Towards a Grid Information Knowledge Base*

Wei Xing[1], Marios D. Dikaiakos[1], and Rizos Sakellariou[2]

[1] Department of Computer Science, University of Cyprus, Cyprus
[2] School of Computer Science, University of Manchester, UK
xing@ucy.ac.cy, mdd@ucy.ac.cy, rizos@cs.man.ac.uk

Abstract. In this paper, we present our work on building a Grid information knowledge base, which is a key component of a semantic Grid information system. A Core Grid Ontology (CGO) is developed for building a Grid knowledge base; and the SPARQL query language is adopted to query the knowledge base.

1 Introduction

Information Services are regarded as a vital component of the Grid infrastructure. They address the problem of the discovery and ongoing monitoring of the existence and characteristics of resources, services, computations and other entities of value to the Grid [1]. As Grids grow larger and gain widespread use, there is an increasing need for Grid information systems to support complex queries, such as:

1. Is there a VO providing exclusive access to a shared-memory multiprocessor system with at least 16 processor, 8 GB of main memory, and a usage charge of not more than 100 euros per CPU time?
2. Find services running Quantum Chromo-Dynamics calculations (QCD) using F90 and MPI?
3. Locate Grid-sites that offer access to a LAPACK software library installed on a shared-memory multiprocessor with 16 to 64 processors?
4. Find the pricing and prior clientele of Grid service that provide access to the XYZ workflow for real-time oil refinery simulations?

However currently no information system can answer the above queries. The main problem of the existing information systems (e.g. MDS, RGMA, BDII) is that they are actually designed and developed for providing particular information to specific Grid sub-systems [2]. For instance, MDS of Globus is designed and developed to support resource discovery; BDII of LCG is used mainly for job scheduling. This makes them not adequate to provide information about whole aspects of Grid systems, such as Grid entities, capability of Grids, Grid resources, Grid middleware, Grid services, Grid applications, and Grid users.

* Work supported in part by the European Commission under the CoreGrid project.

W. Lehner et al. (Eds.): Euro-Par 2006 Workshops, LNCS 4375, pp. 186–190, 2007.

In brief, the main limitations of those systems are: (1) limitations on absorbing heterogeneous information sources; (2) inadequate in supporting information management, retrieval, and sharing in a large-scale multi-Grid systems.

To tackle these issues, we propose a semantic approach, which builds Grid information knowledge bases for Gris information services. The knowledge bases contain semantic metadata of Grid entities, resources, middleware, services, applications, and users. In this paper, we present the work of building a Grid information knowledge base. By using a Core Grid Ontology, we propose a ontology-driven method to create a Grid knowledge base [3]. We also adopt SPARQL query language to support the complex query to the Grid knowledge base [4].

The remaining of this paper is organized as follows. In Section 2, we describe the Core Grid Ontology. In Section 3, we introduce our work of building a Grid information knowledge base. Then, we describe how to query the knowledge base using SPARQL in Section 4. Finally, we conclude our work in Section 5.

2 A Core Grid Ontology Framework

The main issue is how to build, update and manage the Grid knowledge base. A Grid knowledge base may be built based on Grid ontologies, which define fundamental Grid-specific concepts, and the relationships between them. Hence, a Grid ontology is needed in order to build a Grid knowledge base. The main problem for building an ontology for Grids is that there is currently a multitude of proposed Grid architectures and Grid implementations, which are comprised Grid entities, services, components, and applications. It is thus very difficult, if at all feasible, to develop a complete Grid ontology that will include all aspects of Grids. Furthermore, different Grid sub-domains, such as Grid resource discovery and Grid job scheduling, normally have different views of, or interests about a Grid entity and its properties. This makes the definition of Grid entities and the relationships between them very hard. To tackle these issues, we propose a Core Grid Ontology (CGO) that defines fundamental Grid-specific concepts, and relationships. One of our main goals was to make this Core Grid Ontology general enough and easily extensible to be used by different Grid architectures or Grid middleware, so that the CGO can provide a common basis for representing Grid knowledge about Grid systems, including resources, middleware, services, applications.

A Core Grid Ontology (CGO) is proposed to define fundamental Grid-specific concepts, and the relationships between them. One of the key goals is to make this Core Grid Ontology general enough and easily extensible to be used by different Grid architectures or Grid middleware, so that the CGO can provide a common basis for representing Grid knowledge about Grid systems, including Grid resources, Grid middleware, services, applications, and Grid users.

The Core Grid Ontology is designed and developed based on a general model of Grid infrastructures, and described in the Web Ontology Language OWL [3]. Such an ontology can play an important role in building Grid-related Knowledge bases and in supporting the realization of the Semantic Grid. We adopt the CGO as the key building block for the GriSen. It is used for both the creation of a Grid knowledge base and knowledge-based query.

3 Building a Grid Knowledge Base Using CGO

A Grid knowledge base is normally comprised of a set of Grid Ontology classes, the relationships and constraints among those ontology classes, and instances of the classes (i.e. Individuals). In reality, the knowledge base may contain a large number of instances of different CGO classes. To build a Grid knowledge base, creation and updating of the instances of the Grid knowledge base is most important and difficult work. Traditionally, the instances of a knowledge base are created by a manual process with ontology editor. However, the manual process to build and maintain the instances is impossible or difficult for a Grid system. First of all, Grids contain large number of different Grid entities. Hence, the number of their instances is large. It is impossible to generate those instances manually. Secondly, Grids are characterized as dynamic. Consequently, the metadata information about them is also changed frequently. To catch up those changes by hand is very difficult. To cope with these issues, we design an ontology-driven approach that can fetch the information from grid information sources, and represent heterogeneous information about Grids in OWL format.

4 Querying a Grid Knowledge Base

Another important consideration is how to query the Grid knowledge base. Grids are complicate distributed system, which may comprise of a set of interacted components and massive heterogeneous resources. Hence, a Grid user normally does not know exactly what to ask about to the Grid knowledge base. To this end, we design a ontology based query service that supports Grid information navigation based on the definitions and relationships of the Grid entities in the CGO. It can process user requests, and generate queries according to the knowledge of CGO and users' willing.

We adopt the SPARQL as the query language to query the metadata in the Grid knowledge base [4]. SPARQL is a query language for getting information from RDF graphs. It provides facilities to: 1) extract information in the form of URIs, blank nodes, plain and typed literals; 2) extract RDF sub-graphs; 3) construct new RDF graphs based on information in the queries graphs.

The SPARQL query language is based on matching graph patterns. The simplest graph pattern is the triple pattern, which is like an RDF triple , but with the possibility of a variable instead of an RDF term in the subject, predicate or object positions. The query consists of two parts, the SELECT clause and the WHERE clause. The SELECT clause identifies the variables to appear in the query results, and the WHERE clause has one triple pattern.

We design a OntoQuery service, which can help users make a SPARQL query according to user's questions. We illustrate how to query Grid knowledge base using SPARQL with examples as follows:

```
(1) Is there a VO providing exclusive access to a shared-memory
    multiprocessor system with at least 16 processors, 8 GB of main
    memory, and a usage charge of not more than 100 euros per CPU
    time?

PREFIX  cgo: <http://grisen.grid.ucy.ac.cy/cgo/0.1/>
FROM    <grisen.owl>
SELECT  ?VO
WHERE   { ?x cgo:hasName ?VO .
          OPTIONAL { ?x cgo:hasCPUnum ?number . FILTER (?number > 64)}
          OPTIONAL { ?x cgo:hasCPUType ? .    FILTER (?number > 8)}
          OPTIONAL { ?x cgo:price  ?price  . FILTER (?price =< 100)}

(2) Find services running Quantum Chromo-Dynamics calculations (QCD)
    using F90 and MPI.

PREFIX  cgo: <http://grisen.grid.ucy.ac.cy/cgo/0.1/>
FROM    <grisen.owl>
SELECT  ?Service
WHERE   { ?x cgo:runningService ?service   .
  OPTIONAL { ?x cgo:hasName "QCD"  .
                     ?y cgo:installedOn     .
                     ?z cgo:hostsSoftware  . }

(3) Find the pricing and prior clientelle of Grid services that
    provide access to the XYZ workflow for real-time oil refinery
    simulations.

PREFIX  cgo:  <http://grisen.grid.ucy.ac.cy/cgo/0.1/>
FROM    <grisen.owl>
SELECT  ?Service ?Workflow
WHERE   { ?x cgo:runsOn ?service .
  OPTIONAL { ?x cgo:hasName "PPC" . }
          UNION    { ?y cgo:access ?Workflow .
          OPTIONAL { ?z cgo:hasName ‘‘XYZ’’ .
                     ?s cgo:simulation   ‘‘Oil’’ .}
        }
```

The above examples show that the SPARQL is capable of querying a Grid system, in particular, supporting complex queries about "the particular properties&values of the Grid entities". Since SPARQL is a RDF-based query language, we currently also investigate how it can be used to query the complicate relationships among the CGO classes which are represented in OWL.

5 Conclusions

In this paper, we present our on-going work on building a Grid Information knowledge base, and querying the knowledge of Grid resources, Grid middleware, services, applications, and Grid users based on a Core Grid Ontology.

Next step, we plan to implement a set of Grid services that can be used to build and update a Grid information knowledge base automatically and dynamically.

References

1. M. D. Dikaiakos, R. Sakellariou, and Y. Ioannidis, "Information Services for Large-scale Grids: A Case for a Grid Search Engine," in *Engineering the Grid: status and perspective*, J. Dongarra, H. Zima, A. Hoisie, L. Yang, and B. DiMartino, Eds. American Scientific Publishers, January 2006.
2. S. Campana and A. S. M. Litmaath, "LCG-2 Middleware Overview," LCG Technical Document, https://edms.cern.ch/file/498079//LCG-mw.pdf.
3. W. Xing, M. Dikaiakos, and R. Sakellariou, "A Core Grid Ontology for the Semantic Grid," in *Proceedings of 6th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2006).* Singapore: IEEE Computer Society., May 2006, pp. 178–184.
4. E. Prud'hommeaux and A. Seaborne, *SPARQL Query Language for RDF*, W3C Working Draft, July 2005.