Semantic-Based Data Access Services on the Grid

He Huang[1,2] Zhongzhi Shi[1] Yong Cheng[1,2] Lirong Qiu[1,2] Xiaoxiao He[2,3]

[1]Institute of Computing Technology, the Chinese Academy of Sciences, Beijing, China 100080
[2] Graduate School of the Chinese Academy of Sciences, Beijing, China 100039
[3] Wuhan Institute of Physics and Mathematics, the Chinese Academy of Sciences, Wuhan, China 430071

Abstract

Service-oriented approaches have attracted considerable attention recently for distributed application in e-Business and e-Science. There are many research and development activities focusing on consistent access to and coordinated use of databases from grid applications. However, the lack of explicit and formal expression of data semantics prevents the grid-enabled database systems from expanding into more sophisticated applications. To remedy it, ontology grids are proposed to explore important aspects of service-oriented and semantic-based data management, that is, how to integrate ontologies with a service-oriented grid and support semantic-based data access and integration services on the grid.

Keywords : ontology grid, semantic -based data access, semantic-based data integration.

1. Introduction

In an increasing number of scientific disciplines, there is a huge amo unt of databases which are important community resources. The combination of large data set size, geographic distribution of resources and users, and sophisticated applications on data challenges most of the existing methods for coordinated data sharing and manipulation. Currently, service-oriented approaches, e.g. Web Services and the Open Grid Services Architecture (OGSA), have attracted more and more attention. Research [1][2] has focused on integrating databases into Grids and creating a service-oriented "Virtual Database System". Two important toolkits OGSA-DAI [8] and OGSA-DQP [9] have been standardized. The OGSA-DAI uses OGSA facilities to assist with data access and integration from separate data sources via the grid. OGSA-DQP extends the OGSA-DAI architecture with two new services: the Grid Distributed Query Service (GDQS) and the Grid Query Evaluation Service (GQES).

In these methods and middleware toolkits, data access and integration functions are mainly based on metadata catalog managed by MCAT. However, the use of metadata is currently very simple. As the grid-enabled database system expands into more sophisticated applications, more powerful metadata systems and tools will be required. This result is also in accordance with the tenet of Semantic Web [5] and Semantic Grid [7], both of which have already addressed the importance of the well-defined meaning. Ontology is an effective way to describe the meaning formally and explicitly. It can be considered as semantically rich metadata which abstracts over specific data. This paper proposes an ontology grid architecture built on top of OGSA, OGSA-DAI and OGSA-DQP. By using ontologies to structure data semantics, the ontology grid constructs a knowledge space for data objects and supports the services of semantic-based data access and integration.

The rest of this paper is structured as follows. Related work is examined in section 2. Section 3 describes the framework of ontology grid, followed by a conclusion in section 4.

2. Related Work

Many research activities have focused on data handling on the Grid. The SDSC Storage Resource Broker (SRB) [11] is a data grid infrastructure that provides a uniform interface for connecting to heterogeneous data resources over a network and accessing replicated data sets. OGSA-DAI [8] and OGSA-DQP [9], built on top of OGSA, deliver high-level data management functionality for the Grid. The Metadata CATalog (MCAT) is adopted in these methods to provide a way to access data sets and resources based on their attributes and/or logical names rather than their names or physical locations. However, at least two problems are left [10]: firstly, although it is possible to uniquely name a billion objects, it will be very difficult to discover a certain object without a context associated with it; secondly, only the naming mechanism is not sufficient to support semantic-based data integration of multiple data repositories, which is often needed by researchers to combine data on concepts to gain more complete pictures of or understanding on data. The deficiency of the grid-enabled data handling system is mainly caused by the lack of the representation of data semantics.

The semantic grid is an initiative of the UK EPSRC/DTI Core e-Science Program. It aims to enable scientists to generate, analyze, share and discuss their insights, experiments and results in such a way that there is a high degree of easy-to-use and seamless automation, and flexible collaborations and computations on a global scale [7]. The acquiring and using knowledge in grid environment is the key differentiator between the current grid and the semantic grid.

Ontology can bring explicitly defined and machine-processable semantics to data and support human-machine and machine-machine communication based on semantics exchange [6]. In this paper, the ontology grid intends to take advantages of ontologies and OGSA. They can be viewed as a distributed ontology-driven knowledge management system in the sense that ontologies and their underlying data are distributed on sites.

3. The Ontology Grid

Defined on top of OGSA, OGSA-DAI and OGSA-DQP, the ontology grid focuses on integrating ontologies with the gird, provides a virtual ontology view on multiple ontologies from different locations, and supports semantic-based data access and integration. The ontology grid can:

- support queries over *Grid Services* (GSs) available on the Grid;
- use the facilities of the OGSA to dynamically obtain the resources necessary for semantic-based data access and integration;
- use the standard of OGSA-DAI and OGSA-DQP to provide consistent access to database and distributed query processor for the grid;
- adapt techniques of knowledge management to grid-enabled knowledge management.

In this architecture, a *high level O-Grid layer* is added and a set of services are defined by using the existing Grid Service (GS) standard. These defined services are themselves GSs, and thus can be discovered and invoked in the same way as other GSs. The *high level O-Grid layer* is depicted in figure 1.

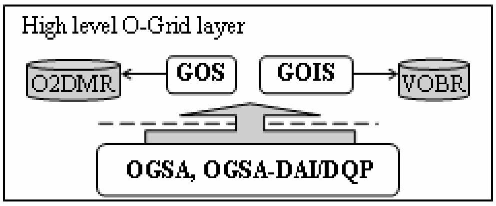

Fig. 1: Ontology Grid.

3.1. Grid Ontology Service (GOS)

The main aim of GOS is to give access to ontologies and support semantic-based data access. I extends GSs with the three portTypes: GOT, GOE and GOS.

The **Grid Ontology Transport** (GOT) portType enables the transport of ontology, together with or without extension, between GOSs and between clients and GOSs. This portType allows ontology to be pushed or pulled by **putOntology** and **getOntology** operations respectively.

The **Grid Ontology Extension** (GOE) port type is used to build ontology-to-data mappings (O2D mappings) between ontologies and their underlying data. The primary operation in the GOE port type is **extend**. The operation **extend** receives a requirement of building O2D mappings between an ontology and data repositories. The mappings are encoded into an XML document and stored in Ontology-to-Data Mapping Repository (O2DMR). The information of the ontology, its O2DMR, and resource metadata of the underlying data repositories is contained in *Service Data Elements* (SDEs). An example of an O2D mapping between a concept and a relational DB is shown in following:

```
<O2DMappings srcOntology="Onto1">
  <O2DMapping>
    <concept name="InstructionBook"/>
    <mappingExpression>
      <queryStatement name="s1" destSource="DS1">
        <expression>
          select * from Table1 where type="Instruction"
        </expression>
        <objectKey>
          <keyAttribute name="ISBN" type="string"/>
        </objectKey>
      </queryStatement>
    </mappingExpression>
</O2DMapping >
      …
</O2DMappings>
```

The **Grid Ontology Query** (GOQ) portType is the core of Grid Ontology Service. This portType accepts a request for ontology querying. The primary operation is **perform**, which takes a request document and returns a response document. Before writing a query the user may wish to browse the intensional description of ontologies and their O2O mappings. A GUI tool, called OntoSphere [12], can accommodate this need. OntoSphere offers an interface for entering queries, shown in figure 2. Then the request document is constructed on the fly. An example of the XML fragment of a query request is shown in following:

```
<gridOntologyServiceRequest>
  <Header> … </Header>
  <Body>
    <rdqlQueryStatement name="s1" srcOntology="Onto1">
```

```
        SELECT  ?title
        WHERE (?x  rdf:type  Publication)
              (?x  authors  "Tim Burners-Lee")
              (?x  subjects  "Semantic Web" )
              (?x  title  ?title )
   </rdqlQueryStatement>
   <Delivery name="delivery" from="s1" to="consumer">
     <Mechanism type="bulk"/>
     <Mode type= "full"/>
   </Delivery>
 </Body>
</gridOntologyServiceRequest>
```

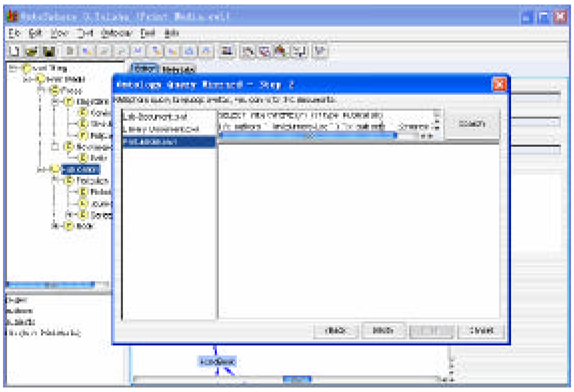

Fig. 2: The interface for query input.

An ontology query should be decomposed into two groups of sub-queries: sub-queries on the intension and sub-queries on the instance base (semantic-based data queries). For each sub-query in the second, it is needed to translate it, by referring to the O2DMR, into a query for the underlying data repositories, and then send the data-based queries to GDQS in OGSA-DQP.

GOS also extends GSs with the two services. Firstly, a *Grid Ontology Service Registry* (GOSR), by implementing the **ServiceGroup** portType in GS, is a facility for the publication of GOSs. Services are registered with a GOSR via an **add** operation. Registered services can be found using a **findServiceData** call. Additionally, a GOS is able to report registered service changes to service instances if they have subscribed for this via a **subscribe** call. Secondly, a *Grid Ontology Service Factory* (GOSF), by implementing the **Factory** portType in GS, is to create GOS instances as requested. A new GOS instance is created by **createService** call.

Figure 3 illustrates the interaction for a ontology query. We adopt the way used in [1] to draw the figure. Circled numbers denote the order of an interaction in a sequence. Solid arrows denote invocation. Dashed ones denote instantiation. (1) A GOSF registers itself to GOSR by **add** call. (2) A client discovers the GOSF by passing a **findServiceData** request to GOSR. (3) A GOS instance is create by **createService** call. (4) The client submits a query by calling **perform** in the GOS portType. (5) The sub-queries about data are translated and sent to OGSA-DQP. (6) The query results is obtained from OGSA-DQP. (7) The final result is delivered to the Client.

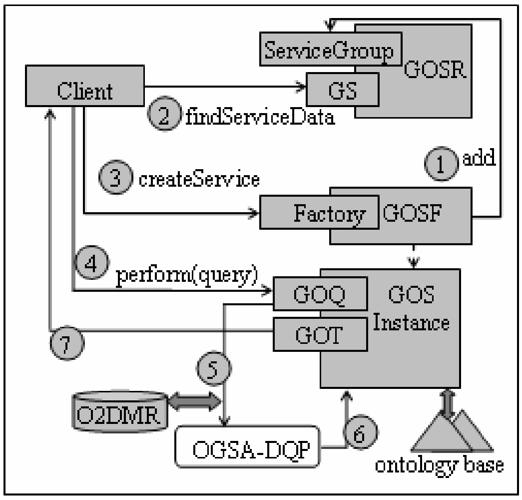

Fig. 3: GOS Interaction for ontology query.

3.2. Grid Ontology Integration Service (GOIS)

The main aim of GOIS is to integrate ontologies. It extends GOS with the **Grid Ontology Integration** (GOI) portType. It accepts a request for integrating two ontologies into a new composite ontology, which can semantically cover its component ontologies. The new generated composite ontology can be viewed as a virtual ontology stored in *virtual ontology base repository* (VOBR). A virtual ontology might provide the illusion that a single ontology is being accessed, whereas in fact, several ontologies is being accessing. The primary operation in the GOI portType is **ontologyIntegrate**. It accepts a request document for integrate two ontologies and import them to the GOIS instance. OntoSphere is invoked to implement three important phrases in ontology mediation: finding similarities between terms, specifying ontology mappings and integrate the two into a virtual ontology.

The GOQ port type of GOIS can support queries on the integrated ontology. The processing of such a query includes optimization, decomposition and local evaluation. Optimization is to optimize the query into a distributed query plan. Decomposition is to decompose the query into groups of sub-queries so that each group is on one real ontology. Then all the groups are sent to corresponding GOS instances.

GOIS also has its corresponding registry, a *Grid Ontology Integration Service Registry* (GOISR), and factory, a *Grid Ontology Integration Service Factory* (GOISF). Similar to GOSR and GOSF of GOS respectively, GOISR is responsible for publication of ontology mapping tools, while GOISF for creation of GOIS instances .

Figure 4 illustrates the interaction of GOIS. The registry of GOISF, discovery of GOISF, and creation of a GOIS instance are similar to those of GOS shown in figure 4. Therefore the three operations are omitted from figure 3. (1) The client submits an integration request. (2) The GOI portType import ontologies. (3) The integrated ontology is stored in VOBR. (4) A query request is sent. (5) Sub-queries are sent to GOS instance. The result is retrieved and delivered to client, which is not pictured in figure 4.

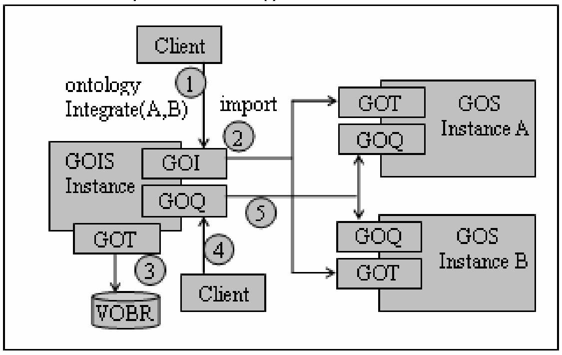

Fig. 4: GOIS Interaction.

4. Conclusion

The ontology grid, built on top of OGSA, OGSA-DAI and OGSA-DQP, provide an infrastructure to manage knowledge resources modeled by ontologies. It inherits many features from its underlying infrastructures, such as service-oriented architecture, and coordinated use of multiple databases. It allows for creating a global knowledge management system that spans heterogeneous and independent data repositories. It serves an interoperability mechanism for turning remote, heterogeneous resources into a globally well-ordered knowledge space.

5. Acknowledgements

This work has been funded by the National Basic Research and Development Plan of China (Grant No. 2003CB317000) and the National High-Tech Program of China (Grant No. 2003AA115220).

6. References

[1] M. Nedim Alpdemir, Arijit Mukherjee, Norman W. Paton, and Paul Watson, "Service-Based Distributed Querying on the Grid", in the Proceedings of the First International Conference on Service Oriented Computing, pages 467-482. Springer, 15-18 December 2003.

[2] Paul Watson, "DATABASES AND THE GRID", In F. Berman, G. Fox, and T. Hey, editors, *Grid Computing: Making The Global Infrastructure a Reality*, John Wiley & Sons Inc., 2003.

[3] Ian Foster, Carl Kesselman, and Steven Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International J. Supercomputer Applications*, 15(3), 2001.

[4] Ian Foster, Carl kesselman, Jeffrey M. Nick, and Steven Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", *Open Grid Service Infrastructure WG, Global Grid Forum*, June 22, 2002.

[5] Tim Berners-Lee, James Hendler, Ora Lassila, "The Semantic Web", *Scientific American*, May 2001.

[6] J. Davies, D. Fensel, and F. van Harmelen, "Towards the Semantic Web: Ontology-Driven Knowledge Management", New York: Wiley, 2003.

[7] David De Roure, Nicholas R. Jennings and Nigel R. Shadbolt, "The Semantic Grid: A Future e-Science Infrastructure", In F. Berman, G. Fox, and T. Hey, editors, *Grid Computing: Making The Global Infrastructure a Reality*, John Wiley & Sons Inc., ISBN: 0-470-85319-0 2003.

[8] OGSA-DAI, http://www.ogsadai.org.uk/.

[9] OGSA-DQP, http://www.ogsadai.org.uk/dqp/.

[10] Reagan W. Moore, "Knowledge-based Grids", Eighteenth IEEE Symposium on Mass Storage Systems and Technologies (MSS'01), pages: 29-39, April 17-20, 2001.

[11] Arcot Rajasekar, Michael Wan, Reagan Moore, Wayne Schroeder, "Storage Resource Broker - Managing Distributed Data in a Grid", *Computer Society of India Journal, Special Issue on SAN*, Vol. 33, No. 4, pp. 42-54 Oct 2003.

[12] OntoSphere 3.0 alpha, http://www.intsci.ac.cn/users/huangh/ontosphere/.