An Approach to Structured Knowledge Representation of Service-oriented Grids

Pin Hu, Lingfen Sun and Emmanuel Ifeachor
School of Computing, Communications and Electronics,
University of Plymouth, Plymouth, PL4 8AA, UK
{pin.hu; L.Sun; E.Ifeachor}@plymouth.ac.uk

Abstract

Service-oriented grid is a type of grid environment which makes extensive use of existing and emerging standards from all segments of the Web service community. Ontology, as a knowledge representation tool, can normally be used to semantically represent such type of grid. However, knowledge representation of a grid environment is not an easy task since the concept of grid covers many computing and communication areas. In this paper, we present a new approach to use and organise descriptions of grid entities to represent service-oriented grid environments. The main objectives are 1) to describe an extended service-oriented grid model; 2) to propose a schematic multiplex description architecture for managing individual descriptions to represent complex and various knowledge of grid; and, 3) to demonstrate the implementation of a kernel ontology for high-level representation of service-oriented grids.

1. Introduction

Grid as a new distributed computing paradigm is expected to provide high speed, high quality, and seamless resource sharing environments for computation, data and knowledge intensive scientific applications. In the development of grid, an important field known as the semantic grid, deals with an extension of the current grid where information and services are given well-defined meaning through the use of machine-processable descriptions to maximize the potential for sharing and reuse [1].

In semantic grid, the use of ontology is common [2 & 3]. Conceptually, ontology is a concept developed in the area of artificial intelligence and is used to capture knowledge in a domain. A grid-related ontology normally defines the concepts of grid entities (e.g. resources, services, virtual organisations, applications and users) and the relationships between them. A complete grid ontology, which covers detailed knowledge of all grid sub-domains, is not practical in most cases (e.g. it is difficult for maintenance and extension) since the concept of grid can cover dozens of computing and communication areas.

Until now, in grid-related ontology, there is still no clear strategy for organising ontologies to semantically represent an entire grid environment, which we believe is significant in knowledge-based global coordination between different grid components (e.g. resources, applications, security, quality of service and users). Most efforts so far have focused on specific sub-domains of grid (e.g. ontologies related to “grid resource” [4, 5 & 8] and ontologies related to “grid service” [6, 17 & 18]). They are useful in the development of grid ontologies, but none can be directly used to represent the general concept of grid. Xing et al [7] claimed the proposal of the first general grid ontology, named Core Grid Ontology (CGO), in 2006. The CGO describes certain basic grid concepts and their relationships. However, its design is static. It defines property restrictions to classes into hard-coded manner (e.g. the use of the properties “descriptor” and “status” with value constraint “owl:allValuesFrom” and cardinality constraint “owl:cardinality” used to describe the class “GridService”). This may result in attribute overlaps when integration with grid sub-domain ontologies is required. As the changing nature of grid and semantic technologies can lead to frequent technical changes in the grid ontologies, the static nature of the CGO may cause considerable difficulty in terms of interoperability between the CGO and other grid-related ontologies.

Moreover, to present a grid, we may also face a heterogeneous description environment. That is to say, the description of an instance of a defined grid entity can be based on various techniques and/or languages (e.g. pure XML, RDF and OWL). How to manage this kind of description environment still remains difficulty (e.g. how can we integrate descriptions of grid entities with different description language and/or format together; how can we develop schemes to let the whole description system be easily extended?).

The aim of this paper is to present a new approach in using and organising descriptions of grid entities to represent service-oriented grid environments. The main objectives of this paper are: 1) to describe an extended service-oriented grid model. The model is designed based on the convergence of Service Oriented Architecture (SOA) and the general concept of grid. It extends the model introduced by the well-known OGSA architecture [9] and focuses on both provisioning and the use of services; 2) to propose a schematic multiplex description architecture, which offers a new approach in dealing with how to organise descriptions to represent complex and variable knowledge of grids; and, 3) to demonstrate the implementation of a kernel ontology for service-oriented grids. The ontology aims to represent the high-level structure of generic service-oriented grid environment, and the relationships between defined classes based on the proposed service-oriented grid model. It is expected to provide a high-level knowledge representation of service-oriented grids, better interoperability with other grid-related descriptions and a basis for the development of scientific grids (e.g. BIOPATTERN grid [10 & 16]).

The remainder of the paper is organised as follows. Section 2 describes the extended service-oriented grid model. Section 3 proposes a schematic multiplex description architecture for knowledge representation of grids. Section 4 demonstrates the implementation of a kernel ontology for service-oriented grids. Finally, Section 5 concludes the paper.

2. A Service-Oriented Grid Model

Grid is a vague term, which may be simply described as resource sharing across different administrative domains. The term service-oriented grid we quote here normally means a type of grid environment, which makes extensive use of existing and emerging standards from all segments of the Web services community [11]. In a service-oriented grid, service plays a key role in sharing of distributed resources. In order to capture an abstract view of service-oriented grid, we consider it from two points of view – the provisioning and the use of services. Figure 1 presents an overview of the proposed service-oriented grid model.

From the provisioning of services point of view, services are interfaces to or represents of virtualised resources (e.g. data, computation, information and knowledge resources). The virtualisations are carried out by grid middleware in a tightly-coupled manner with underlain physical resources (e.g. databases, knowledge-bases and high performance or throughput clusters). Services are held by service containers, which can be regarded as a kind of nodes in a grid environment. Another major type of grid nodes is grid access point, which can provide interfaces for Virtual Organisation (VO) users or administrator to use or control VOs. A group of grid nodes along with the connections between them can then compose of a VO for sharing resources within a specific community (Note that a grid node might be held by multiple VOs). In order to provide security functionalities (e.g. access control) and deliver user required Quality of Service (QoS), security and QoS mechanisms and policies can be built into the abovementioned grid entities and/or treated as resources to be accessed from services.

From the use of services point of view, a service-oriented grid is administrated by certain types of VO administrators, and used by different kinds of VO users, who might be a user or a developer of an application. The use of a grid is based on the pre-designed or automatically configured grid applications and via diverse grid jobs to solve specific problems. When security or QoS functionalities provided by a VO do not reach the corresponding requirements of a grid application (e.g. a VO cannot guarantee a specific level of QoS required by a grid application), additional security or QoS mechanisms will need to be deployed in the VO.

Note that the interactions between some grid entities described above may have various

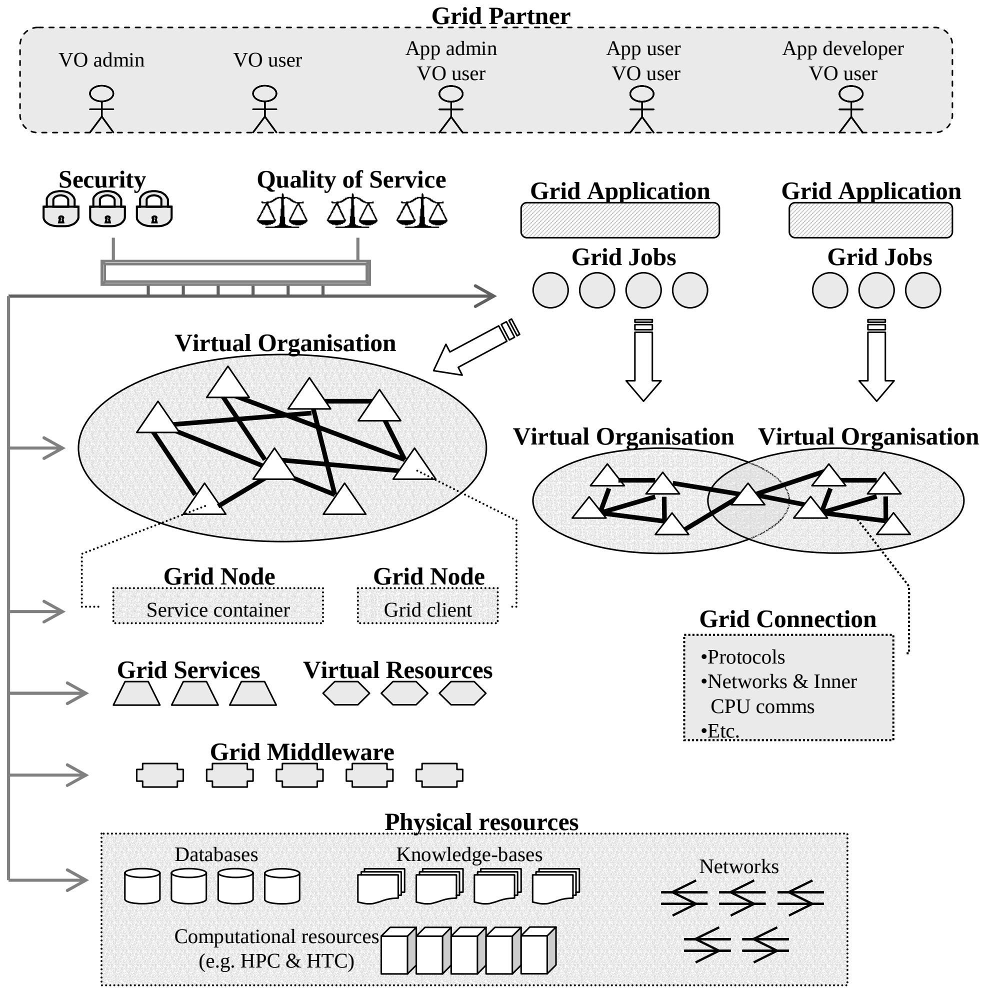

Figure 1. Overview of the Extended Service-Oriented Grid Model

behaviours. For example, a VO may not have any QoS policies and implementations; and in some cases, a virtual resource can be treated as a part of a grid service. Moreover, in the model, services are physically unlayered, but logical layered. That is to say, a group of services can be held by the same type of service containers, but some services may be logically built on top of other services and interact between them. It is also worth to mention that the grid entities presented in the model can be viewed as different sorts of resources. We classify them into different types of grid entities only for the purpose of the ease of understanding and organisation of grid.

3. A Schematic Multiplex Description Architecture for Knowledge Representation of Grids

The knowledge representation of an entire grid environment is not an easy task since the concept of grid covers many computing and communication areas, such as virtualisation, data storage and management, distributed computation, networking, security and QoS. Further, to present such an environment, we may also face a heterogeneous description environment, since the description of an instance of a defined grid entity can be based on various techniques and/or languages. For example, a grid service can be described using only WSDL [22], or with the support of OWL-S [12]; a simple security management scheme may be described using pure XML, and a more complex one may be described using OWL.

In this paper, we propose a Schematic Multiplex Description Architecture (SMDA), which is targeted to overcome the aforementioned problems, and eventually to support knowledge representation of any grid environments. The term “multiplex description” here we mean two aspects of practice. One aspect is the separation of concerns. That is to say, the description of the concept of an entire grid environment should be based on separated descriptions of grid sub-domains with distinct features, and their interoperation only relies on additional integration schema/ontologies. Therefore, the knowledge of each defined grid sub-domain can be managed independently. Also, the change of the knowledge in a sub-domain will not lead a series of changes in other sub-domains.

The other aspect of practice is the use of diverse description techniques and/or languages (e.g. pure XML, RDF and OWL) for grid knowledge representation. This means that the SMDA should be flexible enough to work with different formats and types of description of defined grid entities. The integration between them should ideally be loosely-coupled in order to avoid huge amount of work on attribute and element mapping and description transformation.

Figure 2 presents a brief overview of the proposed SMDA for organising description blocks to semantically represent grid environments. The architecture consists of certain categories of descriptions, where:

Resource Descriptions represent (semantic) descriptions of physical and/or virtual resources, which include the descriptions of resource capabilities, locations, access interfaces, etc. This sort of description normally integrates with security and QoS descriptions in order to provide low-level security and QoS functionalities, such as database access control and cluster monitoring.

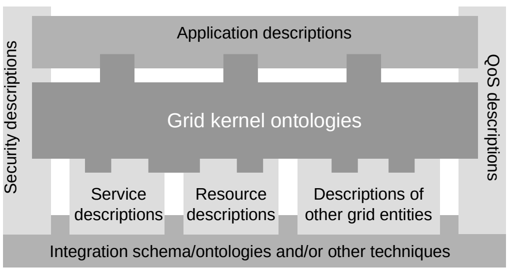

Figure 2. Schematic Multi-ontology architecture for knowledge representation of grids

Service Descriptions describe services in grids. A service can be either tightly-coupled or loosely coupled to other services. For those loosely-coupled services, from OWL-S [12] point of view, the overall ontology structure for service descriptions should contain three major parts, as 1) the service profile for advertising and discovering services; 2) the process model, which gives a detailed description of a service's operation; and 3) the grounding, which provides details on how to interoperate with a service, via messages [12].

Security Descriptions facilitate security depictions in provisioning of common grid security vocabulary, expression of grid security concepts and relationships between such defined concepts. A well organised group of security descriptions must link the top level of applications' and users' security requests with the bottom level of detailed descriptions of security actions taken in grids.

QoS Descriptions mainly cover those knowledge representation areas, as descriptions of user and application QoS requests and semantics of monitoring and management of grid jobs, services and resources, which include ontologies for resource and service matching, job scheduling, etc. Most QoS descriptions work closely with descriptions of other grid entities, such as resource, service and application.

Application Descriptions (semantically) describe any computation-related applications. The description of an application may include the roles of the application users and administrators, workflows of the application, what the security and QoS requirements of the application are, etc. The ontology representation of a data model can be an example of this type of description.

Grid Kernel Ontologies are the heart of the whole architecture. This kind of ontology links different types of grid sub-domain ontologies together, and defines high-level relationships between them. For example, a grid kernel ontology may bridge specified application and service descriptions to help users and/or systems to find out relevant services in grids based on required functionalities offered by applications (e.g. analysis of a medical images or retrieve a patient's historical information). Moreover, the relationships between the grid kernel ontologies and the above-described types of descriptions should, in our opinion, be loosely-coupled. That is to say, in a grid kernel ontology, we may only define the description instance and related format of a grid entity (e.g. service, resource, QoS and security) instead of describing the grid entity in detail to maximally reduce the work of interoperation between the grid kernel ontology and other grid sub-domain descriptions (e.g. mapping of different terms and semantically bridge different concepts).

Integration schema/ontologies provide (semantic) additional integration functionalities between descriptions of different grid sub-domain when a single grid kernel ontology cannot deliver.

Other grid-related descriptions may include 1) descriptions for mapping VO offers with application requirements in sub-domains like security, QoS, service, partners, etc.; 2) descriptions for grid job, which may define semantics for job description format, relationships between grid job and application workflow and between grid job and service, etc.

4. Implementation of a Kernel Ontology for Service-Oriented Grids

In order to prove the concept of the proposed multiplex description architecture for knowledge representation of entire grid environment, we first demonstrate an implementation of a key ontology – kernel ontology for service-oriented grids based on the proposed service-oriented grid model, as described in Section 2.

The aim of the grid kernel ontology is to represent at a high level the semantics for the generic architecture of service-oriented grids (including defining classes of main grid entities, relationships between them and constrains of the classes, etc.) and to link grid sub-domain descriptions to support the knowledge-based coordination between grid components. One important feature of the kernel ontology is that it uses a loosely-coupled way in the integration with grid sub-domain ontologies, which is expected to offer componentised development of ontologies in different grid domains and maximally reduce the ontology interoperation work in adapting the rapid development of grid technologies.

In the kernel ontology development, Web Ontology Language (OWL) [13], as a well known W3C recommendation, is used for the implementation. The selection of OWL makes more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and the ontology easier to extend. Moreover, in the implementation, Protégé [19] is used as the ontology editor. OWLViz [20] is employed for ontology visulisation. RacerPro [21] is used as

Figure 3. Class hierarchy of the kernel ontology for service-oriented grids

the inference system. The latest version of our implemented grid kernel ontology is downloadable at [15].

4.1 Class Definition and Hierarchy

In the proposed service-oriented grid model, we described a set of grid entities. The two top-layer entities are VO and application. Others are the associated components, which can be constructed to support them. For the design of the kernel ontology, we map all the described grid entities into main abstract classes and their relationships into properties and constraints. Table 1 presents the main (abstract) classes and their descriptions defined in the kernel ontology. Some underneath sub-classes are shown in Figure 3.

As it can be seen in the design, we use subclasses: “Named***”, “***Description” and “DescriptionFormat” (under the class “Support”) instead of directly defining grid sub-domain specific properties and/or constraints to main grid entities to avoid attribute overlapping with grid sub-domain ontologies. For example, the class “Service” contains two subclasses as NamedGridService, and GridServiceDescription. Each named grid service can be described by one or more service description(s), and each service description is accompanied with a kind of description format. Thus, the functions offered by the kernel ontology and service descriptions can be distinguished. The kernel ontology is then able to work with various types and formats of service descriptions. The technical change of service descriptions will thus not lead to intensive mapping between them. In other words, a user or a system will be able to find a required service in the kernel ontology, but obtain the detailed descriptions of the various implementations of the service from service ontologies or other knowledge representation techniques based on the instances of GridServiceDescription and DescriptionFormat. When technical changes of service descriptions happen, we only need to add or modify the instances of the above two subclasses in the grid kernel ontology. This avoids huge amount of work on attribute and element mapping and description transformation.

Moreover, in order to support the integration with application descriptions (e.g. bioinformatics data model), we defined a main class: GridApplication and two major (sub)classes: Workflow (under the class

Class	Description
PhysicalResource	Any computing-related resources, such as a PC, a HPC, a binary file, a data storage and communication networks.
GridMiddleware	Software stacks designed to virtualise and provide access to physical resources.
VirtualResource	Logical representation of a/multiple physical resource(s).
Security	The control of risks related to the access of a VO, a grid service, data, etc.
QoS	Quality measure and control in order to provide different priority to different grid users, or guarantee a certain level of performance to grid applications.
GridService	Software component, which provides platform-independent protocols and standards used for exchanging data between applications. It can be either stateful or stateless, as compared with a typical Web service.
GridJob	A paticular unit of work to solve a/multiple problem(s) defined by a grid application.
GridNode	Nodes in a grid environment, including service containers, grid clients, etc.
GridConnection	Logical connections between grid nodes.
Partner	Specified role in the access to resources and/or services within a VO or a grid application.
GridVO	A type of administrative domain for sharing resources across different institutions and/or individuals in order to achieve a specific goal.
GridApplication	A collection of work items that can carry out complex computing tasks by using grid services and resources.
Support	Additional entities which provide global support for descriptions of grid components.

Table 1. Defined main (abstract) classes of the kernel ontology

GridJob) and GridApplicationPartner (under the class Partner). The subclass Workflow can be regarded as the class for process descriptions detailed in the application descriptions. A Workflow description instance can normally be translated into a grid job description instance, which can then be used to realise the application-specific process in grids. The subclass GridApplicationPartner is for the definitions of the individual roles of using and managing applications. For instance, for bioprofiling [10], we may have certain categories of users and administrators, such as patient, researcher, administrator and clinician.

The proposed grid kernel ontology also contains two important (sub)classes: QoSPolicy and SecurityPolicy, which are used to hold all "policy" instances for other grid entities to invoke. For example, a NamedVO instance may have a NamedSecurityPolicy instance for access control, and a NamedGridApplication instance may require a NamedQoSPolicy instance for the statement of its QoS requirements.

4.2 Properties and Constraints

In order to represent relationships between and restrictions to the determined classes, we have also defined a set of properties and constraints.

In the design, "hasDescription" and "hasDescriptionFormat" are the two types of property which describe the relationships between class "Named***" and "***Description", and between class "***Description" and "DescriptionFormat", respectively. The value constraint "owl:allValuesFrom" for property "has***Description" links the restriction class such as NamedGridService and GridServiceDescription, meaning a NamedGridService instance can only have description instance(s) from the class GridServiceDescription. The cardinality constraint "owl:cardinality" for property "hasDescriptionFormat" indicates each "***Description" instance only has one description format.

Other properties and constraints are defined mainly based on the relationships between the grid entities described in the service-oriented grid model. For example, the relationships between named grid application and other grid entities can be expressed as:

– *hasGridApplicationPartner: NamedGridApplicationPartner,*
– *hasWorkflow: NamedWorkflow,*
– *requireGridService: NamedGridService,*
– *hasQoSPolicy: NamedQoSPolicy,*
– *hasSecurityPolicy: NamedSecurityPolicy;*

and between named VO and other grid entities as:

– *hasGridService: NamedGridService,*
– *hasGridConnection: NamedGridConnection,*
– *hasVirtualResource: NamedGridResource,*
– *hasGridNode: NamedGridNode,*
– *hasGridMiddleware: NamedGridMiddleware,*
– *hasQoSPolicy: NamedQoSPolicy,*
– *hasSecurityPolicy: NamedSecurityPolicy;*

We note that most properties defined in the kernel ontology are global to classes. This enables them be easily reused and extended in order to build various relationships between the defined and other additional grid entities.

5. Conclusion and Future work

In this paper, we presented our work in developing ontologies for structured knowledge representation of service-oriented grids. The main contributions are the extended service-oriented grid model, the schematic multi-ontology architecture and the implementation of a kernel ontology for service-oriented grids.

In the future, we will further improve the kernel ontology for service-oriented grids. We will also investigate into the integration of the kernel ontology, existing grid sub-domain ontologies and the bioprofile data model [14] to semantically support tracking, secure access, integration and dynamic analysis of bioprofile data over the BIOPATTERN grid [10 & 16].

Acknowledgement

We acknowledge the financial support of the European Commission (The BIOPATTERN Project, Contract No. 508803) for part of this work.

Reference

[1] D.D. Roure, N.R. Jennings and N.R. Shadbolt, "The Semantic Grid: Past, Present, and Future" Proceedings of the IEEE Volume 93, Issue 3, March 2005

Page(s): 669 – 681.

[2] T. Berners-Lee, K. Hendler, O. Lassila, "The Semantic Web", Scientific American, pp 35-43, Issue 284 (3),17th May 2001.

[3] C. Goble, and D.D. Roure, "The Grid: An Application of the Semantic Web", *ACM SIGMOD Record* 31(4) pp. 65-70, 2002.

[4] J. Brooke, K. Garwood, and C. Goble. "Semantic Matching of Grid Resource Descriptions", In M. D. Dikaiakos, editor, Proceedings of Second European Across Grids Conference (AXGrids 2004), LNCS 3165, pages 240–249, Nicosia,Cyprus, 2004.

[5] P. Alper, O. Corcho, I. Kotsiopoulos, P. Missier, S. Bechhofer, D. Kuo, and C. Goble, "S-OGSA as a Reference Architecture for OntoGrid and for the Semantic Grid", The 3rd GGF Semantic Grid Workshop, GGF16, 2006.

[6] S. Miles, J. Papay, V. Dialani, M. Luck, K. Decker, T. Payne, and L. Moreau, "Personalised Grid Service Discovery", IEE Proceedings Software: Special Issue on Performance Engineering, 150(4):252–256, 2003.

[7] W. Xing, M. D. Dikaiakos, and R. Sakellariou, "A Core Grid Ontology for the Semantic Grid", ccgrid, pp. 178-184, Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), 2006

[8] L. Pouchard, L. Cinquini and G. Strand, "The Earth System Grid Discovery and Semantic Web Technologies", ISWCII, Sanibel Island, FL, October 20, 2003.

[9] D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Siebenlist, R. Subramaniam, J. Treadwell and J.V. Reich, "The Open Grid Services Architecture, Version 1.0", GFD-I.030, OGSA WG, January 2005.

[10] L. Sun, P. Hu, C. Goh, B. Hamadicharef, E. Ifeachor, I. Barbounakis, M. Zervakis, N. Nurminen, A. Varri, R. Fontanelli, S. Di Bona, D. Guerri, S. La Manna, K. Cerbioni, E. Palanca, and A. Starita, "Bioprofiling over Grid for eHealthcare", HealthGRID 2006, Valencia, Spain, June 7th-9th, 2006.

[11] L. Srinivasan and J. Treadwell, "An Overview of Service-oriented Architecture, Web Services and Grid Computing", HP Software Global Business Unit, November 3, 2005.

[12] The OWL Services Coalition, D. Martin (ed.), "OWL-S: Semantic Markup for Web Services", Technical Report, 27 December 2003, http://www.daml.org/services/owl-s/1.0/owl-s.html

[13] D.L. McGuinness and F.V. Harmelen, Editors, W3C Recommendation, 10 February 2004. http://www.w3.org/TR/owl-features/

[14] E. Ifeachor, A. Anastasiou, C. Goh, N. Outram, M. Zervakis, S.V. Huffel, D. Timmerman, P. Lisboa, A. Taktak, A. Starita, K. Cerbiono, F. Ferreira and G. Balls, "A Data Model to Support Bioprofiling", White Paper, BIOPATTERN Project, Jan, 2007

[15] Kernel Ontology for Service-Oriented Grids, https://141.163.121.185:8080/download/kernelOntologyForServiceOrientedGrids.owl

[16] P. Hu, L. Sun, C. Goh, B. Hamadicharef, E. Ifeachor, I. Barbounakis, M. Zervakis, N. Nurminen, A. Varri, R. Fontanelli, S. Di Bona, D. Guerri, S. La Manna, K. Cerbioni, E. Palanca and A. Starita, "The BIOPATTERN Grid – Implementation and Applications", 5th All-Hands Meeting, Nottingham, UK, September 18-21, 2006.

[17] M. Babik, E. Gatial, O. Habala, L. Hluchy, M. Laclavik, and M. Maliska, "Semantic Grid Services in K-Wf Grid", Second International Conference on Semantics, Knowledge and Grid, GuiLin, GuangXi, China, Nov, 2006.

[18] C. Zhu, Z. Liu, W.M. Zhang, W.D. Xiao and J.C. Huang, "An Efficient Decentralized Grid Service Discovery Approach based on Service Ontology", Web Intelligence 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on Volume , Issue , 20-24 Sept. 2004 Page(s): 570 – 573

[19] The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/

[20] OWLViz: A Visualisation Plugin for the Protégé OWL Plugin, http://www.co-ode.org/downloads/owlviz/co-ode-index.php

[21] RacerPro: An OWL Reasoner and Inference Server for the Semantic Web, http://www.racer-systems.com/

[22] E. Christensen, F. Curbera, G. Meredith and S. Weerawarana, "Web Service Description Language (WSDL) 1.1", W3C Note, 15 March 2001, http://www.w3.org/TR/wsdl