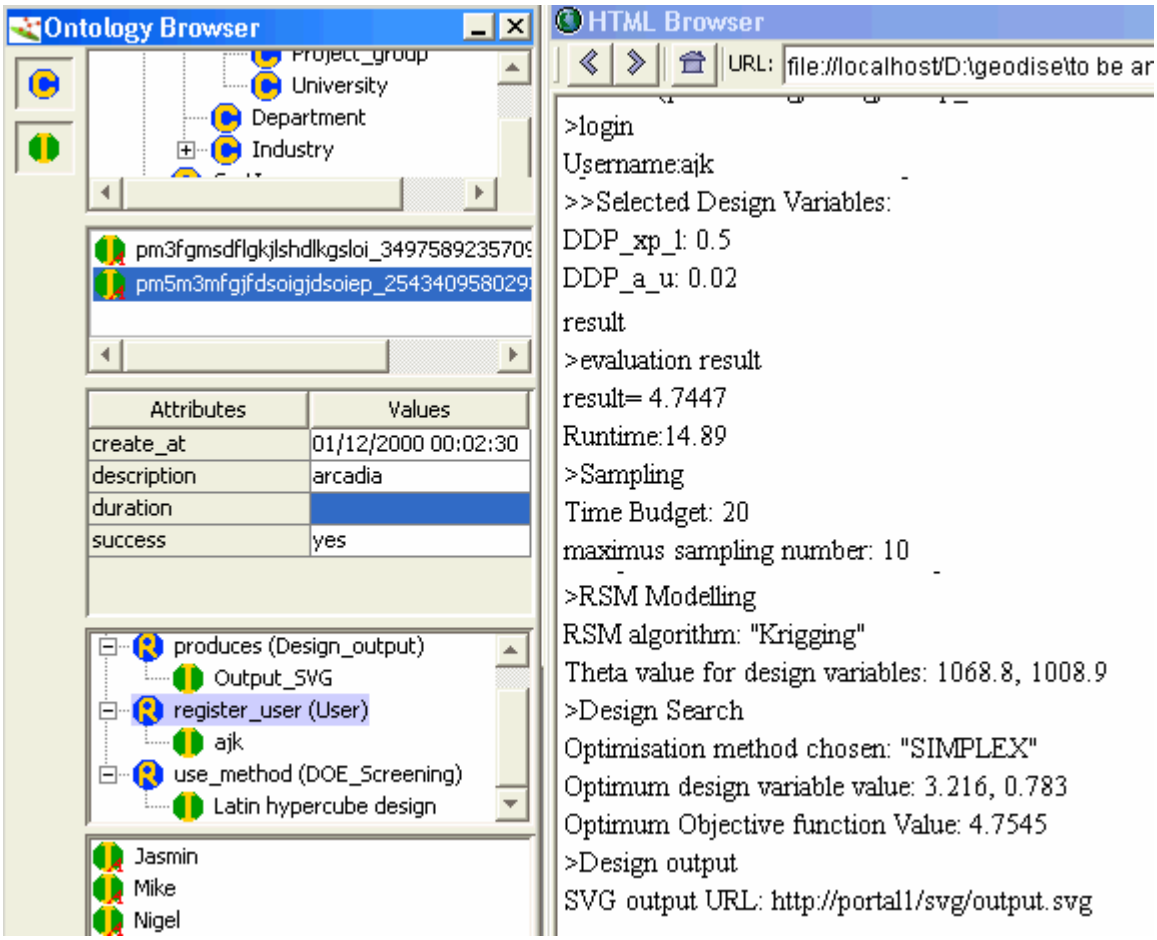

FIG 8: Geodise log file annotation using OntoMat

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

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

FIG 9: Annotated log file with semantic information

The resulting semantically enriched log files can be built into a knowledge repository, which can then be

queried, indexed and reused. This can either guide inexperienced users to carry out design or improve the current design process using methods such as case-based reasoning to find appropriate or suggestive solutions to the current problem based on previous experiences.

We are currently implementing another knowledge intensive system - a knowledge-based ontology-assisted workflow construction assistant (KOWCA). In this system the generic knowledge about design search and optimisation will be converted into a rule-based knowledge base. The task models of the design process will be represented as a task ontology. In the interface of KOWCA, a user can load in the task ontology through the ontology services. Workflow can be constructed by dragging task models from ontological concepts and dropping it into the workflow editor area. Workflow components are configured through ontology instantiation. The underlying knowledge base system will check the consistency of the workflow and/or give advice on what should be done next during the process of workflow construction. It can also run a workflow any time during the construction process to test the intermediate results. It is expected that KOWCA will enable engineers, both novice and experienced, to carry out design in a more controlled and effective way.

5. CONCLUSION AND FUTURE WORK

Grid applications need knowledge support to enhance resource sharing. It is also needed so as to increase the degree of automation available which in turn can facilitate cooperation and collaboration. We have proposed and partially implemented an integrated service-oriented architecture for knowledge engineering over the Internet. The context is design optimisation but it could be applied to many other types of application. We believe that such an approach holds great promise. We are seeking to integrate methods and techniques from Grid Computing, Web Services, and Knowledge Engineering. This is a complex and challenging task.

At the time of writing the Geodise project has been underway for less than a year. We already have prototypes that are capable of running design optimisation in a Grid based environment. We have developed database technologies that can be richly annotated using semantic mark-up languages. We have acquired and modelled a substantial amount of EDSO domain knowledge and are currently capturing content relating to other aspects of the overall design and analysis process. We have presented here a general framework within which these various components will be brought together. There are many topics that we have not touched upon such as knowledge maintenance, the exact nature of knowledge-based decision support and how inference services will be supported under the OGSA architecture. This is work for the future.

ACKNOWLEDGEMENTS

This work was undertaken as part of the EPSRC funded Geodise (GR/R67705/01). The authors gratefully acknowledge the contributions from and discussion with EPSRC projects MyGrid (GR/R67743/01) and AKT (GR/N15764/01(P)).

REFERENCES

[1] Foster, I. and Kesselman, C. 1999. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann.
[2] Chinnici, R., Gudgin, M., Moreau, J. and Weerawarana, S.. (2002) Web Services Description Language (WSDL) 1.2, W3C Working Draft, 9 July 2002. http://www.w3.org/TR/wsdl12/.
[3] Foster, I., Kesselman, C., Nick, J. and Tucke S. 2002. The Physiology of the Grid. An Open Grid Service Architecture for Distributed Systems Integration. http://www.globus.org/ogsa/
[4] Allcock, W., Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. Journal of Network and Computer Applications, 23:187-200, 2001. http://grid.web.cern.ch/grid/
[5] Cox, SJ., Fairman, MJ., Xue, G., Wason, JL. and Keane, AJ. 2001. The GRID: Computational and Data Resource Sharing in Engineering Design Optimisation and Design Search, pp. 207-212 in Proc. Int. Conf. Parallel Processing, ed. T.M. Pinkston, IEEE, ISBN 0-7695-1260-7, Valencia.
[6] Berners-Lee, T., Hendler, J. and Lassila, O. 2001. The Semantic Web, Scientific American, May 2001.
[7] Schreiber S., Akkermans H., Anjewierden A., Hoog R., and Shadbolt N. 1999. Knowledge Engineering and Management. The MIT Press, London.
[8] Advanced Knowledge Technologies (AKT) project http://www.aktors.org/
[9] Buckingham Shum, S., De Roure, D., Eisenstadt, M., Shadbolt, N. and Tate, A. 2002 CoAKTinG:

Collaborative Advanced Knowledge Technologies in the Grid. Proceedings of the Second Workshop on Advanced Collaborative Environments, Eleventh IEEE Int. Symposium on High Performance Distributed Computing (HPDC-11), Edinburgh, Scotland. http://www.aiai.ed.ac.uk/project/akt/coakting/
[10] Uschold, M. and Gruninger, M. 1996. Ontologies: Principles, Methods and Applications. Knowledge Engineering Review 11(2).
[11] Geodise project http://www.geodise.org/
[12] MyGrid project http://mygrid.man.ac.uk/index.shtml
[13] Roure, D., Jennings, N. and Shadbolt, N. 2001. Research Agenda for the Future Semantic Grid: A Future e-Science Infrastructure http://www.semanticgrid.org/v1.9/semgrid.pdf
[14] The UK Research Councils e-Science Core Programme http://www.research-councils.ac.uk/escience/
[15] Keane, AJ. OPTIONS Design Exploration System http://www.soton.ac.uk/~ajk/options/welcome.html
[16] Shadbolt, N.R, O'Hara, K. and G. Schreiber (1996) Advances in Knowledge Acquisition. Springer-Verlag.
[17] PC PACK Knowledge tool http://www.epistemics.co.uk/products/pcpack/
[18] Protégé ontology and knowledge editor http://protege.stanford.edu/index.html
[19] Bechhofer, S., Horrocks, I., Goble, C. and Stevens, R. 2001. OilEd: a Reason-able Ontology Editor for the Semantic Web DL2001, 14th International Workshop on Description Logics, Stanford, USA. http://oiled.man.ac.uk/.
[20] DAML+OIL http://www.daml.org.
[21] Motta, E. (1999) Reusable Components for Knowledge Modelling, IOS Press.
[22] Breuker, J. and van de Velde, W. (1994) CommonKADS Library for Expertise Modelling Reusable Problem Solving Components. IOS Press, Amsterdam.
[23] Crow, L. and Shadbolt, N. R. (2001) Extracting Focused Knowledge from the Semantic Web, International Journal of Human Computer Studies, Vol. 54 (1) pp 155-184.
[24] Sheila A. McIlraith, Tran Cao Son, Honglei Zeng (2001) Semantic Web Services IEEE Intelligent Systems March/April 2001 (Vol. 16, No. 2).
[25] Horrocks, I., Sattler, U. and Tobies, S. (1999). Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99), no.1705 in Lecture Notes in Artificial Intelligence, pp.161-180. Springer-Verlag.
[26] OWL Web Ontology Language 1.0 Reference, http://www.w3.org/TR/owl-ref/
[27] Bechhofer, S. and Goble, C. 2001. Towards Annotation using DAML+OIL. K-CAP 2001 workshop on Knowledge Markup and Semantic Annotation, Victoria B.C.
[28] OntMat-Annotizer Annotation Tool http://km.aifb.uni-karlsruhe.de/annotation/ontomat.html